В. Ф. ОЧКОВ,
Е. П. БОГОМОЛОВА,
Д. А. ИВАНОВ

ФИЗИКО-МАТЕМАТИЧЕСКИЕ ЭТЮДЫ С MATHCAD И ИНТЕРНЕТ

УЧЕБНОЕ ПОСОБИЕ

*Издание второе,
исправленное и дополненное*

ЛАНЬ®
САНКТ-ПЕТЕРБУРГ • МОСКВА • КРАСНОДАР
2017

ББК 32.81я73
О 95

Очков В. Ф., Богомолова Е. П., Иванов Д. А.

О 95 Физико-математические этюды с Mathcad и Интернет: Учебное пособие. — 2-е изд., испр. и доп. — СПб.: Издательство «Лань», 2017. — 484 с.: ил. — (Учебники для вузов. Специальная литература).

ISBN 978-5-8114-2127-5

Излагаются основы применения математических методов, современных вычислительных средств (Mathcad, Maple, SMath и др.) и Интернета для решения типовых задач математики, физики, химии и других школьных и вузовских дисциплин. Рассматриваемые задачи затрагивают вопросы решения уравнений (алгебраических, дифференциальных, интегральных), программирования, статистики, обработки изображений, криптографии, решения головоломок, создания анимаций кинематических и динамических объектов, нечеткой логики, нечетких множеств, оптимизации и др.

Книгу можно рассматривать как пособие к новому зарождающемуся учебному курсу «Физико-математическая информатика», объединяющего в школах и вузах преподавание информатики, математики, физики, химии, литературы, культурологии, истории, изобразительного искусства и других дисциплин (STEAM: Science, Technology, Education, Art and Mathematic) в эпоху всеобщей компьютеризации.

ББК 32.81я73

Рецензенты:
А. В. ГИГЛАВЫЙ — кандидат технических наук, старший научный сотрудник, зам. директора ГБОУ «Лицей № 1533 (информационных технологий)», г. Москва;
Е. В. НИКУЛЬЧЕВ — доктор технических наук, профессор, проректор по научной работе, зав. кафедрой программных систем НОУ ВО «Московский технологический институт».

Обложка
Е. А. ВЛАСОВА

© Издательство «Лань», 2017
© Коллектив авторов, 2017
© Издательство «Лань», художественное оформление, 2017

ПРЕДИСЛОВИЕ

Какую цель преследует публикация данной книги?

Эта книга фактически писалась и публиковалась по частям, начиная с 1985 г., в виде статей в различных журналах, отрывков книг, докладов на конференциях.

Почти все эти публикации открыты на сайте:

http://twt.mpei.ac.ru/ochkov/work2.htm.

Зачем же нужно издавать их отдельной книгой? Тем более, что в настоящее время традиционное бумажное научно-техническое книгоиздательство переживает тяжелые времена и фактически постепенно умирает[1]. Издательства публикуют в основном только те книги, затраты на выпуск которых заранее покрыты либо самим автором, либо каким-либо грантом или бюджетом государства, университета, НИИ или какой-нибудь коммерческой организации. Но и в этом случае экономить приходится на всем: на рецензентах, научных и литературных редакторах, корректорах. Из-за этого книги, увы, часто выходят в редакции автора, т. е. с многочисленными неточностями и «очепятками». Качество книг из-за этого резко снижается[2]. Да, издавать книги за счет собственных средств, а потом ждать возврата денег из книжных магазинов, а также из библиотек научно-технические издательства уже не могут себе позволить. Это происходит, в том числе и из-за конкуренции с электронными (интернетовскими) изданиями. Из-за нечестной конкуренции, когда «честная» книга какого-либо издательства появляется в открытой продаже, кем-то сканируется и размещается в Интернете, что полностью или частично блокирует ее продажи.

Но авторам все-таки удалось издать эту книгу, раз читатель держит ее в руках. И не просто издать, а издать с качественной редакторской и корректорской правкой. Какой ценой — это другой вопрос. Авторы не думают о коммерческом успехе книги. Главное, чтобы она была интересна и полезна читателям в бумажном и в электронном виде, в том числе и в виде живых расчетов в Интернете и анимаций. И еще более «главное» в том, что авторы этой вышедшей из печати книгой подведут некий промежуточный итог своей профессиональной деятельности и получат некоторое «творческое удовлетворение».

Итак, почему же эта книга вышла в бумажном виде?

Во-первых, «нет ничего приятней запаха только что отпечатанной книги». Тем более, если в списке ее авторов видишь и свою фамилию. При всей оперативности и доступности Интернета традиционная техническая книга еще не скоро сдаст свои позиции. Ее (сшитые бумажные страницы в мягком или твердом переплете) можно подарить коллегам, друзьям и близким, представить на творческий конкурс (на конкурс по замещению должности в вузе или НИИ, на получение ученого звания и т. д. и т. п.), что пока нельзя сделать с сайтом Интернета, у которого свои неоспоримые преимущества, а именно: возможность оперативного исправления ошибок и опечаток, дополнения новыми материалами, цветными иллюстрациями, трехмерными схемами, анимацией и живыми расчетами по формулам. Не станем спорить, что лучше — сайт или книга, а будем активно использовать преимущества этих двух разных по форме, но единых по содержанию медийных продуктов и стараться обходить их недостатки. В пользу бумажной книги обычно

[1] Когда появилось кино, то показалось, что театр скоро умрет. Когда появилось телевидение, то показалось, что кино скоро умрет и т. д. Но все пока живы-здоровы. Пожелаем же этого и традиционной бумажной книге!

[2] Тут часто говорят так: «снижается до уровня Интернета». Но в Интернете есть много вполне приличных изданий и по содержанию, и по оформлению. Авторы надеются, что это можно будет сказать и о возможной интернетовской версии этой книге.

приводят еще один довод — с такой книгой можно валяться на диване, открыть ее в дороге и т. д. Но в настоящее время появились электронные книги и планшетные компьютеры, смартфоны («таблетки»), с которыми тоже можно «валяться на диване, открыть в дороге», а также дополнительно делать в них электронные закладки, вести оперативный поиск и даже расчеты. Не хватает только «запаха только что отпечатанной книги», но и это дело времени. Авторы одновременно и опасаются, и надеются, что после выхода этой книги в свет она будет кем-то отсканирована и размещена в Интернете (см. выше). Опасаются потому, что это бьет по карману авторов и издателя. Надеются потому, что это косвенный признак того, что книга интересна и востребована читателем. Плохих книг, вернее, тех книг, какие не читают, в Интернете пиратским способом не публикуют.

В книге читатель не найдет традиционного детального описания задач. Авторы, во-первых, в эти описания ничего нового внести не могли, а могли только переписать существующие тексты, доступные в «более солидных» изданиях[3]. «Изюминка» книги — это то, как использовать современные компьютерные средства, в частности Mathcad и Интернет, для решения некоторых задач математики, физики, химии и других учебных дисциплин, на которых базируется современное школьное и высшее техническое образование.

В книге собраны *этюды*. Что это такое? На языке музыкантов и шахматистов этюд — это небольшое упражнение для отработки некоторых элементов игры. В этюдах книги ставилась цель не *научить* желающих работать в программной среде Mathcad (это все равно, что учить плавать вдали от бассейна), а показать *идеологию* работы с Mathcad, «отработать некоторые элементы игры» в среде этой математической программы. Делалось это, по возможности, на несложных и понятных примерах, затрагивающих актуальные вопросы вычислительной математики (решение уравнений и систем — алгебраических, дифференциальных, интегральных; дифференцирование, интегрирование, оптимизация, статистические расчеты, математическое моделирование, техника символьных преобразований, построение графиков и анимаций, обработка изображений, перебор вариантов, программирование и т. д.). Еще одна музыкальная аналогия. Алгоритм решения задачи можно уподобить нотной записи музыкального произведения. Реализация же алгоритма на компьютере — это исполнение мелодии. При этом могут быть использованы разные «музыкальные» инструменты: языки программирования, электронные таблицы, математические программы... Авторы выбрали Mathcad по причинам, о которых будет сказано во Введении.

У художников слово «этюд» (вернее, «эскиз»[4]) означает зарисовку, представляющую собой часть будущей картины. Этюды книги являются не только упражнениями, но и своего рода зарисовками, где внимательный читатель сможет найти немало полезных советов и интересных идей, которые пригодятся ему в дальнейшем при написании больших Mathcad-полотен, связанных не только с математикой, физикой или химией.

Этюды книги — это самостоятельные физико-математические заготовки, не всегда логически связанные между собой. Поэтому авторы просят у читателей прощения за возможные повторы. Книгу, кстати говоря, можно начать читать с любого этюда. Кроме того, слово «этюд» попало в название книги и из-за того, что эти «зарисовки» сильно разнятся между собой и по форме написания, и по времени создания, и по глубине проработки рассматриваемых физико-математических и прочих чисто компьютерных вопросов. В последующих изданиях книги, если они, конечно, будут, авторы постараются все более-менее систематизировать чтобы получилось более строгое учебное пособие.

---

[3] Вся классическая математика, физика, химия и др., которую преподают в школах и технических вузах, описана еще в XVIII и XIX вв. Все вышедшие позже книги по сути переписаны со старых учебников и монографий.

[4] Но таскают эти работы не в эскизниках, а в этюдниках!

Слово этюд в названии книги перекликается с названием очень интересного Интернет-проекта Николая Андреева «Математические этюды» (см. www.etudes.ru). Но этот проект больше нацелен на популяризацию математики, на математическую сторону задач, на показ готовых анимаций и представлений в виде 3D-решений математических задач. Этюды же этой книги ставят своей целью в первую очередь детализацию процесса решения задач с помощью математических пакетов.

Две книги одного из авторов, кстати говоря, оформлены в виде именно этюдов [1, 2]. Бог любит троицу! Посмотрим, как читатель примет эту третью этюдную книгу. «Компьютерно-писательская судьба» этого автора, кстати, началась именно с шахматного этюда — см. рис. 10.6 на стр. 246 и текст на стр. 244.

На оформление книги (шрифты в тексте, в формулах и рисунках, запятая или точка в десятичном разделителе чисел, русские или международные единицы измерения и прочие оформительские нюансы) сильно повлиял... компьютер. Авторы старались, чтобы «листинги программ» и их описания в тексте совпадали, но этого, к сожалению, не везде удалось добиться, в том числе и из-за того, что в книге приводятся расчеты, сделанные в разных «поколениях» Mathcad: Mathcad 15 и Mathcad Prime. Эти нюансы, касающиеся также и взаимоотношения автора с редактором, описаны в [3].

В книге помещено много ссылок на внешние Интернет-ресурсы: Mathcad-файлы, онлайн расчеты, анимации, литературные и другие источники. Интернет-адреса этих ресурсов, конечно, не нужно набирать вручную. Достаточно зайти на сайт книги и «кликнуть» по соответствующей ссылке. Была идея разместить в книге некий мини-словарь используемых в текстах терминов. Но от нее отказались, так как в настоящее время все эти термины хорошо описаны в Интернете и справку по ним несложно найти с помощью поисковиков Интернета, используя, конечно, надежные источники[5]. Авторы надеются, что читатели будут присылать свои комментарии и новые решения рассмотренных в книге задач, а также и новые задачи, которые будут размещены на сайте книги. Сами авторы планируют расширять описания и решения задач книги, вводить в нее новые этюды и новые комментарии к ним. Эти дополнительные материалы сначала будут опробованы на сайте книги, а затем перейдут в возможные новые издания.

Сайт книги — **https://www.ptcusercommunity.com/groups/etudes**, на котором читатели найдут также и координаты для связи с авторами. Сайт интерактивен: читатели могут зарегистрироваться на нем и дополнять его своими решениями и комментариями самостоятельно или при участии авторов. Статьи и другие интерактивные материалы можно также найти на другом авторском сайте **http://twt.mpei.ac.ru/ochkov/work2.htm**. В этот сайт вложено много сил и средств, включая и финансовых. На развитие сайта повлияли бюджетные средства, отпущенные на развитие Московского энергетического института как национального исследовательского университета.

Надеемся, что сайт книги станет еще одним востребованным форумом пользователей современных программных средств для решения задач математики, физики, химии...

Во втором издании книги исправлены первые 18 этюдов и добавлены 7 новых.

[5] Знать бы еще — какой источник надежный, а какой нет. Но можно просматривать разные источники и стараться делать вывод самому.

Введение

ПРЕПОДАВАНИЕ ИНФОРМАТИКИ, МАТЕМАТИКИ, ФИЗИКИ, ХИМИИ И МАТЕМАТИЧЕСКИЕ ПАКЕТЫ, ИЛИ МЕЖДУ СЦИЛЛОЙ И ХАРИБДОЙ

История развития методов решения задач по математике, физике, химии в школе и вузе — это, помимо прочего, и история борьбы с... *вычислительными средствами*. Сначала (на уроках устного счета, например) запрещали пользоваться бумагой и ручкой, затем (при изучении счета «столбиком» ручкой на бумаге) — калькулятором и, наконец, при решении более сложных задач — компьютером с современными математическими программами: Mathematica, Maple, Matlab, Mathcad, SMath и др. Вернее, прямо не запрещали и не запрещают, а говорили и говорят, что, мол, решение на компьютере школьной или вузовской задачи по математике равносильно решению задачи для устного счета (65 помножить на 9, например) на калькуляторе. Этот запрет касается не самих вычислительных средств, которые можно и нужно осваивать с помощью специально подобранных примеров на занятиях по информатике и программированию, а применения этих компьютерных инструментов для решения задач, придуманных для уроков и семинарских занятий по арифметике, тригонометрии, линейной алгебре, математическому анализу... Но современные школьники и студенты этого не понимают. Более того, они уже не мыслят учебы, а иногда, к сожалению, и досуга без компьютера.

С устным счетом здесь все более-менее ясно. Эти упражнения — прекрасная *гимнастика для ума* (см. этюд 6). Использование на таких занятиях калькулятора равносильно дооборудованию спортивного тренажера... гидроусилителями. Счет «столбиком» также можно рассматривать как гимнастику для ума и хорошую моторику для рук. Но тут подмешивается еще один довод. Считать в уме нужно уметь, если под рукой не окажется карандаша и бумаги, считать карандашом на бумаге нужно уметь, если под рукой не окажется калькулятора и т. д. Но... добывать огонь трением «нужно уметь, если под рукой не окажется» спичек или зажигалки, определять стороны света по деревьям в лесу «нужно уметь, если под рукой не окажется» компаса или навигатора и т. д. и т. п. К сожалению или к счастью, с развитием цивилизации мы разучились считать[1] в уме, добывать огонь трением, ориентироваться на местности по природным приметам и т. д. и т. п. Школьный учитель одного из авторов этой книги не уставал повторять на занятиях по арифметике, что если мы, его ученики, не научимся быстро и точно считать в уме или, по крайней мере, столбиком на бумаге, то нас будут обвешивать и обсчитывать в магазинах. Сейчас эта мотивация освоения устного счета уже не работает, т. к. в современных супермаркетах практикуют современные же методы «обвешивания и обсчитывания».

Противники использования современных компьютерных средств решения школьных и вузовских задач по математике, физике, химии также опираются на ряд других доводов, о которых они, правда, открыто не говорят.

Во-первых и к сожалению, многие школьные учителя и преподаватели вузов, просто-напросто не умеют работать с современными компьютерными математическими программами и/или не знают об их возможностях. Эти преподаватели освоили компьютер, но на уровне офисных программ (текстовый редактор, табличный процессор, электронная почта, работа в Интернете) и азов операционной (файловой) системы, но дальше идти не хотят или не могут, оправдывая это и тем, что, мол, математические программы вредны для учащихся (см. выше).

[1] Быстро и безошибочно считали в уме, как правило, сугубо отрицательные литературные герои: Чичиков из «Мертвых душ» Гоголя, Корейко из «Золотого теленка» Ильфа и Петрова.

Во-вторых, внедрение этих программ в учебный процесс требует кардинального пересмотра содержания и методов преподавания, а также переписывания учебников и задачников по математике, физике, химии или, по крайней мере, существенной их переработки. Примеры в задачниках, конечно, переписываются. Прежде там были, например, фунты и аршины (см. этюд 6), а теперь метры и килограммы. Раньше в задаче было: «Землекоп выкопал столько-то метров канавы», а сейчас — «компьютер имеет такой-то объем памяти», но суть задач и методика их решения при этом, увы, не меняются.

В-третьих, вышеупомянутые компьютерные программы довольно дороги. Их не в состоянии купить многие наши школы и вузы. Но эта проблема, сразу скажем, решаема. Фирмы-разработчики математических программ предоставляют учебным заведениям существенные скидки, а в ряде случаев передают программы бесплатно. Но с корыстной, конечно, целью. Учащиеся, освоив бесплатную программу, после окончания вуза, купят ее сами или попросят своего работодателя купить программу. Преподаватели должны не сетовать на дороговизну программ или на невозможность по разным причинам работать с пиратскими копиями, а искать пути решения этой проблемы, связываться, например, с разработчиками программ и их дилерами. Кроме того, нужно помнить, что есть бесплатные версии программ. Так, например, фирма PTC — разработчик программы Mathcad дает возможность месяц работать с полной версией Mathcad Prime, а потом с ее укороченной версией Mathcad Express, позволяющей, тем не менее, решать довольно сложные математические задачи.

И еще один очень важный момент.

В настоящее время в образовательный процесс технических вузов интенсивно внедряются мощные программные средства для компьютерного моделирования физических процессов. Если говорить о вузах энергетического профиля, в одном из которых работают авторы этой книги, то в них стали широко использовать программы типа ANSYS или COMSOL (моделирование процессов тепло- и массообмена), ThermoFlow (расчеты термодинамических циклов), FlowVision (моделирование процессов гидрогазодинамики) и др. На базе подобных программ создаются новые лабораторные работы, проводятся семинары и даже читаются курсы лекции во многих продвинутых вузах энергетического и энергомашиностроительного профиля.

При этом данные программы зачастую используются как некие «черные ящики», куда закладывают исходные данные, закрывают крышку ящика (нажимают кнопку *Рассчитать*), открывают крышку и вынимают результат: таблицы с цифрами, графики, трехмерные виртуальные объекты (которые затем можно даже распечатать на 3D-принтере), анимацию и т. д. При этом у студентов имеются очень смутные (если вообще имеются) представления о том, что происходит в этих «черных ящиках». Ведь эти программы создавались не для образовательных, а для коммерческих целей, а сам расчетный код в них надежно спрятан от посторонних взглядов (защищен от промышленного шпионажа). Нередки случаи, когда этот код скрывают и из-за того, что он весьма старый и примитивный, а «современность» программы видна только по ее интерфейсу (весь пар ушел в гудок). Эффект от такого образовательного процесса зачастую оказывается близким к нулю. Есть, правда, образовательные технологии, когда в рамках лабораторной работы сначала проводится реальный эксперимент (например, нагрев или охлаждение физического тела с замером температуры на его поверхности с помощью тепловизора) с последующим моделированием и отображением этого процесса на компьютере. Сравнение картинок с тепловизора и с экрана дисплея становится предметом обсуждения вопросов, относящихся к сути физического явления, методам и ошибкам средств инструментального контроля, а также ограничений и допущений компьютерных программ, это явление моделирующих.

Мы сейчас обрисовали некую Сциллу, подразумевая под этим «чудищем-чудовищем» не сами пакеты типа ANSYS или COMSOL, а то, как они внедряются в учебный процесс. Сейчас расскажем о втором «чудище» — о некой Харибде.

В настоящее время накопилось большое число программ на так называемых языках программирования высокого уровня (Fortran, Pascal, BASIC, C и т. д.) для расчетов процессов, которые моделируются в вышеописанных «черных ящиках». Эти программы (коды) преподаватели старшего поколения пытаются «переподать» современным студентам для объяснения сути расчетов. Но для большинства студентов языки этих программ непонятны. И главное, неясна мотивация их изучения основной массой студентов, которые никогда не будут по серьезному программировать (тем более на этих языках), а будут использовать готовые программные продукты (те же «черные ящики») или математические пакеты для решения своих частных задач: расчетов по сложным формулам, решения уравнений и систем (алгебраических и дифференциальных), построения графиков, создания анимации, обработки статистических данных и т. д. и т. п.

Но многие преподаватели по-прежнему стараются обучить студентов работе именно со своими программами. Этому есть много объяснений. Во-первых, не пропадать же товару — созданному с такими усилиями и основательно протестированному десятилетиями коду, например, на «старом добром фортране». Во-вторых, многие преподаватели, повторяем, не знают о возможностях новых программных средств или считают, что они (преподаватели) уже устарели для изучения чего-то нового. В-третьих, обращение к «старому доброму коню», пардон, фортрану или бейсику, «который борозды не испортит», объясняется и тем, что во многих организациях, где придется работать студентам по окончании вуза, тот же фортран является тем «конем», на котором до сих пор «пашут», руководствуясь помимо прочего и принципом, что лучшее — враг хорошего.

Проплыть между Сциллой и Харибдой, на наш взгляд, можно, если при обучении использовать математические пакеты, в частности, Mathcad для моделирования физических явлений с детализацией их сути и методов реализации на компьютере. Это и есть главная идея данной книги.

Но почему Mathcad? Чем же он хорош для такого рода расчетов? Во-первых, конечно, тем, что один из авторов хорошо знает этот пакет и даже написал несколько книг по нему [1, 2, 4–18].

Но есть, конечно, и объективные показатели удобства работы с Mathcad[2]. Перечислим их.

1. Хорошая документированность расчетов. Расчет, сделанный в среде Mathcad, можно распечатать и отдать на проверку или рецензирование человеку (преподавателю), который никогда не работал на компьютере[3]. Mathcad-расчет полностью повторяет расчет, сделанный на бумаге, благодаря режиму WYSIWYG — What You See Is What You Get — «что ты видишь (на экране дисплея), то ты и получишь (на бумаге принтера)». Распечатки решений, полученные в среде Mathcad, можно оставить в архиве и через 50–100 лет прочитать их и понять, что там написано, а затем, если понадобится, без особых усилий воспроизвести в новых программных средах, которые появятся к тому времени. А это очень важно.

Если говорить о компьютерных информационных технологиях, то в этой области в настоящее время наблюдается кризис, которого мы уже частично коснулись выше и который условно называют «кошмаром унаследованного программ-

[2] Расчеты книги слегка разбавлены решениями в альтернативных математических средах: SMath (русский клон пакета Mathcad — см. рис. 1.2 на стр. 16), Maple (рис. 5.12 на стр. 115, рис. 17.3 на стре. 369) и Mathematica (Интернет-версия — см. рис. 1.1 на стр. 13, рис. 7.35 на стр. 172, рис. 9.24 на стр. 223 и рис. 9.30 на стр. 229).

[3] В конце этюда 9 мы покажем одно из отличий в маткадовской и ручной формулах. Там же будет сказано, что эти отличия постепенно исчезают.

ного обеспечения». Что это такое? Представим себе некую фирму, корпорацию или университет, которые за 30–40 лет интенсивного использования компьютеров разработали и накопили большое количество программ различной степени сложности для расчета и моделирования разнообразных процессов, аппаратов и технологий в тех или иных областях науки и техники, а также для создания и управления базами данных (знаний). На смену морально и физически устаревшим компьютерам приходят новые с новой «начинкой» и новыми операционными системами. Компьютеры объединяются в локальные сети, которые, в свою очередь, не остаются в стороне от процесса «глобализации» и интегрируются в Интернет — проводной и беспроводной. Развитие беспроводной связи компьютеров с серверами стало причиной возникновения «облачных» информационных технологий. Модернизация компьютерного парка часто приводит к тому, что старые прикладные программы перестают запускаться на новых или обновленных компьютерах — рабочих станциях и серверах. Иногда просто невозможно считать программу со старых носителей (перфокарт, перфолент, дискет различного диаметра, устаревших «флешек» и др.), т. к. новые компьютеры не имеют соответствующих считывающих устройств. Приходится либо совсем отказываться от таких программ и создавать с нуля новые, либо тратить время и средства на создание или приобретение неких утилит (эмуляторов) для запуска старых программ на новых или обновленных компьютерах.

Конкретный пример, касающийся этой книги. В нее вставлены ссылки на online расчеты по технологии Mathcad Application Server (MAS, Mathcad 11 — файлы с расширением mcd) и на такие же сетевые расчеты по технологии Mathcad Calculation Server (MCS, Mathcad 14/15 — файлы с расширением xmcd). Так вот, файлы с расширением mcd в ряде случаев перестают работать в Интернете из-за того, что сетевая версия Mathcad 11 установлена на Windows Server 2002, который уже давно не поддерживается корпорацией Microsoft. Но и технология Mathcad Calculation Server (Mathcad 14/15) когда-то сойдет на нет. Ей на смену с марта 2016 г. идет технология PTC Mathcad Gateway, опирающаяся на Mathcad Prime 3.1 (см. **http://www.ptc.com/engineering-math-software/mathcad/gateway**). Пользователи Mathcad могут делать подписку на этот расчетный сервер и слать туда свои файлы, чтобы по ним можно было считать без самого пакета Mathcad Prime 3.1. Авторы книги не успели еще воспользоваться этой новой технологией.

Но это еще полбеды. Настоящая беда начинается тогда, когда из «фирм, корпораций, университетов» уходят специалисты, которые создавали и поддерживали эти программы, а пришедшая им на смену молодежь не может модернизировать их, адаптировать к новым требованиям. Виной тому и недостаточное документирование кодов, и отсутствие соответствующих инструментальных программных средств, и простая невозможность правильно прочесть программу, написанную на «мертвых» языках программирования. Если же все-таки с большим трудом удается воссоздать или модернизировать старую программу, используя старые или даже новые языки программирования, то «кошмар унаследованного программного обеспечения» через некоторое время всплывет вновь.

Хорошая документированность и детализация расчетов очень важны в образовательной сфере. Не секрет, что многие преподаватели высших учебных заведений настроены против использования студентами компьютеров для расчетов по учебным задачам. Мы уже касались этой проблемы. Эти преподаватели вполне обоснованно считают, что такие задачи нужно решать сугубо «ручками» без использования компьютеров, или, по крайней мере, теми средствами, которые они сами использовали, будучи студентами. Но эти преподаватели имеют в виду использование компьютеров как «черных ящиков» (см. выше), в которые «закладывают» исходные данные и из которых «вынимают» готовые ответы. Пакет Mathcad — это отнюдь не тот «черный ящик», который противопоказан студентам.

2. Работа с единицами измерений [3]. Электронные таблицы и языки программирования сыграли с нами злую шутку — отучили нас работать с физическими величинами. Вернее, приучили нас работать с величинами, лишенными размерности (приведенными к базовым СИ), а единицы их измерения (единицы СИ — метры, секунды, паскали, кельвины, джоули, ватты и т. д.) держать в уме, что очень неудобно и чревато ошибками в расчетах. Да и сами основные единицы СИ неудобны: базовая единица давления (паскаль) очень мала и всегда требует множителей кило или мега, температура в кельвинах плохо «чувствуется» и требует перевода в градусы (шкалу) Цельсия и т. д. Эта проблема будет более подробно рассмотрена в этюде 1. Эта особенность позволяет называть Mathcad не просто математическим, а физико-математическим пакетом. Он создавался, кстати говоря, физиком-ядерщиком по образованию — Алланом Раздоу (Allan Razdow).

3. Гибкая система имен переменных. Переменные и функции в среде Mathcad за редким исключением имеют те же имена, которые закрепились за ними в тех или иных научно-технических дисциплинах задолго до появления компьютеров. Например, греческой буквой η с различными индексами в термодинамике обозначают коэффициент полезного действия (термический, внутренний относительный и т. д.), а другой греческой буквой λ — теплопроводность. Это наряду с использованием традиционного написания математических констант, операторов и функций, а также верхних и нижних индексов делает язык Mathcad доступным всем непосвященным (см. п. 1 выше) без особых дополнительных комментариев. Нотация в среде Mathcad практически совпадает с общепринятой математической нотацией: интеграл, производная, модуль, степень и т. п.

Остальные четыре особенности присущи и другим математическим пакетам (MATLAB, Maple, Mathematica, SMath, Derive и др.), но мы их, тем не менее, перечислим.

4. Численная и символьная математика, позволяющая использовать при решении задач богатую библиотеку численных методов, предваряя или дополняя их попытками (удачными или менее удачными) аналитического решения задачи. Mathcad изначально создавался как пакет численной математики, к которому впоследствии (в 5-й версии) было прикреплено ядро символьной математики из пакета Maple, которое затем (в 13-й версии) было заменено на ядро символьной математики из пакета MuPAD. Такая же примерно «биография» и у «численного» пакета Matlab, где также вначале стоял символьный движок из ядра Maple, который потом был заменен таким же движком из пакета MuPAD. Пакеты же Maple и Mathematica изначально были пакетами символьной математики с элементами численных расчетов.

5. В пакет Mathcad встроены довольно мощные и гибкие инструменты создания плоской и объемной графики, а также анимации. Это позволяет легко и быстро визуализировать исходные, промежуточные и итоговые данные без вызова внешних процедур или написания вспомогательных программ, что способствует лучшему пониманию сути расчета, выявлению в нем возможных ошибок и ложных путей решения поставленной задачи.

6. Программирование. В среде Mathcad математические действия на рабочем листе выполняются как на обычном листе бумаги слева направо и сверху вниз [19]. Но иногда такой порядок счета необходимо изменить — не выполнять, например, какую-то часть операторов, а выполнять какую-то другую или выполнять выделенную группу операторов несколько раз. Такая возможность (программирование) в среде Mathcad предусмотрена, и ею успешно пользуются не только продвинутые пользователи, но даже и те, кто когда-то ошибочно считал, что он никогда не будет программировать. Инструменты программирования Mathcad позволяют решать довольно сложные задачи, не умещающиеся в узких рамках последовательного алгоритма (слева направо и сверху вниз).

7. В пакете Mathcad предусмотрена возможность расширения списка доступных функций тремя путями:

- путь первый — прикрепление к Mathcad через механизм DLL (Dynamic Link Library — динамически подключаемая библиотека) функций, написанных, например, на языке программирования C;
- путь второй — ссылка из рабочего документа на другие Mathcad-документы. После такой ссылки в рабочем документе станут доступными (видимыми, как говорят программисты) пользовательские переменные и функции, хранящиеся в документе (файле), на который была сделана ссылка. Эти и другие Mathcad-документы можно загрузить на свой компьютер или в локальную сеть и использовать как шаблоны. Но можно это и не делать, если компьютер имеет надежную и высокоскоростную связь с Интернетом;
- путь третий — инструменты Mathcad позволяют самому пользователю быстро написать и отладить функции, возвращающие, например, свойства материалов, опираясь на формулы, таблицы или графики, взятые из внешнего источника — из бумажной или электронной книги, а также из Интернета.

Эти три возможности можно и нужно использовать для того, чтобы в расчетном документе стали доступными, например, функции, имеющие в качестве аргументов параметры конкретной точки термодинамического цикла и возвращающие нужные теплофизические свойства рабочего тела в этой точке: удельную энтальпию, удельную энтропию, плотность, удельную теплоемкость (изобарную или изохорную), теплопроводность, вязкость и т. д. [2]. Без этого невозможно рассчитывать теплотехнические процессы (специализация одного из авторов).

И последнее. Расчеты, созданные в среде Mathcad, можно без особых хлопот разместить на сервере (в Интернете или в локальной сети) для работы без самого пакета Mathcad. Так, например, со многими расчетами, описанными в этой книге, можно поработать через Интернет по технологии Mathcad Calculation Server, не устанавливая на свой компьютер каких-то дополнительных программ.

В книге все эти возможности будут показаны на конкретных физико-математических примерах.

Этюд 1

ВЫЧИСЛИТЕЛЬНЫЙ РЕНЕССАНС, ИЛИ РУЖЬЕ НА СЦЕНЕ

В этюде будет рассказано, как можно докопаться до физической сути формул, приводящихся в различных справочниках, и оставить эту физическую суть в обновленной формуле, подготовленной для публикации в сетевых открытых интерактивных справочниках с «живыми» формулами.

Математика: погрешность вычисления, точность вычисления, порядок малости, порядок роста, замена переменных, производная, частная производная, определенный интеграл, свойства логарифмов, аналитическое решение обыкновенного дифференциального уравнения, решение трансцендентного уравнения.

Физика: колебание маятника, единицы физических величин, обезразмеривание физических величин, сила звука, КПД электростанции, теплотворная способность топлива, концентрация растворенного вещества, термодинамический цикл, энтропия, внутренняя энергия, мощность насоса, плотность воды, ускорение свободного падения.

Информатика: интерактивные расчеты в Интернете, числа с плавающей точкой, расчеты по эмпирическим и псевдоэмпирическим формулам, компьютерные средства аналитических преобразований (символьная математика), облачная функция.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127829**

Начнем этюд с трех небольших «зарисовок с натуры» — трех, так сказать, подэтюдов в первом этюде.

Первая зарисовка. На кафедре физики Alma Mater авторов этой книги студенты (вчерашние школьники) проводят лабораторные работы на реальном физическом стенде. Затем эти студенты должны подсесть к компьютеру, установленному в лаборатории, зайти на специальный расчетный сайт Интернета (например, на такой, какой показан на рис. 14.16 на стр. 321[1]), ввести данные, снятые со стенда, обработать их методом наименьших квадратов и получить кривую на экране дисплея, «обвешанную» экспериментальными точками. Так вот некоторые студенты (а их из года в год становится все больше и больше) не подсаживаются к компьютеру, а вытаскивают из карманов и сумок смартфоны, планшеты и другие «гаджеты», беспроводно заходят в Интернет на нужный сайт и вводят в него буквы и цифры, работая... двумя большими пальцами рук. Или только одним большим пальцем, если это смартфон. На вопрос преподавателя, почему они не используют лабораторный компьютер с полноразмерной и удобной клавиатурой, следует примерно такой ответ: «А я не умею работать со стандартной клавиатурой компьютеров и ноутбуков — я на ней набираю тексты и цифры очень медленно — одним пальцем, долго отыскивая нужную клавишу. Я уже привык к «клаве» моего смартфона, где можно набирать тексты даже на весу, не кладя гаджет на стол или колени». В транспорте (в московском метро, например, где сейчас есть сеть Wi-Fi) часто можно видеть молодых, да и не очень молодых людей, ловко набирающих тексты одним большим пальцем. Проводятся даже чемпионаты мира по такому набору текстов. Часть текста этой книги, кстати говоря, тоже набралось авторами не на стандартной «клаве», пардон, клавиатуре, а тыканьем электронного карандаша (стилуса) в виртуальную (экранную) клавиатуру планшета, или простым рукописным написанием текста самим этим стилусом на экране планшета. Кстати, о «простом рукописном написании». Многие бывшие школьники, а теперешние студенты, пишут «как курица лапой».

[1] В этом этюде будет рассказано о проведении линии вблизи экспериментальных точек (о статистической обработке опытных данных).

Их рукописный текст можно разобрать, если только он написан печатными буквами, как в анкетах или бланках ЕГЭ, ориентированных на чтение компьютером. И в этом винят компьютер. А на подходе массовые устройства, переводящие голосовые сообщения в текст. Тогда мы совсем разучимся писать, как сейчас разучились считать в уме (см. Введение).

Вторая зарисовка. Эти же и другие студенты на лекциях по математике, физике, химии довольно часто заходят со своих смартфонов-планшетов-ноутбуков на математические сайты, вводят алгебраические и дифференциальные уравнения (пример на рис. 1.1[2]), находят их аналитические или численные решения и... проверяют выкладки лектора, записанные мелом на доске. Тем же способом они проверяют правильность промежуточных и итоговых вычислений, проделываемых студентами у доски при решении задачи на семинарах. Ну и, естественно, сама лекция зачастую конспектируется этими же студенческими большими пальцами... Если вообще конспектируется. Некоторые студенты просто включают диктофон и время от времени фотографируют доску и преподавателя, задачи, решенные на доске другими студентами, и собственные решения на доске.

На рисунке 1.1, кстати говоря, можно видеть кнопку Step-by-step solution. Она позволяет проследить пошаговое решение задачи, что очень важно в образовательном процессе.

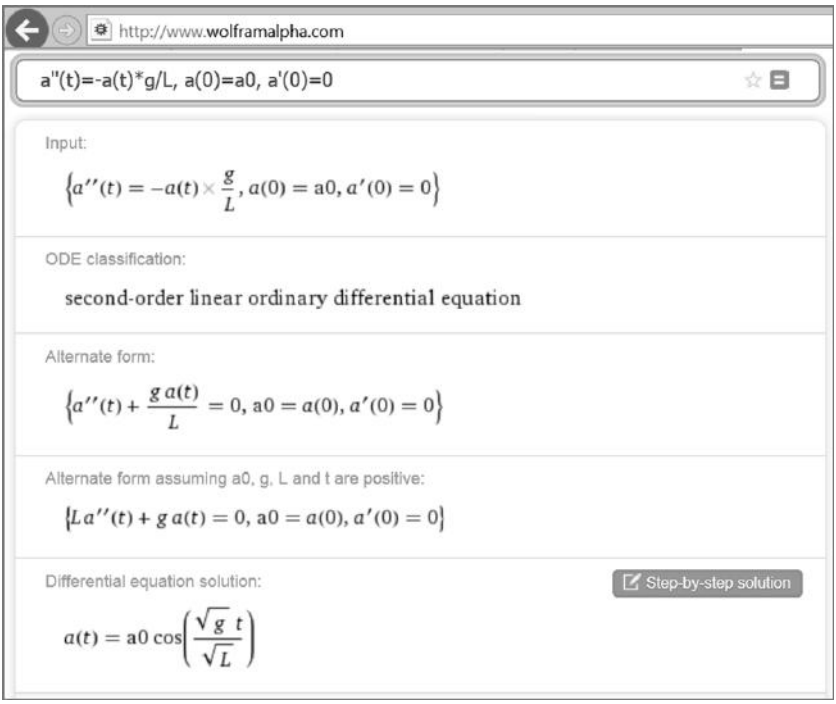


Рис. 1.1

Пример сайта с решением математических задач — показано нахождение аналитического решения обыкновенного дифференциального уравнения второго порядка с начальными условиями (уравнение математического маятника)

Третья зарисовка. В том же родном университете авторов на семинарские занятия по уравнениям математической физики один из студентов приносит ноут-

[2] В книге есть и другие «картинки», снятые с этого математического сайта фирмы Wolfram — см., например, рис. 4.22 на стр. 83 или рис. 9.24 на стр. 223. Это, а также авторские и другие Интернет-расчеты, позволили в название книги вставить «и Интернет».

бук и, получив в тетради в результате решения задачи формулу, задающую форму колебания струны (предварительно проверив правильность вычисления интегралов компьютерным интегрированием), тут же создает анимацию, визуализирующую эти колебания. Это позволяет прямо в процессе занятия усложнять задачу, снимать с нее некоторые ограничения. Компьютер заставляет менять содержание и методику даже в самой, наверное, консервативной вузовской учебной дисциплине — в математике, и мы уже отметили эту тенденцию во Введении.

Наши три зарисовки поведения современных студентов — это преамбула к основной теме первого этюда книги. А она касается намечающейся *революции* в содержании и технологии публикации научно-технических статей, справочников, монографий и учебников, содержащих *расчетные формулы*. И вообще *работы* с математическими формулами.

В чем суть этой революции?

Когда-то давно авторы сдавали в редакции журналов и книжные издательства свои рукописи статей и книг в виде машинописных текстов с формулами, вписанными в них от руки или впечатанными на пишущих машинках с латинским алфавитом[3]. Греческие же буквы приходилось почти всегда вписывать вручную. Вручную также размечался текст — курсив, полужирный, тире, дефис и т. д. и т. п. Правда, в какой-то период времени появились печатные машинки со сменными литерами (кириллица, латиница, греческие и спецсимволы), насаженными, например, на съемный поворотный шар (пример — машинка IBM), но их быстро вытеснили компьютеры с принтерами, способными печатать любой шрифт.

Еще раньше, до эры печатных машинок, в редакции сдавались для набора просто рукописные тексты[4], по которым в типографиях набирались так называемые гранки, с которыми далее работали авторы, редакторы и корректоры. От эпохи рукописного оформления научных работ в математике остались два важных обозначения[5]: «о» малое — $\overset{=}{o}(x)$, обозначающее бесконечно малую по сравнению с x величину, и «О» большое — $\underline{\underline{O}}(x)$, обозначающее величину того же порядка, что и x. И еще вместо дефиса ставили знак равно, что должно было отличать дефис (кое=где, когда=нибудь и т. д.) от тире.

В настоящее время в издательства для публикации передаются файлы с текстами, набранными, как правило, в Word'е, и формулами, набранными специальными компьютерными средствами (LaTeX или WordEquation, например) или представляющими из себя сканы (рисунки) программ, где эти формулы были созданы (пример — данная книга). Появились также статьи, целиком набранные в среде какой-либо математической программы с работающими формулами и сохраненные для публикации в виде PDF-файлов.

---

[3] «Лев Никулин <...> был фронтовым корреспондентом. А может быть, политработником. В оккупированной Германии проявлял интерес к бронзе, фарфору, наручным часам. Однако более всего хотелось ему иметь заграничную пишущую машинку. Шел он как-то раз по городу. Видит — разгромленная контора. Заглянул. На полу — шикарный ундервуд с развернутой кареткой. Тяжелый, из литого чугуна. Погрузил его Никулин в брезентовый мешок. Думает: «Шрифт в Москве поменяю с латинского на русский». В общем, таскал Лев Никулин этот мешок за собой. Месяца три надрывался. По ночам его караулил. Доставил в Москву. Обратился к механику. Тот говорит: — Это же машинка с еврейским шрифтом. Печатает справа налево. Так наказал политработника еврейский Бог.» Сергей Довлатов «Соло на ундервуде».

[4] Софья Андреевна по нескольку раз переписывала и правила романы своего мужа Льва Толстого перед сдачей их в типографию. Смех мешал наборщикам в типографии набирать повести Гоголя.

[5] Согласно редакционной разметке все рукописные строчные буквы подчеркивались двумя черточками сверху, а прописные — двумя черточками снизу.

Третья нарождающаяся революция[6] заключается в «возвращении к смыслам». «Ренессанс» тут просматривается в том, что в формулах «возрождается» их суть. А она в том, что формулы создаются и публикуется в статье или книге для того, чтобы по ним можно было *считать*. Это, кстати, касается графиков и таблиц. Тем более, что сейчас все чаще и чаще статьи и книги читаются и изучаются не с бумажного листа, а с экрана компьютера, планшета, смартфона, где этот расчет несложно реализовать.

Для чего это делается?

Во-первых, для того чтобы исключить опечатки и ошибки в формулах. Тексты книг и статей, как правило, тщательно вычитываются самими авторами, редакторами и корректорами (реальными и/или компьютерными). Формулы же, увы, обделены таким вниманием. Редакторы и корректоры обычно проверяют только стиль переменных и констант в них, особо не вникая в их суть. Проверяется, например, чтобы латинская буква была прописана курсивом, а аналогичная буква кириллицы — прямым шрифтом и т. д. На этот счет есть целый полиграфический стандарт, который сейчас несколько размывается [3]. Но по разным объективным причинам (человеческий фактор!) опечатки неистребимы. И если от текстовых опечаток урон не очень велик, то формульные опечатки могут приводить к серьезным негативным последствиям. Вот конкретный пример работы с опечатками в формулах. Одному из авторов попала на рецензию статья о теплопроводности водного раствора NaCl. В статье помимо текстовых описаний был набор формул и коэффициентов (так называемая формуляция) для расчета этой теплофизической величины в зависимости от давления, температуры и концентрации соли. Студентам было поручено оживить формулы по технологии Mathcad Calculation Server [20–22]. Когда это было сделано, оказалось, что формулы выдают не те ответы, что были приведены в статье для контроля.

Разбор формуляции показал, что некоторые формулы и коэффициенты в статье были набраны с... «очепятками». Когда опечатки были исправлены, все правильно заработало. Статья вышла не только без досадных ошибок[7], но и со ссылкой на живой расчет [23].

Во-вторых, тут уместно вспомнить и переиначить крылатую чеховскую фразу о том, что если на сцене висит ружье, то оно должно выстрелить: если в статье или книге помещена формула, то она должна считать. Читатели (те же студенты с двумя пальцами) должны иметь возможность непосредственно во время чтения изменить исходные данные и получить новый ответ. Это касается не только формул, но таблиц и графиков, заключающих в себя некую функциональную зависимость или корреляцию (см. этюд 14). Один из авторов по гранту РФФИ писал книгу о способах «оживления» справочников [24] и «оживил» справочную серию «Теплотехника и теплоэнергетика» [25]. Во время этой работы было найдено много ошибок и опечаток в формулах, которые были исправлены в очередном издании этой справочной серии. Кроме того, была зафиксирована такая интересная особенность — формула правильная, но выдает неверный или неполный ответ при ее реализации на компьютере. В чем тут дело?

Современные средства оживления формул такие, например, как математические пакеты Mathcad, Maple и SMath, работают с единицами измерения физических величин. Это очень удобно: ведется автоматический пересчет величин с разными единицами измерения при их вводе и выводе, а также контроль над соответствием размерностей. Калькуляторы, языки программирования и табличные процессоры взяли на себя счет. Теперь же современные программные средства делают за нас и пересчет единиц измерения. Но тут возникают некоторые нестыковки.

---

[6] Революций обычно бывает три. Последующие, как правило, уже не нумеруют.

[7] Опечатка в числе или формуле существенно опаснее, чем опечатка в тексте. В Интернете можно найти много описаний случаев, когда неверный счет и/или пересчет единиц измерения приводил к тяжёлым последствиям.

Как известно, формулы в справочниках бывают двух типов — *физические* и *эмпирические*. Первые оперируют физическим величинам, в них можно вставлять эти величины с любыми единицами измерения — примеры таких формул: $e = m \cdot c^2$, $F = m \cdot a$ и т. д. Соответствующие математические пакеты все правильно пересчитают или выдадут сообщение о несоответствии физических величин. Второй тип формул (а они, как правило, возникают в процессе статистической обработки опытных (эмпирика!) данных — см., например, этюд 14) требуют, чтобы физические величины вводились в формулу с определенными единицами измерения. Самая, наверное, простая и известная эмпирическая формула — это формула, связывающая рост среднестатистического человека с его весом (массой): рост человека в сантиметрах — это его вес в килограммах плюс 100. Вот так будет выглядеть эта формула в среде SMath (русский клон пакета Mathcad — [26]) с привлечением механизма размерных величин (рис. 1.2).

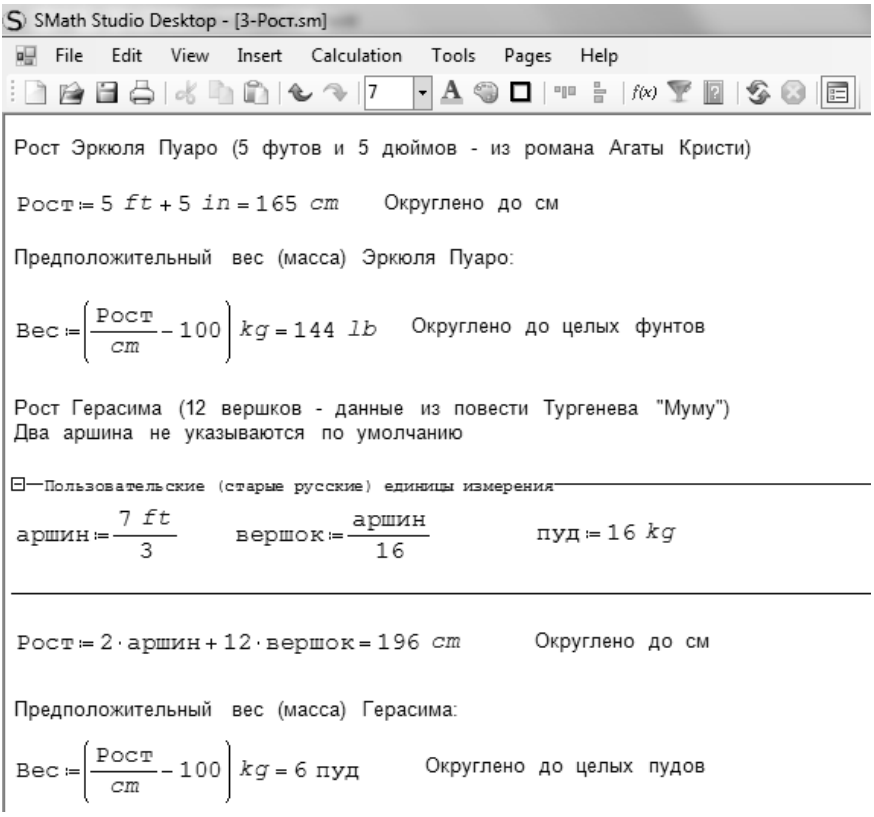


Рис. 1.2
Работа с простейшей эмпирической формулой

Примечание. На рисунке 1.2 вывод результатов делается с *округлением*, хотя сами переменные содержат до 15 знаков в мантиссе. Эта деталь вскрывает важный момент процесса оживления формул. Дело в том, что в бумажных справочниках, монографиях, задачниках и учебниках расчет по связанным формулам ведется с округлением промежуточных результатов и с переносом соответствующих округленных величин в последующие формулы. При расчете же на компьютере в переменных сохраняется максимальная точность. Расчет при этом будет несколько отличаться от того, что приведен в бумажном источнике. См. также этот вычислительный нюанс в расчете условного расхода топлива, показанного на рисунке 1.4 ниже.

Но иногда для повышения точности расчетов необходимо числа… округлять. Вот интересный пример, иллюстрирующий этот тезис.

При расчетах может возникнуть ошибка (погрешность), видимая только на 15-м знаке в мантиссе. Эта особенность Mathcad и других вычислительных систем с плавающей десятичной точкой проявляется в том, что 1 + 1 не всегда равно точно двум, а 2.000000000000001 или 1,99999999999999. На рисунке 1.3 эта особенность Mathcad мешает правильно решать задачу о построении двух числовых последовательностей *x* и *y* рекуррентным способом (см. этюд 10). Из рисунка 1.3 видно, что ошибка (сравните содержимое векторов $\mathbf{x}_1$ и **x**) «слегка» проявляется на 12-м шаге (1-st Error), а на 3000-м шаге становится катастрофической (last Error): 0.26… (вектор $\mathbf{x}_1$) вместо ожидаемого — 0.1… (вектор **x**): это вам не 1,9999999999999 вместо 2. Исправить эту ошибку можно, вставив в расчет встроенную в Mathcad функцию округления числа round, которая будет «глушить шум», связанный с тем, что в Mathcad нет целочисленной арифметики, а есть только арифметика с плавающей точкой.

$n := 3000 \quad i := 1 .. n$

$x1_0 := -0.1 \quad y1_0 := 0 \quad \begin{pmatrix} x1_i \\ y1_i \end{pmatrix} := \begin{pmatrix} 1 - y1_{i-1} + |x1_{i-1}| \\ x1_{i-1} \end{pmatrix}$

$x_0 := -0.1 \quad y_0 := 0 \quad \begin{pmatrix} x_i \\ y_i \end{pmatrix} := \mathrm{round}\left[\begin{pmatrix} 1 - y_{i-1} + |x_{i-1}| \\ x_{i-1} \end{pmatrix}, 14\right]$

x1 =

	0
11	-0.200000000000000
12	0.700000000000001
13	1.90000000000000
14	2.20000000000000
15	1.30000000000000
16	0.099999999999999
17	-0.200000000000000
18	1.10000000000000
19	2.30000000000000
2994	-2.26792947738698
2995	-1.37016420130499
2996	4.63809367869198
2997	7.00825787999697
2998	3.37016420130499
2999	-2.63809367869198
3000	0.267929477386982

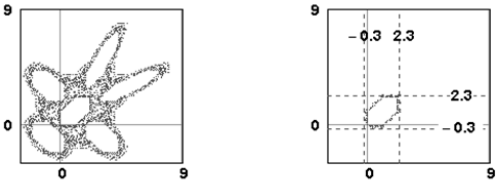
1-st Error

2-d Error

Last Error

x =

	0
11	-0.200000000000000
12	0.700000000000000
13	1.90000000000000
14	2.20000000000000
15	1.30000000000000
16	0.100000000000000
17	-0.200000000000000
18	1.10000000000000
19	2.30000000000000
2994	-0.100000000000000
2995	0.100000000000000
2996	1.20000000000000
2997	2.10000000000000
2998	1.90000000000000
2999	0.800000000000000
3000	-0.100000000000000



max(x1) = 7.906
min(x1) = –2.832
max(y1) = 7.906
min(y1) = –2.832
Calculation with Error

max(x) = 2.3
min(x) = –0.3
max(y) = 2.3
min(y) = –0.3
Calculation without Error

Рис. 1.3
Накопление ошибки в вычислениях

Примечание. «Шум» арифметики плавающей точки может быть красивым — см. левый график на рис. 1.3. Это так называемый Gingerbreadman — пряничный человечек, свалившийся набок. Правый график — это «портрет» наших двух векторов без ошибки. Можно менять значения первых элементов векторов **x** и **y** (а можно добавить третий вектор и смотреть «красоту» уже в объеме, а не на плоскости), а также значения коэф-

фициентов рекуррентных соотношений и наблюдать «картинки» с ошибкой и без ошибки. Более подробно эта особенность вычислений описана в [27].

Эмпирическая формула при вводе ее в компьютер требует, чтобы переменные были поделены на нужные единицы измерения, а итоговый ответ был умножен на соответствующую единицу измерения. Без такого «утяжеления» формула не будет правильно работать.

Как правило, в справочниках базовые единицы измерения отмечаются в описаниях их физических формул. Это делается в видах ручного счета или счета на компьютерах без привлечения инструментария физических величин. Тривиальный пример: $F = a \cdot m$, где F — это сила (ньютоны), a — ускорение (m/s^2), а m — масса (kg). Для современных компьютерных средств расчета такое уточнение излишне, т. к. в случае небазовых единиц измерения будут автоматически сделаны соответствующие пересчеты.

Нельзя забывать о ряде физических задач, чьё решение в размерных величинах приводит к числам, выходящим за допустимый диапазон вычислений на компьютере. Задачи также могут одновременно содержать как очень большие, так и очень малые параметры. А тогда мы можем получить некорректные округления, связанные с одновременной обработкой больших и малых чисел. Алгоритмы решения уравнений математических моделей также могут оказаться непригодными, если уравнения модели содержат переменные, значения которых отличаются по порядкам. Так, погрешности при определении параметров, порядки которых велики, могут быть не значимы для них самих. В то же время, эти погрешности будут сильно искажать значения параметров малых порядков. В таких случаях перед началом численного решения следует осуществить переход к вычисляемым величинам с разумным порядком малости или роста, а также по возможности сократить число арифметических операций, накапливающих ошибки при работе расчетной программы. В задачах динамики существенным является также фактор времени: многие процессы настолько быстротечны, что приходится вводить масштабированное, так называемое «медленное» время.

Все указанные факторы приводят к необходимости предварительной специальной замены переменных — к *обезразмериванию физических величин*. В результате этой операции все переменные математической модели будут иметь практически одинаковый порядок. Сами уравнения в результате исключения из них «лишних» констант и величин, как правило, сильно упрощаются, при этом выделяется существенный набор параметров, фактически определяющих рассматриваемый процесс. Идея обезразмеривания заключается в переходе от абсолютных значений — расстояний, скоростей, времен, теплопроводностей, напряженностей и т. п. — к относительным. Отношения берутся к величинам, характерным, для данной физической модели. Обычно это: длина, время, энергетические или механические характеристики.

Многие физические константы по определению являются безразмерными. Это, например, число Маха (критерий сжимаемости), число Рейнольдса (критерий вязкости), коэффициент Пуассона (упругие свойства материала), число Прандтля (связь вязкости и теплопроводности) и др. Угол, измеренный в радианах, — это тоже безразмерная величина, рассчитываемая как отношение одной длины (дуги окружности) к другой длине (радиусу окружности). Рассмотрим процедуру обезразмеривания физических переменных на примере уравнения поперечного движения стержня (балки). Вывод уравнения поперечных колебаний стержня дан [28] в предположении, что ось ox проходит по оси симметрии балки, длина оси стержня не меняется при изгибе, то есть, $dl = dx$, а деформации на малом участке длины тоже малы. Вывод уравнения опирается на закон Гука, и уравнение имеет

вид $\frac{\partial^2 U}{\partial t^2}+\frac{EJ}{PS}\cdot\frac{\partial^4 U}{\partial X^4}=\frac{Q(t,X)}{PSL}$, где $y=U(t,X)$ (м) — прогиб стержня в точке с координатой x (*м*) в момент времени t (с), $J=\frac{BH^3}{12}$ (м^4) — момент инерции прямоугольной балки относительно оси ox, $S=BH$ (м^2) — площадь сечения прямоугольной балки, $\frac{Q(t,X)}{PSL}$ (м/с^2) — внешняя сила, отнесённая к массе балки. «Характерные» величины: L^* (м) — длина балки, P^* (кг/м^3) — плотность металла, E^* (н/м^2) — модуль Юнга (упругости) металла, C^* (м/с) — скорость распространения волны растяжения-сжатия в металле (является физической характеристикой среды распространения волны). Например, в случае стальной балки из незакалённой стали имеем: $P^*=P=7800$ кг/м^3, $E^*=E=2\cdot10^{11}$ н/м^2, $C^*=C=5870$ м/с, $\nu=0.3$ — коэффициент Пуассона. Безразмерные величины вводятся следующим образом: $\rho=\frac{P}{P^*}$ — безразмерная плотность, $l=\frac{L}{L^*}$, $h=\frac{H}{L^*}$, $b=\frac{B}{L^*}$ — безразмерные длина, высота и ширина балки соответственно, $x=\frac{X}{L^*}$, $u=\frac{U}{L^*}$ — безразмерные координата и прогиб соответственно, $\varepsilon=\frac{E}{E^*}$ — безразмерный модуль Юнга, $\tau=\frac{t\cdot C^*}{L^*}$ — безразмерное («медленное») время. После замены переменных получаем содержащее единственный параметр уравнение $\frac{\partial^2 u}{\partial\tau^2}+a^2\cdot\frac{\partial^4 u}{\partial x^4}=q$ относительно безразмерных величин, в котором $a^2=\frac{H^2}{12L^2\cdot d}$ и $q=\frac{Q\left(\frac{\tau L^*}{C^*},xL^*\right)}{BHEd}$ при $d=\frac{1-\nu}{(1+\nu)(1-2\nu)}$.

Можно процедуру обезразмеривания провести и для задачи колебания груза на пружине. Пусть в момент времени t_0 имеем: начальное растяжение x_0 (м), k (Н/м), частоту υ_0 (Гц) и массу m (кг). Здесь характерное время будет порядка $\sqrt{\frac{m}{k}}$, и если пружина растягивается мгновенно, то шаг вычислений будет настолько мал, что сравнится с машинной точностью, при этом характерные значения x могут быть настолько велики, что тоже приведёт к потере точности вычислений. Обезразмерим величины, введя новые переменные $\tilde{x}=x/x_0$, $\tilde{t}=t\sqrt{k/m}$, $\tilde{\upsilon}=\upsilon\sqrt{m/k}/x_0$. Система уравнений в новых переменных будет иметь вид: $\begin{cases}\dot{\tilde{x}}=\tilde{\upsilon}\\ \dot{\tilde{\upsilon}}=-\tilde{x}\end{cases}$. Здесь характерные значения всех переменных сравнимы с единицей, а поэтому систему можно с требуемой точностью решить численно. Кроме этого, решив один раз эту систему, мы получим решение целого класса задач, для различных величин x_0, k и m.

Как задачу вернуть обратно к размерности? Сделать обратную замену переменных.

Итак, обезразмеривание заключается в том, чтобы от абсолютных физических единиц перейти к относительным единицам, естественным именно для данного физического процесса. Кроме того, переход к безразмерным критериям был обусловлен положениями теории подобия. Конкретный пример. Переход от ламинарного течения к турбулентному происходит при определенном значении безразмер-

ного критерия Рейнольдса, учитывающего скорость потока, геометрические размеры и вязкость среды. Это справедливо и для трубы большого диаметра, по которой прокачивается газ, и для капилляра с протекающей вязкой жидкостью.

С другой стороны, обезразмеривание расчетных величин приводило и приводит к тому, что, например, «Рейнольдса можно сложить с Прандтлем» и не получить при этом компьютерного сообщения об ошибке. Но вязкость с теплопроводностью сложить никак нельзя, вернее, нельзя сложить на компьютере с контролем размерностей.

Эта и другие причины обусловливают возврат к размерным физическим величинам, отход от их «теней» — от безликих, пардон, безразмерных величин, ущербность которых люди давно чувствовали, прибавляя им псевдоединицы измерения — градусы, например. И современные компьютерные средства позволяют осуществить такой «ренессанс» размерностей.

«Обезразмериванием в квадрате» можно считать использование децибелов. Бел — это десятичный логарифм отношения двух одноразмерных физических величин (раз), а децибел — это, соответственно, одна десятая бела (два — «квадрат»). Измеряя величину в децибелах, мы тем самым выстраиваем некую логарифмическую шкалу значений физической величины. Обычно децибелами измеряют интенсивность звука. Интенсивность звуковых волн сравнивается с наименьшей интенсивностью звука I_0, слышимого человеческим ухом (обычно $I_0 = 0.01$ Вт/м^2). Можно рассчитать непосредственно отношение измеряемой интенсивности звука I к наименьшей, однако диапазон значений этого отношения столь широк, что вызывает неудобства, которые устраняются введением логарифмической шкалы с десятичным множителем деци — $0.1 \cdot \lg(I/I_0)$. Очень громкий звук отбойного молотка характеризуется уровнем 80 дБ, разговор в помещении — 60 дБ, а едва слышимый звук вроде шелеста листьев — примерно 10 дБ. Децибелы, как и прочие «обезразмеренные» величины, были введены в том числе и для облегчения счета, но сейчас, повторяем, облегчать счет на компьютере не имеет особого смысла.

В свое время при переходе на СИ издательства и редакции журналов перестали принимать рукописи, где использовались иные, но, тем не менее, до сих пор широко используемые единицы измерения. Особенно больно эта ломка шла в научно-технических кругах, где в ходу фунты, футы и прочие «заокеанские» единицы. Да и у нас в России и во всей Европе есть свои «футы-фунты»: атмосферное давление, к примеру, мы никогда не научимся оценивать в паскалях, вернее, в гектопаскалях, а всегда будем тут оперировать миллиметрами ртутного столба. Сейчас издательства отходят от жесткого правила, предписывающего использовать в статьях и книгах только единицы СИ.

В журналах и книгах публикуются формулы, которые условно можно назвать *псевдоэмпирическими*. Эти формулы физические по своей сути, но эмпирические по способу компьютерного счета по ним, требующего ввода в формулы единиц измерения (см. рис. 1.2). Конкретный пример из теплоэнергетики. Если известен КПД (η) электростанции, то удельный расход условного топлива (b) можно подсчитать по такой простой формуле: $b = 123/\eta$: на электростанции с КПД 34% сжигается 362 (123/0.34) грамма условного топлива на выработку одного киловатт-часа электроэнергии. Но эта простенькая формула, реализованная в среде Mathcad, будет выдавать неполный ответ — ответ, лишенный единиц измерения (см. первую строку расчета, показанного на рис. 1.4). В чем тут дело? А дело тут в том, что эту формулу когда-то давно адаптировали для ручного счета и для счета по программам без единиц измерения — для языков программирования и электронных таблиц. Но если вспомнить, что такое условное топливо и какова его теплотворная способность $Q_{ут}$, то можно восстановить исходную формулу $b = 1 / (Q_{ут} \cdot \eta)$ и без проблем использовать ее в Mathcad или SMath (рис. 1.4). Ответ при этом не только будет с единицами измерения, но и окажется более точным. Работники электростанции, сделав расчет по исходной, а не по адаптированной

формуле, могут констатировать, что у них сжигается почти на полграмма топлива меньше. А в масштабах всей страны это огромная экономия (энергосбережение).

$$\eta := 34\% \qquad b := \frac{123}{\eta} = 361.765$$

Правильное значение, но без единиц измерения

$$Q_{ут} := 7000\ \frac{kcal}{kg}$$

Теплотворная способность условного топлива

$$\frac{1}{Q_{ут}} = 122.835\ \frac{gm}{kW \cdot hr}$$

$$b := \frac{1}{Q_{ут} \cdot \eta} = 361.28\ \frac{gm}{kW \cdot hr}$$

Правильный ответ с единицами

Рис. 1.4
Работа с псевдоэмпирической формулой

Случай, отображенный на рисунке 1.4, довольно тривиальный. Отсутствие единиц измерения в ответе, показанном в первой строке на рисунке 1.4, указывает на псевдоэмпиричность этой формулы, если так можно выразиться. А вот в следующем расчетном случае такая псевдоэмпиричность не столь очевидна. Это формула пересчета видов концентрации, взятая из справочника и вставленная в среду Mathcad, выдает ответ с правильными единицами измерений и, что самое неприятное, с правдоподобным ответом.

Задача: известна *молярность* (отношение количества растворенного вещества к объему раствора) водного раствора NaCl — найти его *моляльность* (отношение количества растворенного вещества к массе растворителя). Формулу для пересчета несложно найти в многочисленных справочниках по химии и в Интернете (рис. 1.5).

Из таблицы, показанной на рисунке 1.5, можно скопировать (мы об этом копировании еще упомянем ниже) формулу для расчета моляльности (переменная L) по молярности (переменная M). Для этого нужно еще ввести дополнительные данные: значения молярной массы NaCl (переменная wp[8]), а также плотности раствора (переменная q[9]). Скопировать, ввести и получить... неверный ответ — см. вторую строку расчета, показанного на рисунке 1.6.

Примечание. В расчете на рисунках 1.4, 1.6 и 1.8 единица массы грамм прописана как gm вместо более привычного g. Это связано с тем, что идентификатор g зарезервирован под ускорение свободного падения. Это и физическая константа и единица ускорения. В среде Mathcad Prime они по умолчанию отличаются цветом — синий у константы и зеленый у единицы измерения.

[8] Эту величину можно уточнить на авторском расчетном сайте по адресу **http://twt.mpei.ac.ru/MCS/Worksheets/Chem/MW.xmcd**.

[9] Для этого опять есть авторский сайт с адресом **http://twt.mpei.ac.ru/TTHB/1/Water-chem/C-NaCl.html**, где плотность раствора определяется интерполяцией табличных данных, взятых из тех же химических справочников. Получаются эмпирические формулы, о которых мы упоминали выше.

http://erichware.info/sposob/ximia/ximikonc.htm

	0 < K < 100 % процент по массе	T титр	L моляльность	M молярность
M =	$\frac{10 * q * K}{wp}$	$\frac{1000 * T}{wp}$	$\frac{1000 * q * L}{1000 + wp * L}$	$\frac{1000 * yp}{v}$
L =	$\frac{1000 * K}{wp * (100 - K)}$	$\frac{1000 * T}{wp * (q - T)}$	$\frac{1000 * yp}{mb}$	$\frac{1000 * M}{1000 * q - wp * M}$
T =	$\frac{q * K}{100}$	$\frac{mp}{v}$	$\frac{q * wp * L}{1000 + wp * L}$	$\frac{wp * M}{1000}$
K =	$\frac{mp * 100\,\%}{mb + mp}$	$\frac{100 * T}{q}$	$\frac{100 * wp * L}{1000 + wp * L}$	$\frac{wp * M}{10 * q}$

wp – молярная масса растворенного вещества (г/моль)
mb – масса растворителя (г) mp – масса растворенного вещества (г)
yp – количество растворенного вещества (моль) v – объем раствора (см^3)
q – плотность раствора (г/см^3)

Рис. 1.5
Формулы пересчета видов коцентрации

$M := 2\ \frac{mol}{L}$ $q := 1.076\ \frac{gm}{cm^3}$ $wp := 58.44\ \frac{gm}{mol}$ Исходные данные

$L := \frac{1000 \cdot M}{1000\ q - wp \cdot M} = 1.859\ \frac{mol}{kg}$ Похожий, но неверный ответ

$L := \frac{1000\ \frac{M}{mol \cdot L^{-1}}}{1000\ \frac{q}{gm \cdot cm^{-3}} - \frac{wp}{gm \cdot mol^{-1}} \cdot \frac{M}{mol \cdot L^{-1}}} \cdot \frac{mol}{kg} = 2.085\ \frac{mol}{kg}$ Эмпирическая формула

$L := \frac{M}{q - wp \cdot M} = 2.085\ \frac{mol}{kg}$ Правильный ответ - физическая формула

Рис. 1.6
Работа с псевдоэмпирической формулой пересчета видов концентрации

Дело тут в том, что формулы, приведенные в таблице на рис. 1.5, — это все те же псевдоэмперические формулы, приспособленные для ручного расчета с конкретными небазовыми единицами плотности (г/см^3, а не кг/м^3, например), молярной массы (г/моль, а не кг/моль), массы (г, а не кг), объема (см^3, а не м^3) и т. д. Формулу для определения моляльности по молярности можно, конечно, ввести в расчет по правилам работы с эмпирическими формулами (см. выше), добавив нужные единицы измерения (см. третью строку на рис. 1.6), и получить правильный ответ. Но лучше тут вернуться к исходной физической формуле, убрав пересчетные коэффициенты 1000 (см. последнюю строку на рис. 1.6).

Многие химики, когда им нужно пересчитать концентрацию из одного вида в другой, стараются не использовать готовые формулы — такие, например, какие приведены в таблице на рисунке 1.5. Они пытаются при решении подобных задач

составить некие пропорции и получить ответ уже по ним. Можно также составить и решить алгебраическое уравнение, связывающее, например, количество или массу растворенного вещества при разных видах выражения концентрации. В среде Mathcad есть средства символьной математики, позволяющие оперировать не численными значениями переменных, а самими символами переменных. На рисунке 1.7 показано, как в среде Mathcad можно выводить формулы для пересчета видов концентрации растворов, решая алгебраические уравнения. Это, можно сказать, и есть настоящее возрождение (ренессанс!) формул в справочниках: приведены не только сами формулы, но и уравнения (физические законы), из которых они вытекают.

В этюдах 4 и 9 показаны и другие примеры, когда поиск в справочниках заменяется составлением и решением уравнения — см. рисунок 4.14 (формула золотого сечения) и рисунок 9.24 (формула колебания маятника).

Расчет моляльности по молярности (from molarity to molality)

$$\mathrm{L} := \mathrm{M}\cdot v = \mathrm{L}\cdot \left| \begin{array}{l} \text{“Масса растворителя”} \\ \left| \begin{array}{l} \text{“Масса раствора”} \\ v\cdot q \end{array} \right| - \left| \begin{array}{l} \text{“Масса растворенного вещества”} \\ \mathrm{M}\cdot v\cdot wp \end{array} \right| \end{array} \right| \xrightarrow{solve,\,L} \frac{M}{q - M\cdot wp}$$

Расчет молярности по моляльности (from molality to molarity)

$$\mathrm{M} := \mathrm{M}\cdot v = \mathrm{L}\cdot \left| \begin{array}{l} \text{“Solvent mass”} \\ \left| \begin{array}{l} \text{“Solution mass”} \\ v\cdot q \end{array} \right| - \left| \begin{array}{l} \text{“Solute mass”} \\ \mathrm{M}\cdot v\cdot wp \end{array} \right| \end{array} \right| \xrightarrow{solve,\,\mathrm{M}} \frac{L\cdot q}{L\cdot wp + 1}$$

Расчет моляльности по титру (from titre to molality)

$$\mathrm{L} := \frac{\mathrm{T}\cdot v}{wp} = \mathrm{L}\cdot \left| \begin{array}{l} \text{“Solvent mass”} \\ \left| \begin{array}{l} \text{“Solution mass”} \\ v\cdot q \end{array} \right| - \left| \begin{array}{l} \text{“Solute mass”} \\ \mathrm{T}\cdot v \end{array} \right| \end{array} \right| \xrightarrow{solve,\,\mathrm{L}} \frac{\mathrm{T}}{wp\cdot(q - \mathrm{T})}$$

Расчет титра по моляльности (from molality to titre)

$$\mathrm{T} := \frac{\mathrm{T}\cdot v}{wp} = \mathrm{L}\cdot \left| \begin{array}{l} \text{“Solvent mass”} \\ \left| \begin{array}{l} \text{“Solution mass”} \\ v\cdot q \end{array} \right| - \left| \begin{array}{l} \text{“Solute mass”} \\ \mathrm{T}\cdot v \end{array} \right| \end{array} \right| \xrightarrow{solve,\,\mathrm{T}} \frac{L\cdot q\cdot wp}{L\cdot wp + 1}$$

Рис. 1.7

Генерация формул пересчета видов концентрации

На рисунке 1.8 показано, как может быть решена наша задача о пересчете моляльности по молярности с «возрожденной» формулой, базирующейся на физическом законе сохранения количества вещества, расписанного в выкладках, отображенных на рисунке 1.7.

На рисунке 1.9 показан авторский сайт Интернета для пересчета видов концентраций с показом использованной формулы. Как это делалось, можно узнать в этюде 5. Сайт создавался довольно давно. Теперь же его стоит модернизировать — показывать не только саму формулу, но и уравнение, из которого она получается (см. рис. 1.8).

Расчет моляльности водного раствора NaCl по его молярности

$$M := 2\ \frac{mol}{L} \qquad q := 1.076\ \frac{gm}{cm^3} \qquad wp := 58.44\ \frac{gm}{mol}$$ Исходные данные

$$\mathrm{clear}_{sym}(M, q, wp)$$

$$L := M \cdot v = L \cdot (v \cdot q - M \cdot v \cdot wp) \xrightarrow{solve,\,L} \frac{M}{q - M \cdot wp} = 2.085\ \frac{mol}{kg}$$ Ответ

Рис. 1.8
Расчет моляльности по «возрожденной» формуле

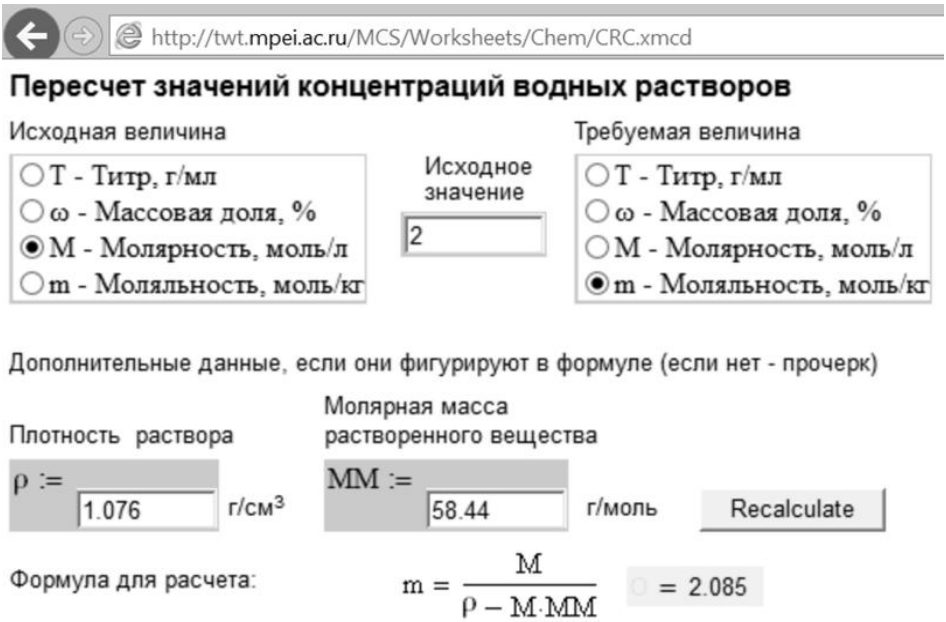


Рис. 1.9
Сайт для пересчета видов концентрации

Псевдоэмпирических формул, подобных тем, что показаны на рисунках выше, в справочной литературе огромное количество, и это приводит к сбоям и ошибкам при необдуманном их использовании. Старая услуга по упрощению формул в настоящее время превратилась в «медвежью»[10].

[10] При наборе текста этой книги планшет подсказывал полное написание очередного слова после набора его первых трех-четырех букв. Эта услуга компьютера иногда тоже оказывалась медвежьей — в текст вставлялось не то слово, какое нужно. Если читатель увидит в этой книге опечатку, то это может быть опечатка и не самих авторов, а компьютера. Планшет, например, вместо Пиковой дамы, пожелал вставить Поисковую даму, а вместо Мещанина во дворянстве — Механика во дворянстве. Такое своеобразное чувство юмора у компьютера.

Примечание. В таблице на рис. 1.5 переменная K отмечена как «процент по массе», но не уточняется, по какой массе — по массе раствора или по массе растворителя. Из анализируемой таблицы можно понять, что имеется в виду масса именно раствора, а не растворителя. Но в других случаях такое уточнение отсутствует и это может приводить к существенным ошибкам в пересчетах. Так, например, в популярном справочнике по аналитической химии Ю. Ю. Лурье [29] растворимость некоторых соединений в воде дается как отношение массы растворенного вещества к массе растворителя. На соответствующем авторском расчетном сайте **http://twt.mpei.ac.ru/MAS/Worksheets/Chem/solutions.html** эта величина дается в двух единицах (отношение к массе растворителя и к массе раствора), что должно исключить ошибки. И подобных умолчаний в справочниках можно найти много. В этом заключается одна из опасностей при работе с безразмерными величинами, и мы еще упомянем об этом в этюде 13, обсчитывая треугольник. Одно из умолчаний связано с температурой. Типичный пример: дана температура на входе в теплообменник 25°C, найти температуру на его выходе, если теплоноситель нагревается на 5°C. Ответ 30°C. Но если эту задачу записать в среде Mathcad, то ответ будет парадоксальным 25°C + + 5°C = 303.15°C. Все встает на свои места, если принять во внимание, что 5°C равно 5 кельвинам, но 25°C равно 298.15 кельвинам. Человек, увы, далеко не всегда, но все-таки понимает такое умолчание, пронизывающее многие справочники, а вот компьютер — нет. В подобных расчетах также дополнительно необходимо учитывать, по какой температурной шкале дана температура — 1968 или 1990 гг. [2, 30].

За простой формулой может скрываться довольно сложный алгоритм. Конкретный пример из термодинамики: расчет термодинамического КПД цикла двигателя внутреннего сгорания — цикла Отто.

Он состоит (в идеальной версии) из двух изохор (сгорание топливно-воздушной смеси в цилиндре двигателя и выхлоп отработавших газов[11]) и двух изоэнтроп[12] (сжатие топливно-воздушной смеси в цилиндре и расширение газов при рабочем ходе поршня) — см. рисунок 1.14. В справочниках и учебниках по термодинамике можно найти формулу, по которой рассчитывается термический КПД цикла Отто. На рисунке 1.10 отображен фрагмент авторской интернет-страницы с этой формулой для интерактивного ее использования: можно поменять значение степени сжатия r (отношения максимального объема к минимальному объему цилиндра двигателя внутреннего сгорания), C_p и C_v (изобарная и изохорная) теплоемкости рабочего тела цикла Отто, привязанные к универсальной газовой постоянной R), и получить новое значение η_t с фиксацией точки на графике зависимости КПД от степени сжатия.

Подобные формулы для других циклов двигателей внутреннего сгорания (Дизеля, Брайтона, Тринклера, Калины и др.) можно найти в Интернете (см., например, **http://en.wikipedia.org/wiki/Thermodynamic_cycle**).

Откуда взялась формула, показанная на рисунке 1.10? На рисунке 1.11 представлен расчет параметров газа в четырех узловых точках цикла Отто с опорой на базовые понятия (определения) термодинамики: уравнение Клапейрона — Менделеева ($pv = RT$), энтропия S и внутренняя энергия U.

Удельная энтропия — это сумма двух интегралов: интеграла по температуре от T_0 (тут можно взять любое значение[13]) до T (температура рабочего тела в рассматриваемой точке) и интеграла по давлению от p_0 (тут опять же можно взять любое значение) до p (давление в рассматриваемой точке). Удельная внутренняя энергия — это интеграл удельной изохорной теплоемкости по температуре от T_0 до T.

[11] Изохора отображает процесс с постоянным объемом. В цикле Дизеля сгорание топливно-воздушной смеси идет при постоянном давлении (изобара).

[12] Процесс с постоянной энтропией.

[13] Отсюда, кстати, и идут расхождения в данных по термодинамическим свойствам веществ в различных справочниках. Но сама энтропия нам не интересна. Важно знать, как она меняется.

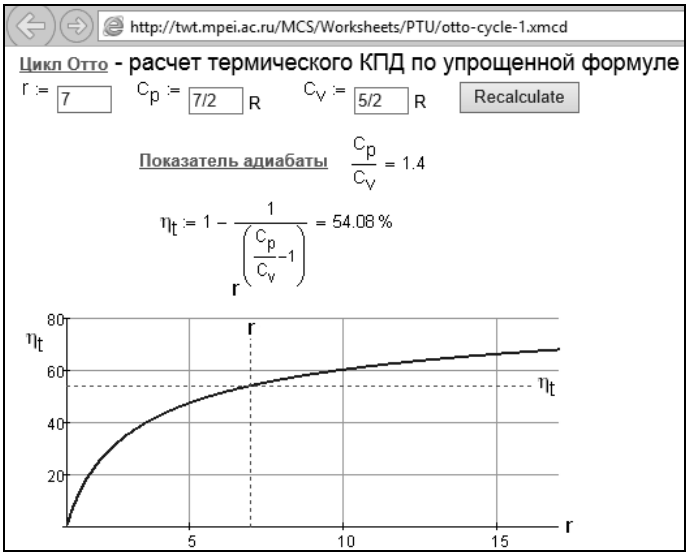


Рис. 1.10
Термический КПД цикла Отто

В расчете, показанном на рисунке 1.11, в точках *2* (окончание процесса сжатия топливно-бензиновой смеси) и *4* (окончание процесса рабочего хода поршня), решается уравнение, в котором неизвестными величинами являются пределы интегрирования — значения температуры и давления. В конце расчета, показанного на рисунке 1.11, выдаются два ответа: полученные через поточечный расчет (он в рамке) и по формуле, полученной в результате предварительного символьного интегрирования. Оба ответа совпали.

Если подынтегральное выражение содержит постоянный сомножитель — константу (а в нашем расчете принято, что C_p и C_v не зависят от температуры), то эту константу можно вынести за знак интеграла и взять его без константы, а потом на нее умножить. На рисунках 1.12 и 1.13 показано, как это сделал символьный процессор Mathcad.

Интересен расчет удельной энтропии с точки зрения использования единиц измерения — температуры и давления. Символьный процессор Mathcad выдал ответ, содержащий разность логарифмов размерных величин (давления), что привело к ошибке. Чтобы эту формулу использовать для численных расчетов, нужно вспомнить, что разность логарифмов равна логарифму дроби (см. рис. 1.12), и подправить формулу.

Если в расчете, показанном на рисунке 1.11, продолжить преобразования и упрощения типа тех, которые представлены на рисунках 1.12 и 1.13, то мы, в конце концов, получим формулу, показанную на рисунке 1.10.

Но следует помнить о том, что удельная изобарная и изохорная теплоемкости реального рабочего тела зависят от температуры и давления, а сам состав рабочего тела в реальном тепловом двигателе меняется. Поэтому расчет, показанный на рисунке 1.11, хотя и сложен, но более интересен в плане его совершенствования и приближения к реальному термодинамическому процессу.

Расчет цикла Отто

$T_0 := 1\ K \qquad p_0 := 1\ Pa$

Исходные данные:

$p_1 := 1\ atm \qquad T_1 := 300\ K \qquad n := 7 \cdot \frac{m^3}{m^3} \qquad T_3 := 1200\ K \qquad c_p := \frac{7}{2} \cdot R \qquad c_v := \frac{5}{2} \cdot R$

Point 1 $\quad v_1 := \frac{R \cdot T_1}{p_1} \qquad s_1 := \int_{T_0}^{T_1} \frac{c_p}{T}\,dT - R \cdot \int_{p_0}^{p_1} \frac{1}{p}\,dp \qquad u_1 := \int_{T_0}^{T_1} c_v\,dT = \left(6.215 \cdot 10^3\right) \frac{J}{mol}$

Point 2 $\quad v_2 := \frac{v_1}{n} \qquad s_2 := s_1$ Кривая 1-2 - изоэнтропа

Solve

Первое приближение: $p_2 := n \cdot p_1 \qquad T_2 := T_1 + 200\ K$

Система Уравнений: $p_2 \cdot v_2 = R \cdot T_2 \qquad s_2 = \int_{T_0}^{T_2} \frac{c_p}{T}\,dT - R \cdot \int_{p_0}^{p_2} \frac{1}{p}\,dp \qquad u_2 := \int_{T_0}^{T_2} c_v\,dT = 1.356 \cdot 10^4 \frac{J}{mol}$

Решатель: $\begin{bmatrix} p_2 \\ T_2 \end{bmatrix} := \mathbf{find}\left(p_2, T_2\right)$

Point 3 $\quad v_3 := v_2$ Кривая 2-3 - изохора $\quad p_3 := \frac{R \cdot T_3}{v_3}$

$s_3 := \int_{T_0}^{T_3} \frac{c_p}{T}\,dT - R \cdot \int_{p_0}^{p_3} \frac{1}{p}\,dp \qquad u_3 := \int_{T_0}^{T_3} c_v\,dT = \left(2.492 \cdot 10^4\right) \frac{J}{mol}$

Point 4 $\quad v_4 := v_1 \qquad s_4 := s_3$

Solve

$p_4 := p_2 \qquad T_4 := T_1$

$p_4 \cdot v_4 = R \cdot T_4 \qquad s_4 = \int_{T_0}^{T_4} \frac{c_p}{T}\,dT - R \cdot \int_{p_0}^{p_4} \frac{1}{p}\,dp \qquad u_4 := \int_{T_0}^{T_4} c_v\,dT = \left(1.143 \cdot 10^4\right) \frac{J}{mol}$

$\begin{bmatrix} p_4 \\ T_4 \end{bmatrix} := \mathbf{find}\left(p_4, T_4\right)$

$q_1 := u_3 - u_2 \qquad q_2 := u_4 - u_1 \qquad \boxed{\eta_t := \frac{q_1 - q_2}{q_1} = 0.541}$

Расчет по формуле

$1 - \frac{1}{n^{\frac{c_p}{c_v} - 1}} = 0.541$

Рис. 1.11

«Поточечный» расчет термического КПД цикла Отто

$$\int_{t_0}^{t} \frac{c_p}{T}\,\mathrm{d}T - R \int_{p_0}^{p} \frac{1}{P}\,\mathrm{d}P \rightarrow c_p \cdot \left(\ln(t) - \ln(t_0)\right) - R \cdot \left(\ln(p) - \ln(p_0)\right)$$

$c_p := \frac{7}{2} R = 29.101 \frac{J}{mole \cdot K} \qquad t_0 := K \qquad p_0 := Pa$

$t := 18\ °C \qquad p := 1\ atm$

$s := c_p \cdot \left(\ln(t) - \ln(t_0)\right) - R \cdot \left(\ln(p) - \ln(p_0)\right)$

Эти единицы измерения несовместимы.

$s := c_p \cdot \ln\left(\frac{t}{t_0}\right) - R \cdot \ln\left(\frac{p}{p_0}\right) = 69.279 \frac{J}{mole \cdot K}$

Рис. 1.12

Расчет удельной энтропии идеального газа

$$\int_{t_0}^{t} c_v \, dT \rightarrow c_v \cdot (t - t_0)$$

$$c_v := \frac{5}{2} R = 20.786 \frac{J}{mole \cdot K}$$

$$t_0 := K$$

$$t := 18 \ °C$$

$$u := c_v \cdot (t - t_0) = 6031.1 \frac{J}{mole}$$

Рис. 1.13
Расчет внутренней энергии идеального газа

На сайте книги читатель найдет и другие интерактивные страницы по расчету цикла Отто. На рисунке 1.14 показан сайт, где посетитель может:

- выбрать рабочее тело цикла;
- указать его параметры (p_1, T_1, n и T_3);
- выбрать вид диаграммы, отображающей процесс;
- указать точку, по которой будет показано положение поршня (здесь приведены все рисунки, но на сайте демонстрируется один из них с отображением соответствующей точки на диаграмме цикла Отто).

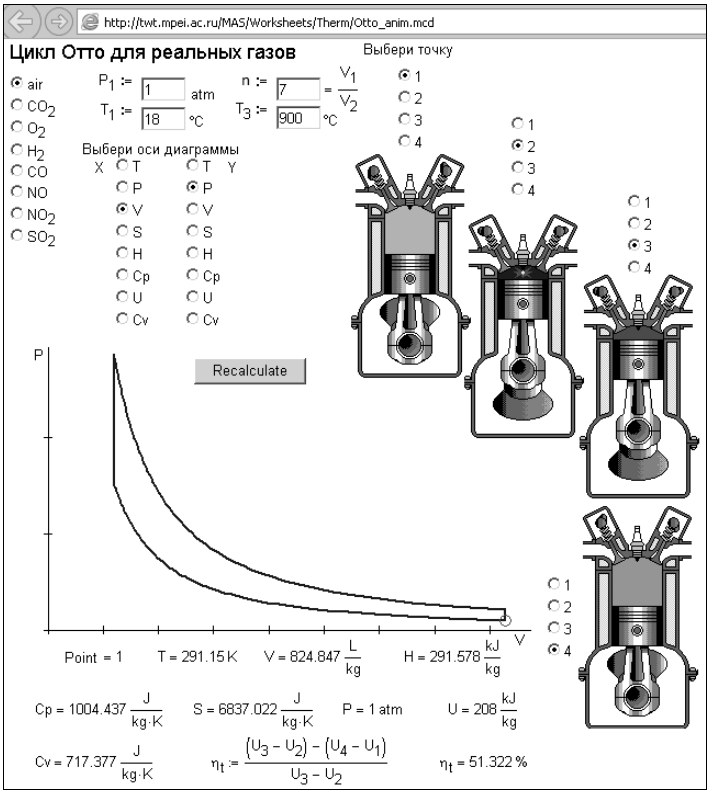


Рис. 1.14
Сайт с расчетом и графическим отображением цикла Отто

Сайт с «живыми» расчетами работает в МЭИ уже почти десять лет. За это время на сайте **www.trie.ru** размещено более двадцати тысяч подобных расчетов, касающихся в первую очередь энергетики, энергосбережения и энергоэффективности, свойств рабочих тел, теплоносителей и конструкционных материалов. Не обделены вниманием и общетехнические научные и учебные дисциплины: математика, физика, химия (см. выше), тепломассообмен, термодинамика, гидрогазодинамика и др. дисциплины классического технического университета. Расчеты дополнены функциями, на которые можно ссылаться и которые после такой ссылки становятся видимыми в расчетном рабочем документе. Кроме того, многие формулы продублированы в виде текстовых строк, которые без труда кликами мышки можно копировать в ячейки электронных таблиц и в коды языков программирования.

Иногда расчеты приходится дополнять не просто формулами, по которым они сделаны, а целым программным кодом — см. как пример расчет (перевод чисел в разных системах счисления), показанный на рисунке 3.2 (стр. 53).

Авторская технология «живых расчетов» в настоящее время реализована Международной ассоциацией по свойствам воды и водяного пара, на сайте которой **www.iapws.org** утвержденные расчетные стандарты этой наиважнейшей субстанции стали публиковаться не только в виде «мертвых» pdf-файлов, но и в виде «живых» расчетов. На российском аналогичном сайте размещены дополнительно ссылки на облачные функции и шаблоны с соответствующими функциями. Авторы также сотрудничают с издательством Elsevier по оживлению формул справочников (см. на сайте **www.knovel.com** инструмент Equation Library).

Фирма Elsevier (**www.elsevier.com**) является крупнейшим в мире электронным издательством, публикующим на своем сайте журналы и книги по многим областям знаний[14]. Фирма Knovel — это тоже электронное издательство, входящее теперь в Elsevier, но ориентированное на публикацию технической литературы с возможностью интерактивной работы с графиками, таблицами и формулами. На сайт **www.knovel.com** можно выйти непосредственно из среды Mathcad 15 (рис. 1.15).

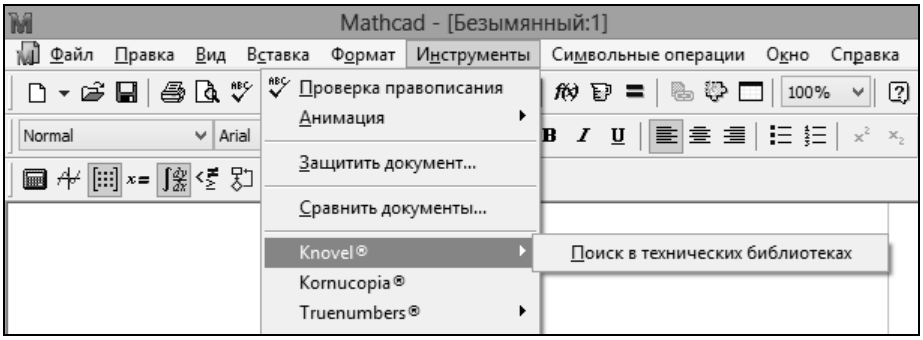


Рис. 1.15
*Выход на сайт **www.knovel.com** из Mathcad 15*

Пользователи Mathcad перед решением новой задачи могут поискать готовое решение на сайте Knovel, не выходя из Mathcad.

Авторы данной книги активно сотрудничают с фирмой Knovel и даже передали для размещения на сайте **www.knovel.com** некоторые свои расчеты, касающиеся и теплотехники (рис. 1.16).

[14] Инструмент оценки публикационной активности ученых Scopus — это одна из разработок фирмы Elsevier.

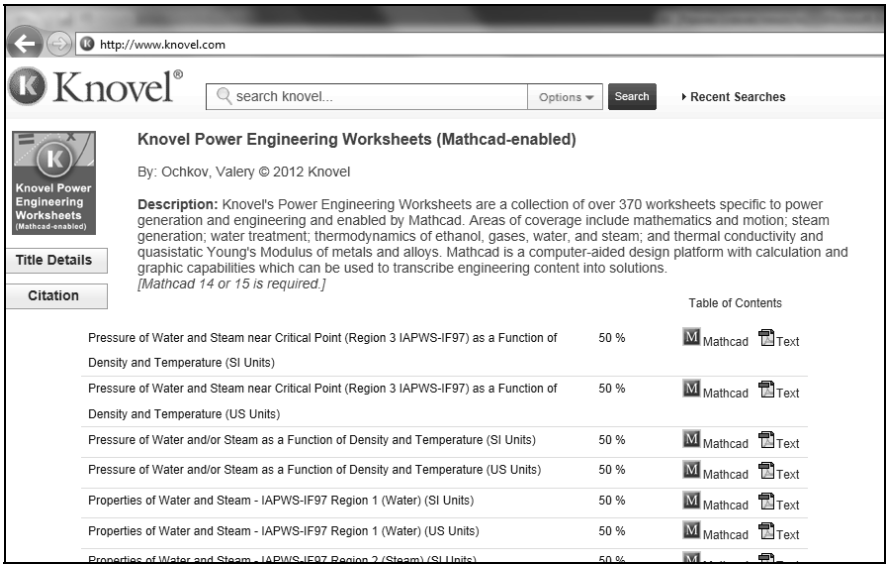


Рис. 1.16
Авторские Mathcad-расчеты на сайте Elsevier/Knovel

В 2012 г. фирма Knovel стала частью фирмы Elsevier. На рисунке 1.17 можно видеть ссылку на сайт **www.knovel.com** в списке онлайн-инструментов фирмы Elsevier.

На сайте фирмы Knovel размещены собственные онлайн-инструменты, среди которых есть так называемая *библиотека уравнений* (Interactive Equations — рис. 1.18).

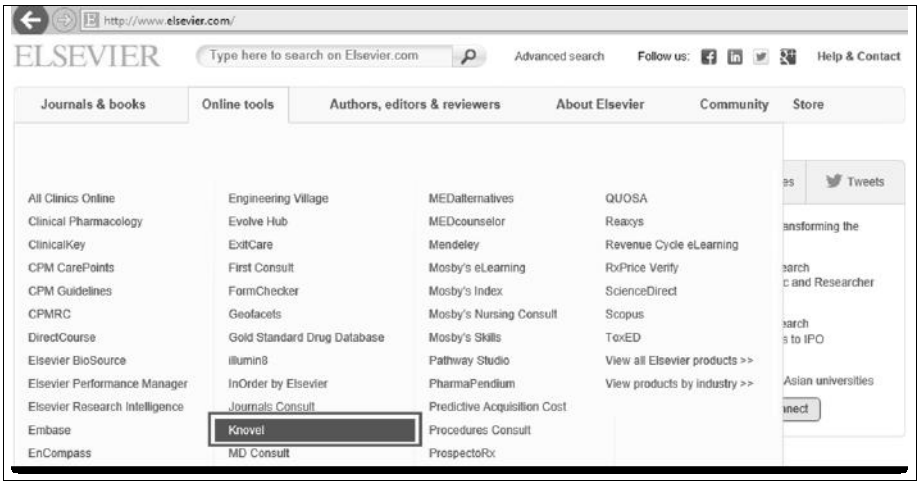


Рис. 1.17
Стартовая страница сайта электронного издательства Elsevier со списком онлайн-инструментов

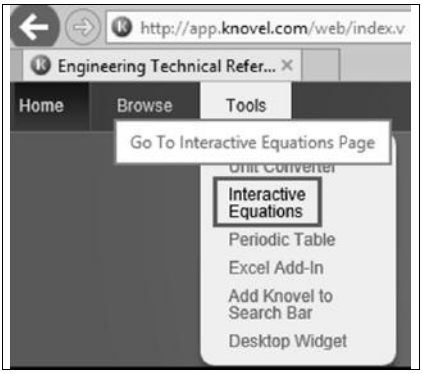


Рис. 1.18
Интерактивные инструменты фирмы Knovel

Библиотека уравнений фирмы Knovel позволяет не только просматривать те или иные формулы, но и интерактивно работать с ними примерно так, как это делается на Mathcad Calculation Server, который поддерживают авторы данной книги и о котором много написано в этой книге. На март 2014 г. в библиотеке уравнений фирмы Knovel было 129 формул: 39 по химии и химическим технологиям, 77 по механике и 13 по свойствам материалов (рис. 1.19), а к началу 2016 г. уравнений должно быть более 3000.

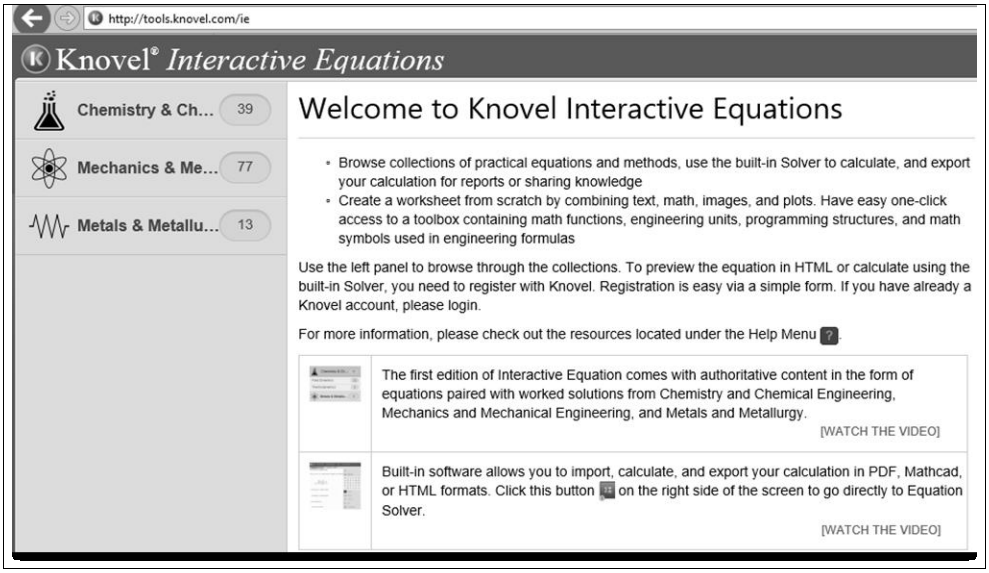


Рис. 1.19
Стартовая страница библиотеки уравнений фирмы Knovel

У авторов книги есть большие планы сотрудничества с фирмой Knovel в плане наполнения библиотеки теплотехническими и прочими расчетами. Старт этой работе дала публикация расчета мощности насоса в зависимости от массового расхода воды и перепада давления на нем (рис. 1.20–1.22) с опорой на функцию, возвращающую плотность воды в зависимости от давления и температуры, созданную по формуляции IAPWS (см. выше).

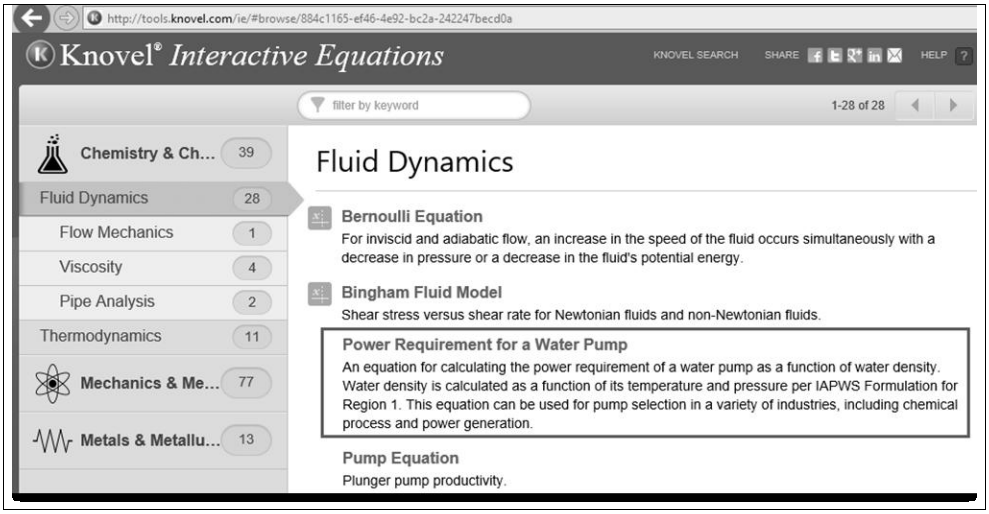


Рис. 1.20

Ссылка на расчет мощности насоса в библиотеке уравнений фирмы Knovel

Расчет, показанный на рисунке 1.21, сделан в среде пакета SMath, который бесплатно скачивается с сайта **www.smath.info.** И это главное. Фирма Knovel не стала делать ставку на Mathcad из-за того, что это довольно дорогой коммерческий продукт другой фирмы — фирмы PTC.

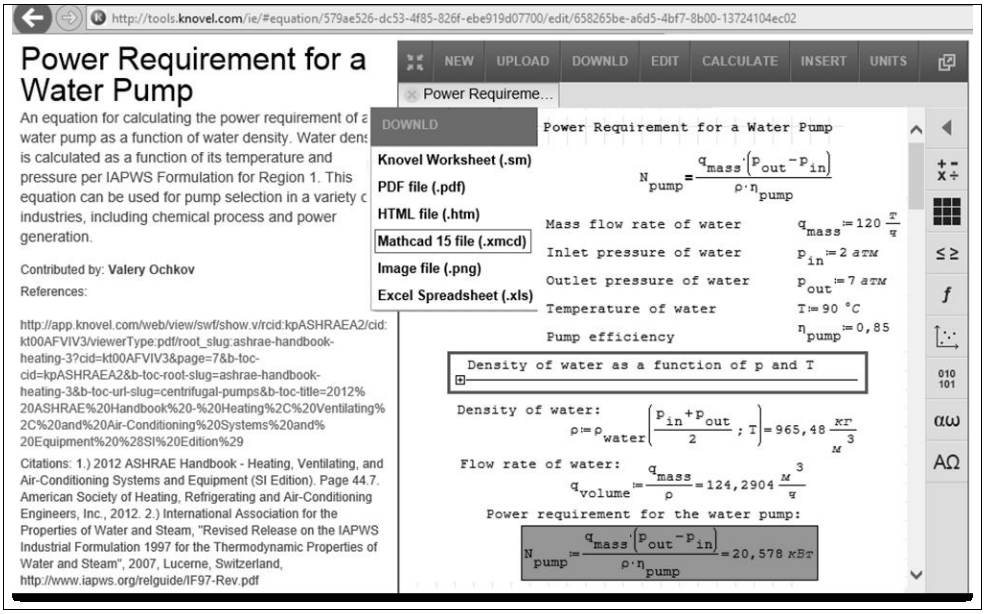


Рис. 1.21

Интерактивный расчет мощности насоса на сайте фирмы Knovel

В интерактивном SMath-расчете, показанном на рисунке 1.21, есть скрытая область с функцией, возвращающей плотность воды в зависимости от давления и температуры. Это область раскрыта и частично показана на рисунке 1.22. Эту и

многие другие подобные функции можно скачать с авторского сайта **http://twt.mpei.ac.ru/rbtpp/Region**. Там она имеет имя wspD1PT. Если в имени этой функции единицу заменить двойкой, то она будет возвращать плотность уже не воды, а водяного пара, свойства которого определяются не первой, а второй областью формуляции IAPWS-IF97. Если в имени функции будет тройка, то она будет относиться к околокритической области. Если там будет пятерка, то к области водяного пара высокой температуры. Четвертая область формуляции IAPWS-IF97 определяет свойства воды и водяного пара на линии насыщения от тройной точки воды до критической точки. Mathcad-файл с именем H2O и расширением mcd, mcdx или mcdxz, хранящийся в Интернете по адресу **http://twt.mpei.ru/tthb/**, содержит функции с именами wspD1PT, wspD2PT, wspD3PT и wspD5PT. Кроме того, там есть функция с именем wspDPT, которая определяет, в какой области формуляции IAPWS-IF97 находятся параметры (давление и температура), и вызывает нужную пронумерованную функцию.

```
Density of water as a function of p and T

ρwater(p ; T) := "Revised Release on the IAPWS Industrial Formulation 1997"
                 "for the Thermodynamic Properties of Water and Steam"
                 "see http://www.iapws.org/relguide/IF97-Rev.pdf"
                 "Density of water (Region 1) as function of pressure and temperature"

γ(w ; τ) := "Gibbs free energy"
            I := (0 0 0 0 0 0 0 0 1 1 1 1 1 1 2 2 2 2 2 3 3 3 4 4 4 5 8 8
            J := (-2 -1 0 1 2 3 4 5 -9 -7 -1 0 1 3 -3 0 1 3 17 -4 0 6 -5
            n := (1,4632971213167·10^-1 -8,4548187169114·10^-1 -3,7563603
            Σ_{i=1}^{34} ( n_i · d/dw ((7,1 - w)^{I_i}) · (τ - 1,222)^{J_i} )

R := 0,461526 кДж/(кг·К)

w := p / (16,53 МПа)

τ := 1386 К / T

1 / ( (R·T)/p · w · γ(w ; τ) )
```

Рис. 1.22

Функция интерактивной библиотеки фирмы Knovel, возвращающая плотность воды (показана часть функции)

При публикации «живых» формул их, как правило, дублируют с заменой переменных их численными значениями примерно так.

Энтальпию воды в конце необратимого процесса сжатия ее в насосе $h_{зд}$ вычислим по формуле для расчета относительного внутреннего КПД насоса:

$$h_{зд} = h_{2'} + (h_з - h_{2'})/\eta_{oi}^{н} = 121{,}4 + (133{,}9 - 121{,}4)/0{,}85 = 136{,}1 \text{ кДж/кг}.$$

Такое дублирование ведется для дополнительного контроля, но уже не самой формулы, а правильности численных значений переменных формулы и счета по ним. Предполагается, что человек (некий «сертификатор» расчета), имеющий под рукой калькулятор (арифмометр, логарифмическую линейку, специальные расчетные таблицы, счеты, абак и т. д., если идти вглубь истории), может проверить

арифметические выкладки, не разыскивая в расчете численные значения переменных, вычисленных ранее.

В среде некоторых математических пакетов также требования стандартов учитываются: там можно дублировать формулы с подстановкой численных значений переменных. Вот пример подобного оформления расчета в среде Mathcad 15 (рис. 1.23).

Вместимость конуса: $r := 20mm$ $h := 3.5cm$

$$V := \frac{\pi}{3} \cdot r^2 \cdot h = \frac{\pi}{3} \cdot (20\ mm)^2 \cdot (3.5\ cm) = 14.661\,mL$$

Рис. 1.23
«Оцифровка» формулы в среде Mathcad 15

Подобный стиль счета предусмотрен и в некоторых специализированных пакетах — в пакете NormCAD (**http://normcad.ru**), например, предназначенном для расчетов по российским (машино-) строительным нормам и правилам (СНиПы). Вот пример расчета в среде NormCAD.

Площадь арматуры по грани нормальной оси X:

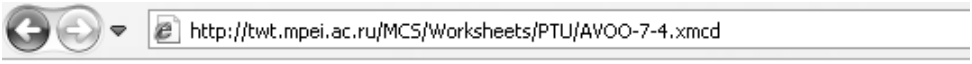
$$A_{sx} = A_{s1,\ x} \cdot (n_x + 1) + (2 \cdot A_{so} - A_{s1,\ x} - A_{s1,\ y}) \cdot b / (1 + b) =$$
$$= 0 \cdot (0+1) + (2 \cdot 0{,}0008 - 0 - 0{,}000314159) \cdot 0{,}50695 / (1 + 0{,}50695) = 0{,}000432567\ м^2$$

Но такая «оцифровка» формул была желательна при «ручном» ведении расчетов с помощью калькулятора (арифмометра и т. д., см. выше), но стала, как понимает читатель, совершенно излишней при компьютерном счете. Тем не менее, многие устаревшие, но действующие и поныне стандарты оформления расчетов требуют «оцифровки» формул, даже если они велись полностью на компьютере. «По одежке встречают, по уму провожают» — этим правилом руководствуются в различных сертифицирующих организациях, где ваш расчет не примут на проверку, если он оформлен неправильно.

«Оцифровку» расчетных формул требуют многие преподаватели, принимающие типовые курсовые и дипломные проекты в высших и средних технических учебных заведениях. Некоторые преподаватели, принимая от студента такой расчет, кладут около себя калькулятор (чуть раньше они «брали в руки логарифмическую линейку», а еще раньше — счеты) и проверяют все или по крайней мере некоторые ключевые математические выкладки, допуская, что студент тут мог ошибиться, забывая при этом о том, что эти расчеты делал не студент, а компьютер, который не может ошибиться в арифметических выкладках.

Студент тут может ошибиться, если он выберет не ту формулу или подставит в нее неправильное значение. Механическая проверка счета по данной формуле тут не поможет... Когда-то в МЭИ первые компьютеры стали печатать ведомости на зарплату — колонки цифр с итоговой суммой внизу. Главный бухгалтер МЭИ эти ведомости должен был подписывать. Но прежде чем это сделать он... на счетах проверял правильность числа, стоящего внизу ведомости на зарплату, — распечатки компьютера, суммировал все числа. Когда этому главбуху говорили, что в итоговой сумме не может быть ошибки, что ошибка может быть только при начислении той или иной зарплаты тому или иному работнику, то ответ был примерно таков — компьютер в случае чего не посадят в тюрьму, а меня посадят, если я подпишу неправильную ведомость... Так и до самого ухода на пенсию этот главный бухгалтер суммировал числа в ведомостях на зарплату перед их подписанием... Так и некоторые преподаватели требуют «оцифровки» формул в студенческих расчетах, не понимая, что этим возможную ошибку не выявить...

Процесс «оцифровки», зафиксированный в вышеприведенных формулах, требует значительного ручного оформительского труда и не всегда приводит к желаемому результату: к нужному числу знаков в ответе с нужной единицы измерения и т. д. Чтобы «волки были сыты и овцы целы», чтобы удовлетворить проверяющего расчет и не перегружать себя лишней и трудоемкой работой, можно оформить расчет так — дать формулу с итоговым ответом, а под ней «распечатать» переменные формулы с нужным форматом числа и с нужной единицей измерения (см. рис. 1.24).



Энтальпию воды в конце необратимого процесса сжатия ее в насосе $h_{3д}$ вычислим по формуле для расчета относительного внутреннего КПД насоса:

$$h_{3д} := h_{2'} + \frac{h_3 - h_{2'}}{\eta_{oi}^{н}} = 136.1\,kJ/kg$$

где $h_{2'} = 121.4\,kJ/kg$ - энтальпия воды на входе в насос,

$h_3 = 133.9\,kJ/kg$ - энтальпия воды на выходе из насоса при обратимом процессе сжатия

и $\eta_{oi}^{н} = 0.85$ - внутренний относительный КПД насоса.

Рис. 1.24

Формула со списком переменных и их численных значений

«Умный» преподаватель, проверяющий расчет, показанный на рис. 1.24, видя не совсем или совсем неправильный ответ по формуле, не станет проверять счет по этой формуле. Ему достаточно проверить, какие переменные фигурируют в формуле и какие значения эти переменные имеют. Вот тут-то и будет очень полезна распечатка значений переменных анализируемой формулы так, как это показано на рис. 1.24. Если же возникнет потребность проверки правильности счета по формуле, к примеру, для контроля правильности округления ответа, то около формулы лучше поместить значения переменных с базовыми единицами измерения — см. рис. 1.25 и сравните его с рис. 1.23.

Исходные данные:

$$p := 0.7\ atm \qquad v := 7100\ \frac{L}{kg} \qquad m := 14\ \frac{gm}{mole}$$

Ответ $$T := \frac{m \cdot p \cdot v}{R} = 847.94\ K$$

где $$m = 0.014\ \frac{kg}{mol} \qquad p = 70927.5\ Pa$$

$$v = 7.1\ \frac{m^3}{kg} \qquad R = 8.314\ \frac{J}{K \cdot mol}$$

Рис. 1.25

Расчет по формуле с переводом единиц измерения к базовым единицам СИ

Другая помощь проверяющему будет заключаться в том, что вблизи формулы размещаются не только значения переменных формул, но и значения некоторых фрагментов формулы. Эта особенность показана на рис. 1.26.

Термический КПД паро-газового энергетического блока:

$\eta_{ПГУ} := \left(m \cdot l_{ГТУ} + l_{ПТУ}\right) / q_1 = 63.05\,\%$

где $m = 8.26498\,\frac{kgg}{kgws}$ (отношение расхода газа к расходу воды и водяного пара),

$l_{ГТУ} = 423.911\,kJ/kgg$ (удельная работа газовой турбины),

$m \cdot l_{ГТУ} = 3503.62\,kJ/kgws$ (работа газовой турбины на кг воды и водяного пара),

$l_{ПТУ} = 1322.94\,kJ/kgws$ (удельная работа паровой турбины),

$m \cdot l_{ГТУ} + l_{ПТУ} = 4826.56\,kJ/kgws$ (удельная работа газовой и паровой турбины) и

$q_1 = 7655.13\,kJ/kgws$ (удельная теплота, подведенная к ПГУ).

Рис. 1.26

Расчет по формуле с «распечаткой» значений переменных и фрагментов формулы

Когда авторы видят в статье, книге или Интернете более-менее интересную формулу, они сразу поручают своим студентам «оживить» ее и разместить на сайте, адрес которого находится на первых местах поиска по ключу «Расчеты в Интернете» почти всех поисковых машин: Google, Yandex, Yahoo, Bing, Rambler и т. д.

Технология издания научно-технических статей и книг претерпевает существенное изменение. Традиционные бумажные тиражи уже не печатаются, а ведется дистрибуция электронных версий с различными сервисами: автоматическая связь с авторами, ссылки на работы которых приведены в статье или книге, закладки, заметки, обсуждение на форумах, возможность работы с видео и анимацией и т. д. и т. п. В этот ряд прекрасно вписывается и возможность непосредственного счета по формулам — тема данного этюда. Если же кому-то необходимо работать именно с бумажной версией книги — нет проблем: ее можно сделать самому или заказать в специальных полиграфических центрах.

ДИВЕРТИСМЕНТ. КАК ВЫСОКО МЫ ПОДНЯЛИСЬ В СВОБОДНОМ ПАДЕНИИ?

На авторском расчетном сервере можно найти не только онлайн расчеты (см. пример выше на рис. 1.9 (стр. 24), рис. 10.9–10.11 (стр. 249) и 10.14 на стр. 227–231), но ссылки, вставив которые в текущий расчет, можно сделать видимыми дополнительные константы и функции.

Конкретный пример. В Mathcad встроена константа *g* — ускорение свободного падения. Мы с этой константой будем работать в этюде 9 при расчете, например, гравитационного поезда (рис. 9.27, 9.28 и рис. 9.30–9.34 на стр. 227–231). Но эту важную физическую величину константой можно назвать только условно. Она зависит как минимум от высоты над уровнем моря и широты местности. Есть формулы, описывающие эту зависимость. Одна из них открыта для интерактивного использования на авторском сайте **http://twt.mpei.ac.ru/MCS/Worksheets/g-h-psi.xmcd**. Но на этот же сайт можно сделать ссылку — «облачную» ссылку так, как это показано на рис. 1.27.

После такой ссылки идентификатор g будет уже не константой, а функцией двух аргументов, которую можно вызывать для решения различных задач. Например, для оценки, на какой высоте над Москвой ускорение свободного падения будет равно величине, встроенной в Mathcad (см. последний оператор на рис. 1.27).

Еще один пример, связанный с высотой над уровнем моря.

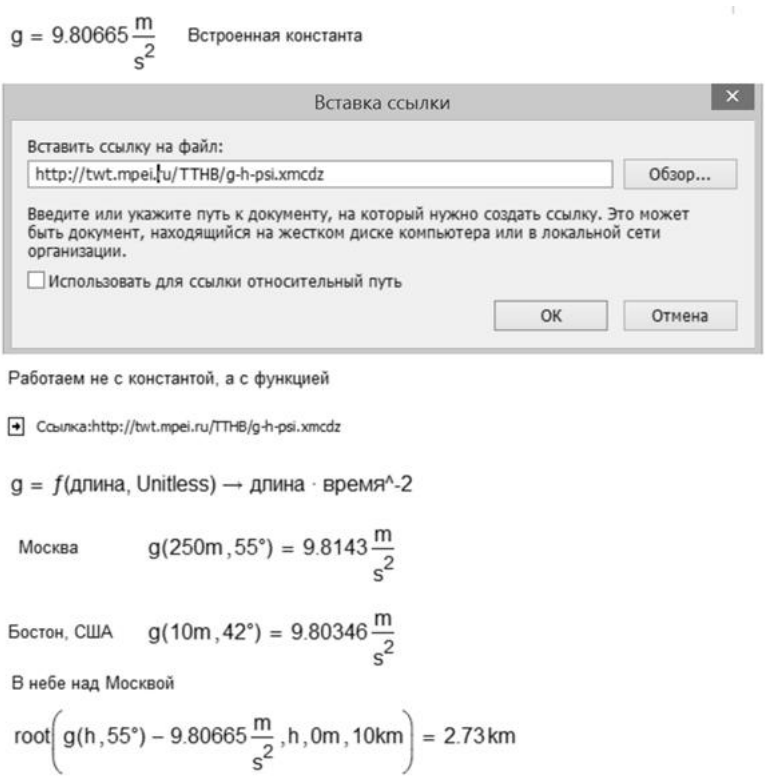


Рис. 1.27
Работа с облачной функцией ускорения свободного падения

В книге «Странствования за границей» Марк Твен в своем юмористическом духе описывает случай во время альпийского путешествия:

«*Неприятности наши кончились; поэтому люди могли отдохнуть, а у меня, наконец, явилась возможность обратить внимание на научную сторону экспедиции. Прежде всего я хотел определить посредством барометра высоту места, где мы находились, но, к сожалению, не получил никаких результатов. Из моих научных чтений я знал, что не то термометр, не то барометр следует кипятить для получения показаний. Который именно из двух, — я не знал наверное и потому решил прокипятить оба.*

И все-таки не получил никаких результатов. Осмотрев оба инструмента, я увидел, что они вконец испорчены: у барометра была только одна медная стрелка, а в шарике термометра болтался комок ртути...

Я отыскал другой барометр; он был совершенно новый и очень хороший. Полчаса кипятил я его в горшке с бобовой похлебкой, которую варил повар. Результат получился неожиданный: инструмент совершенно перестал действовать, но суп приобрел такой сильный привкус барометра, что главный повар — человек очень умный — изменил его название в списке кушаний. Новое блюдо заслужило всеобщее одобрение, так что я приказал готовить каждый день суп из барометра. Конечно, барометр был совершенно испорчен, но я не особенно жалел о нем. Раз он не помог мне определить высоту местности, значит, он больше мне не нужен».

Серьезно разбирать юмористические тексты — дело несерьезное, но все же попробуем сделать это. Тем более, что такая попытка была уже предпринята Я. Перельманом в знаменитой «Занимательной физике».

Ясно, что ни термометр, ни кипящая вода для определения высоты над уровнем моря не нужны. Необходим только барометр и... барометрическая формула, связывающая давление воздуха и высоту над уровнем моря. Но если иметь под рукой только термометр, то определить давление воздуха можно, «сварив из него суп». На авторском интерактивном, открытом, сетевом расчете, размещенном на ранее упомянутом сайте Elsevier/Knovel, показан расчет, связывающий температуру, давление и высоту — см. рис. 1.28.

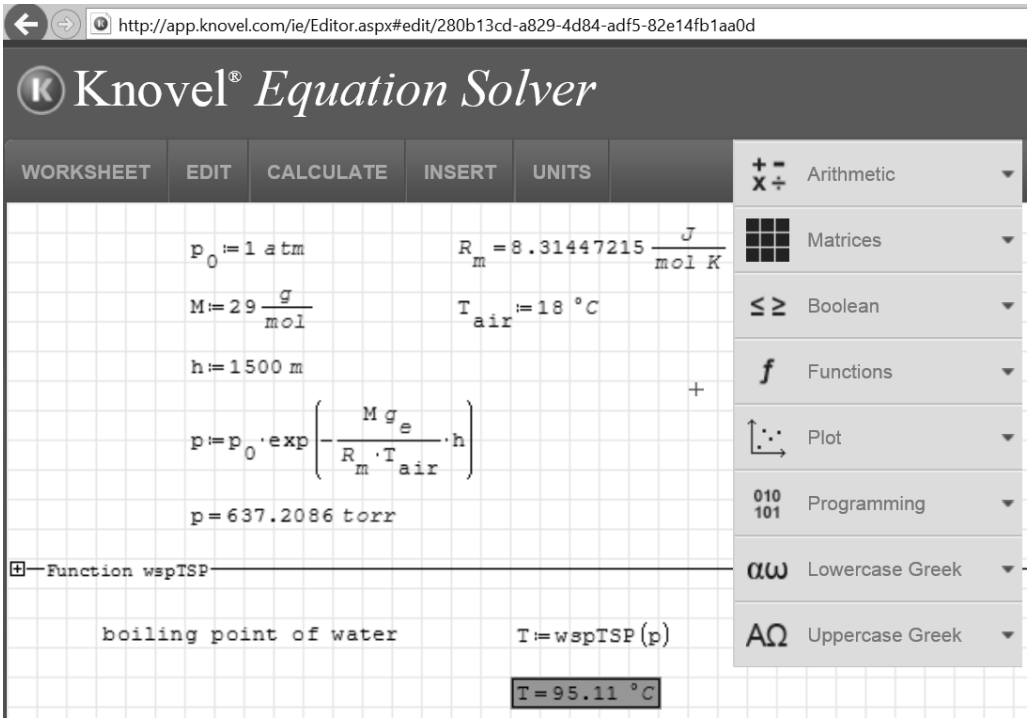
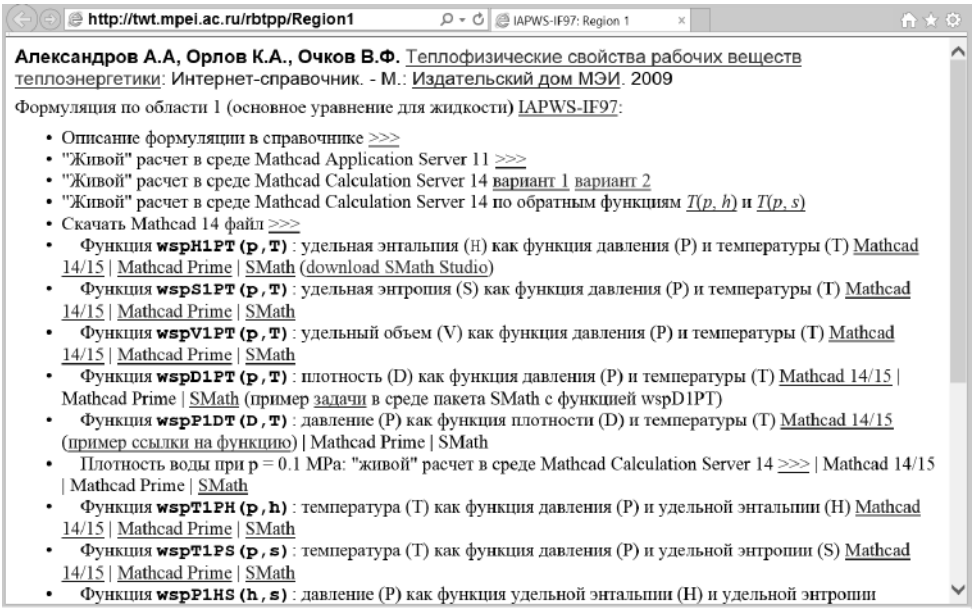


Рис. 1.28
Работа с облачной барометрической функцией

На высоте полтора километра вода будет кипеть при температуре 95 градусов по шкале Цельсия, а давление воздуха будет составлять 637 мм ртутного столба. Но тут следует помнить, что связь температуры кипения воды с давлением — это *функция*, а давления с высотой над уровнем моря — это *корреляция*, т. к. барометрическая формула не только слишком груба, но и не учитывает ряд других факторов. О корреляции мы подробнее поговорим в этюде 14.

В расчете на рисунке 1.28 используется функция с именем wspTSP, возвращающая температуру (T) насыщения (S — saturated) воды и водяного пара в зависимости от давления (P). Эта функция входит в авторский пакет облачных функций для Mathcad с именем H2O.xmcd, который хранится по адресу **http://twt.mpei.ru/tthb** и который можно вставлять в расчеты, где требуются знание теплофизических свойств воды и водяного пара по технологии, показанной на рис. 1.27. Подробная информация об этом размещена на сайтах **http://twt.mpei.ac.ru/rbtpp** и **www.wsp.ru**. На рисунке 1.29 показана одна из страниц авторского интернет-справочника, с помощью которой можно не только вести online расчеты свойств воды, но и скачивать нужные функции, возвращающие эти свойства в средах популярных расчетных программ. Файл для Mathcad 14/15 можно не скачивать, а делать на него ссылку.



http://twt.mpei.ac.ru/rbtpp/Region1 IAPWS-IF97: Region 1

Александров А.А, Орлов К.А., Очков В.Ф. Теплофизические свойства рабочих веществ теплоэнергетики: Интернет-справочник. - М.: Издательский дом МЭИ. 2009

Формуляция по области 1 (основное уравнение для жидкости) IAPWS-IF97:

- Описание формуляции в справочнике >>>
- "Живой" расчет в среде Mathcad Application Server 11 >>>
- "Живой" расчет в среде Mathcad Calculation Server 14 вариант 1 вариант 2
- "Живой" расчет в среде Mathcad Calculation Server 14 по обратным функциям *T(p, h)* и *T(p, s)*
- Скачать Mathcad 14 файл >>>
- Функция **wspH1PT(p,T)** : удельная энтальпия (H) как функция давления (P) и температуры (T) Mathcad 14/15 | Mathcad Prime | SMath (download SMath Studio)
- Функция **wspS1PT(p,T)** : удельная энтропия (S) как функция давления (P) и температуры (T) Mathcad 14/15 | Mathcad Prime | SMath
- Функция **wspV1PT(p,T)** : удельный объем (V) как функция давления (P) и температуры (T) Mathcad 14/15 | Mathcad Prime | SMath
- Функция **wspD1PT(p,T)** : плотность (D) как функция давления (P) и температуры (T) Mathcad 14/15 | Mathcad Prime | SMath (пример задачи в среде пакета SMath с функцией wspD1PT)
- Функция **wspP1DT(D,T)** : давление (P) как функция плотности (D) и температуры (T) Mathcad 14/15 (пример ссылки на функцию) | Mathcad Prime | SMath
- Плотность воды при p = 0.1 MPa: "живой" расчет в среде Mathcad Calculation Server 14 >>> | Mathcad 14/15 | Mathcad Prime | SMath
- Функция **wspT1PH(p,h)** : температура (T) как функция давления (P) и удельной энтальпии (H) Mathcad 14/15 | Mathcad Prime | SMath
- Функция **wspT1PS(p,s)** : температура (T) как функция давления (P) и удельной энтропии (S) Mathcad 14/15 | Mathcad Prime | SMath
- Функция **wspP1HS(h,s)** : давление (P) как функция удельной энтальпии (H) и удельной энтропии

Рис. 1.29
Сайт по свойствам воды

Задание читателям

1. Сгенерировать формулы пересчета видов концентрации для всех выражений, приведенных на рис. 1.5, по образцам, показанным на рис. 1.7.

2. Найти в справочниках типичные псевдоэмпирические формулы, касающиеся вашей профессиональной деятельности, и преобразовать их в чисто физические.

3. Переделать расчет, показанный на рис. 1.11, так, чтобы рассчитывался цикл Дизеля (Карно, Тринклера, Калины и т. д.).

4. Помочь персонажу Марка Твена — вывести формулу, по которой можно рассчитать высоту над уровнем моря в зависимости от температуры кипящей воды.

5. В плоском вакуумном диоде зависимость силы тока от напряжения имеет вид $I = cU^{3/2}$, где c — некоторая постоянная (так называемый «закон трех вторых»). Во сколько раз увеличится сила давления на анод, возникающая из-за ударов электронов о его поверхность, если напряжение на диоде увеличить в два раза? Начальной скоростью электронов, вылетающих с катода, пренебречь.

Этюд 2

ЛЕБЕДЬ, РАК И ЩУКА, ИЛИ СКАЛЯР И ВЕКТОР В КОМПЬЮТЕРНЫХ ВЫЧИСЛЕНИЯХ

В этюде будет рассказано о таких важных понятиях, как скаляр и вектор. Будут даны примеры использования этих величин в физических расчетах на компьютере и при программировании, когда необходимо вводить в расчет массивы данных.

Математика: скаляр, вектор, матрица, матрица-столбец, матрица-строка, двумерное, трехмерное и n-мерное пространство, абсолютная величина, модуль (длина, норма) вектора, линейная функция, уравнение прямой на плоскости, линейная интерполяция, система линейных алгебраических уравнений (СЛАУ), обыкновенное дифференциальное уравнение, начальное условие.

Физика: скалярная и векторная физическая величина, плотность водного раствора соли, сопротивление воздуха.

Информатика: одномерный, двумерный и вложенный массив, численное решение дифференциального уравнения с физическими величинами, работа с плоской графикой, структурное программирование.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127828**

На лекциях по информатике и математике в Московском энергетическом институте авторы иногда делают опросы (см. также начало этюда 14) для того, чтобы уточнить, что студенты изучали раньше в школе и что они изучают сейчас в вузе на параллельных курсах физики, химии, термодинамики, теоретической механики и других учебных дисциплин. На вопрос о том, знают ли они, что такое *скалярная* физическая величина и *векторная* физическая величина, все студенты хором отвечают: «Знаем! Нам все уши в школе на уроках физики прожужжали, объясняя это!» «Знаем» они выкрикивают вслух, а про «уши», конечно, умалчивают. Но в студенческой реакции на вопрос можно услышать данное продолжение ответа... Все школьники и студенты безошибочно знают, что масса — это скалярная величина, а вес (сила) — векторная и что задачи по физике нужно решать с учетом этих фундаментальных понятий.

Скаляр и вектор присутствует и в языках программирования, электронных таблицах и математических пакетах, которые в настоящее время широко используются при преподавании информатики и других учебных дисциплин в школе и вузе.

В языках программирования есть понятие *массива* (array) — одномерного, двумерного и многомерного. Электронные таблицы (SuperCalc, VisiCalc, Excel и др.) создавались именно для того, чтобы хранить и обрабатывать эти самые таблицы — одну из разновидностей массива данных. В имени языка программирования технических расчетов Matlab, который часто пытаются сравнить с Mathcad, — Mat означает матрицу (а **Mat**rix), а lab — лабораторию (а **Lab**oratory). В математических пакетах, в частности в пакете Mathcad, понятие массива прописано в явном виде. Пользователь может ввести в расчет *массив* (матрицу) с одним столбцом, который в среде Mathcad обозначен как *вектор*. Но при этом нужно четко понимать, что маткадовский вектор несет в себе как минимум две смысловые нагрузки, которые мы попытаемся уяснить на примере решения двух несложных задач.

На рисунке 2.1 показан расчет ускорения тела с заданной массой, если известны три силы, приложенные к телу в трех направлениях, — по оси x, y и z (наша задача решается в трехмерном пространстве). Но мы можем перевести ее и на плоскость (двумерное пространство) и на линию. А можно перейти и в четырех-

мерное и «болеемерное» пространство, которое часто описывается в научной фантастике, но которое реально присутствует в математике.

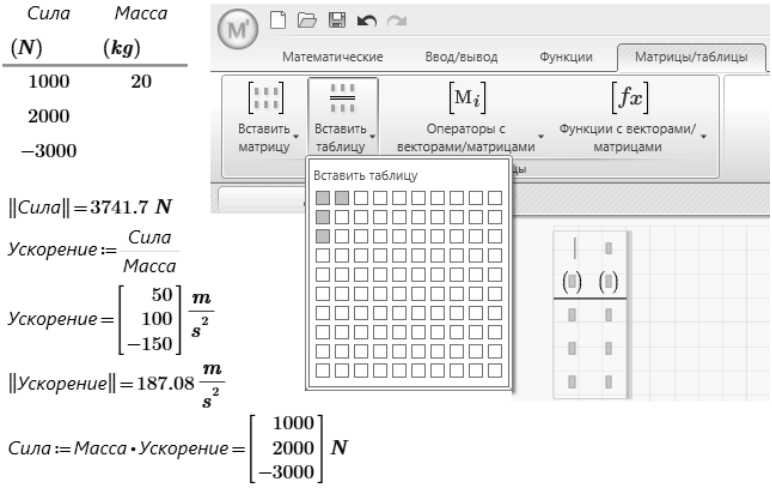


Рис. 2.1
Задача о движении тела в пространстве (Mathcad Prime)

В расчете, показанном на рис. 2.1, мы ввели две исходные величины — `Сила` в виде вектора с тремя компонентами (1000, 2000 и минус 3000 ньютон) и `Масса` в виде скаляра со значением 20 кг. Затем мы рассчитали длину вектора `||Сила||` — определили абсолютное значение силы, приложенной к телу, и вектор `Ускорение` (три его составляющие по осям x, y и z). Многие пользователи Mathcad ошибочно полагают, что длина вектора — это число компонентов в нем, а не квадратный корень из суммы квадратов значений компонентов. Эта путаница связана и с тем, что в среде Mathcad число компонентов вектора подсчитывает встроенная функция с именем длина (см. рис. 2.2 ниже). Оператор же вычисления «настоящей» длины вектора (нормы вектора) $\|\mathbf{v}\|$ появился только в Mathcad Prime.

Примечание. Мы воспользовались одним удобным инструментом — *таблицей*, введенной в Mathcad Prime. Введенную прямоугольную таблица размером 3 на 2 (три строки и два столбца — см. правую часть рис. 2.1) мы заполнили частично, дав силе три значения (вектор), а массе только одно (скаляр).

На рисунке 2.2 показано решение в среде предыдущей версии Mathcad — Mathcad 15 нашей задачи об ускорении тела.

В варианте решения задачи, показанном на рис. 2.2, значение силы вводится не вектором с тремя компонентами, как на рис. 2.1, а покоординатно: Сила_x, Сила_y и Сила_z. Для этого в выделенной области расчета (см. верхнюю часть рис. 2.2), во-первых, системной переменной `ORIGIN`[1] присвоено значение единицы (по умолчанию оно равно нулю — почему, скажем чуть позже), а во-вторых, переменным x, y и z присвоены значения 1, 2 и 3 соответственно. Это сделано для того, чтобы компоненты вектора `Сила` обозначать не как Сила_1, Сила_2 и Сила_3, а как Сила_x, Сила_y и Сила_z, что более соответствует «физике» задачи: разложение вектора по трем координатам: x, y и z.

Глядя на рис. 2.1 и 2.2 можно представить себе реальную силу (стрелку-вектор) в 3741.7 N (ньютон), приложенную к телу и раскладывающуюся по координатам x, y и z со значениями 1000, 2000 и минус 3000 N соответственно. Такие векторы-стрелки часто рисуют в учебниках и задачниках по физике, иллюстрируя

[1] В книге описан только один расчет, где системная переменная ORIGIN равна не нулю и не единице, а двойке — см. рис. 16.2 на стр. 358.

силы (см., например, рис. 7.7 на стр. 146), направления движения, электрические и магнитные поля и прочие векторные физические величины. Кстати, в различных учебниках и популярных изданиях по физике к векторам силы привязывают... Лебедя, Рака и Щуку из знаменитой басни Крылова и анализируют, смог бы воз сдвинуться с места[2]. В этюде 9 мы будем рассматривать силы, приложенные к планете или спутнику вследствие действия закона всемирного тяготения. Эти силы будут разложены по двум или трем осям пространства.

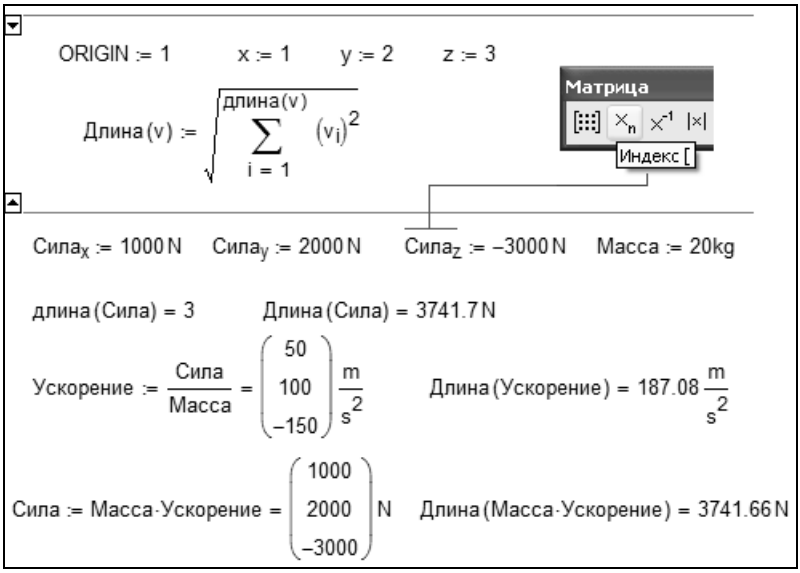


Рис. 2.2

Задача о движении тела в пространстве (Mathcad 15)

Второй тип данных, который в среде Mathcad имеет также форму вектора, но не является по сути (физической) таковым — *это массив данных в одном столбце*. Что это такое, мы поясним рисунком 2.3, где показано решение задачи линейной интерполяции по точкам.

Суть задачи такова. Имеются дискретные значения массовой доли (отношения массы растворенного вещества к массе раствора) водного раствора поваренной соли ω и его плотности ρ. Необходимо создать непрерывную функцию, которая бы возвращала значение плотности раствора соли в промежуточных точках. Такая функция нужна при пересчете видов концентраций — см. рис. 1.6 и 1.7 (стр. 22–24). Подробнее об этой статистической операции будет рассказано в этюде 14.

Координаты сглаживаемых точек в задаче на рис. 2.3 хранятся в двух массивах данных (векторах) с именами ω и ρ. Эти массивы по форме, повторяем, являются векторами, но это не векторы в «физическом» смысле того слова, который был подчеркнут на рис. 2.1 и 2.2. У векторов ω и ρ в задаче, показанной на рис. 2.3, конечно, можно посчитать длину (квадратный корень из суммы квадратов компонентов), но эта величина не будет иметь никакого физического смысла, того смысла, который ясно просматривается в задаче, показанной на рис. 2.1 и 2.2. На рис. 2.3 фактически показана *база данных* с двумя *полями* (ω и ρ) и с 8 *записями*, которую мы храним в виде двух векторов одного размера. Эту базу данных можно отображать графически, обрабатывать статистически (что и показано

[2] Авторов данной книги можно уподобить этим трем персонажам. Но если у Крылова «воз и ныне там», то авторам все-таки удалось издать книгу, несмотря на то, что они «тянули» ее в разные стороны: в сторону математики, в сторону физики и в сторону информатики.

на рис. 2.3), сортировать по полям, фильтровать по записям и т. д., но к какому-то n-мерному пространству эти «векторы» никакого отношения не имеют.

Плотность водного раствора NaCl при 20°C в зависимости от массовой доли

ω $\left(\frac{gm}{kg}\right)$	ρ $\left(\frac{gm}{cm^3}\right)$
0	0.998
1	1.005
2	1.013
3	1.020
10	1.071
15	1.109
20	1.148
27	1.2

$\rho_{NaCl}(x) := \mathrm{linterp}(\omega, \rho, x)$ $\quad x := 0\,\frac{gm}{kg}, 0.01\,\frac{gm}{kg} .. 27 \cdot \frac{gm}{kg}$

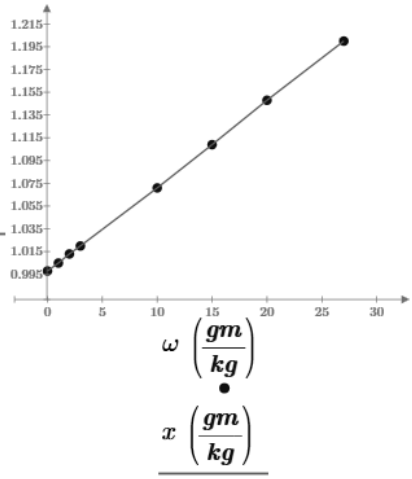


Рис. 2.3
Сглаживание точек в среде Mathcad

Примечание, касающееся и этюда 1. Через точки, показанные на рис. 2.3, можно провести прямую линию и описать зависимость плотности и концентрации водного раствора поваренной соли простой формулой $\rho = a + b \cdot \omega$ (см. описание этой процедуры в этюде 14). Но при этом будет потеряна физическая суть формулы, базирующаяся на таблице замеренных значений плотности и концентрации. Так что лучше оставить формулу (программу) в виде, показанном на рис. 2.3, вставив в функцию два вектора ω и ρ, и сделав ее тем самым автономной. В функции хорошо будет виден диапазон применимости формулы — от 0 до 27% по массовой доле, что помогает исключить ошибку экстраполяции.

Второе отличие массивов, показанных на рис. 2.1 и 2.2, с одной стороны, и на рис. 2.3, с другой стороны, в том, что нумерация *компонентов* вектора, как правило, начинается с единицы, а *элементов* массива — с нуля. На рисунке 2.3 эта установка (`ORIGIN = 0`) зафиксирована умолчанием. Третье отличие: составляющие векторов, показанных на рис. 2.1 и 2.2, обычно называют *компонентами* (координатами вектора), а векторов, показанных на рис. 2.3, — *элементами* (элементами матрицы-столбца). Но это не столь важно.

Как, наверное, уже отметил читатель, задачи, отображенные на рис. 2.1, 2.2 и 2.3, решаются с использованием *единиц физических величин*: метров (m), секунд (s), килограммов (kg) и др. Это очень мощный инструмент пакета Mathcad [4], позволяющий эффективно контролировать правильность расчетов (не складывать метры с килограммами, грубо говоря) и организовать удобный ввод-вывод численных значений переменных. Но тут встает интересный вопрос в технологическом и смысловом плане — может ли массив/вектор Mathcad иметь элементы с различной размерностью, когда, например, один элемент имеет размерность длины, второй — времени, а третий вообще безразмерный. Традиционный Mathcad, например, Mathcad 15 такой возможности не допускал. В среде Mathcad Prime это стало допустимым. Технологическая сторона вопроса тут решена, а что со смысловой стороной? Есть ли реальные математические задачи, решение которых требует использования векторов и массивов с различной размерностью элементов?! На

рисунке 2.4 такая задача показана: даны две точки на плоскости с координатами x_1–y_1 и x_2–y_2, необходимо найти значения коэффициентов a и b уравнения прямой $y(x) = a + b \cdot x$, проходящей через эти две точки. На рисунке 2.4 показано, как можно эту задачу решить в среде Mathcad Prime.

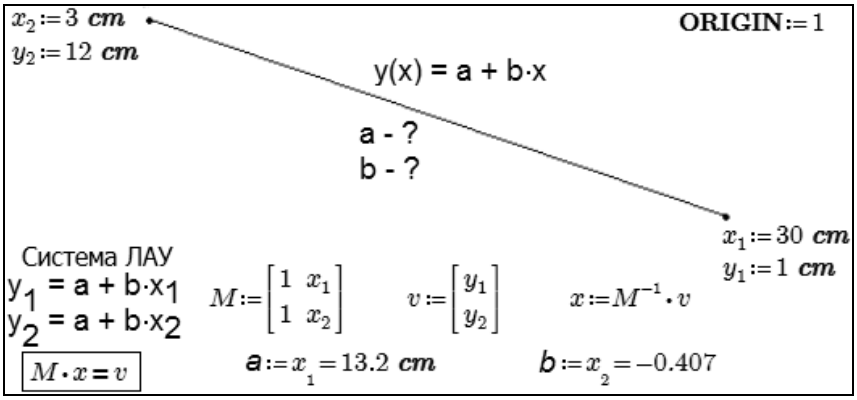


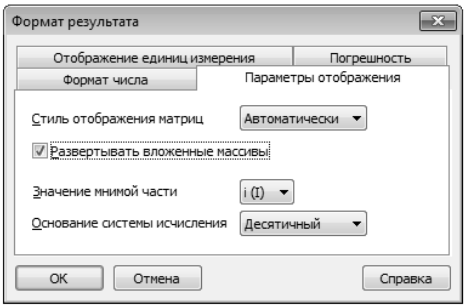
Рис. 2.4
Задача об уравнении прямой линии

Задачу об уравнении прямой линии, проходящей через две точки на плоскости[3], можно свести к решению системы двух линейных алгебраических уравнений (СЛАУ) в векторно-матричной форме `M·x = v`, где `M` — квадратная матрица коэффициентов при неизвестных `x`, а `v` — вектор свободных членов. Решение системы — это нахождение значений вектора `x`, состоящего из двух компонентов, первый из которых хранит величину с размерностью длины, а второй безразмерный. Матрица `M` также хранит разноразмерные величины, одни из которых (первый столбец) безразмерные, а другие (второй столбец) имеют размерность длины. Здесь также возникает вопрос — можно ли конструкции `v` и `x`, задействованные в задаче на рис. 2.4, называть векторами с тем «физическим» смыслом, который просматривается в этом слове в задаче, показанной на рис. 2.1 и 2.2. Нет, конечно! Тем не менее, в курсе высшей математики под названием «Линейная алгебра», один из разделов которого посвящен решению систем линейных алгебраических уравнений, конструкции x и `v` называют векторами, а конструкцию M — матрицей.

Вектор по критериям классического программирования (BASIC, Pascal и др. языки) — это просто одномерный массив, т. е. массив с одним индексом, а матрица — двумерный массив, т. е. массив с двумя индексами. В программировании используют и *многомерные* (трех-, четырех- и т. д.) массивы. Если скаляр можно уподобить точке, вектор — отрезку прямой линии, а матрицу — прямоугольнику, то трехмерный массив — параллелепипеду. В среде Mathcad эту объемную конструкцию можно создать в виде *вложенного массива*, пример которого показан на рис. 2.5, где показано хранение дискретных значений функции `f(x, y, z)`: `f = 0.81` при `x = 0,5, y = 15` и `z = 0.2, f = 0.48` при `x = 1, y = 30` и `z = 0.2` и т. д. На рисунке 2.5 показано также вывод на печать матрицы M в свернутом виде (информация в фигурных скобках — число строк и столбцов у свернутого вектора или матрицы) и вывод на печать двух отдельных элементов вложенного массива, которые были отмечены выше: `f(0.5, 15, 0.2) = 0.48` и `f(1, 30, 0.2) = 0.64`.

[3] Мы эту задачу будем решать в этюде 8 (см. рис. 8.6 на стр. 184), когда будем моделировать движение дворников автомобиля, закрепленных в двух точках.

$$M := \begin{bmatrix} \begin{pmatrix} \text{"x1="} \\ 0.2 \end{pmatrix} & \begin{pmatrix} \text{"y\z"} & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.8 & 1 \\ 5 & 0.63 & 0.56 & 0.49 & 0.42 & 0.35 & 0.28 & 0.21 & 0.14 \\ 10 & 0.7 & 0.56 & 0.49 & 0.42 & 0.42 & 0.49 & 0.56 & 0.63 \\ 15 & 0.7 & 0.63 & 0.56 & 0.49 & 0.49 & 0.49 & 0.42 & 0.42 \\ 20 & 0.56 & 0.56 & 0.49 & 0.49 & 0.42 & 0.42 & 0.35 & 0.35 \\ 30 & 0.49 & 0.42 & 0.35 & 0.42 & 0.49 & 0.42 & 0.35 & 0.35 \end{pmatrix} \\ \begin{pmatrix} \text{"x2="} \\ 0.5 \end{pmatrix} & \begin{pmatrix} 5 & 0.81 & 0.72 & 0.63 & 0.54 & 0.45 & 0.36 & 0.27 & 0.18 \\ 10 & 0.9 & 0.72 & 0.63 & 0.54 & 0.54 & 0.63 & 0.72 & 0.81 \\ 15 & 0.9 & 0.81 & 0.72 & 0.63 & 0.63 & 0.63 & 0.54 & 0.54 \\ 20 & 0.72 & 0.72 & 0.63 & 0.63 & 0.54 & 0.54 & 0.45 & 0.45 \\ 30 & 0.63 & 0.54 & 0.45 & 0.54 & 0.63 & 0.54 & 0.45 & 0.45 \end{pmatrix} \\ \begin{pmatrix} \text{"x3="} \\ 1 \end{pmatrix} & \begin{pmatrix} 5 & 0.72 & 0.64 & 0.56 & 0.48 & 0.4 & 0.32 & 0.24 & 0.16 \\ 10 & 0.8 & 0.64 & 0.56 & 0.48 & 0.48 & 0.56 & 0.64 & 0.72 \\ 15 & 0.8 & 0.72 & 0.64 & 0.56 & 0.56 & 0.56 & 0.48 & 0.48 \\ 20 & 0.64 & 0.64 & 0.56 & 0.56 & 0.48 & 0.48 & 0.4 & 0.4 \\ 30 & 0.56 & 0.48 & 0.4 & 0.48 & 0.56 & 0.48 & 0.4 & 0.4 \end{pmatrix} \end{bmatrix}$$



$$ORIGIN := 1$$

$$M = \begin{pmatrix} \{2,1\} & \{6,9\} \\ \{2,1\} & \{6,9\} \\ \{2,1\} & \{6,9\} \end{pmatrix}$$

$$(M_{2,2})_{3,3} = 0.81 \qquad (M_{3,2})_{2,3} = 0.64$$

Рис. 2.5
Пример вложенного массива в Mathcad

Вложенный массив — это вектор или матрица («простой» массив), один, два или более элементов которого это не скаляр, а новый вектор или матрица. Если говорить о «физике» решаемых в среде Mathcad задач, то тут нужно упомянуть *тензор*, который соответствует трех- и более мерному массиву. Типичный вложенный массив в среде Mathcad — это вектор, хранящий три матрицы — интенсивность трех цветов изображения (см. рис. 5.7 на стр. 111).

Если вернутся к началу этюда, где упомянут опыт общения авторов со студентами, изучающими информатику и математику в школе и вузе, то можно отметить такой феномен. Первый автор этой книги читает студентам курс информатики почти четверть века (см. **http://twt.mpei.ac.ru/ochkov/Potoki.htm**). Двадцать лет назад бывшие школьники, а теперь студенты были очень заинтересованы в изучении программирования, они, как они выражались, «тащились» от программирования. Теперь же у многих студентов на программирование возникла некая «аллергия». Причин у этого печального явления, как можно предположить, две. Первая в том, что это общешкольная проблема, связанная с неприятием того, что преподают в школе. У многих людей, например, наблюдается стойкое отвращение к литературным произведениям, которые изучали в школе. Двадцать лет назад программирование в школе почти не изучали, поэтому в те времена у студентов был огромный интерес к этому предмету. А теперь его, увы, нет.

Вторая причина потери у школьников, а потом и у студентов интереса к программированию заключается в том, что преподаватели часто дают студентам для программирования уж очень «занудные» задачи. Эти задачи в прошлом веке, может быть, были сами по себе интересны. Но сейчас для решения таких задач появились новые средства, исключающие традиционное программирование. И школьники-студенты это интуитивно чувствуют или даже прямо знают об этом.

Конкретный пример. На рисунке 2.2 можно видеть, как современными средствами Mathcad (оператором суммы) была создана функция пользователя с именем «длина», возвращающая «физическую» длину вектора. Но многие преподаватели информатики в школе и в вузе по-прежнему предлагают школьникам и студентам создавать такие и другие подобные примитивные функции средствами традиционного программирования (BASIC, Pascal, C и др.), не зная или умалчивая о том, что в современных программных оболочках многое уже давно реализовано встроенными функциями и операторами. Кстати, функцию пользователя с именем

«длина» можно создать средствами программирования Mathcad (см. рис. 2.6) с помощью цикла for (цикла с параметром).

```
ORIGIN ≡ 1

Length(v) := | "Длина вектора"
             | Sum ← 0
             | for  i ∈ 1 .. длина(v)
             |   Sum ← Sum + (v_i)^2
             | return √Sum

Сила := ( 1000 N )
        ( 2000 N )
        ( -3000 N )

длина(Сила) = 3
Length(Сила) = 3741.66 N
```

Рис. 2.6
Программа в среде Mathcad

В Mathcad есть, например, очень интересная функция `match(z, A)`, возвращающая координаты элемента со значением `z` в матрице `A`. Если элементов со значением `z` в матрице `A` окажется несколько, то функция `match` вернет тот самый вложенный массив (см. рис. 2.5) — вектор, элементы (или компоненты) которого будут новыми векторами с двумя элементами — с номером строки и номером столбца матрицы A, где «сидит» величина `z`. Эту функцию мы будем использовать в этюде 15 в программе решения задачи коммивояжера (см. рис. 15.10 на стр. 345). Можно, конечно, поручить школьникам или студентам создать функцию `match` через программирование с циклами и альтернативами. Но лучше давать студентам более занимательные задачи, те задачи, у которых решения еще нет или оно еще не так «вылизано». Заставлять студентов создавать функцию `match` это все равно, что заставить их написать программу расчета синуса через разложение в ряд Тейлора, умалчивая при этом, что синус есть в любом калькуляторе и встроен в любой язык программирования.

Эту проблему можно поставить шире и перенести ее на другие школьные и вузовские дисциплины — физику [31], математику [32], химию [5].

Дело в том, что изучение многих учебных дисциплин ориентировано на докомпьютерные и даже докалькуляторные методы решения задач. И это мы отмечали во Введении и этюде 1. Возьмем, к примеру, физику, с которой начался данный этюд. Решая задачу по этой дисциплине, школьник или студент должен вспомнить или найти в учебнике формулу или набор формул, по которой можно решить конкретную задачу. Сами же эти формулы являются, как правило, частными случаями решения систем уравнений (алгебраических, дифференциальных, интегральных и др.), отображающих фундаментальные законы сохранения вещества, энергии; перехода энергии из одной формы в другую и т. д. и т. п. Упор на использование готовых решений, а не на постановку задачи в общем виде был сделан когда-то давно по простой и понятной причине. В те времена, когда создавались методики решения этих задач, не было эффективных и доступных средств решения уравнений и систем, отображающих фундаментальные законы физики, химии и других научных (учебных) дисциплин. Теперь же такие средства появились. И это требует кардинального пересмотра методик решения задач и переписывания почти всех учебников и задачников. Более подробно мы эту тему описали в этюде 1.

Конкретный пример. Нужно рассчитать, на какую высоту поднимается брошенный вертикально вверх камень [31]. Решить данную задачу можно через «выуженную» из памяти или справочника (учебника, задачника) формулу, где эта высота высчитывается с учетом начальной скорости и ускорения свободного падения. Если же задачу усложнить — бросать камень не строго вертикально, а под

некоторым углом к вертикали, то нужно будет искать новые формулы. Но можно подойти к задаче по-иному — видеть в ней систему дифференциальных уравнений и решать ее аналитически (получить те же формулы с частными решениями) или численно.

В этом случае не потребуется искать формулы — нужно будет только знать фундаментальные законы природы, уметь переводить их на язык математики (составлять те же дифференциальные и прочие уравнения) и решать их современными компьютерными средствами. Еще одно преимущество данного подхода состоит в том, что с задач можно снять многие ограничения, снижающие ценность результатов. Так, в задаче о полете камня можно учесть сопротивление воздуха, изменение его плотности по высоте, изменение значения ускорения свободного падения по высоте, кривизну земной поверхности и многое другое. Все это можно учесть, если перейти от использования набора формул к решению исходных уравнений. На рисунке 2.7 в качестве примера показано, как можно «изящно» решить задачу о разгоне автомобиля, имеющего массу m, площадь поперечного сечения s и движущемся в воздухе с плотностью ρ при коэффициенте трения k.

В задаче при необходимости можно также учесть изменение движущей силы N во времени, сопротивление качения колес по дороге и другие факторы. В этюде 4 будет решена подобная задача для моторной лодки (см. рис. 4.21 на стр. 82), а в этюде 9 для спутника Земли (см. рис. 9.20 на стр. 219, рис. 9.21 и 9.22 на стр. 220).

И последнее. Один из авторов когда-то рассказывал учителям одной очень хорошей московской школы о своих наработках по использованию математических пакетов в преподавании математики, физики, химии. Так вот, эти учителя сказали, что все это очень хорошо и заманчиво, но наша школа сейчас называется по-другому — не «Средняя общеобразовательная школа номер такой-то», а «Курсы по подготовке к сдаче ЕГЭ по математике, физике, химии, информатике...» Вопросы же для ЕГЭ специально составляются так, чтобы при ответах на них не нужно было использовать не только компьютер, но даже и простейший калькулятор... Хотя в настоящее время студент, инженер, ученый и даже простой офисный работник не может обойтись без этих вычислительных средств. (Это к вопросу об отрыве школы от реальной жизни.) Раньше, до эры ЕГЭ учитель мог ориентироваться на сильных в математике учащихся и давать на уроках интересные, творческие задачи, выходящие за рамки учебников. Слабым же ученикам можно было, несколько кривя душой, ставить «тройки», особо не напрягая их. Главное, чтобы они не мешали остальным. Теперь же приходится всех «стричь под одну гребенку»: натаскивать всех без исключения на сдачу ЕГЭ, по результатам которых судят о работе самого учителя и о шансах учеников на поступление в тот или иной вуз. Это еще один камень в огород ЕГЭ. Да, система ЕГЭ подтянула слабых учителей и учеников к среднему уровню. Но она также, увы, опустила сильных учителей и учеников к этому самому пресловутому среднему уровню.

Да, для страны, являющейся сырьевым придатком развитых стран, такая образовательная модель является оптимальной. Но если же мы не на словах, а на деле хотим слезть с «нефтяной иглы» и сохранить статус научной державы, то образовательную модель, которую нам навязывают чиновники от образования, нужно менять.

Представим себе такой ЕГЭ по математике, физике, химии... Тестируемый (школьник, студент или кто-то другой, занимающийся самообразованием, в том числе и через Интернет) подсаживается к компьютеру, который выдает задание. Выполнить же это задание можно всеми возможными способами: решить задачу в уме или на листе бумаги, решить задачу в среде какой-нибудь компьютерной программы, поискать аналогичные решения в Интернете, вывесить задание на каком-либо форуме и ждать ответа, послать задание другу, организовать голосование Интернет-сообщества по вариантам ответов, если они даны и т. д. и т. п. Компьютер же после всестороннего анализа действий тестируемого должен сам решить, какую оценку ему поставить.

N	m	S	ρ	k	v_0	t_{end}
(kgf)	(kg)	(m^2)	$\left(\frac{kg}{m^3}\right)$	$\left(\frac{N}{N}\right)$	(kph)	(s)
300	1200	2.5	1.2	2	0	30

Solve

Ограничения

$v(0\ s) = v_0$

Инерция + Сопротивление воздуха = Сила мотора

$$m \cdot v'(t) + k \cdot \rho \cdot S \cdot v(t)^2 = N$$

Решатель

$v := \mathbf{odesolve}(v(t), t_{end})$

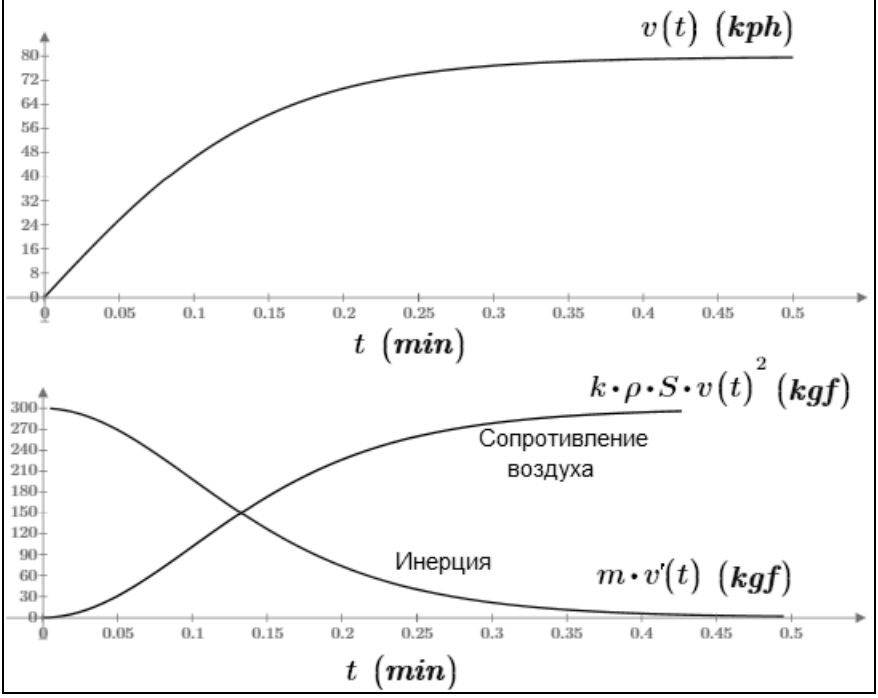


Рис. 2.7
Задача о старте автомобиля

Без четкого понимания таких базовых понятий, как скаляр, вектор, матрица, таблица, массив (одномерный, двумерный и многомерный), база данных, база знаний и т. д. освоение информатики будет идти большим трудом. На «курсах по подготовке к сдаче ЕГЭ по математике, физике, химии, информатике...», т. е. в средней школе можно и нужно организовать кружки под названием «Курсы по подготовке к сдаче ЕГЭ по математике, физике, химии, информатике... с помощью компьютера», на которых можно будет разбирать задачи, подобные тем, ка-

кие приведены в этой книге. Кстати, в этюде 14 будет рассмотрена корреляция итогов сдачи ЕГЭ по математике и реальных математических знаний первокурсников [33].

Задания для читателей

1. В треугольнике, показанном на рис. 13.1 (стр. 293), стороны 1–2, 2–3 и 1–3 рассматривать как векторы, выходящие из соответствующих вершин треугольника. Определите углы между этими векторами, используя встроенные функции Mathcad работы с векторами и матрицами.

2. Дополнить анимации, описанные в этюдах 8 и 9, графическим изображением векторов скорости перемещаемых объектов.

3. В Интернете можно найти задачи на равновесие разных плоских и объемных систем — см., например,

http://fiziku5.ru/uchebnye-materialy-po-fizike/ravnovesie-proizvolnoj-ploskoj-sistemy-sil

На рисунке 2.8 схематично показана одна такая задача. Решите эту задачу в среде Mathcad и нарисуйте в реальном масштабе вектора сил в опорных точках. Придумайте такую же, но объемную задачу.

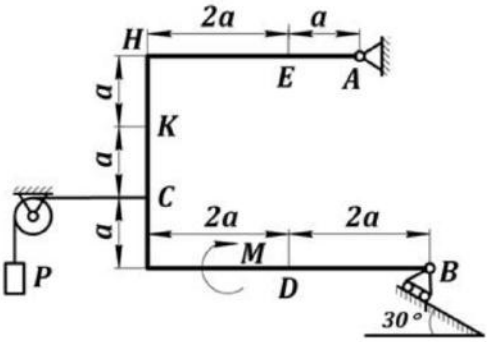


Рис. 2.8
Задача на равновесие сил

Этюд 3

ЗОЛОТОЙ ЖУК, ИЛИ РИМСКО-АРАБСКАЯ КРИПТОГРАФИЯ

В этюде будет рассказано о системах счисления, применяемых в математике, о работе с переменными, хранящими цепочки символов (тексты), о простых и сложных методах шифрования информации.

Математика: системы счисления: позиционные и непозиционные, вероятность, система линейных алгебраических уравнений (СЛАУ), дроби, наибольший общий делитель (НОД), простые множители, производная.

Физика: нагрев пергамента.

Информатика: двоичный, восьмеричный, десятеричный и шестнадцатеричный формат чисел, цикл с предпроверкой, строковые переменные и функции работы с ними.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127827**

Система счисления — это способ записи чисел и соответствующие ему правила действий над ними. Люди придумали несколько систем счисления. Провести их классификацию довольно сложно, поскольку сравнивать их приходится по разным свойствам.

Первое свойство — это нотация, т. е. набор символов, служащий для записи чисел. Нотация может быть цифровой (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), алфавитной (I, V, X, L, C, D, M), значковой (иероглифы, клинопись и другие значки), а также смешанной (цифры и буквы или буквы со значками). Например, в древнерусской системе для записи чисел использовались буквы, дополненные различными значками — тильдами, стрелочками, сплошными и прерывистыми окружностями и т. п. В начале XVIII в. на Руси применялась смешанная система записи чисел, состоящая из кириллических и арабских цифр. Например, на некоторых медных полушках (монетах достоинством 1/4 копейки) отчеканены такие даты 17К (1720) и 17К1 (1721).

Второе свойство — это основание системы счисления, т. е. тот базисный набор чисел (в различной нотации), которого достаточно для выражения всех числовых величин, практически значимых для жизни человека соответствующей цивилизации и культуры. Известны единичная (одна палочка), двоичная (0 и 1), пятеричная, восьмеричная, десятичная, двенадцатеричная, двадцатеричная, шестидесятеричная и другие системы счисления. Например, в шестнадцатеричной системе основанием служит число 16, которое обозначается как «10», а ряд натуральных чисел от 1 до 16 записывается так: 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10.

Третье свойство — это набор правил действий с символами записи числа. Различают аддитивные и мультипликативные системы.

Четвертое свойство — это значение символа записи числа. Есть системы непозиционные. В них значение символа не зависит от его положения в записи числа. Например, в римском числе XX (двадцать) символ X встречается дважды и в каждом случае обозначает одну и ту же величину — число 10, два раза по 10 в сумме дают 20. Но есть системы позиционные. В них значение символа однозначно зависит как от его положения в записи числа, которая является поразрядной, так и от правила чтения числа (слева направо или наоборот). В этом случае «вес» каждого символа изменяется в зависимости от его положения (позиции) в последовательности символов, изображающих число. Например, в арабском десятичном числе 353.3 первая тройка означает три сотни, вторая — три единицы, а третья — три десятых доли единицы. В некоторых позиционных системах счисления есть особые правила прочтения и изображения числа. Так, в древнерусской глаголической системе счисления числа за исключением чисел от 11 до 19, которые произ-

носились справа налево (например, 17 — семь-на-двадцать). Некие «сбои» в десятичности можно наблюдать и в настоящее время. Современные немцы, например, говорят не «двадцать один», а «один и двадцать», не «семьдесят четыре», а «четыре и семьдесят». Вспомним знаменитую «древнегреческую фразу» Аркадия Райкина «Айн унд цванцих, фир унд зибцих». Тот же немец никогда не скажет «тысяча двести», а скажет «двенадцать сотен». Французы число 75 выговаривают как 60 и 15. И таких примеров в разных языках можно привести множество.

Примитивная система счета с помощью счетных палочек является непозиционной, единичной и аддитивной. Какую бы палочку из набора мы не взяли, она всегда обозначает «один». Но эту систему можно усложнить, раскрасив палочки в различные цвета и придав разным цветам разный »вес».

Примерно в третьем тысячелетии до нашей эры древние египтяне придумали свою числовую систему, в которой для обозначения ключевых чисел 1, 10, 100 и т. д. использовались специальные значки — иероглифы. Все остальные числа составлялись из этих базовых чисел при помощи операции сложения. Система счисления Древнего Египта является десятичной, непозиционной и аддитивной.

Одна из важных тем уроков информатики в школе — это системы счисления: *двоичная*, *восьмеричная*, *десятичная*, *шестнадцатеричная*... Школьники должны уметь, как минимум, перевести десятичное число в двоичное и наоборот. Пакет Mathcad 15 имеет встроенные инструменты для работы с такими числами. На рисунке 3.1 можно видеть число, составленное из трех чисел в разной форме представления (`2.73o` — восьмеричное число, `9h` — шестнадцатеричное число и `100b` — двоичное число). Оно выведено «на печать» в четырех встроенных в Mathcad форматах — двоичном, восьмеричном, десятичном и шестнадцатеричном. Если число не заканчивается суффиксом b (**b**inary — двоичное), o (**o**ctal — восьмеричная) или h (**h**exadecimal — шестнадцатеричная), то это десятичное число — см. первый ответ на рис. 3.1.

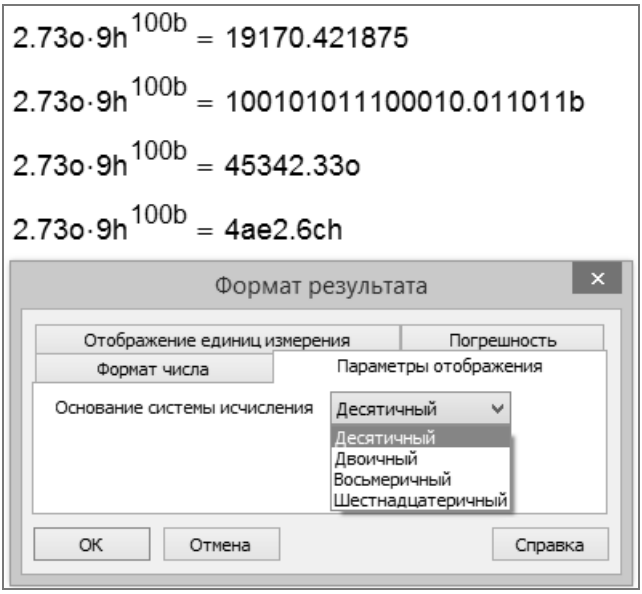


Рис. 3.1
Системы исчислений в среде Mathcad 15

С шестнадцатеричной формой записи числа связана такая интересная ошибка: `h := 3 2h = 2`. В чем тут дело, почему 2, умноженное на 3 (на переменную `h`), равно двум, а не шести. Многих это приводит в недоумение и заставляет предпола-

гать, что или пакет Mathcad сломался или он сам... go to bugs, как говорят американцы, — сошел с ума. На форуме PTC Community по адресу **https://www.ptcusercommunity.com/groups/mathcad-paradoxes** открыт авторский сайт «Mathcad Mad House — Сумасшедший дом Mathcad», где собраны подобные казусы[1]. Дело в том, что выражение `2h` — это не только двойка, умноженная на то, что хранится в переменной `h` (а Mathcad тем и хорош, что он подстраивается под общематематическое правило, разрешающее не ставить знака умножения между константой и переменной), но и просто числовая константа — двойка в шестнадцатеричном формате, то есть просто двойка в десятичном формате. Отсюда вывод — знак умножения нужно ставить и там, где его математики обычно не ставят — между числовой константой и переменной.

В среде Mathcad 15 несложно создать пользовательскую функцию перевода записи чисел из одной системы счисления в другую. На рисунке 3.2 показан такой расчет для целых чисел, открытый в Интернете и позволяющий представлять числа в различных системах счисления: от двоичной до шестнадцатеричной. Так, троичное число 020202 эквивалентно семеричному числу 350, что видно из рис. 3.2. Этот расчетный сайт создан в русле идеи, изложенной в этюде 1, — выдаем не только результат счета, но и все формулы, по которым они выполнены.

С числами в двоичном, восьмеричном и прочих представлениях в обыденной жизни мы дела почти не имеем[2]. Решая задачу на компьютере или просто с помощью калькулятора, мы вводим привычные десятичные числа. Компьютер переводит их в двоичный формат, проводит с ними нужные действия, получает двоичный ответ, который переводится в десятичный формат и выводится на экран дисплея.

Поговорим теперь о нотациях (т. е. обозначениях) чисел. Более привычным для нас является *римское* представление чисел — *римские числа*. С ними, как это не покажется странным, мы имеем дело очень часто — практически каждый день. И не только тогда, когда смотрим на часы с римскими числами на циферблате, но и тогда, когда... снимаем деньги с карточки в банкомате, например. В банкомат мы вводим арабское число (7 126 руб., например), а банкомат выдает нам... римское число (рис. 3.3): пять тысяч, две тысячи, сотню, две десятки, пятерку и рубль[3]. «Римский» счет денег мы ведем и тогда, когда нас просят записать в финансовых документах денежную сумму не числами, а словами, прописью: «Семь тысяч сто двадцать шесть рублей».

На рисунке 3.3 показана программа-функция, написанная для Mathcad Prime, которая переводит сумму денег в набор банкнот и монет[4]. Если в этой программе матрицу денег заменить на матрицу римских чисел (рис. 3.4), то она будет переводить арабские числа в римские.

---

[1] Другой подобный очень распространенный казус с оператором умножения такой. Если ввести два оператора `f := 2 f(1+2)` и после второго оператора нажать клавишу «=», то мы получим не численный ответ (6), а сообщение об ошибке: пакет Mathcad посчитает, что мы пытаемся вызвать пользовательскую функцию с именем `f`, а не перемножить значение переменной с именем `f` на то, что записано в скобках. Пакет Mathcad часто дописывает за пользователя знак умножения, но эта услуга иногда оказывается медвежьей.

[2] Поэтому-то наверно в Mathcad Prime пока нет недесятичных форматов ввода и вывода чисел.

[3] Банкомат такие мелкие деньги, конечно, не выдает. Пусть это будет, например, продавщица в магазине или автомат по продаже билетов на электричку, выдающий сдачу.

[4] О банкнотах и монетах, а также о других интересных числовых рядах будет рассказано в этюде 10.

http://twt.mpei.ac.ru/MCS/Worksheets/base2base.xmcd

$Num_{in} :=$ 020202 $\quad base_{in} :=$ 3 $\quad base_{out} :=$ 7 $\quad$ Recalculate

$Num_{in} := if(IsString(Num_{in}), Num_{in}, num2str(Num_{in}))$

$Num_{out} := base2base(Num_{in}, base_{in}, base_{out}) \qquad Num_{out} = \text{"350"}$

```
base2base(Num_in, base_in, base_out) ≡ | NBD ← "0123456789ABCDEF"
                                       | InputNumberLength ← strlen(Num_in)
                                       | NBD_base ← substr(NBD, 0, base_in)
                                       | DecimalValue ← 0
                                       | for  j ∈ 0 .. InputNumberLength − 1
                                       |   | k ← substr(Num_in, j, 1)
                                       |   | cif ← search(NBD_base, k, 0)
                                       |   | error("Found a digit in the number that isn't in this base")  if  cif = −1
                                       |   | DecimalValue ← DecimalValue + floor(cif·base_in^(InputNumberLength−j−1) + 1/2)
                                       | Out ← ""
                                       | while  DecimalValue > 0
                                       |   | X ← floor(mod(DecimalValue, base_out))
                                       |   | Out ← concat(substr(NBD, X, 1), Out)
                                       |   | DecimalValue ← floor(DecimalValue / base_out)
                                       | Out
```

Рис. 3.2

Сайт для перевода записи чисел из одной системы счисления в другую

```
f(n) := | M ← [ 5000   1000   500   100   50   10   5   2   1
                "5000p" "1000p" "500p" "100p" "50p" "10p" "5p" "2p" "1p" ]
        | R ← ""
        | while n > 0
        |   | while n ≥ M_(0,i)
        |   |   | R ← concat(R, M_(1,i))
        |   |   | n ← n − M_(0,i)
        |   | i ← i + 1
        | R
```

$f(7126) = \text{“5000p 1000p 1000p 100p 10p 10p 5p 1p ”}$

Рис. 3.3

Программа-функция для разложения суммы денег на банкноты и монеты

$$Arab2Rim(n) := \left\| \begin{array}{l} M \leftarrow \begin{bmatrix} 1000 & 900 & 500 & 400 & 100 & 90 & 50 & 40 & 10 & 9 & 5 & 4 & 1 \\ \text{“M”} & \text{“CM”} & \text{“D”} & \text{“CD”} & \text{“C”} & \text{“XC”} & \text{“L”} & \text{“XL”} & \text{“X”} & \text{“IX”} & \text{“V”} & \text{“IV”} & \text{“I”} \end{bmatrix} \\ \text{Далее как на рисунке 3} \end{array} \right\|$$

$$Arab2Rim(1828) = \text{“MDCCCXXVIII”} \qquad Arab2Rim(2014) = \text{“MMXIV”}$$

Рис. 3.4

Перевод арабского числа в римское

Алгоритм перевода арабского числа в римское довольно прост [1, 34, 35]: из арабского числа n изымаются «римские банкноты и монеты» (вторая строка матрицы `M`) до тех пор, пока «арабские» деньги не кончатся (`while n > 0`). При этом проверяется, есть ли для этой операции соответствующая i-я банкнота или монета (`while` $n \geq M_{0,i}$). Встроенная в Mathcad функция concat собирает воедино цепочку символов — номинал банкнот и монет.

Сложнее написать обратную функцию — функцию перевода римского числа в арабское. Такую операцию мы проводим в уме, когда видим старинное здание с «зашифрованным» годом постройки в виде римского числа на фронтоне или памятник с «римскими» датами рождения и смерти. Задачу можно упростить, если воспользоваться прямой (исходной) функцией перевода арабского числа в римское (рис. 3.3 и 3.4): перебираются все арабские числа от единицы, далее они переводятся в «римский» вид, который сравнивается с исходным римским числом. Если произошло совпадение, то найденное арабское число выводится «на печать» (см. рис. 3.5а).

$$Rim2Arab(R) := \left\| \begin{array}{l} n \leftarrow 1 \\ \text{while } Arab2Rim(n) \neq R \\ \quad \| \; n \leftarrow n+1 \\ n \end{array} \right\|$$

$$Rim2Arab(\text{“MDCCCXXVIII”}) = 1828$$

$$Rim2Arab(\text{“MMXIV”}) = 2014$$

a

```
RA(n) := | AV ← (1000 900 500 400 100 90 50 40 10 9 5 4 1)^T
         | RV ← ("M" "CM" "D" "CD" "C" "XC" "L" "XL" "X" "IX" "V" "IV" "I")^T
         | (A ← 0  i ← 0)
         | while  strlen(n) > 0
         |   | while  RV_i = substr(n, 0, strlen(RV_i))
         |   |   (A ← A + AV_i  n ← substr(n, strlen(RV_i), strlen(n) − strlen(RV_i)))
         |   | i ← i + 1
         | return A

RA("MCMXLVIII") = 1948      RA("MCMXCVIII") = 1998      RA("MCMXXVIII") = 1923  ???
```

b

Рис. 3.5

Перевод римского числа в арабское:

a — с использованием ранее заданной прямой функции; *b* — без использования ранее заданной прямой функции (почему такой вид — см. первое задание читателям в конце этюда).

Один из авторов видел у себя на даче[5] одно необычное использование римских чисел. На соседнем участке строилась баня — собирался сруб из бревен, которые были помечены зарубками: парами римских чисел: I, II, III, IV и т. д. Первая зарубка означала номер стены бани от первой (I) до четвертой (IV), а вторая — номер бревна в этой стене от самого нижнего до самого верхнего. Этот сруб был изготовлен где-то далеко и перевезен на новое место. Перед его разборкой плотники пометили стены... римскими числами потому, что арабские числа топором или «болгаркой» написать было бы довольно сложно. Это были довольно грамотные плотники. Ведь можно было сделать просто зарубки I, II, III, IIII[6], IIIII, IIIIII и т. д.

К школьной форме записи чисел можно отнести и простые дроби, о которых мы ещё поговорим (см. рис. 3.7 ниже).

Римские числа можно рассматривать как некие зашифрованные арабские числа. В связи с этим можно предложить школьникам на уроках информатики такую игру: создать матрицу, подобную той, что показана на рис. 3.4, и по ней шифровать арабские числа, которые другие школьники будут пытаться расшифровывать [36].

Информатика — это наука о работе с информацией. В частности, о ее защите от посторонних взглядов. Защищать письменные тексты помогает раздел информатики под названием *криптография* или *тайнопись* [37]. Расшифровка одного такого криптографического текста красочно описана в рассказе Эдгара По «Золотой жук». Этого американского писателя по праву называют родоначальником детективного жанра в литературе.

Сюжет рассказа незамысловат. Один человек находит на берегу моря пергамент, приносит его домой и случайно оставляет у огня. На пергаменте от нагрева проступают таинственные символы:

53‡‡†305))6*;4826)4‡.)4‡);806*;48†8¶60))85;1‡(;:‡*8†83(88)5*†;46(;88*96*?;8)*‡(;485);
5*†2:*‡(;4956*2(5*—4)88*;4069285);)6†8)4‡‡;1(‡9;48081;8:8‡1;48†85;
4)485†528806*81(‡9;48;(88;4(‡?34;48)4‡;161;:188;‡?;

Подобные цепочки «таинственных» символов мы иногда можем видеть на дисплее компьютера, когда выбрана неверная кодировка текста.

Человек, нашедший пергамент, расшифровывает текст и находит по нему запрятанный пиратами клад.

Шифр был основан на замене букв исходного текста другими буквами или символами[7]. Это один из самых древних и самых простых способов шифрования текстов. К настоящему времени разработано большое количество систем шифрования различной степени сложности. Новый импульс этому процессу дало развитие компьютерных информационных технологий. Пересылая информацию даже в пределах компьютера, мы ее «шифруем» — переводим привычные буквы и цифры в нули и единицы: в цепочки «таинственных символов». Компьютеры, пересылая информацию друг другу по локальным сетям или по Интернету, могут использовать специальные алгоритмы шифрования для того, чтобы передаваемую информацию не мог прочесть тот, кому это не положено.

Расшифровка текста, основанного на замене символов, состоит из двух этапов — рутинного и творческого. Рутинный этап — это подсчет частоты использования тех или иных символов в исходном тексте. Герой рассказа Эдгара По сделал это вручную. Мы же можем подсчитать это на компьютере, написав несложную программу на языке Mathcad, — см. рис. 3.6.

[5] Дача автора — это один из источников тем этюдов этой книги: см. этюд 7 про кирпич, подвешенный на электрические провода, или этюд 12 про пожарное ведро.

[6] Римскими числами иногда четыре пишут как `IIII`, но никогда сорок не пишут как `XXXX`.

[7] На такой замене базируются некоторые головоломки — см. этюд 18.

```
Текст := "53‡‡†305))6*;4826)4‡.)4‡);806*;48†8¶60))85;1‡(;:‡*8†83(88)5*†;46(;88*96*?;8)*‡(;4

v := | [i ← 1   v[1,1] ← substr(Текст, 1, 1)   v[1,2] ← 1]                    strlen(Текст) = 203 ⇨
     | for j ∈ 2..strlen(text)
     |   | [Символ ← substr(Текст, j, 1)   НовыйСимвол ← "Да"]
     |   | for k ∈ 1..i
     |   |   | if Символ = v[k,1]
     |   |   |   | [НовыйСимвол ← "Нет"   v[k,2] ← v[k,2] + 1]
     |   | if НовыйСимвол = "Да"
     |   |   | [i ← i + 1   v[i,1] ← Символ   v[i,2] ← 1]
     | csort(v, 2)

v^T = | "–"  "."  "¶"  "?"  ":"  "3"  "9"  "2"  "0"  "†"  "1"  "("  "6"  "5"  "*"  "‡"  ")"  "4"  ";"  "8" |
      |  1    1    2    3    4    4    5    5    6    8    8   10   11   12   13   16   16   19   26   33 |
```

Рис. 3.6

Программа подсчета частоты встречи символов в тексте

В переменную `Текст` (см. рис. 3.6) вставлена (скопирована из Интернета) цепочка таинственных символов из рассказа Эдгара По. Показано только начало этой цепочки, в которой 203 символа. Подсчет числа символов в переменной, хранящей строку, ведет встроенная в Mathcad функция `strlen` (str — string, строка и len — length, длина). Нам понадобится еще одна строковая функция Mathcad — функция `substr` (подстрока), возвращающая часть строки, начиная с нужного места (второй аргумент функции `substr`), и нужной длины (третий аргумент). Эти три встроенные функции использовались также и в программе перевода римского числа в арабское — см. перевернутый рис. 3.5*b*.

В программе, показанной на рис. 3.6, в матрице `v` будут формироваться два столбца — сам символ и частота его встречи в тексте. Первой строкой программы записывается первая строка этой матрицы — в нашем случае цифра 5 и единица. Далее в программе циклом с параметром `j` ведется перебор переменной-строки `Текст` со второго его элемента (первый уже зафиксирован в матрице `v`) до последнего. Принимается, что очередной символ текста, зафиксированный в переменной `Символ`, ранее не встречался: в переменную `НовыйСимвол` заносится значение «`Да`». Во втором (вложенном) цикле с параметром `k` ведется проверка этого предположение перебором ранее сформированных строк матрицы `v`. Если окажется, что очередной символ уже встречался в тексте (`Символ = v`$_{k,1}$) и наше предположение станет ошибочным (переменная `НовыйСимвол` принимает значение «`Нет`»), то значение элемента матрицы `v`$_{k,2}$ увеличивается на единицу. Если же очередной символ из переменной `Текст` в матрице `v` еще не зафиксирован (переменная `НовыйСимвол` сохранила значение «`Да`» по выходу из цикла с параметром `k`), то в матрице `v` создается новая строка (`i ← i + 1`), куда заносится сам новый символ (`v`$_{i,1}$ ← `Символ`) и единица (`v`$_{i,2}$ ← `1`). Возвращает программа отсортированную и транспонированную матрицу `v` — см. нижнюю часть рисунка.

Наш шифрованный текст, как уже было посчитано, содержит 203 знакомест. Отдельных же символов в тексте всего 20, из которых наиболее часто (33 раза) встречается восьмерка. Такой короткий текст можно, конечно, проанализировать и без компьютера, как это сделал герой рассказа Эдгара По (рассказ опубликован впервые в 1848 г.). Но через нашу программу можно пропустить более объемные тексты. Это позволит узнать, какие буквы в том или ином языке используются чаще, а какие реже. Эту же работу можно провести, беря за основу не буквы, а от-

дельные слова. В английском языке, на котором был написан рассказ «Золотой жук», самая «частая» буква — это буква е, а самое частое слово — это определенный артикль the. Это и послужило ключом к расшифровке текста в рассказе Эдгара По (творческая часть дешифровки). Как?! Прочтите или перечитайте рассказ еще раз [38].

Рассказ Эдгара По «Золотой жук» в свое время (середина девятнадцатого века) вызвал большой интерес общества к криптографии. Тут можно упомянуть и знаменитый рассказ Артура Конан Дойля «Пляшущие человечки» [39], сюжет которого также основан на расшифровке цепочек символов.

Но если раньше тайнописью занимались в основном только военные и дипломаты, то в наше время с этим важным разделом информатики прямо или косвенно приходится сталкиваться всем и повсеместно, разговаривая по сотовому телефону или снимая деньги с банкомата...

Мы надеемся, что наша простенькая программа, помогающая дешифровке простого кода, станет некой стартовой точкой для изучения более сложных алгоритмов криптографии. А пока можно поиграть в такую игру. Двое разрабатывают для переписки код, основанный на замене одних символов другими, а третий «перехватывая» и анализируя эти сообщения, раскрывает «шифровки». В таких сообщениях, как правило, опускаются пробелы, знаки препинания и игнорируются большие буквы[8]. Это было сделано в шифровке рассказа Эдгара По. В рассказе же Артура Конан Дойля пляшущий человечек с флажком в руке отмечал начало слова. Сколько понадобится для этого перехваченных сообщений и времени — это отдельный вопрос. Вернее, задание на создание программы для раскрытия ключа шифра. Часть этой программы показана на рис. 3.6.

Шифр из рассказа Эдгара По основан на замене букв исходного текста другими буквами или символами. Это один из самых древних и самых простых способов шифрования текстов. Такой шифр является **шифром замены**. А вот в **шифре перестановки** символы меняют местами, соблюдая некоторую закономерность.

К настоящему времени, повторяем, разработано огромное количество систем шифрования различной степени сложности. В работе «Математическая теория секретной связи» Клод Шеннон обобщил накопленный до него опыт разработки шифров. Он выяснил, что в любых сложных шифрах в качестве типичных компонентов можно выделить шифры замены, а также шифры перестановки или их сочетания.

При использовании цифровых компьютеров удобно представлять информацию в виде последовательностей нулей и единиц. С другой стороны, слова в любом алфавите можно легко перевести в двоичные слова. Рассмотрим, например, телеграфный код — старое техническое применение двоичной системы счисления. Он состоит из 32 символов — двоичных слов длины 5. Сопоставим каждой букве двоичное слово длины 5 следующим образом:

пробел → 00000, А → 00001, Б → 00010, В → 00011, Г → 00100, Д → 00101, ..., Ю → 11110, Я → 11111.

Легко составить программу по телеграфной шифровке и дешифровке. Поручим это сделать читателю.

Вопрос о том, какова наименьшая длина слова при телеграфной шифровке, для каждого алфавита решается очень просто. Каждая буква в алфавите имеет свой порядковый номер. В латинском алфавите (базовый вариант) всего 26 букв. В двоичной записи имеем $26 = 16+8+2 = 2^4+2^3+2^1 = 11010$. Снова мы получили пятисимвольное двоичное слово. А сколько же должно быть букв в алфавите, чтобы телеграфный код содержал только четырехбуквенные слова? Это легко сосчитать: $1111 = 1+2+4+8 = 15$.

[8] Так малограмотно, вернее, без уважения к собеседнику сейчас, увы, часто пишут некоторые посетители различных форумов.

В реальных устройствах при передаче сигналов, состоящих из нулей и единиц, в силу различных помех ряд цифр может измениться. Тогда правильная расшифровка может быть проведена лишь с некоторой вероятностью.

Пусть при передаче символа «0» этот же символ будет получен с вероятностью p_1, а при передаче символа «1» этот же символ будет получен с вероятностью p_2. Пусть в переданном сообщении m нулей и n единиц. Тогда вероятность того, что сообщение будет принято абсолютно верно, находится по формуле

$$P = p_1^m \cdot p_2^n.$$

Поскольку любая вероятность равна числу от 0 до 1, а натуральные степени таких чисел (не равных 1 и 0) меньше самого числа, то становится понятно, что с увеличением количества букв в сообщении вероятность получить абсолютно точное сообщение уменьшается.

Мы опять же попросим читателя составить программу, например, для $p_1 = 0{,}98$ и $p_1 = 0{,}97$, взять m и n до 10, от 10 до 100, больше 100 (разные).

В реальной жизни вероятность выше 0,95 считается достаточной. Можно узнать, при какой максимальной длине сообщения и для каких вариантов значений величин m и n искомая вероятность будет не меньше 0,95. Для этого достаточно методом перебора решить неравенство $p_1^m \cdot p_2^n \geq 0.95$. Очевидно, вручную это сделать очень сложно, а вот с помощью компьютера достаточно легко. Читатель! За работу!

Если передается логически связное сообщение, например, бытовой или литературный текст на каком-то языке, то малое число дефектов в принятом сообщении не является препятствием к пониманию. Например, при разговоре по мобильному телефону или по рации при наличии небольших помех на линии мы можем понять собеседника даже по отрывкам фраз[9]. Но вот если передаются логически несвязанные символы или наборы цифр, то неверно принятое сообщение может не только быть бесполезным, но и способно нанести серьезный ущерб. Чтобы подстраховаться в таких случаях, сообщения дублируют, т. е. передают его два, три и большее число раз. Тогда при пересылке сообщения k раз вероятность того, что, по крайней мере, одна из попыток передачи приведет к абсолютно точному результату, вычисляется по формуле

$$P = 1 - (1 - p_1{}^m \cdot p_2{}^n)^k.$$

Очевидно, что при большом числе попыток вероятность будет близка к единице. Но в огромном числе попыток нет смысла, правильнее поискать оптимальный вариант. Например, для $P \geq 0.99$ получаем, что $(1 - p_1{}^m \cdot p_2{}^n)^k \leq 0.01$.

Читатель может составить соответствующую программу, подобрав для нее интересный случай.

Рассмотрим другой тип шифров. Шифр перестановки основан на перестановке букв в исходном тексте. Обычно этот текст разбивается на отрезки равной длины, и каждый отрезок шифруется (т. е. в нем переставляются буквы) независимо.

Простейшим шифром перестановки является следующий. Берем прямоугольную таблицу подходящего размера (матрицу), вписываем в нее без пробелов исходный текст «по строкам», а шифрованный текст считываем «по столбцам». Оче-

[9] «Причиной приезда Максимилиана Андреевича в Москву была полученная им позавчера поздним вечером телеграмма следующего содержания: «Меня только что зарезало трамваем на Патриарших. Похороны пятницу, три часа дня. Приезжай. Берлиоз». Максимилиан Андреевич считался, и заслуженно, одним из умнейших людей в Киеве. Но и самого умного человека подобная телеграмма может поставить в тупик. Раз человек телеграфирует, что его зарезало, то ясно, что его зарезало не насмерть. Но при чем же тогда похороны? Или он очень плох и предвидит, что умрет? Это возможно, но в высшей степени странна эта точность — откуда он так-таки знает, что хоронить его будут в пятницу в три часа дня? Удивительная телеграмма! Однако умные люди на то и умны, чтобы разбираться в запутанных вещах. Очень просто. Произошла ошибка, и депешу передали исковерканной. Слово «меня», без сомнения, попало сюда из другой телеграммы, вместо слова «Берлиоза», которое приняло вид «Берлиоз» и попало в конец телеграммы.». Михаил Булгаков. «Мастер и Маргарита».

видно, что следует подобрать размер сообщения так, чтобы было несколько вариантов матрицы. Поскольку все строки должны быть заполнены, то в конец истинного сообщения можно приписать нужное число различных букв, взятых произвольно. Например, берем сообщение, состоящее из 114 букв. Так как $114 = 6 \cdot 19$, то разбиваем текст на 6 или на 19 строк и зашифровываем указанным образом. Но можно дописать в конец любые 6 букв, и тогда у нас будет больше вариантов, поскольку

$$120 = 2 \cdot 60 = 3 \cdot 40 = 4 \cdot 30 = 5 \cdot 24 = 6 \cdot 20 = 8 \cdot 15 = 10 \cdot 12 \text{ и т. п.}$$

Любой из вариантов годится. Читатель может составить такую несложную программу сам.

Возможны и более хитрые способы выписывания и считывания.

Если требуется зашифровать n натуральных чисел $x_1, \ldots, x_n$, то можно поступить следующим образом. Вместо этих чисел можно передать другие натуральные числа $b_{11}, \ldots, b_{nn}, a_1, \ldots, a_n$, договорившись предварительно о том, что $x_1 b_{11} + \ldots + x_n b_{1n} = a_1$ и т. д. В этом случае исходные числа однозначно находятся как решение системы n линейных уравнений с n неизвестными. Это еще одно задание для читателя!

ДИВЕРТИСМЕНТ. О МАТЕМАТИКЕ И МАТЕМАТИКАХ

Есть такая книга, вернее, сборник «О математике: проблемы преподавания» [40]. Сборник очень неровный. Более половины текста сборника — это воспоминания о прекрасных педагогах-математиках и об уникальных школах и интернатах с математическим уклоном советских времен. Есть и другие очень интересные статьи, а есть явные спекуляции.

Основная идея книги («плач Ярославны») — такая: в СССР была лучшая в мире система математического образования, которую «демократы» и «либералы» постепенно разрушают.

Слов о том, что именно компьютеры помогают нашим «демократам» и «либералам» добивать «наше самое лучшее математическое образование», в явной форме в сборнике нет, но они в завуалированном виде пронизывают многие статьи сборника. В частности, утверждается, что калькулятор и компьютер работают только с десятичными дробями, а начинать изучать математику (арифметику) нужно сугубо с простых дробей. Если этого не сделать, то никаких хороших знаний по математике школьникам уже не привить. Но современный школьник имеет под рукой калькулятор и не понимает, зачем его заставляют учить эти самые простые дроби и в чем принципиальная разница между простой и десятичной дробью. Отсюда же идет и непонимание рационального и иррационального числа, алгебраического и трансцендентного уравнения (см. этюд 4) и т. д. и т. п.

Что тут можно возразить, точнее, добавить?! Когда первый автор этой книги получал образование — среднее и высшее, дела обстояли так. Из десяти восьмых классов школы, где учился этот автор, в девятый класс переходило только 5–10%. Остальные шли работать на производство, поступили в техникумы или ПТУ. Через два года из десятого класса этой же школы в вузы поступало примерно 30% выпускников. Можно сказать, что в те годы, о которых ностальгируют авторы сборника (1950–1960-е гг.), старшие классы школы и вузы были *элитными* учебными заведениями, где посчастливилось учиться далеко не всем. А были еще и спецшколы и интернаты для одаренных детей, в частности, в области математики. Были также и «элитные» вузы типа МГУ (alma mater большинства авторов сборника), МФТИ, МИФИ... Теперь же в вузы, а точнее — в псевдовузы, которых сейчас расплодилось несметное количество, поступает 90% и более тех, кто десять-одиннадцать лет назад пошел в первый класс. О каком качественном математическом образовании тут может идти речь?! Нужно в десятки раз сокращать количе-

ство вузов, оставляя только те, где дают качественное высшее образование. А на это наше общество идти не готово и не пойдет.

И о дробях, простых и десятичных. Пакет Mathcad 15 имеет средства работы с простыми дробями и при желании их можно успешно использовать в начальных классах школы. Основное действие при работе с простыми дробями, при их, например, сложении — это нахождение наибольшего общего делителя (НОД) знаменателей. В Mathcad есть такая функция — `gcd` (**g**reatest **c**ommon **d**ivisor). Кроме того, в среде Mathcad число можно с помощью оператора `factor` разложить числа на простые множители (69 = 3·23 и 57 = 3·19). На рис. 3.7 показано, как можно в полуавтоматическом режиме (цепочкой ручных присвоений, определив перед этим наибольший общий делитель НОД и/или простые множители знаменателей) правильно сложить в среде Mathcad две простые дроби. (Кстати, в упомянутом сборнике в ряде статей с ехидцей констатируется, что американские школьники, решая такую задачу, просто-напросто сложат числители и знаменатели двух дробей и получат 181/126 и что скоро так будут делать все наши школьники, если не предпринять срочных мер по спасению математического образования.)

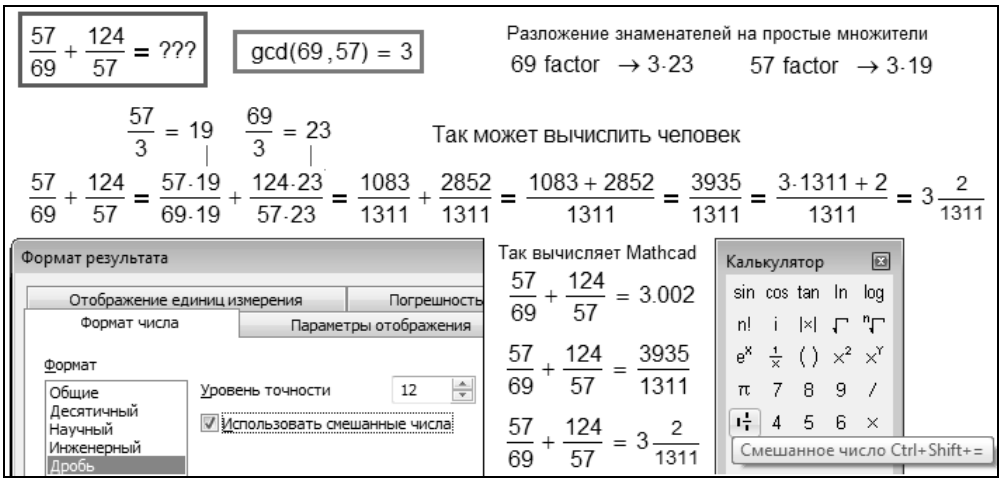


Рис. 3.7
Работа в среде Mathcad 15 с простыми дробями

На рис. 3.7 показано также диалоговое окно форматирования результата, где есть позиция «Дробь», позволяющая выводить ответ в виде простой дроби. Этим инструментом, кстати, нужно пользоваться осторожно — можно вывести и отформатировать значение числа π в виде 22/7 и утверждать, что эта математическая константа является рациональным числом. А это не так. И об этом «знает» даже сам Mathcad — см. рис. 12.1 на стр. 273. На рис. 3.7 показана также панель операторов Калькулятор, в котором есть кнопка ввода в расчет числа в виде целой части и части в виде простой дроби (смешанное число). Это позволяет очень продуктивно (и в смысле решения практических задач и в смысле образования) работать с простыми дробями в среде Mathcad — вернуться, так сказать, в начальные классы школы.

Оператор символьного преобразования `factor`, использованный в задаче на рис. 3.7, имеет прямое отношение к криптографии — основной теме этого этюда. Есть система кодирования, основанная на разложении числа на простые множители (см. рис. 15.2 на стр. 339): если в зашифрованном перехватываемом сообщении удастся это сделать, то сообщение можно расшифровать. Но весь фокус здесь в том, что довольно большое число можно раскладывать на несколько простых очень и очень долго. В этом можно убедиться, поработав с оператором factor.

Но есть в сборнике и зачаточные мысли о том, что компьютер при умном к нему подходе со стороны учеников, а главное, со стороны преподавателей может и возродить математическое образование — «я тебя погубил, я тебя и спасу!» В одной статье сборника говорится об использовании на лекциях по математике компьютера, Интернета, проектора, большого экрана и т. д., но, увы, ничего не говорится о современных математических компьютерных пакетах. Эту мысль мы и постарались развить в данной книге. Ведь преподаватели математики для многих школьников и студентов — это не только «прекрасные педагоги», о которых много написано в сборнике, а также и «мучителей толпа», заставляющая заучивать теоремы и считать в уме. Многим людям до сих пор снятся в кошмарных снах наибольшие общие делители, наименьшие общие кратные, простые дроби и прочие «орудия пытки». Компьютер может на уроках математики взять на себя рутинную работу (убрать нудную сторону математики, эти самые «орудия пытки»), оставляя учителю и ученикам простор для творчества.

И еще один аспект сборника, перекликающейся с темой книги. В сборнике много внимания уделяется анализу различий в подходах к решению задач физиками и математиками, вопросам привязки занятий по математике в вузах, вернее, во втузах к будущей инженерной специальности. Ведь студенты изучают математику не только для общего развития («гимнастика для ума»), но и для сугубо практических целей. Дискуссионный вопрос сборника — какие задачи решать на занятиях по математике: абстрактные или привязанные к профилю студента: электрика, теплотехника, химика и т. д. У авторов этой книги в этом плане есть некая «красная тряпка», которой они дразнит математиков — найти корни уравнения $x + x^2 = 0$. Математики чувствуют подвох в этом вопросе и дают ответ (ноль и минус единица) только после дополнительных уговоров и разъяснений. Но когда они слышат, что нельзя складывать величину и ее квадрат, что это равносильно сложению метров с килограммами, математики тут просто взрываются. Но все противоречия исчезают, если вспомнить о том, что по умолчанию умножение на единицу в математике не записывается. А если вести записи по канонам XVI в., то указанное уравнение получит вид $1 \cdot x + x \cdot x = 0$, т. е. на самом деле никто не складывает величину и ее квадрат, поскольку уравнение решается либо в безразмерных, либо в одинаковых $1(\text{ед.}) \cdot x(\text{ед.}) + x(\text{ед.}) \cdot x(\text{ед.}) = 0(\text{ед.})^2$ единицах.

Другой пример (типовая задача математического анализа): есть функция (дан ее вид), требуется определить значения аргументов, при которых производная больше самой функции. Возражения в том плане, что это опять же «сложение метров с килограммами, вернее, расстояния и скорости», воспринимаются опять же в штыки. Математики очень болезненно реагируют на математические ошибки физиков, но очень легко прощают себе свои «физические ошибки», считая, и правильно считая, что математика совершенно свободна от реалий нашего физического мира и вольны помимо одной физической интерпретации математического объекта давать другую. Или вообще, отказавшись от интерпретации физической, дать интерпретацию алгебраическую, геометрическую либо абстрактно-структурную. В случае с функцией и ее производной, например, можно сравнивать значения некоторой безразмерной величины (функция) и значения другой безразмерной величины — тангенса угла наклона касательной к оси абсцисс (производная).

Из задачника в задачник кочует задача, которую списали еще с глиняных табличек Древнего Вавилона (она приводится и в сборнике как иллюстрация того, откуда взялись квадратные уравнения — еще один кошмарный сон многих, но, к счастью, не всех бывших школьников): *Я перемножил длину и ширину, получил площадь; излишек длины над шириной, сложенный с площадью, равен 183; сумма длины и ширины равна 27; найти длину, ширину и площадь.* Вопрос о том, можно ли складывать длину («излишек длины над шириной») с площадью, математиков (и древних и современных) совершенно не волнует. На рис. 3.8 показано решение этой задачи в среде Mathcad.

$$\begin{bmatrix} a\cdot b = S \\ (a-b)+S = 183 \\ a+b = 27 \end{bmatrix} \text{solve}, \begin{pmatrix} S \\ a \\ b \end{pmatrix} \rightarrow \begin{pmatrix} 180 & 15 & 12 \\ 182 & 14 & 13 \end{pmatrix}$$

$a := 15m \quad b := 12m \quad S := a\cdot b = 180\,m^2 \quad a + b = 27\,m$

$(a-b)+S = \blacksquare$

Значение имеет единицы измерения: длина^2,
а должно иметь единицы измерения: длина.

Рис. 3.8
Задача о площади прямоугольника

Задача на рис. 3.8 решена — найдены два решения на выбор, но перевод чисел `a` и `b` в размерные величины (длина) и проверка ответа (а ее всегда нужно делать при решении задачи и на компьютере и при ручном решении) выявила «физический» дефект задачи, вытекающий из ее некорректной формулировки. Ведь если только в условии задачи фразу *«излишек длины над шириной, сложенный с площадью, равен 183»* заменить фразой *«численное значение излишка длины над шириной, сложенное с численным значением площади, равно 183»*, то все сразу же встает на свои места. И таких задач в задачниках по математике уйма! Эти «авгиевы конюшни» нужно чистить! Неким своеобразным «Гераклом» тут может выступить компьютер, в паре, конечно, с математиками, ведущими занятия в школах и вузах. Нужно прекратить наконец-то «плакать как Ярославна», а принять компьютер в «свои ряды». Мы должны наконец-то перестать мучить школьников и студентов работой, которую прекрасно может сделать компьютер и которую 99% школьников и студентов считают занудством, а только 1% математических талантов и гениев или просто одаренных людей — творческой работой. Нужны новые математические задачи и для развития «мозгов» и для практических целей. И задачи эти должны решаться в связке «человек — компьютер».

Задания читателям (дополнительно к тем, какие даны в тексте этюда)

1. Создать программу, переводящую римское число в арабское не через использование обратной программы (см. рис. 3.5), а напрямую. Если будут затруднения, то можно взглянуть на рис. 3.5*b*, перевернув книгу вверх ногами.

2. Составить программу определения частоты встречи того или иного слова в тексте. Текст же хранить не в переменной в виде строки (см. рис. 3.6), а текстовым файлом на диске, считывая его, используя предусмотренные для этого встроенные инструменты Mathcad.

3. Составить программу шифрования чисел на основе чисел Фибоначчи.

Этюд 4

РЕШАТЕЛИ, ИЛИ ВЕЛИКОЛЕПНАЯ СЕМЕРКА MATHCAD

В этюде будет рассказано, какие средства есть у математического пакета Mathcad для решения уравнения и систем уравнений, какие есть уравнения, как они решались в докомпьютерную эру и как они решаются сейчас.

Математика: корень уравнения, нуль функции, максимум функции, система линейных алгебраических уравнений, полином, золотое сечение, максимум функции, типы уравнений, методы аналитического решения уравнений, оптимальная цепная линия.

Физика: путь, время, скорость, уравнение движения, равномерное движение, переменное движение, трение скольжения.

Информатика: аналитическое и численное решение уравнений, компьютерные аналитические преобразования, метод Ньютона (касательных), метод половинного деления, инструменты трассировки численного решения, трассировка и увеличение графика, точность решения систем уравнений, цвет в программах.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127826**

Многие задачи по математике, физике, химии, механике, термодинамике и другим школьным и вузовским дисциплинам сводятся к решению уравнений и систем уравнений [41]. Поэтому полезно будет узнать, какие инструменты для решения такого рода задач есть у пакета Mathcad, очень популярного у школьников, студентов инженеров и ученых. Эти инструменты объединены в группу «Решение уравнений» встроенных функций Mathcad, которые используют различные *численные методы*. В среде Mathcad 15 названия этих методов можно узнать, если на имени некоторых функции, их реализующих, нажать правую кнопку мыши. Через это же нажатие можно изменить некоторые установки численных методов.

В группе «Решение уравнений» традиционно находятся *семь* функций (см. второе название этюда). Есть такой классический вестерн «Великолепная семерка», голливудская адаптация философской драмы Акиры Куросавы «Семь самураев». В американском фильме «главным» в семерке ковбоев, защищавших крестьян от бандитов, был Крис Адамс, которого сыграл Юл Бриннер. Семерка — это некое сакральное число не только в культуре и истории (семь древних мудрецов, семь чудес света, семь дней недели, семь нот в гамме и т. д.), но и в естествознании — семь цветов радуги, семь базовых единиц измерения международной системы СИ и т. д.

Есть еще в среде Mathcad и оператор `solve` для символьного (аналитического) решения уравнений и их систем. Описание этих инструментов будет сделано на несложных школьных «водных» примерах.

Задача 1. Моторная лодка прошла по реке в одну сторону (`L = 10 km`), а потом вернулась в исходную точку, затратив на этот «круиз» 1 ч 45 мин (`t`). Спрашивается, какова скорость течения воды в реке (неизвестная `x`), если собственная скорость лодки (`v` — скорость в стоячей воде) равна 12 км/ч (`kph` — мы, следуя современному тренду, будем стараться использовать международное написание единиц измерения).

Раньше подобные «школьные» задачи решались в несколько действий. Но не всякая задача может быть решена пошагово. Поэтому-то люди и придумали алгебру. В древние времена, например, пока не была выведена формула корней квадратного уравнения, не каждое такое уравнение можно было решить пошагово, причем решения были очень хитроумными. Кстати, нашу задачу о моторной лодке тоже сходу нельзя решить пошагово. Читатель, найди, если сможешь, по-

шаговое решение этой задачи и сравни найденное решение с тем, какое приведено ниже. Первый шаг такого решения может быть таким: `2 · 10 km / 12 kph = 1 hr + 40 min`: движение в текучей, а не в стоячей воде увеличило время пути на 5 минут. Многие студенты, подлаживаясь под несколько устарелые требования преподавателей, проводят пошаговые вычисления на компьютере в среде того же пакета Mathcad, и переписывают ответ в расчетную записку, имитируя ручной счет.

Сейчас в связи с широким использованием компьютеров в образовательной сфере принято составлять, а затем решать уравнения, выбирая их подходящие корни, и об этом мы рассуждали в этюде 1. Пойдем и мы по этому пути, но, составив уравнение, попробуем решить его не на бумаге, а на компьютере в среде математической программы Mathcad.

В нашей задаче о моторной лодке время в пути `t` — это суммарное время, затраченное на поездку в одну сторону `L / (v + x)` (условно будем считать, что это движение по течению реки), и в обратную сторону (против течения) `L / (v - x)`. Поэтому наше уравнение будет иметь вид:

$$(L / (v + x)) + (L / (v - x)) = t$$

0. SOLVE

Начнем с решения полученного уравнения средствами *символьной математики* Mathcad. Формальное, более правильное и более длинное название символьной математики — *компьютерные аналитические преобразования*, но у нас прижилась калька с английского — *symbolic math*. Это название мы и будем использовать далее.

Если *численная математика* (которая, повторяем, тоже есть в среде Mathcad и составляет его основу) оперирует числами, хранящимися в переменных, то *символьная математика* работает сначала с самими переменными-символами, а потом уж — с числами, какие в них хранятся.

На рисунке 4.1 показано решение уравнения прямого и обратного движения моторной лодки по реке с помощью оператора `solve` *символьной математики* Mathcad (на этом и некоторых других рисунках будут показаны позиции меню и панели инструментов Mathcad Prime и Mathcad 15 для решения описываемых задач).

Из полученного общего аналитического решения (из вектора с двумя элементами-формулами — см. рис. 4.1) можно скопировать один элемент, подставить в него исходные значения переменных `L`, `t` и `v` (см. рис. 4.2) и получить численный ответ — скорость течения воды в реке. Ответ будет выдан в метрах, деленных на секунду (Mathcad по умолчанию ориентирован на СИ — на международную систему исчислений), и подправлен пользователем на более уместные тут километры в час (`kph`). Mathcad, повторяем, — это не просто математический, а физико-математический пакет [4]: переменные Mathcad хранят не числа, а физические величины (длину, время, силу, массу и т. д.), что очень полезно при расчетах в задачах с физическим смыслом. Это существенно ускоряет и упрощает расчеты, позволяет избежать ошибок в них, а также автоматизирует соответствующие преобразования единиц измерения.

Спрашивается, для чего же тогда в пакете Mathcad есть и численная математика, если задачу можно просто и красиво решить с помощью символьной математики? Дело в том, что символьная математика, нацеленная на выдачу всех решений в виде формул (абсолютная точность!), часто не справляется с более-менее сложной задачей, и это показано на рисунках 4.3 и 4.4.

$$\frac{L}{v+x}+\frac{L}{v-x}=t \xrightarrow{solve,x} \begin{bmatrix} \dfrac{\sqrt{t\cdot v\cdot(t\cdot v-2\,L)}}{t} \\ -\dfrac{\sqrt{t\cdot v\cdot(t\cdot v-2\,L)}}{t} \end{bmatrix}$$

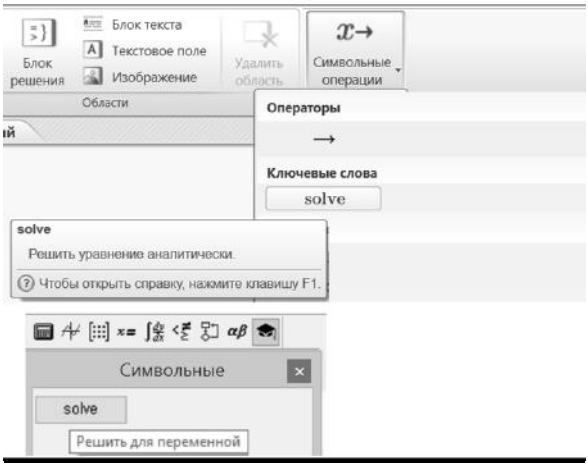


Рис. 4.1
Аналитическое решение задачи о движении моторной лодки

$$L:=10\ km \qquad t:=1\ hr+45\ min \qquad v:=12\ kph$$

$$\frac{\sqrt{t\cdot v\cdot(t\cdot v-2\,L)}}{t} = 2.619\ kph$$

Рис. 4.2
Решение задачи о моторной лодке по найденной на рис. 4.1 формуле

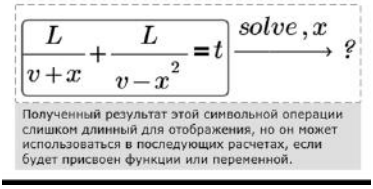


Рис. 4.3
Поиск корня уравнения: очень объемный скрытый ответ

На рисунке 4.3 в уравнении движения моторной лодки один из иксов был возведен в квадрат. Физический смысл уравнения пропал (складывается скорость с квадратом скорости).

Примечание. Такими «нефизическими» формулами заполнены все современные учебники и задачники по математике. И это не очень хорошо, вернее, совсем плохо. Хорошо тогда, когда за формулой скрывается какая-нибудь физическая реальность. Такое направление математики условно называют «натуральная математика». Проблема размерностей исчезнет сразу же, как только мы вспомним, что еще во времена Франсуа Виета господствовал принцип однородности, который требовал в подобных случаях умножать икс на некую единицу, получая везде квадраты скорости. Но, как водится, у современных математиков, умножение на единицу не производится, а размерности игно-

рируются. Типичная задача, часто предлагаемая школьникам и студентам для решения в рамках курса математического анализа, о которой мы уже писали в этюде 3 — это сравнение самой функции и ее производной. Но сравнение функции с ее первой производной равносильно физически некорректному сравнению пути (длины) и скорости — разных физических величин.

Но сейчас главное не физический смысл уравнения, показанного на рис. 4.3. Важно то, что пакет Mathcad, решив это чуть усложненное уравнение, не смог вывести на дисплей ответ — настолько тот оказался громоздким. Но это еще полбеды, которую можно исправить, прочитав комментарий на рис. 4.3. В этюде 8 на рис. 8.13 (стр. 191)показано, как вытаскивается такой объемный ответ символьной математики. Настоящая «беда» показана на рис. 4.4 для еще более усложненного уравнения. Если, например, один икс возвести в квадрат, а другой в куб, то символьная математика Mathcad «поднимет руки вверх» и скажет: «Сдаюсь!»

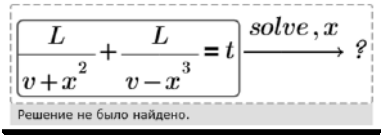


Рис. 4.4
Поиск корня уравнения: решение не найдено

Если в константы этого «нефизического» уравнения подставить безразмерные численные значения известных величин, то хотя бы один действительный корень у этого уравнения зафиксировать удастся — см. рис. 4.5, где данная задача решена графически.

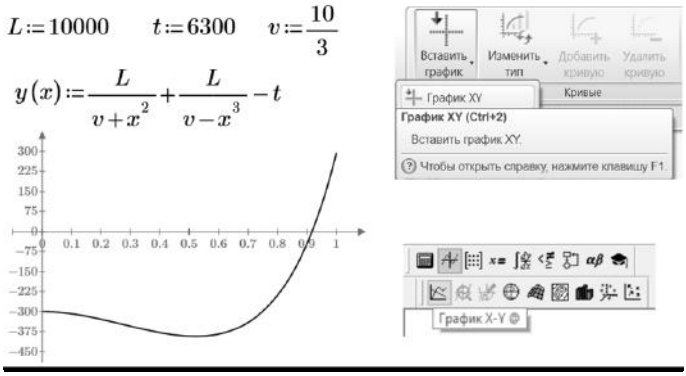


Рис. 4.5
Графический поиск нуля функции

1. ROOT+ROOT

Из рисунка 4.5 видно, что у нашего уравнения, превращенного в функцию пользователя переносом переменной `t` в левую часть, есть как минимум один действительный корень в районе числа 0.9. Уточнить численное значение этого корня поможет встроенная в Mathcad функция `root` — см. рис. 4.6 и 4.7.

На рисунке 4.6 показан вызов функции `root` с четырьмя аргументами, а на рис. 4.7 с двумя. В обоих случаях ответ выведен с тремя знаками после десятичной точки. Но можно вывести и большее число знаков — до 15. В первом случае (рис. 4.6) нуль функции `y(x)` ищется методом деления пополам на интервале, заданном третьим и четвертым аргументами функции `root` (см. авторскую анимацию этого метода на сайте **http://communities.ptc.com/videos/1468**). Во втором случае

(рис. 4.7) нуль функции рассчитывается методом секущих с опорой на первое предположение `x := 1` (анимация — **http://communities.ptc.com/videos/1466**). В среде Mathcad для вычисления нуля функции пользователя фактически есть две одинаковые по имени, но разные по своей сути встроенные функции `root`.

$$\mathbf{root}(y(x), x, 0.9, 1) = 0.918$$

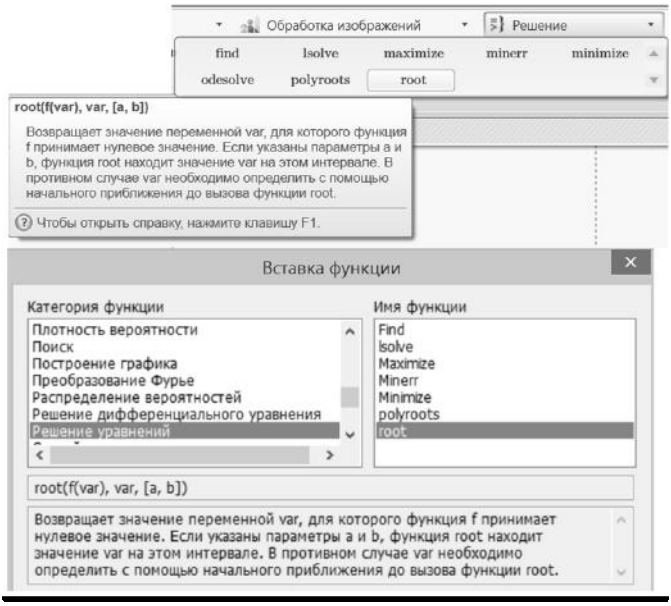


Рис. 4.6
Работа в среде Mathcad встроенной функции ***root*** *с четырьмя аргументами*

$$x := 1 \quad \mathbf{root}(y(x), x) = 0.918$$

Рис. 4.7
Работа в среде Mathcad встроенной функции root с двумя аргументами

На рисунке 4.8 показана работа функции root на более сложном примере — с функцией пользователя `sin(x)/x`, которая имеет бесконечное число нулей. На отрезке от 2 до 7 функция `y(x)` имеет два нуля (π и 2π), но четырехаргументная функция root ответа не выдала, так как функция `y(x)` имеет одинаковые знаки на концах этого отрезка и функция `root` считает, что там не может быть корней уравнения. На отрезке от 1 до 17 нулей уже пять, один из которых (9,425) найден четырехаргументной функцией `root`. На концах отрезка от 1 до 17 функция `y(x)` имеет разные знаки. При первом приближении, равном 0.01, двухаргументная функция `root` выдала не ближайший нуль (3.14), а «очень-очень дальний»: 298.451. Понять эти особенности применения функции `root` можно только после детального рассмотрения численных методов, заложенных в эту функцию, — метода деления отрезка пополам и метода секущих.

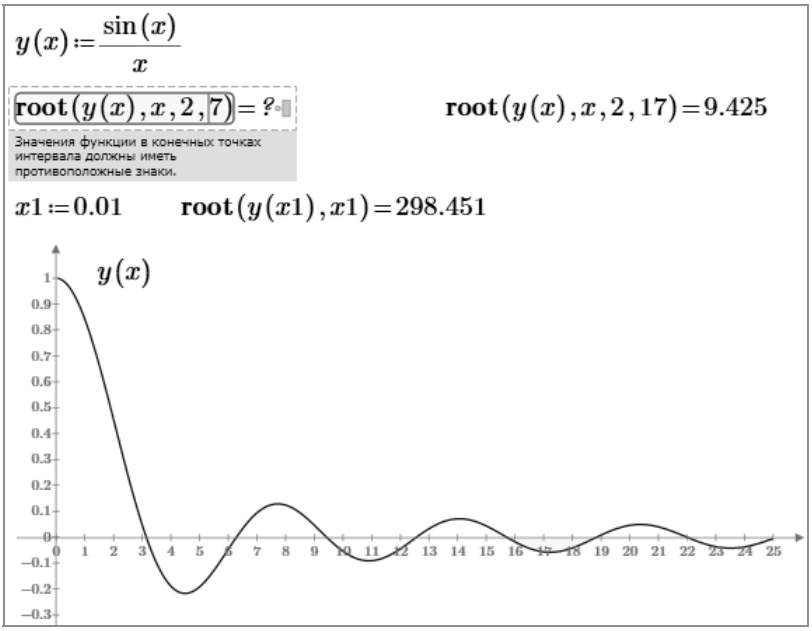


Рис. 4.8
Особенности работы двухаргументной функции root

Ранее мы отметили, что символьная математика Mathcad оперирует не числами, а символами — самими переменными, хранящими или не хранящими числа. Но это не совсем так.

Если какая-либо переменная выражения хранит численное значение, то символьная математика будет работать не с самой переменной (с символом), а с числом, хранящимся в этой переменной. На рисунке 4.3 была показана осечка символьной математики Mathcad при решении довольно простого уравнения. Но если всем переменным этого уравнения, кроме переменной `x`, задать численные значения, то символьный оператор `solve` успешно справится с задачей — см. рис. 4.9.

$$\frac{L}{v+x^2}+\frac{L}{v-x^3}=t \xrightarrow{solve,x} \begin{bmatrix} 0.91753907432064754652 \\ -0.10072745491075931273-0.504473077723681i \\ -0.10072745491075931273+0.50447307772368065275i \\ -0.35804208224956446053-1.432067662218 07i \\ -0.35804208224956446053+1.4320676622180746717i \end{bmatrix}$$

Рис. 4.9
Численный ответ символьного оператора

На рисунке 4.9 показано, что «символьный» оператор `solve` в отличие от «численной» функции `root` выдал все пять корней уравнения (один действительный и четыре комплексных с мнимой частью) без установки интервала (рис. 4.6) или первого предположения (рис. 4.7). Кроме того, если численная математика при выводе ответа «на печать», как мы уже отметили, по умолчанию ограничивается тремя знаками после десятичной точки, то символьный оператор `solve` в этом случае выдал численные решения с двадцатью знаками после запятой. При «численном» ответе количество значащих цифр можно увеличить до 15, а при символьном до 250.

Примечание. Лишить переменную ее численного значения для последующих аналитических преобразований можно операторами: $\mathrm{clear}_{\mathrm{sym}}(a)$ (Mathcad Prime — см. рис. 1.8 на стр. 24) или a : = a (Mathcad 15).

Если наше уравнение с числовыми константами (см. рис. 4.9) и дальше усложнять, то на каком-то этапе оператор solve не сможет найти корни. Функция же root по-прежнему будет выдавать корень, правда, лишь один из многих и с опорой на заданный интервал поиска (рис. 4.6) или на первое приближение (рис. 4.7). При этом задавать интервал поиска придется, исходя из уверенности, что корень на этом интервале имеется. Метод секущих же при неправильном первом приближении вообще не выдаст нужного результата. Это такой своеобразный компромисс. Отсюда общее правило: поставленную математическую задачу нужно стараться сначала решить аналитически в общем виде, не придавая переменным конкретных числовых значений (рис. 4.1) или придавая отдельным или всем переменным числовые значения (рис. 4.9). Если же ответа найти не получается, то придется переходить к поиску частных решений численными методами, дополняя их анализом графиков.

На рисунке 4.10 показано использование графика и функции root в двух ее вариантах для решения нашей задачи о моторной лодке. Интересный факт. Двухаргументная функция root при первом приближении, равном нулю, выдала не ожидаемый положительный, а неожиданный отрицательный корень. Этот нюанс можно понять, если опять же учесть особенности метода секущих при поиске нулей функции и после построения графика не на отрезке от –3 до 3 км/ч, а на отрезке –13 до 13 км/ч, охватывающем точки разрыва, что мы сделаем ниже. Дело в том, что функция root с двумя аргументами работает так. Пользователь задает одну опорную точку поиска (первое *предположение*, но это далеко не всегда первое *приближение* — см. пример на рисунке 4.8). Далее пакет Mathcad **правее** этой точки на расстоянии CTOL (по умолчанию это 0.001 в нашем случае метров, т. е. один миллиметр) фиксирует вторую точку и проводит через две эти точки секущую, которая будет близка к касательной). Эта секущая где-то пересекает ось X — это будет третьей, очередной точкой итерационного поиска корня. Далее реализуется классический метод секущих. Правая точка разрыва нашей анализируемой функции «перекидывает» поиск в область отрицательных значений. Можно сказать, что в функцию root с двумя аргументами заложен гибрид метода Ньютона (касательных) и метода секущих. На авторском сайте **http://communities.ptc.com/videos/1411** можно видеть анимацию метода Ньютона для одиночного уравнения, а на сайте **http://communities.ptc.com/videos/1472** для системы двух уравнений.

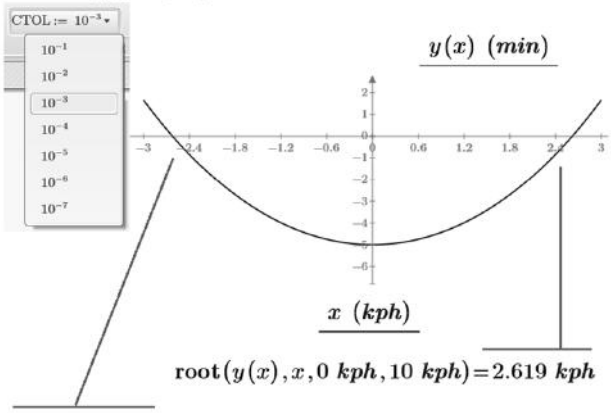
Примечание. Для повышения точности поиска нуля функции с помощью встроенной в Mathcad функции root можно не уменьшать значение системной переменной CTOL (см. список левее графика на рис 4.10), а умножить анализируемую функцию на 10^3, 10^4 и т. д. Выбор точности вычислений — это отдельная задача. С одной стороны, повышенная точность никогда не будет лишней, а с другой стороны, она приводит к замедлению расчетов и срыву их в ряде случаев. Во всем нужна мера!

Уравнение движения моторной лодки на рисунке 4.1 можно преобразовать в квадратное. К такому приему обычно прибегают в школах, т. к. школьников, как правило, учат решать аналитически только квадратные уравнения — см. текст под рисунком 5.19 на стр. 120. Как такое преобразование можно сделать в среде Mathcad, показано на рис. 4.11.

Оператор символьной математики simplify (упростить) приводит левую часть исходного выражения к общему знаменателю, умножает обе части уравнения на полученный знаменатель и переносит все слагаемые в левую часть уравнения (рис. 4.11). Таким способом выделяется функция, которая приравнена к нулю. Оператор coeffs находит коэффициенты этой функции-полинома (в данном случае квадратного). Это квадратное уравнение можно решить оператором solve, но... см. ниже.

$$t := 1\ hr + 45\ min \qquad v := 12\ kph \qquad L := 10\ km$$

$$y(x) := \frac{L}{v+x} + \frac{L}{v-x} - t \qquad x := -3\ kph, -2.99\ kph .. 3\ kph$$



$$\mathrm{root}(y(x), x, 0\ kph, 10\ kph) = 2.619\ kph$$

$$\mathrm{root}(y(x), x, 0\ kph, -10\ kph) = -2.619\ kph$$

$$x := 0\ kph \qquad \mathrm{root}(y(x), x) = -2.619\ kph$$

Рис. 4.10

*Графическое и численное (через функцию **root**) решение задачи о моторной лодке*

$$\frac{L}{v+x} + \frac{L}{v-x} = t \xrightarrow{simplify} \frac{2 \cdot L \cdot v}{v^2 - x^2} = t$$

$$2 \cdot L \cdot v - t \cdot \left(v^2 - x^2\right) \xrightarrow{coeffs, x} \begin{bmatrix} 2 \cdot L \cdot v - t \cdot v^2 \\ 0 \\ t \end{bmatrix}$$

$$t \cdot x^2 + 0 \cdot x + \left(2 \cdot L \cdot v - t \cdot v^2\right) = 0 \qquad a \cdot x^2 + b \cdot x + c = 0$$

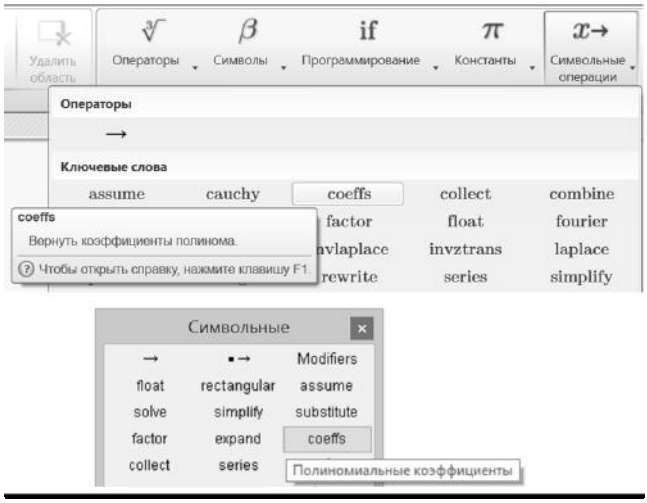


Рис. 4.11

Определение коэффициентов полинома

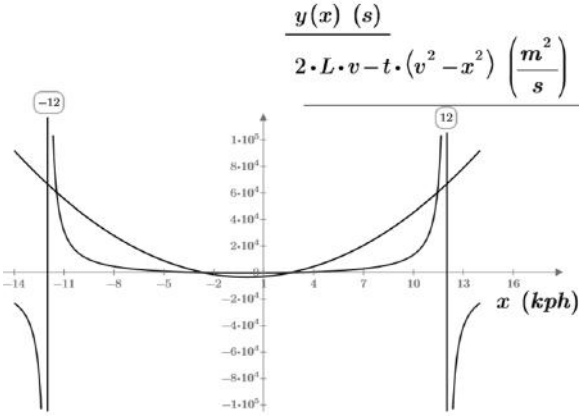


Рис. 4.12
Исходное и квадратное уравнение движения моторной лодки

Примечание. Квадратичная функция, полученная после преобразования исходного уравнения движения моторной лодки, не равна исходной функции, а только имеет с ней два общих корня. В этом можно убедиться, взглянув на графики, показанные на рис. 4.12. Поскольку по условию задачи `v = 12 kph`, а уравнение движения не имеет смысла при `x = ± v`, то функция, стоящая в левой части уравнения, имеет разрывы при `x`, равном `12` и минус `12 kph`. Квадратичная же функция таких разрывов, естественно, не имеет.

2. POLYROOTS

Если выражение представляет собой полином (квадратный, например, см. выше), то можно найти все его нули, использовав еще одну функцию из «великолепной семерки Mathcad» — функцию `polyroots`. Она имеет в качестве аргумента вектор коэффициентов полинома и возвращает его нули (вектор, который на один элемент короче вектора-аргумента), т. е. решение нашей задачи — см. рис. 4.13.

В нашей задаче о движении моторной лодки полином оказался квадратным и его корни, повторяем, можно было найти через оператор символьной математики `solve` (см. рис. 4.1). Но в случае полиномов высокой степени оператор `solve` не сработает. Вот тут-то и пригодится численная встроенная функция `polyroots`.

3. FIND

Показать работу еще одной функции из «великолепной семерки Mathcad» — функции `Find` — поможет нам еще одна дополнительная моторная лодка.

Задача 2. От двух пристаней на прямолинейном участке реки одновременно отходят две моторные лодки и плывут навстречу друг другу. Они встречаются в точке, делящей этот участок реки в соотношении золотого сечения. Найти скорость второй лодки v_2 и скорость течения воды в реке `v`, если известна скорость первой лодки v_1, расстояние между пристанями `L` и время `t` движения лодок до встречи. (Шуточный вариант задачи: от двух станций по одноколейной дороге вышли навстречу друг другу два поезда. И не столкнулись! Почему? Ответ: не судьба!)

Мы в этой задаче используем знаменитое «золотое сечение», т. е. такое деление отрезка на две неравные части, при котором длина меньшей части отрезка так относится к длине большей части, как длина большей части относится к длине всего отрезка (см. рис. 4.14). Это свойство золотого сечения помнят многие, чего не скажешь о формуле золотого сечения. На сайте

http://communities.ptc.com/videos/1521 показана авторская анимация метода золотого сечения при численном поиске на заданном отрезке минимума функции одного аргумента.

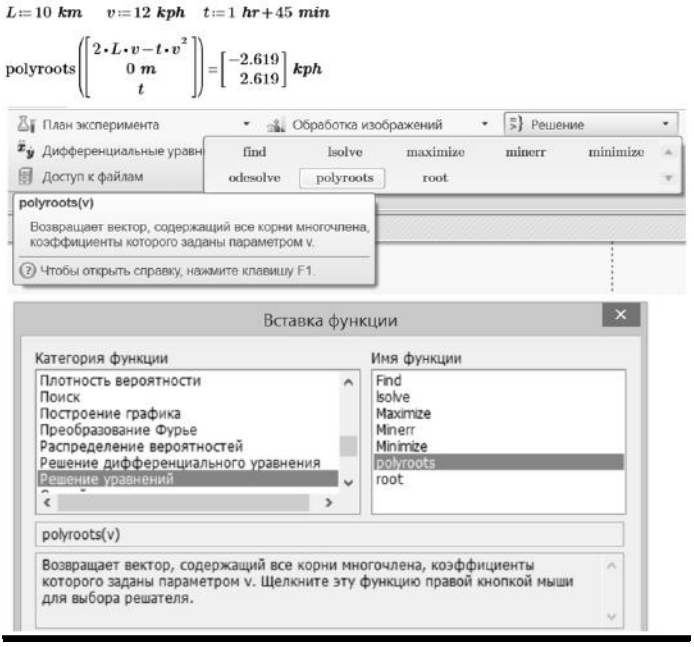


Рис. 4.13
Поиск нулей полинома в среде Mathcad

Золотое сечение в задачу вставлено неслучайно. Можно поискать в своей памяти или в справочниках (бумажных или интернетовских) формулу золотого сечения. Но можно поступить иначе [32] — вспомним основную идею этюда 1: написать в среде Mathcad само уравнение золотого сечения применительно к нашей задаче о моторных лодках и решить его аналитически, получив нужную формулу — см. рис. 4.14.

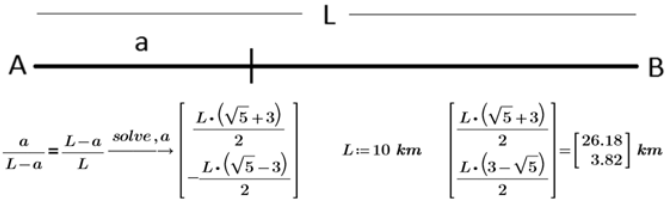


Рис. 4.14
Решение уравнения золотого сечения в среде Mathcad

На рисунке 4.14 оператор `solve` выдал два решения, из которых нам подходит только второе — 3.82 km. Первое же решение (26.18 km) лежит вне рассматриваемого отрезка. Символьная математика, повторяем, выдает все ответы, из которых нужно еще уметь выбрать подходящее. Или уметь заставить оператор `solve` выдать только один нужный ответ — см. рисунок 5.19 в этюде 5.

На рисунке 4.15 показано решение в среде Mathcad Prime и Mathcad 15 задачи о двух моторных лодках, сводящееся к решению системы двух уравнений с двумя

неизвестными. Решение найдено с помощью функции Find, требующей начального приближения к решению.

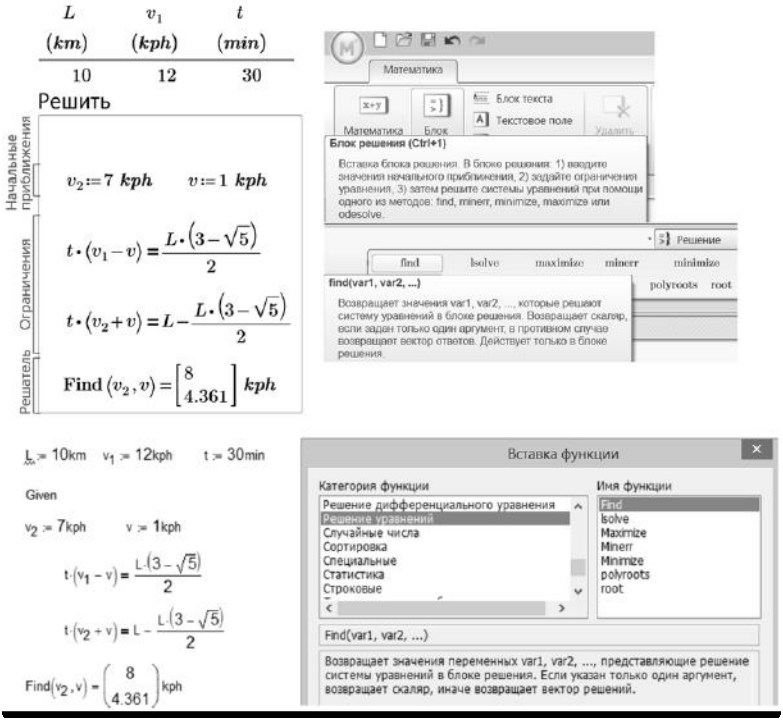


Рис. 4.15
Решение систем уравнений с помощью функции ***Find***

Встроенная функция Find меняет значение своих аргументов, начиная от начального приближения/предположения так, чтобы уравнения системы превратились в тождества. Вернее, почти в тождества. Дело в том, что и обе функции root (рис. 4.7 и 4.9) и функция Find (рис. 4.7) возвращают значения, отличающиеся от точных решений на величину, не превышающую по модулю значения, хранящегося в системной переменной CTOL. Ведь что такое корень уравнения?! Корень — это значение переменной, при котором уравнение превращается в тождество. Но при численном (приближенном!) решении найти точный корень не всегда удается. Да и время тратить на это не целесообразно. Подстановка приближенного значения корня в уравнение приводит к тому, что правые и левые части уравнения отличаются друг от друга на значение, хранимое в переменной CTOL, которое по умолчанию равно 0.001. Это значение можно менять, решая конкретную задачу. На сайте с авторской анимацией **http://communities.ptc.com/videos/1472** можно видеть особенности поиска четырех корней системы двух нелинейных уравнений: уравнения эллипса и уравнения лемнискаты Бернулли (см. также рис. 13.14 из этюда 13). На сайте **http://communities.ptc.com/videos/2418** можно увидеть анимацию, показывающую, как задание первого приближения влияет выбор конкретного корня из нескольких существующих. Более подробно о методах решения, заложенных в функцию Find, можно почитать в работе [42].

Примечание. В среде Mathcad Prime по сравнению с Mathcad 15 существенно изменилась технология решения уравнений с помощью функции Find. В среде Mathcad Prime не нужно больше вводить ключевое слово Given. Достаточно ввести область Решить с тремя подобластями. От ключевого слова Given отказались в том числе и потому, что мно-

гие пользователи после ввода этого слова нажимали клавишу пробела, превращали тем самым это ключевое слово в комментарий и... не понимали, почему Mathcad отказывается решать систему уравнений. На рис. 4.15 показаны для сравнения решения в обеих версиях Mathcad.

Примечание. Переменная `L`, которой на рис. 4.15 присваивается начальное значение 10 km (`L := 10 km`), подчеркнута волнистой чертой, указывающей на некую ненормальную расчетную ситуацию в среде Mathcad 15. Переменная `L` по умолчанию в среде Mathcad 15 хранит значение одного литра (единица вместимости) и это значение пользователь переопределяет. В среде Mathcad Prime эта недоработка (неудобство) исправлена — там можно иметь две независимые переменные `L`: единицу вместимости и отдельную пользовательскую переменную, хранящую, как в нашем случае, расстояние.

4. LSOLVE

Можно понять, что система двух алгебраических уравнений движения двух моторных лодок навстречу друг другу, показанная на рис. 4.15, *линейна*, и применить к ней еще одну функцию из «великолепной семерки Mathcad» — функцию `lsolve`, предназначенную для решения (solve) именно систем линейных (l) алгебраических уравнений (СЛАУ) — см. рис. 4.16.

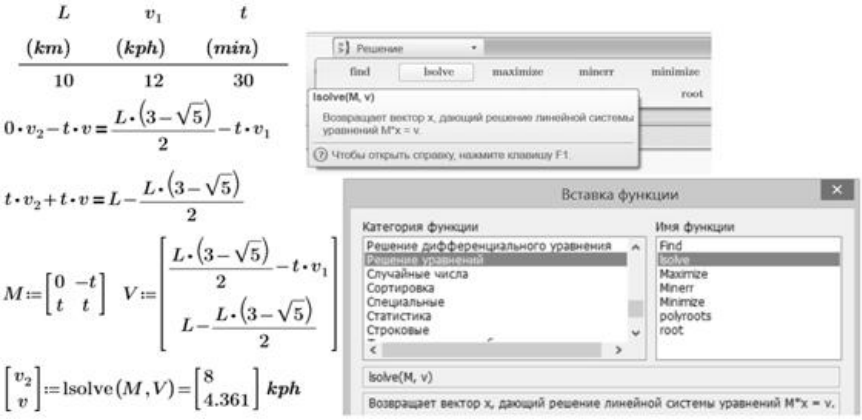


Рис. 4.16
Решение СЛАУ в среде Mathcad

На рисунке 4.16 система уравнений, идентичная системе с рис. 4.15, с помощью тождественных алгебраических преобразований приведена к виду классической линейной системы: слева неизвестные v_2 и `v` со своими коэффициентами, справа — свободные члены. Функция `lsolve` имеет два аргумента: матрицу коэффициентов при неизвестных СЛАУ (у нас это `M`) и вектор свободных членов `V`. Возвращает функция `lsolve` вектор найденных значений неизвестных. При решении СЛАУ с помощью функции `lsolve` (рис. 4.16) начальные приближения (см. рис. 4.15) вводить не надо, т. к. у этой системы либо есть, причем единственное решение, либо решений совсем нет, либо решений бесконечно много. В [32] дан графический анализ этой особенности с привлечением понятия ранга матрицы (см. также рис. 7.25–7.27 в этюде 7).

Примечание. Вторым аргументом функции lsolve может быть не только квадратная (классический случай СЛАУ), но и прямоугольная матрица, отображающая недоопределенную или переоопределенную системы.

5 И 6. MINIMIZE&MAXIMIZE

Об очередной функции «великолепной семерки» — о функции `Minimize`, будет рассказано на примере *задачи оптимизации*, связанной также с «водным транспортом».

Задача 3. Определить крейсерскую скорость судна — скорость, при которой затраты на его эксплуатацию будут минимальны.

Задача предельно упрощена — затраты на эксплуатацию судна состоят из двух частей: почасовой зарплаты экипажа, пропорциональной времени движения судна (обратно пропорциональной скорости судна), и затрат на горючее, пропорциональных квадрату скорости судна (коэффициенты пропорциональности — `a` и `b`). Увеличивая скорость судна, мы экономим на зарплате экипажа, но при этом приходится больше тратить денег на горючее. Попробуем найти тут оптимальное решение!

На рисунке 4.17 показано решение этой типичной задачи оптимизации с помощью встроенной функции `Minimize` с графической иллюстрацией решения.

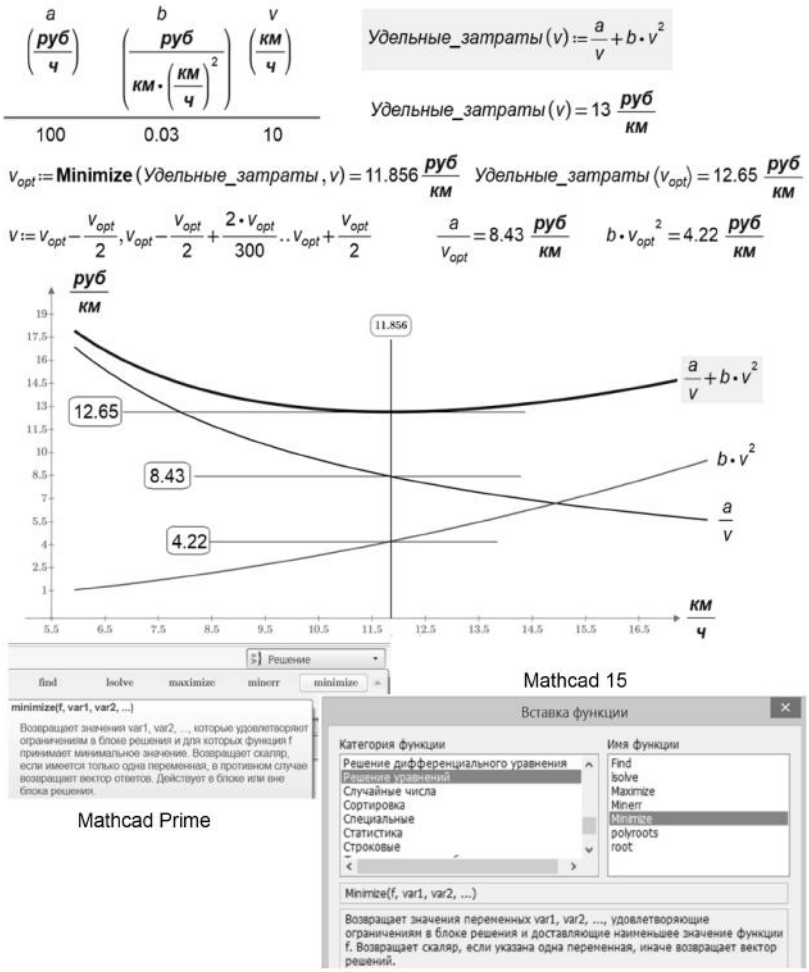


Рис. 4.17
Нахождение крейсерской скорости судна численной математикой Mathcad

Функция `Minimize` меняет значение своего второго аргумента, начиная от заданного предполагаемого значения (у нас это 10 км/ч) так, чтобы значение первого аргумента (целевой функции `Удельные_затраты`) приняло минимальное значение. Если бы мы не минимизировали затраты, а максимизировали, например, прибыль владельца судна, то нужно было бы при решении такой задачи функцию `Minimize` заменить на функцию `Maximize`. В оптимизационных задачах часто присутствуют ограничения — скорость судна, например, не может превышать максимально допустимую. В этом случае функции `Minimize` или `Maximize` нужно будет поместить в область `Ограничения` блока `Решить`, показанного на рис. 4.15.

Найти минимум нашей целевой функции `Удельные_затраты` можно и средствами символьной математики Mathcad, что показано на рис. 4.18.

$$\frac{d}{dx}\left(\frac{a}{x}+b\cdot x^2\right) \xrightarrow{solve,x} \begin{bmatrix} \left(\frac{a}{2\cdot b}\right)^{\frac{1}{3}} \\ -\frac{\left(\frac{a}{2\cdot b}\right)^{\frac{1}{3}}}{2}+\frac{\sqrt{3}\cdot\left(\frac{a}{2\cdot b}\right)^{\frac{1}{3}}\cdot 1i}{2} \\ -\frac{\left(\frac{a}{2\cdot b}\right)^{\frac{1}{3}}}{2}-\frac{\sqrt{3}\cdot\left(\frac{a}{2\cdot b}\right)^{\frac{1}{3}}\cdot 1i}{2} \end{bmatrix}$$

$$\left(\frac{a}{2\cdot b}\right)^{\frac{1}{3}} = 11.856\ \frac{км}{ч}$$

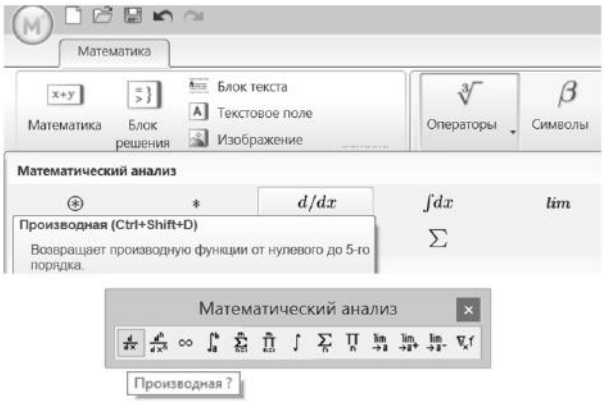


Рис. 4.18
Нахождение крейсерской скорости судна символьной математикой Mathcad

На рисунке 4.18 ведется поиск нулей первой производной функции по удельным затратам на километр пути судна. Но если затраты на топливо будут зависеть от скорости судна, взятой не во второй степени, а в степени n (этот коэффициент, близкий к двойке, уточняют экспериментально), то символьная математика уже не справится с такой усложненной задачей (рис. 4.19), и придется вернуться к численным методам решения задач (рис. 4.17).

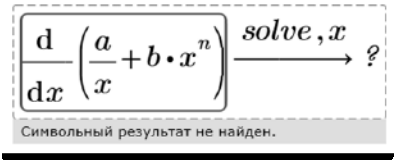


Рис. 4.19
Осечка при работе с символьной математикой Mathcad

7. MINERR

Последняя[1] функция «великолепной семерки Mathcad» — это функция `Minerr` (Minimal Error — минимальная ошибка). Если функция `Find` (см. рис. 4.15) не находит решения системы уравнений, то она возвращает сообщение об ошибке. Функция же `Minerr` в такой ситуации возвратит не сообщение об ошибке, а значения своих аргументов (невязку системы), при которых система уравнений будет максимально приближена к системе тождеств — точку последнего приближения к решению. В старых версиях Mathcad не было функций `Minimize` и `Maximize`, и задачи оптимизации приходилось решать именно через функцию `Minerr`. Попробуем и мы поступить так. На рисунке 4.20 показано, как эта функция решает задачу крейсерской скорости судна: при оптимальном движении затраты на эксплуатацию судна будут максимально приближены к нулю (мечта всех судовладельцев).

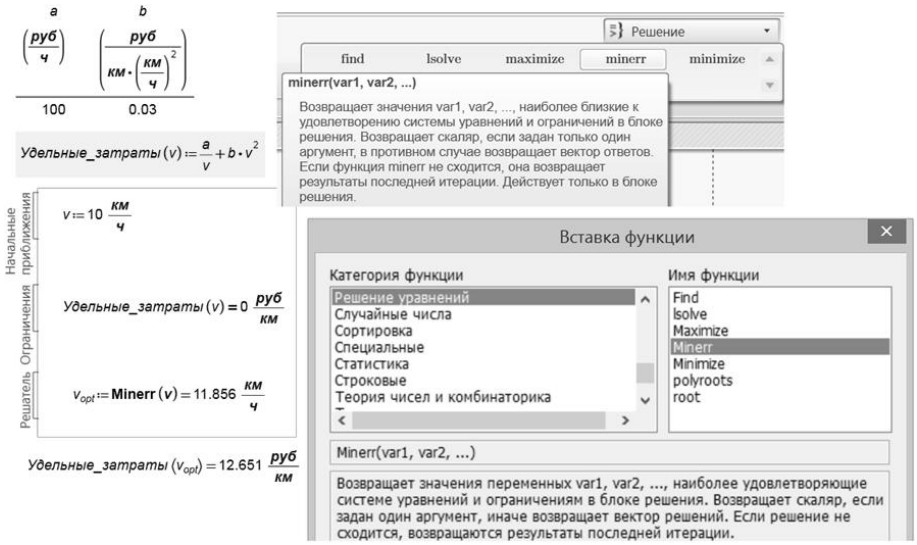


Рис. 4.20
Решение задачи оптимизации с помощью функции Minerr

Функция `Minerr` поможет нам в этюде 8 сделать анимацию прави́ла Чебышёва (см. рис. 8.8 и 8.9) при укороченном одном стержне. Кстати, Чебышёв родился в 1821 г. недалеко от того места, где сейчас дача одного из авторов книги (см. стр. 380).

Функцию `Minerr` можно считать ***главной*** в «великолепной семерке Mathcad», т. к. ею можно заменить и функцию `Find`, и функцию `root` (в двух ее вариантах), и функцию `polyroots`, и функцию `lsolve`, и в ряде случаев функции `Minimize` и `Maximize`. При использовании функции `Minerr` надо обязательно предусматривать проверку реше-

[1] Последняя в списке, но не последняя по важности — смотрим далее.

ний. Нередки случаи, когда решения могут оказаться ошибочными, чаще всего из-за того, что из нескольких корней находится нереальный (или не представляющий интереса) корень. Дело в том, что функция `Minerr` пытается найти максимальное приближение к искомому числу путем минимизации среднеквадратической погрешности решения. Следует заранее убедиться в том, что решение существует, и как можно точнее указать начальное приближение к решению.

С другой стороны, главной (незаменимой) функцией «великолепной семерки Mathcad» можно считать функцию `root`, т. к. она одна остается активированной при окончании месячного пробного использования Mathcad Prime и при переходе к Mathcad Express.

Компьютерная математика с универсальными и скрытыми от пользователей методами аналитических и численных решений заставляет нас забывать о типах уравнений. Вспомним о них!

ТИПЫ УРАВНЕНИЙ

Для того чтобы без проблем и правильно решать уравнения и системы уравнений, нужно знать не только специфику численных методов (см. выше), но и свойства самих уравнений, что поможет решать их аналитически.

Каждое уравнение с одним неизвестным относятся к одному из четырех типов уравнений: *алгебраические*, *рациональные*, *иррациональные* и *трансцендентные*[2]. Есть еще и *интегральные* уравнения, содержащее интегральное преобразование над неизвестной функцией. Если интегральное уравнение содержит также производные от неизвестной функции, то говорят об интегро-дифференциальном уравнении. В этюде 7 мы столкнемся и с так называемым *программистскими* уравнениями, когда часть уравнения представляет собой программный код с циклом с параметром (см. рис. 7.16).

Метод аналитического решения определяется типом решаемого уравнения

Если полином ***n***-й степени приравнять нулю, то мы получим **алгебраическое** уравнение. Основная теорема алгебры говорит о том, что такое уравнение имеет ровно ***n*** корней. Но во-первых, не все корни будут действительными и, возможно, вообще не существует ни одного действительного корня. А во-вторых, корни могут совпадать, т. е. быть кратными. Доказано, что не существует формул для корней алгебраического уравнения выше пятой степени. Но и формулы для `n = 5` настолько громоздки, что их использование лишено какой-либо практической пользы. Mathcad может символьно решать алгебраические уравнения вплоть до четвертой степени.

Если уравнение более высокой степени допускает частичное разложение на множители, то оно тоже может быть символьно разрешимо. Тут уместно вспомнить школьный метод подбора целого корня и теорему Безу. Если алгебраическое уравнение имеет целые коэффициенты, и делители свободного члена известны, то можно подобрать целый корень x_0 (если такой имеется) «вручную», либо использовав Mathcad. Поделив полиномиальную функцию на двучлен $(x - x_0)$, получим алгебраическое уравнение степени на единицу меньше. Если целый корень не подбирает-

[2] По другой классификации уравнения делят только на две группы — алгебраические и трансцендентные (не алгебраические — все остальные, включая рациональные и иррациональные). В русском языке нет слова, объединяющего понятия алгебраическое и трансцендентное (неалгебраическое) уравнения. Сейчас все чаще и чаще под алгебраическим уравнением подразумевают и чисто алгебраическое (полином, приравненный к нулю), и трансцендентное, отделяя тем самым алгебраические уравнение от дифференциальных и интегральных. Другая филологическая аналогия. В русском языке есть слово лес. Лес, в свою очередь, может быть бором, рощей, дубравой и т. д. Но есть языки, в которых нет обобщающего слова лес, а есть только бор, роща, дубрава... Развитие математики, точнее, аналитических методов решения уравнений имело следствием то, что у нас нет одного термина, охватывающего понятия алгебраического и неалгебраического уравнения.

ся, то такое уравнение теряет свои преимущества и становится в один ряд с другими типами уравнений.

Рассмотрим теперь **рациональные** уравнения. Такие уравнения содержат исключительно дроби, в числителях и в знаменателях которых находятся только многочлены. С помощью Mathcad эти уравнения легко формально преобразовать в алгебраические (см. пример на рис. 4.11). Правда, при таких преобразованиях может измениться область допустимых значений преобразуемого уравнения, т. к. знаменатель какой-то дроби может оказаться в числителе. Это порождает проблему посторонних корней, а поэтому решение рационального уравнения требует обязательной проверки (подстановки полученных чисел в **исходное** уравнение). Если все дроби в рациональном уравнении «одноэтажные», то проверку можно заменить предварительным поиском области определения рациональной функции, приравняв нулю все знаменатели. Если дроби «многоэтажные», то такая процедура потребует априорных упрощений. Если хоть в одном из знаменателей находится многочлен третьей или более высокой степени, то поиск допустимых значений оборачивается поиском корней нового алгебраического уравнения. В таком случае проверка — более экономный способ отсечения посторонних корней.

Иррациональными называют такие уравнения, которые помимо рациональных функций содержат радикалы (корни целых степеней — квадратные, кубические и т. п.), а все подкоренные выражения являются рациональными функциями. Известно, что радикалы четных степеней определены не везде в действительной области. Это обстоятельство приводит к необходимости находить область определения прежде, чем решать само уравнение. Фактически само уравнение следует сопроводить неравенствами, которые Mathcad тоже будет решать. Если этого не слелать, то уравнение по умолчанию будет решаться на области комплексных чисел, которые для большинства пользователей, исследующих реальные физические и другие явления, попросту бесполезны.

Вторая проблема, возникающая при решении уравнений с радикалами четных степеней — появление посторонних корней. Ведь основным методом решения иррационального уравнения является метод возведения обеих частей уравнения в нужную степень, а при возведении в четную степень как числа `x`, так и числа `-x`, мы получим один и тот же результат. В итоге получается новое уравнение, строго говоря, не равносильное исходному. Заметим, что есть аналитические способы решения и уравнений с кубическими (и другими нечетными) радикалами, которые тоже приводят к посторонним корням. Но тут снова можно прибегнуть к проверке, т. е. подстановке полученных числовых величин в исходное уравнение. Следует помнить, что лишние корни вполне могут принадлежать области определения функции, приравниваемой нулю.

Класс **трансцендентных** уравнений обширен. В него входят показательные, логарифмические, тригонометрические уравнения, а также уравнения, содержащие различные (а не только степенные) элементарные функции и композиции элементарных функций. Помимо проблем, связанных с областью определения, в таких уравнениях при наличии тригонометрических функций могут возникнуть проблемы с периодичностью решений.

Теперь поговорим о системах

В зависимости от того, сколько неизвестных входят в систему уравнений, можно выделить два типа систем: системы с одним неизвестным и системы с несколькими неизвестными. Классификация всех систем — занятие бессмысленное, поскольку специальные эффективные (матричные) методы решения разработаны только для систем линейных алгебраических уравнений с несколькими неизвестными. Такие системы еще называют линейными алгебраическими системами. Все другие системы решаются с помощью одних и тех же общих численных методов.

Системы с одним неизвестным можно решить так: найти по отдельности решение каждого уравнения системы, а потом выбрать одинаковые для всех уравнений числа. Часто можно поступить проще: сначала решить то уравнение, которое имеет наименьшее число корней, а потом все эти корни подставить в каждое из оставшихся уравнений и оставить лишь те значения, которые удовлетворяют сразу всем уравнениям системы.

Нелинейные системы с несколькими неизвестными решаются численными итерационными методами, требующими задания начального приближенного значения искомых неизвестных.

ВЫБОР МЕТОДА РЕШЕНИЯ УРАВНЕНИЯ

Посмотрим, чем же стоит руководствоваться при выборе метода решения конкретного уравнения или системы.

Для вычисления всех корней алгебраического уравнения не выше пятой степени рекомендуется использовать символьные вычисления, а также функцию `polyroots` (см. рис. 4.13), поскольку она не требует проведения процедуры локализации корней. Во всех остальных случаях придется либо локализовать корень на конкретном отрезке, либо использовать итерационные методы, имея достаточно хорошее начальное приближение и выбрав подходящую точность вычислений.

Прежде чем начать поиск корней уравнения, нужно на нескольких различных интервалах построить график функции, приравниваемой к нулю. Поведение графика даст ответ на несколько вопросов. Имеет ли функция действительный корень? Где он расположен? Сколько всего действительных корней? Отделены ли корни друг от друга, или они имеют некоторую точку скопления? Можно ли считать, что корни периодически повторяются, и чему равен период? Есть ли у функции точки разрыва, и насколько далеко от них лежат корни? Следует ли уменьшить значение системной переменной `CTOL`, чтобы различить два близко расположеных корня?

Когда ответы на все вопросы получены, тогда можно определиться с методом и точностью вычислений.

Заметим, что если проигнорировать этап построения графика функции, то, например, функция `root` может сработать некорректно. Правда, по графику нельзя определить, попадет ли в процессе решения в точку локального минимума невязки последовательность приближений. Если причина ошибки в этом, то нужно задать другое начальное приближение. Чем точнее выбрано начальное приближение корня, тем быстрее итерационный процесс будет сходиться.

Результатом решения системы будет численное или символьное значение вектора неизвестных величин. При символьном решении не надо вводить начальные значения, а при численном — надо. Если в системе всего два неизвестных, то построение трехмерных графиков функций, входящих в систему, позволит удачно подобрать начальное приближение для решения системы. В случае неудачного начального приближения появится сообщение об отсутствии сходимости последовательности итераций. Тогда придется начать все сначала, задав другое первое приближение.

Широкое использование компьютеров для аналитического или численного поиска корней уравнений привело к тому, что многие пользователи перестали чувствовать разницу между алгебраическими, рациональными, иррациональными и трансцендентными уравнениями. Более того, все эти уравнения стали называть просто алгебраическими. Так, в документации Mathcad сказано, что этот пакет может численно решать и системы алгебро-диффренциальных уравнений (DAE) — системы, где присутствуют и алгебраические и дифференциальные (см. ниже) уравнения. Хотя там могут быть и другие типы уравнений — рациональные, иррациональные и трансцендентные.

Раз мы упомянули дифференциальные уравнения, то следует сказать, что в Mathcad Prime к великолепной семерке добавилась еще одна функция — функция `Odesolve`.

7 + 1. ODESOLVE

Наша первая задача о «круизе» моторной лодки (см. рис. 4.1 и 4.2) имела существенное допущение: скорость лодки была постоянной. Но это условие выполнить практически невозможно, т. к. лодка по прибытии в один конец пути должна сбросить скорость, развернуться и пуститься в обратный путь. Можно, конечно, подправить задачу так: лодка достигает конечной точки и в этот момент эстафету принимает другая моторная лодка, движущаяся с такой же собственной скоростью, но в обратном направлении. Приблизить задачу об одной лодке к реальным условиям нам поможет еще одна встроенная функция Mathcad — функция `Odesolve`, предназначенная для решения (solve) обыкновенных (o — ordinary) дифференциальных (d) уравнений (e — equation) и их систем. Если при численном решении алгебраических уравнений мы получаем числа, подстановка которых в уравнения превращает их в тождества или в почти тождества, то при решении дифференциальных уравнений и их систем мы получаем уже не числа, а *функции*, подстановка которых превращает исходные дифференциальные уравнения в тождества. Заметим, что функция `Odesolve` в группе «Решение уравнений» стала восьмой (7 + 1 — см. выше) только в среде Mathcad Prime. В Mathcad 15 в группе «Решение уравнений» ее не было, но она была в другой группе.

Итак, **задача 4.** На моторной лодке, движущейся со скоростью v, заглушили мотор. Спрашивается, как будут меняться во времени пройденный ею путь[3]? Задача предельно упрощена — на лодку действует силы трения воды и воздуха, пропорциональные квадрату скорости лодки (см. рис. 4.17–4.20, где этот квадрат присутствовал). На рисунке 4.21 показано решение этой задачи с помощью функции `Odesolve` и его графическое отображение.

Коэффициент пропорциональности между инерцией и силой трения, записанный в уравнении на рис. 4.21 (масса лодки, помноженная на ускорение — на первую производную скорости по времени), состоит из двух частей, связанных с трением о воздух надводной части лодки и трением о воду ее подводной части. Эти коэффициенты пропорциональны плотности ρ среды (воздуха или воды) и площади поперечного сечения надводной и подводной частей лодки `S`. В уравнение можно добавить силу тяги мотора и моделировать также старт моторной лодки и ее последующее движение с переменной или постоянной тягой мотора. Если скорость лодки станет постоянной, то дифференциальное уравнение превратится в алгебраическое: сила тяги мотора будет уравновешиваться силой сопротивления воды и воздуха.

Задачу об остановке моторной лодки мы решили численно: функция `Odesolve` не ищет аналитического решения уравнения. Она формирует таблицу значений искомой функции с именем `x` (пройденный путь), по которой методом интерполяции создается непрерывная функция, график которой мы построили на рис. 4.21. На сайте **http://communities.ptc.com/videos/1471** можно просмотреть авторскую анимацию численного решения обыкновенного дифференциального уравнения методами Эйлера и Рунге-Кутты, а на сайте **http://communities.ptc.com/videos/1699** помещена анимация похожей задачи — остановки автомобиля под действием сил трения. Другие авторские динамические модели (вращение планет со спутниками, качание маятников, спуск на парашюте, ныряние в воду, движение в туннеле, запуск ракеты с подводной лодки, скатывание с горы и др.), реализованные в среде Mathcad, можно найти здесь **http:// communities.ptc.com/groups/dynamic-models-in-mathcad**.

[3] В этюде 2 мы решали подобную задачу в отношении автомобиля — см. рис. 2.7.

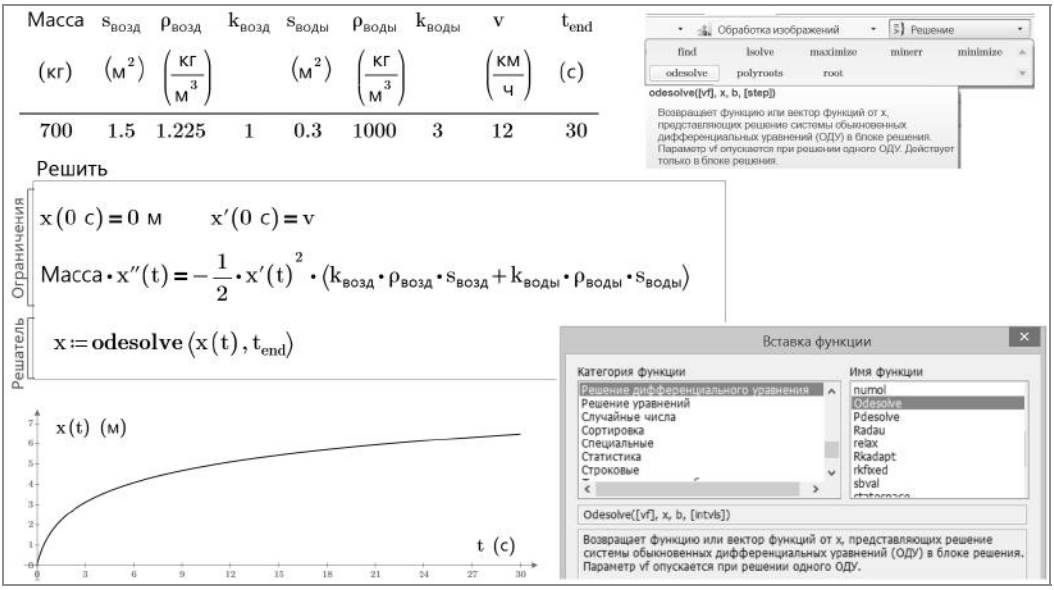


Рис. 4.21
Решение задачи об остановки лодки

В среде Mathcad нет средств аналитического (символьного) решения дифференциальных уравнений. Но их можно поискать и найти в Интернете. На рисунке 4.22 показано аналитическое решение задачи о движении моторной лодки — логарифмическая функция. Оно нашлось, поскольку исходное уравнение было достаточно простым: вторая производная функции пропорциональна квадрату ее первой производной. Но если с нашей задачи о движении лодки начать снимать ограничения, то символьного решения уже не будет, и нам придется возвращаться к численным методам — к функции `Odesolve`. Так, например, при торможении лодки площадь поперечного сечения ее надводной части уменьшается (лодка проседает в воду), а подводной части растет (вспомним, что самые быстроходные суда те, у которых подводная часть минимальна: глиссирующие суда, суда на подводных крыльях или на воздушной подушке). Коэффициенты $k_{возд}$ и $k_{воды}$ (см. блок исходных данных на рис. 4.21), в свою очередь, зависят от скорости и характера движения лодки: они одни при ламинарном обтекании тела и другие при турбулентном движении, когда за лодкой клубятся вихри воды и воздуха. У воды и воздуха разная вязкость, что тоже нужно учитывать при математическом моделировании движении лодки. Этим занимается очень интересная наука под названием *гидрогазодинамика*...

И последнее. После нахождения корня алгебраического уравнения (или решения системы уравнений) всегда, независимо от метода, стоит сделать *проверку* — подставить найденное значение (значения) в уравнение (в систему) и убедиться, что получилось тождество. Или почти тождество, если используются численные (приближенные) методы решения. Проверка решения рассматриваемого дифференциального уравнения может заключаться в построении графика баланса сил и произведения массы на ускорение (если иметь в виду задачу, показанную на рис. 4.21) в зависимости от аргумента полученной функции (`t` — см. рис. 4.21). Этот график должен совпадать с осью `x`. Кроме того, если есть аналитическое решение дифференциального уравнения, то его можно сравнить с численным решением. Мы это сделали в расчетном документе, показанном на рис. 4.22, где Mathcad-документ (численное решение задачи об остановке лодки) дополнен аналитическим решением.

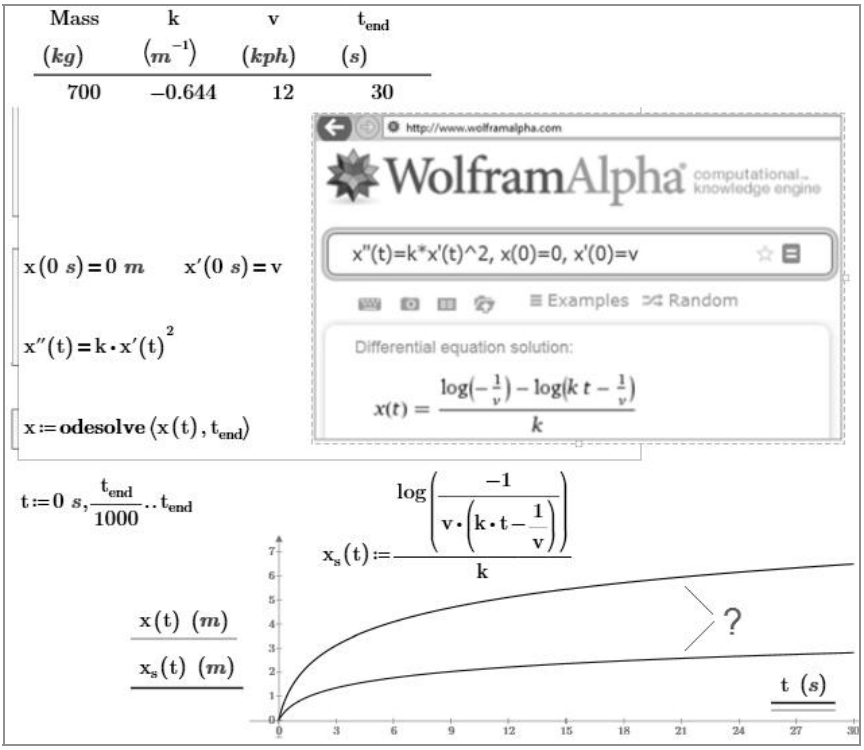


Рис. 4.22

Графическое сравнение аналитического и численного решения ОДУ

Почему на рисунке 4.22 графики торможения моторной лодки не совпали? Авторы долго ломали над этим головы, пока не сообразили, что в пакете Mathematica (а именно он работает на отмеченном на рис. 4.22 сайте) `log` — это натуральный логарифм, а в пакете Mathcad — десятичный. Этим и объясняется различный наклон логарифмических кривых. Если в формуле над кривыми функцию `log` заменить на `ln` или `log(..., e)`, то эти две кривые сольются в одну. В псевдоэмпирических формулах (о них речь шла в этюде 1) можно увидеть коэффициент 2.3, связывающий эти два логарифма.

Дифференциальные уравнения тоже можно разбить на типы. Есть, например, линейные дифференциальные уравнения, решения которых можно свести к решению алгебраических уравнений. Но это тема для отдельного этюда 7.

ВЫВОДЫ

Каждая из рассмотренных функций «великолепной семерки Mathcad» обладает своими особенностями и ограничениями. Прежде чем приступать к решению задачи, следует продумать, какая из этих функций приведет к поставленной цели, причем наилучшим образом.

Школьнику, студенту, инженеру или ученому необходимо (а в ряде случаев и достаточно) освоить «великолепную семерку Mathcad», особенности численных, графических и аналитических методов решения задач, чтобы успешно решать на компьютере свои учебные или профессиональные задачи [2].

ДИВЕРТИСМЕНТ 1. ВАН ГОГ В СРЕДЕ MATHCAD

Для решения алгебраических уравнений, как отмечено выше, предназначены встроенные функции `root` (корень) и `polyroots` (корни полинома). Если быть точным, то функции `root` и `polyroots`, возвращают не *корни* уравнений, а *нули* функций. В случае, когда функции `root` и `polyroots` необходимо применить к поиску корня *(корней)* уравнения, то его следует преобразовать в функцию: перенести все слагаемые в одну часть уравнения, приравняв полученную сумму к нулю, и работать с «ненулевой» частью как с функцией. Например, уравнение

$$a \cdot x = b$$

преобразуется сначала в уравнение $a \cdot x - b = 0$, а затем определяется функция $f(x) := a \cdot x - b$, нули которой совпадают с корнями исходного уравнения.

Функция `root` первоначально имела только два аргумента: первый из них — это анализируемая функция (в полном ее написании `y(x)`, а не просто `y`, как, например, у функции `Minimize` — см. рис. 4.17), у которой ищется нуль, а второй отмечал аргумент (неизвестное), найденное значение которого делает функцию равной нулю. В двухаргументной функции `root` заложен *метод секущих*, требующий первого приближения. Вблизи точки первого приближения определяется вторая опорная точка, равная сумме значений первого приближения и встроенной (системной) переменной `CTOL` (по умолчанию равной `0.001`), умноженной на значение первого приближения. Через эти две точки проводится *секущая*, пересечение которой с осью x дает очередное (третье) приближение. Итерации заканчиваются тогда, когда в очередном приближении значение функции будет отличаться от нуля менее чем на значение встроенной (системной) переменной `CTOL`. Реализацию метода секущих средствами программирования Mathcad, а также пошаговое приближение к корню уравнения (нулю функции) можно увидеть на сайте **http://twt.mpei.ac.ru/mas/worksheets/secant.mcd**. Фрагмент реализации этого метода представлен на рис. 4.23.

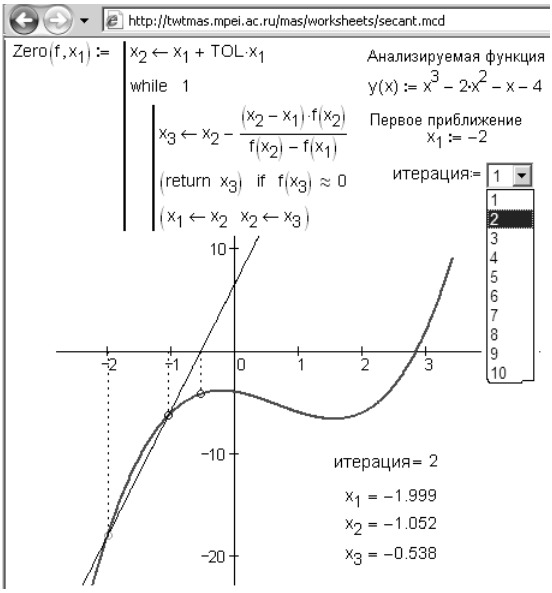


Рис. 4.23
Метод секущих

Посетитель этого сайта указывает в списке номер итерации (1, 2, 3 и т. д.) и видит, как секущая приближается к нулю. Если затем нажимать кнопки обозревателя Интернета «вперед-назад» (см. левый верхний угол рис. 2.23), то можно ускорить перемещение секущей, вызывая картинки из буфера обозревателя Интернета, и получить некую псевдоанимацию.

В Mathcad 13/14/15 появились встроенные средства отладки программ, которые, в частности, позволяют прослеживать работу встроенных инструментов поиска нуля функции. Подробнее об этом можно прочесть в этюде 13. На рис. 4.24 показано, какие промежуточные значения просчитываются при поиске нуля пользовательской функции `y(x)` с помощью встроенной двухаргументной функции `root`.

Рис. 4.24
Трассировка двухаргументной функции root

Чтобы проследить работу функции `root`, нужно анализируемой функции `y(x)` добавить встроенную функцию `trace(x)`, открыть окно трассировки через меню **Вид** и включить режим трассировки соответствующим переключателем окна **Отладка** (см. рис. 4.24). После этого нажатия клавиши <F9> будут приводить к тому, что в окне трассировки станут появляться числа промежуточных значений поиска нуля функции. Эту колонку чисел можно перенести в основное окно Mathcad и при необходимости использовать в расчетах. Это мы сделаем в этюде 13.

Поиск нуля функции, показанный на рис. 4.23, назвать методом секущих можно лишь с некоторой натяжкой. Дело в том, что этот метод требует в качестве первого приближения не одно, а два значения аргумента. При одном значении аргумента (одно значение для первого приближения) у кривой, отображающей на графике анализируемую функцию, можно провести не секущую, а *касательную*. Метод касательных (Ньютона) поиска нуля функции показан на рис. 4.25. Его недостаток состоит в том, что для его реализации требуется знание производной анализируемой функции.

В программе, показанной на рис. 4.25, производная от анализируемой функции определяется численно, что само по себе считается не вполне корректной математической операцией. Во встроенную функцию `root` и пользовательскую `Zero`, показанную на рис. 4.23, заложен метод *секущих — Ньютона*: при первой итера-

ции у кривой проводится (почти) касательная, затем приближения ведутся через секущие.

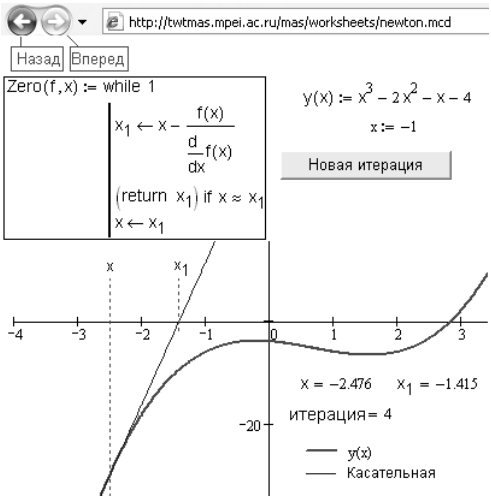


Рис. 4.25
Метод Ньютона

«Непредсказуемость» функции `root` в ее двухаргументном варианте, а также тот факт, что возвращаемый ею нуль зависит от значения переменной, находящейся вне списка аргументов функции `root`, заставила разработчиков Mathcad ввести в седьмую версию этого математического пакета *четырехаргументный* вариант функции `root`. Теперь ее синтаксис выглядит так:

```
root(f(var), var[, a, b])
```

Квадратные скобки в описании функции означают *необязательность* элементов, заключенных в эти скобки. Поменялась не только форма, но и содержание функции `root` — в ее четырехаргументный вариант заложен метод *половинного деления*. Его суть хорошо иллюстрирует старая притча о том, как можно поймать льва в пустыне: «...для этого нужно оградить ее забором, перегородить этот загон пополам и посмотреть, в какой половине оказался лев. Далее половину со львом снова разделить пополам и так поступать до тех пор, пока лев не окажется в некоем ящике». Программу метода половинного деления при «поимке» нуля функции можно увидеть на рис. 4.26.

На рис. 4.26 зафиксирован момент 13-й итерации, когда значение `y((a+b)/2)` стало по модулю меньше значения `CTOL` (решение задачи). На рис. 4.27 показана трассировка четырехаргументной функции `root`.

Функция `root`, реализуя метод половинного деления, уже не требует первого приближения, но предполагает задание интервала `[a, b]`, где необходимо найти нуль функции.

Метод половинного деления, заложенный в функцию `root`, требует, чтобы значения анализируемой функции в точках `a` и `b` имели противоположные знаки. В этом случае, если анализируемая функция непрерывна на отрезке `[a, b]`, то внутри него будет находиться, как минимум, один нуль. В противном случае функция `root` будет возвращать сообщение об ошибке, призывающей пользователя изменить значения `a` и `b`. Если же на отрезке `[a, b]` окажется более одного корня (три, пять, семь и т. д. при, повторяем, непрерывности анализируемой функции), то функция `root` вернет один из них. Какой именно — изображено на рис. 4.28 и

4.29, где показан результат сканирования прямоугольной области `[a, b]` с помощью четырехаргументной функции `root`.

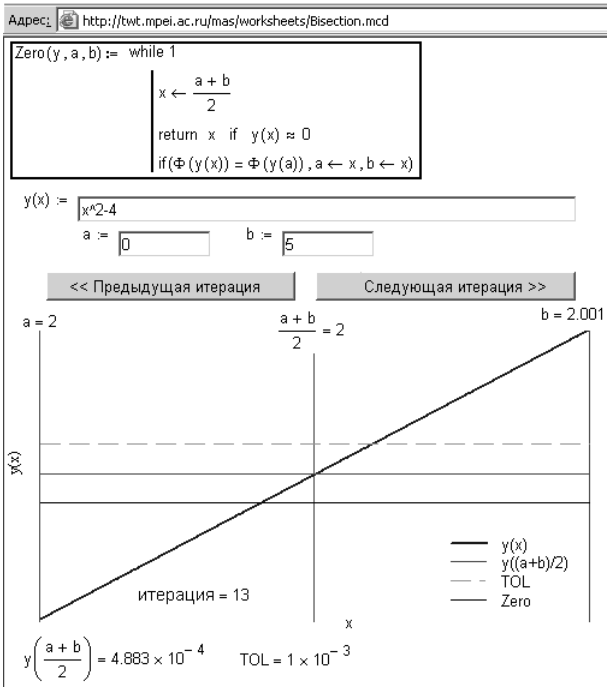


Рис. 4.26
Метод половинного деления

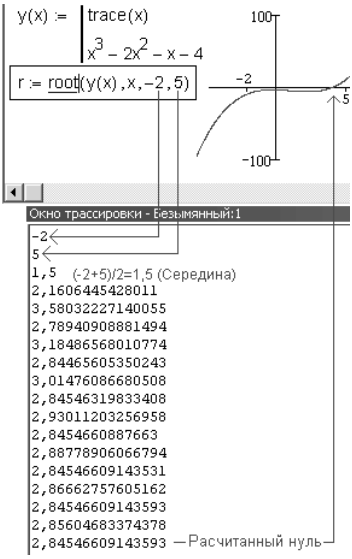
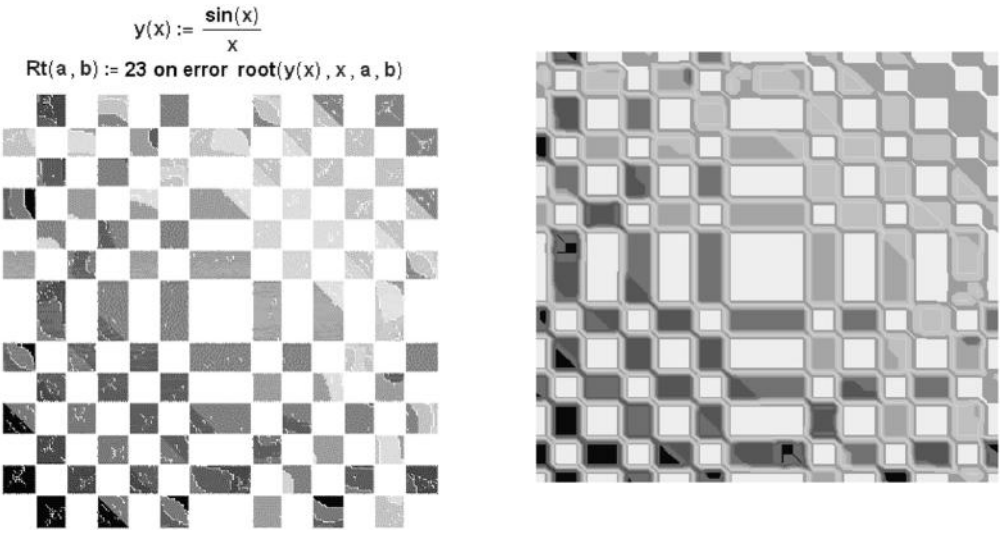


Рис. 4.27
Трассировка четырехаргументной функции root



a *b*

Рис. 4.28
«Отпечаток» нулей четырехаргументной функции root:
a — Mathcad 11; *b* — Mathcad 12–15.

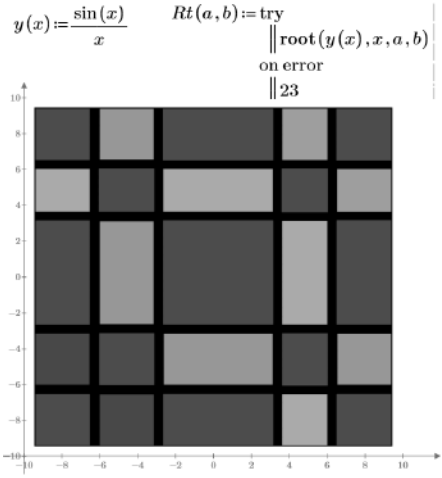


Рис. 4.29
«Отпечаток» нулей четырехаргументной функции root Mathcad Prime 3.

Белые области на рис. 4.28*a* и 4.28*b*, а также красные области на рис. 4.29 отмечают заданные значения переменных `a` и `b`, при которых нуль функции `sin(x)/x` не был найден. Цвет в закрашенных областях на рис. 4.28*a*, *b* фиксирует тот или иной найденный нуль: холодные тона — отрицательные нули, а теплые — положительные (если взглянуть на соответствующий цветной рисунок на сайте книги). На рисунке 4.29 диапазон изменения аргументов `a` и `b` был специально сужен до $[-3\pi, 3\pi]$, чтобы получились ровные четырехугольники. На рис. 4.28*a* и *b* он был равен [−20, 20].

Примечание. Если нуль функции не найден, срабатывает оператор Mathcad `on error`, возвращающий значение своего первого операнда (у нас оно равно 23, т. е. число, пре-

вышающее значение самого большого нуля в рассматриваемом диапазоне), если во втором операнде происходит сбой, зафиксировавший сообщение об ошибке.

Функцию `root` несложно приспособить и для решения *обратных задач*, когда необходимо найти значение аргумента функции по заданному значению самой функции (см. рис. 4.30, где ищется диаметр шара объемом 30 см^3).

$$V_{шар}(d) := \frac{4}{3}\cdot\pi\cdot\left(\frac{d}{2}\right)^3 \qquad V := 30cm^3$$

$$d := root\left(V_{шар}(d) - V, d, 0mm, 100mm\right)$$

$$d = 38.537\,mm$$

$$V_{шар}(d) = 29.967\,cm^3$$

Рис. 4.30
Решение обратной задачи с использованием функции root

Находить нули функции помогает также и *графика* Mathcad. На декартовом графике в среде Mathcad 15 *лупой* или *трассировкой* можно уточнить начальное приближение к нулю и/или сделать проверку правильности решения (рис. 4.31). В среде Mathcad Prime трассировка графика осуществляется перетаскиванием мышкой маркеров.

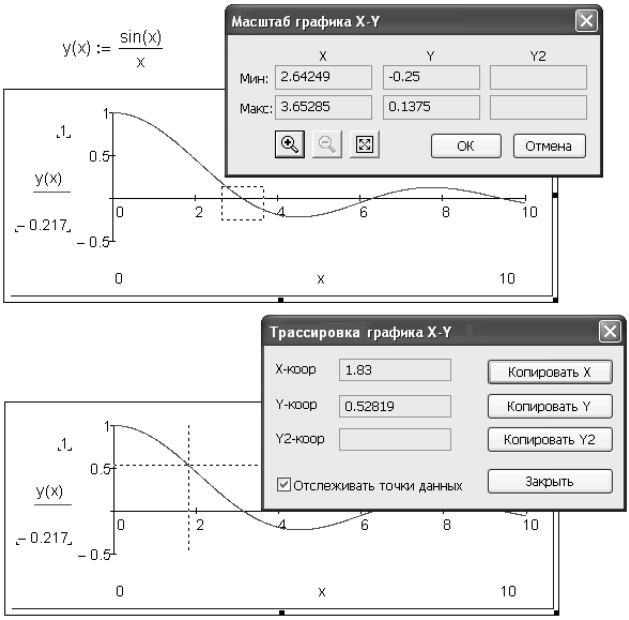


Рис. 4.31
Уточнение нулей с помощью графики

При трассировке графика выбор первого приближение (значение `1.83`, если иметь в виду случай, зафиксированный на рис. 4.31) автоматизирован через кнопку **Копировать X**. При этом значение `1.83` будет сохранено в буфере обмена и его можно будет вставить в сам Mathcad-документ — `x: = 1.83`.

В среде Mathcad 14 и 15 новая символьная математика позволяет находить все нули периодических функций, где `z` — это множество целых чисел (см. рис. 4.32).

Теперь перейдем к разбору методов решения в среде Mathcad *систем* уравнений, отметив при этом еще раз, что некоторые системы можно подстановкой свести к одному уравнению и использовать функцию `root`[4].

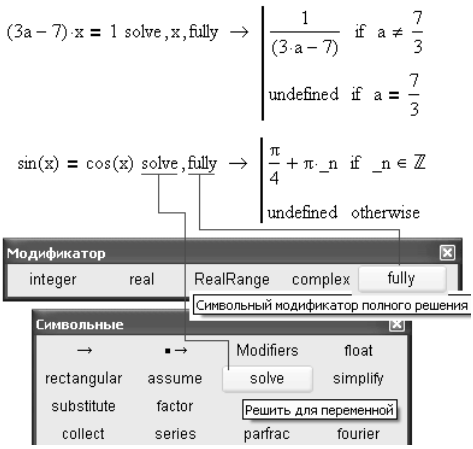


Рис. 4.32

Поиск всех нулей функции средствами символьной математики

На рис. 4.33 и 4.34 показано решение системы двух нелинейных алгебраических уравнений средствами символьной и численной математики Mathcad, соответственно. При решении системы инструментами символьной математики в переменную `All_Sol` (от англ. *all solution* — все решения) заносится матрица, строки которой — это корни системы. Их 24, что подсчитывается функцией `rows`. Из этих корней только четыре действительные. Это можно подсчитать, рассмотрев матрицу `All_Sol` (рис. 4.33) или построив график системы (рис. 4.34). Как был построен этот график, это второй вопрос — главное же, что на нем видны эти четыре точки пересечения двух кривых, отображающих два уравнения нашей системы.

Сама система на рис. 4.34 имеет неизвестные, обозначенные не `x` и `y`, а x_0 и x_1, т. е. в виде двух компонентов вектора `x` (см. этюд 2). К векторной записи неизвестных (x_0, x_1, x_2, ..., а не `x, y, z, ...`) приходится прибегать по ряду причин. Одна из них состоит в том, что, если верить документации Mathcad, блоком `Given/Find` можно решить систему с числом неизвестных до 256, а количество аргументов функции `Find`, как и любой другой встроенной функции с переменным числом аргументов, не может превышать 50. Убрать эту нестыковку и поможет векторная запись неизвестных системы уравнений. Но главное преимущество вектора неизвестных перед неизвестными скалярами состоит в том, что такой подход позволяет делать запись решения более компактной за счет, в том числе, использования операторов суммы и произведения.

Второе отличие решения, показанного на рис. 4.34, от решения, продемонстрированного на рис. 4.33, состоит в том, что функция `Find` не возвращает найденные значения неизвестных, превращающих уравнения в тождества, а формирует пользовательскую функцию с именем `Sol`, через вектор-аргумент которой в решение вводятся нужные первые приближения. Функция `Sol` в Mathcad-документе, показанном на рис. 4.34, вызывается пять раз при разных значениях этого первого приближения и возвращает четыре корня нашей системы и... один сбой, происшедший, когда функция `Find` не смогла решить задачу и вернула встроенное сообщение об ошибке, призывающее пользователя изменить первое приближение

[4] Знаменитый принцип математики, когда новую задачу можно свести к старой, уже разобранной.

и/или значения встроенных (системных) переменных `TOL` (от англ. *tolerance* — погрешность) и/или `CTOL` (от англ. *constrains tolerance* — погрешность ограничений).

Given

$$x^2 \cdot y^3 - 7 \cdot x^3 \cdot y^3 = 7$$

$$\left(x^2 + y^2\right)^2 = 7\left(x^2 - y^2\right)$$

Поиск корней системы

$$\mathrm{All_Sol} := \mathrm{Find}(x, y)^T \ \mathrm{float}, 5 \rightarrow \begin{pmatrix} 1.2014 & -.86820 \\ -.82517 - .63490 \cdot i & -.84785 - .37507 \cdot i \\ -.82517 + .63490 \cdot i & -.84785 + .37507 \cdot i \\ .38908 + .86932 \cdot i & -.66745 - .83448 \cdot i \\ .38908 - .86932 \cdot i & -.66745 + .83448 \cdot i \\ 2.5504 & -.39970 \\ -2.6876 + 6.1184 \cdot 10^{-2} \cdot i & -.17568 - .32065 \cdot i \\ -2.6876 - 6.1184 \cdot 10^{-2} \cdot i & -.17568 + .32065 \cdot i \\ -.28491 + .17760 \cdot i & -5.5662 \cdot 10^{-2} - 2.6751 \cdot i \\ -.28491 - .17760 \cdot i & -5.5662 \cdot 10^{-2} + 2.6751 \cdot i \\ 5.0922 \cdot 10^{-2} - .38482 \cdot i & -2.4418 \cdot 10^{-2} - 2.5597 \cdot i \\ 5.0922 \cdot 10^{-2} + .38482 \cdot i & -2.4418 \cdot 10^{-2} + 2.5597 \cdot i \\ 1.5092 \cdot 10^{-2} - .85613 \cdot i & 4.3778 \cdot 10^{-2} - 1.1628 \cdot i \\ 1.5092 \cdot 10^{-2} + .85613 \cdot i & 4.3778 \cdot 10^{-2} + 1.1628 \cdot i \\ .37739 + .17285 \cdot i & 6.9290 \cdot 10^{-2} - 2.7099 \cdot i \\ .37739 - .17285 \cdot i & 6.9290 \cdot 10^{-2} + 2.7099 \cdot i \\ 2.6909 + 6.5104 \cdot 10^{-2} \cdot i & .18103 - .33220 \cdot i \\ 2.6909 - 6.5104 \cdot 10^{-2} \cdot i & .18103 + .33220 \cdot i \\ -2.5582 & .38389 \\ -.34601 + .85818 \cdot i & .61366 - .86123 \cdot i \\ -.34601 - .85818 \cdot i & .61366 + .86123 \cdot i \\ -1.1363 & .84602 \\ .87730 - .64607 \cdot i & .88230 - .35602 \cdot i \\ .87730 + .64607 \cdot i & .88230 + .35602 \cdot i \end{pmatrix}$$

Подсчет числа корней

$$\mathrm{rows}(\mathrm{All_Sol}) = 24$$

Рис. 4.33

Полное решение системы двух нелинейных уравнений с помощью функций Find при ее аналитическом вызове (Mathcad 11/12/13)

Given $\quad (x_0)^2 \cdot (x_1)^3 - 7 \cdot (x_0)^3 \cdot (x_1)^3 = 7$

$$\left[(x_0)^2 + (x_1)^2\right]^2 = 7 \cdot \left[(x_0)^2 - (x_1)^2\right] \qquad \mathrm{Sol}(x) := \mathrm{Find}(x)$$

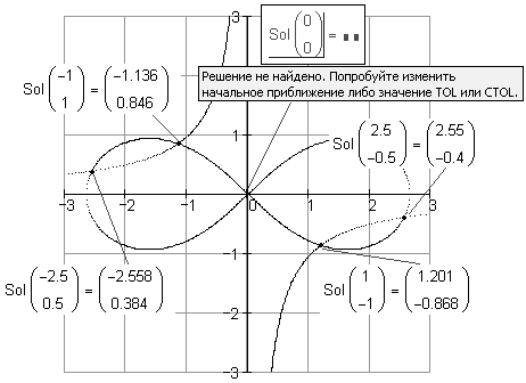


Рис. 4.34

Решение системы двух нелинейных уравнений с помощью функций Find при ее численном вызове

В среде Mathcad 14/15 ее новая символьная математика более строго подходит к полному решению системы нелинейных алгебраических уравнений и дает ответ с альтернативой, малая часть которого показана на рис. 4.35.

Given

$$x^2 \cdot y^3 - 7 \cdot x^3 \cdot y^3 - 7 = 0$$

$$\left(x^2 + y^2\right)^2 - 7 \cdot \left(x^2 - y^2\right) = 0$$

$$\text{Find}(x, y)\ \text{float}, 5 \rightarrow \begin{cases} \dfrac{462937896479782709679614153777 7}{761337209245131063987861344219 89} \\ \text{undefined} \quad \text{otherwise} \end{cases}$$

Рис. 4.35

Решение системы двух нелинейных уравнений с помощью функций Find при ее аналитическом вызове (Mathcad 14/15)

Можно *просканировать* поле первых приближений и узнать, из какой точки функция `Find` возвращает конкретный корень из четырех возможных, а где она дает сбой. Результаты такого сканирования представлены на рис. 4.36, 4.38, 4.40–4.42.

Можно построить декартов график корней, которые возвращает функция `Find` при изменении значения `x` от –10 до 10 и при фиксированном значении `y = 0.5`. Из графика видно, что в районе `x = 0.2÷0.7` функция `Find` дает сбои — возвращает не один корень из четырех возможных, а сообщение об ошибке, которое на графике отображается разрывом. Более интересная и более информативная картинка получается не на линии, а на плоскости при сканировании решения уже не по одному, а по двум аргументам (неизвестным).

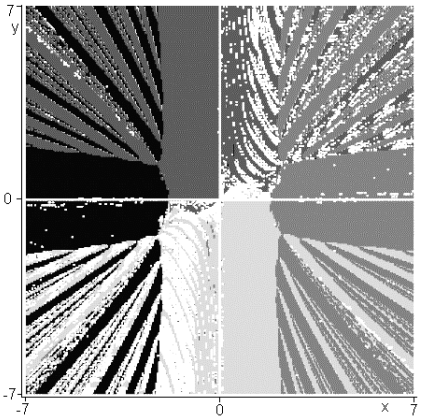


Рис. 4.36

«Картинка» численного поиска корней системы двух нелинейных алгебраических уравнений по алгоритму Левенберга — Маркворта

На рис. 4.36 цветом (оттенками серого, но на сайте книги можно найти и цветную картинку) выделены области на плоскости *Oxy*, принадлежащие одному из четырех корней нашей системы: точки, выбор которых в качестве первых приближений приводит к возврату функцией `Find` соответствующего, одного из четырех корней. Незакрашенные области («белые пятна» на карте) — это точки, старт от которых приводит к сбою при поиске корня, т. е. к возврату не пары чисел, а сообщения об ошибке. При построении картинки, показанной на рис. 4.36, также как и в случае, отображенном на рис. 4.29, был использован оператор `on error` (обработчик ошибок), позволивший гладко просканировать область.

Когда автор показал цветной вариант рис. 4.36 человеку, разбирающемуся в изобразительном искусстве, то тот сказал, что это... «Пейзаж в Овере после дождя» Винсента Ван Гога. Правда, этот «искусствовед» не успел надеть свои очки, а рис. 4.36 был показан ему издали. Но при более детальном сравнении была отмечена схожесть цветовой гаммы и «фактуры» (полоски грядок) рисунков. Так ли это — пусть читатель решит сам, сравнив рис. 4.36 и рис. 4.37, а, еще лучше, не рисунки книги, а их цветные аналоги на сайте, открыв их на экране дисплея в двух окнах одновременно[5].

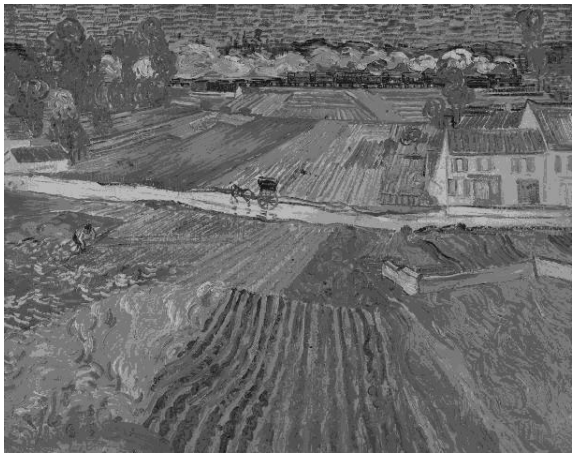


Рис. 4.37
Винсент Ван Гог «Пейзаж в Овере после дождя»

Но оставим в стороне «эстетическую сторону» рис. 4.36 и поговорим о практических делах.

Привязка цвета на графике к найденному корню более четко видна на рис. 4.38, где, во-первых, сужен диапазон сканирования на плоскости xy (не 7–7, а 3–3), а во-вторых, на «пейзаж» наложены графики наших анализируемых функций. Читатель может взять другие два уравнения и, используя данную или иную цветовую гамму корней и иные настройки их поиска (см. далее), попытаться создать новые «произведения изобразительного искусства», самостоятельные или перекликающиеся с известными.

Напомним, что в среде Mathcad 15 реализованы *три метода* решения уравнений и систем: метод Левенберга — Марквардта, метод сопряженных градиентов и квазиметод Ньютона — см. рис. 4.39, где показано локальное меню, появляющееся после нажатия правой кнопки мыши на слове `Find` и позволяющее выбирать нужный метод решения.

[5] Рисунок 4.36 хорошо будет «висеть» в комнате любителя Mathcad, а рисунок 4.29 хорошо «ляжет» на линолеум или кафель. Идея «математики под ногами» не нова. Пол в холле штаб-квартиры Математической ассоциации Америки (MAA) в городе Вашингтон выстлан плиткой в форме пятиугольного паркета. Поиск и классификация многоугольных паркетов является наглядной и интересной задачей теории замощений современной комбинаторной геометрии. К настоящему времени математикам известно, что любым треугольником и четырехугольником можно замостить плоскость, а также то, что существуют только три типа выпуклых шестиугольников, способных это сделать. Выпуклыми фигурами, имеющими более шести сторон, замостить плоскость невозможно. Это же невозможно сделать и при помощи правильных пятиугольников (пентагонов) — выпуклых многоугольников, все пять сторон которых равны друг другу. В настоящее время задача классификации многоугольных паркетов сводится к определению всех типов пятиугольных паркетов. Однако до сих пор математикам не известно точное число типов пятиугольников, способных замостить плоскость (источник **http://lenta.ru/articles/2015/08/20/pentagon**).

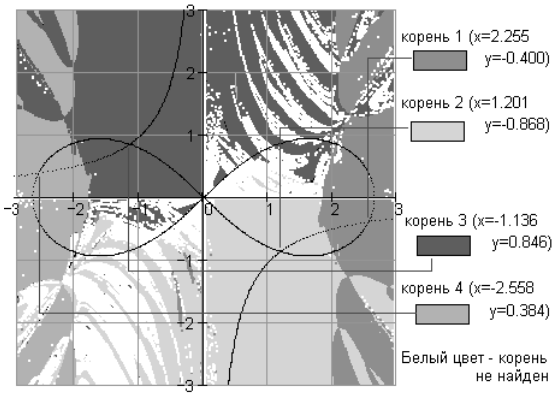


Рис. 4.38
«Пейзаж» численного поиска корней системы двух нелинейных алгебраических уравнений с наложенным графиком уравнений

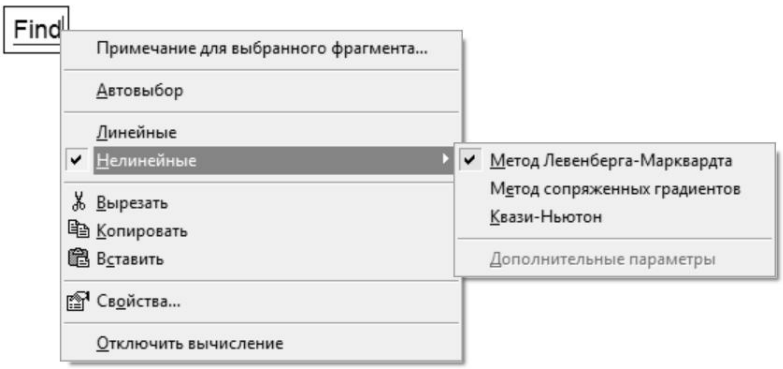


Рис. 4.39
Выбор метода решения системы нелинейных алгебраических уравнений

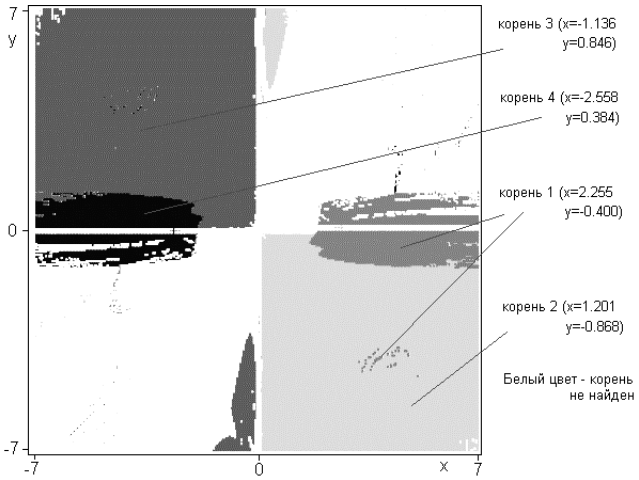


Рис. 4.40
«Пейзаж» численного поиска корней системы двух нелинейных алгебраических уравнений по алгоритму сопряженных градиентов

Замена метода Левенберга — Марквардта на метод сопряженных градиентов или квазиметод Ньютона резко меняет картину поиска корней нашей системы (рис. 4.40 и 4.41) и может служить неким графическим критерием сходимости того или иного алгоритма решения данной вычислительной задачи. В частности, процент «белых пятен», который несложно подсчитать, может давать уже не качественную, а количественную оценку сходимости конкретного метода решения на конкретной задаче.

Резко меняет картину поиска корней системы и изменение значения встроенной в Mathcad переменной `CTOL`, отвечающей за точность расчета. Функция `Find` возвращает значения своих аргументов, при которых невязка системы (разница между значениями левых и правых частей уравнений) не превышает по модулю значения переменной `CTOL`, которая при «создании пейзажей», показанных на рис. 4.36, 4.38, 4.40 и 4.41, была равна `0.001` (значение по умолчанию). На рис. 4.42 «пейзаж» нашей системы двух уравнений «прорисован» для `CTOL = 10`$^{-7}$.

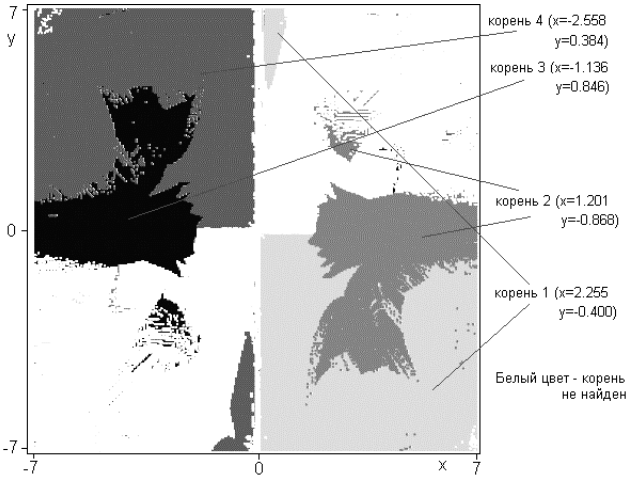


Рис. 4.41
«Пейзаж» численного поиска корней системы двух нелинейных алгебраических уравнений по квазиметоду Ньютона

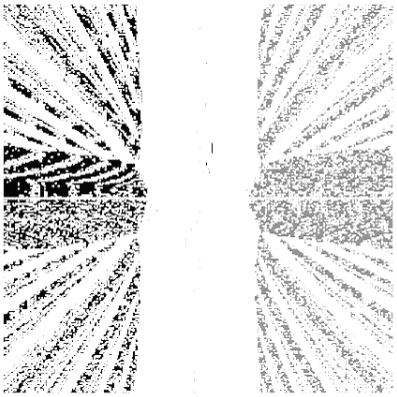


Рис. 4.42
«Пейзаж» численного поиска корней системы двух нелинейных алгебраических уравнений по алгоритму Левенберга — Марквардта с повышенной точностью

Из рисунка 4.42 видно, что повышение точности поиска корней системы уравнений приводит к тому, что только два корня из четырех можно найти. Кроме того, сильно расширяется область «белых пятен» — точек, откуда корни вовсе не рассчитываются.

Сканирование области неизвестных можно провести не только для системы, но и для одного уравнения вида `f(x, y) = 0`. Один пользователь Mathcad на форуме PTC Community попросил помочь ему построить график функции двух переменных, стоящей в левой части уравнения:

$$\sin(x \cdot \sin(x)) - \cos(y \cdot \cos(y)) = 0.$$

Стандартные средства Mathcad позволяют решить эту задачу только через построение линий уровня (Contour Plot — «контурная карта») [43]. Решения уравнения — все точки (и только они) линии нулевого уровня (`f(x, y) = 0` — береговая линия на географической карте). Но беда в том, что по вышеописанному конкретному уравнению четкую линию `f(x, y) = 0` получить никак не удавалось — все было как в тумане[6]: видно, что кривая очень сложная, но какая она именно, неизвестно.

```
f(x,y) := sin(x·sin(x)) − cos(y·cos(y))
x1 := −10   x2 := 10   y1 := −10   y2 := 10   t0 := time(0)
(y(x) x) = | N_points_of_Graph = 0
           | step_x = (x2 − x1)/700
           | step_y = (y2 − y1)/700
           | "vertical scanning of x-y surfase"
           | for x ∈ x1, x1 + step_x .. x2
           |   for y ∈ y1, y1 + step_y .. y2
           |     if (f(x,y)·f(x + step_x,y)) < 0
           |       | X[N_points_of_Graph] = x
           |       | Y[N_points_of_Graph] = y
           |       | N_points_of_Graph = N_points_of_Graph + 1
           | "horizontal scanning of x-y surfase"
           | for y ∈ y1, y1 + step_y .. y2
           |   for x ∈ x1, x1 + step_x .. x2
           |     if (f(x,y)·f(x,y + step_y)) < 0
           |       | X[N_points_of_Graph] = x
           |       | Y[N_points_of_Graph] = y
           |       | N_points_of_Graph = N_points_of_Graph + 1
           | (Y X)
t1 := time(0)   t := t1 − t0   Time of work of program (sec)  t = 124.41
Number of points on Graph  length(x) = 28828
```

Рис. 4.43
Программа сканирования корней уравнения

[6] Туман (Fog), кстати говоря, — это одно из средств повышения объемности трехмерных графиков. Так, на рис. 7.4 (стр. 144) можно видеть еще один «шедевр» автора — «ежик», пардон, мост в тумане. Два других инструмента повышения объемности графиков — это перспектива и освещение. Перспектива наряду с туманом была использована при строительстве моста: полотно проезжей части вдали сужается (Марк Твен говорил, что многие составные немецкие слова так длинны, что их нужно писать, придерживаясь законов перспективы).

На рис. 4.43 программа сканирования плоскости xy несколько усложнена (продублирована): сканирование ведется не только «по вертикали» (`for x ... for y ...`), но и «по горизонтали» (`for y ... for x ...`). Это делается для того, чтобы повысить четкость графиков.

Программа на рисунке 4.43 работает медленно[7], но верно — на рис. 4.44 и рис. 4.45 можно увидеть четкий график линии нулевого уровня нашей функции, сплетающийся в кружево. Получилась квадратная кружевная шаль.

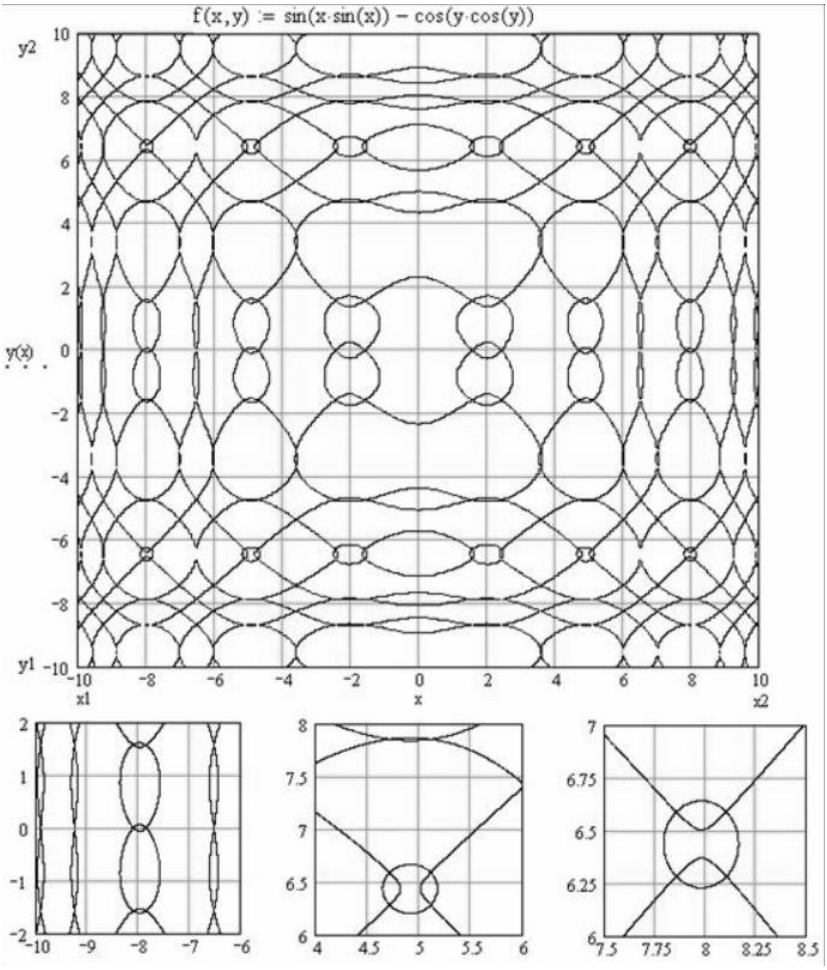


Рис. 4.44
Салфетка, сплетенная компьютером

[7] На рис. 4.43 с помощью встроенной Mathcad-функции time зафиксировано время счета — время генерации двух векторов (в них по 28 828 элементов) — 124 секунды (Pentium II, 233 МГц). Такие программы можно оптимизировать по скорости работы. В частности, сильно тормозит работу то, что внутри двойного цикла с параметром (`for x ... for y...`) дважды вычисляется значение анализируемой функции (в точке `x, y` и в точке `x, y+dy`: сверху и снизу от предполагаемого корня), хотя это можно делать один раз, используя при этом результат предыдущего вычисления. Изменяя программу в этом направлении, мы ускорим ее работу, но сделаем более сложной для понимания читателями.

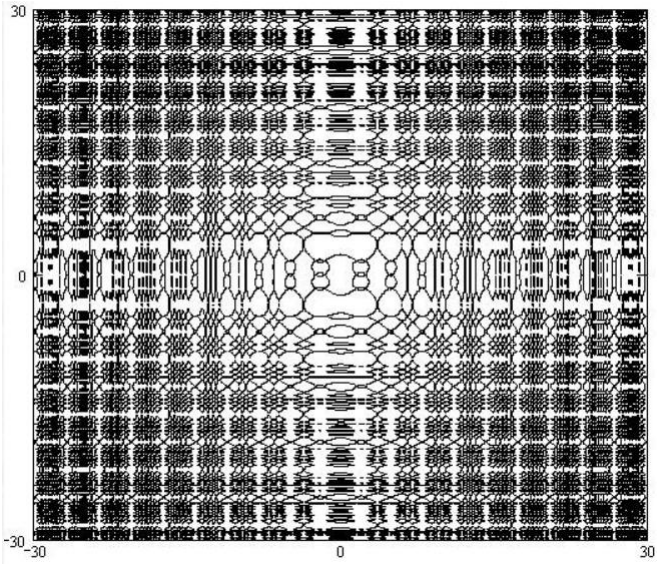


Рис. 4.45
Салфетка, сплетенная компьютером (вариант 2)

На рис. 4.46 можно увидеть полярные графики нашей функции, также кривые, сплетающиеся в кружево — в кружевную салфетку для круглого стола.

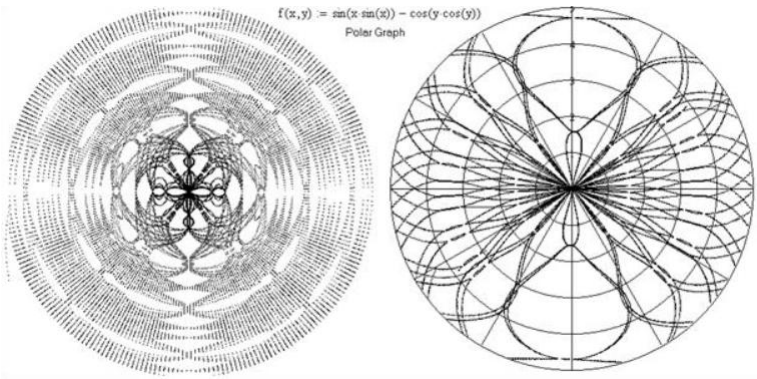


Рис. 4.46
Кружевная салфетка в полярных координатах

Методика построения графиков, изложенная выше, после небольшой доработки (см. программу на рис. 4.47) годится для графического отображения не только решений уравнений, но и решений неравенств.

Изобразительное искусство немыслимо без цвета. Автор расцветил переменные в программе на рис. 4.47:

- черный цвет — встроенные конструкции;
- синий цвет — пользовательские переменные и функции;
- малиновый цвет — пользовательские формальные переменные;
- зеленый цвет — пользовательские локальные переменные.

Примечание. Цветной вариант рис. 4.47 можно найти на сайте книги.

```
f(x, y) := sin(x·sin(x)) − cos(y·cos(y)) ≥ 0

x1 := −7    x2 := 7    y1 := −7    y2 := 7    N := 300    t0 := time( )

(y(x) x) := | (N_points_of_Graph ← 0   step_x ← (x2 − x1)/N   step_y ← (y2 − y1)/N)
            | for  x ∈ x1, x1 + step_x .. x2
            |   for  y ∈ y1, y1 + step_y .. y2
            |     if  f(x, y)
            |       | (X_N_points_of_Graph ← x   Y_N_points_of_Graph ← y)
            |       | N_points_of_Graph ← N_points_of_Graph + 1
            | (X Y)
```

Рис. 4.47
Цветная программа сканирования неравенства

Если график функции можно сравнить с кружевом, то график решения неравенства похож на другой образец народного творчества — на вышивку Ришелье. Правильнее, но менее красиво сказать, не «на вышивку», а «на вырезку Ришелье» — на ткани делают «художественные» дырки и обметывают края (см. рис. 4.48). У этой «вышивки» есть и другое название — broderie anglaise. Константин Левин в «Анне Карениной» подшучивал над своей женой Кити, занятой этим рукоделием: «Добрые люди штопают дыры, а она их нарочно вырезывает».

В детстве одного из авторов дома окружали такие салфетки — ими мама накрывала все, что можно накрыть: подушки на кровати, шкафы и даже телевизор КВН-49. Многое можно отдать, чтобы сейчас мама смогла бы накрыть такой салфеткой компьютер этого автора!

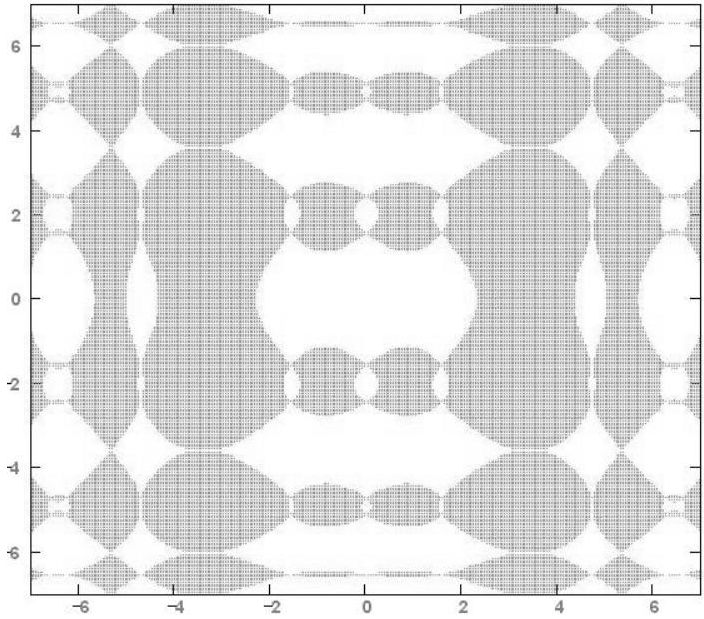


Рис. 4.48
Компьютерная вышивка Ришелье

ДИВЕРТИСМЕНТ 2. О БЕДНОМ СЛОВЕ GIVEN ЗАМОЛВИТЕ СЛОВО

Как отмечено в этом этюде функции Find и Minerr должны работать в паре с ключевым словом Given. Функции же Minimize и Maximize требуют этого слова не всегда, а лишь в тех случаях, когда оптимизация ведется с учетом ограничений. Ключевое слово Given имеет существенный недостаток — его нельзя вставлять в Mathcad-программы. На форумах по этому пакету часто задаются вопросы по этому поводу. Ответ на этот вопрос обычно такой — нужно вне программы создать функцию пользователя с опорой на функции Find или Minerr, а затем в программе вызывать ее. Такую функцию пользователя необходимо создавать и в том случае, если, например, необходимо построить график или вести оптимизацию по какому-либо параметру.

Конкретный пример, который также можно «притянуть за уши» к нашим «водным» задачам: через реку перекидывают цепи, чтобы построить подвесной мост. На рисунке 4.49 показано решение задачи оптимизации такой цепи — поиск длины цепи (S), которую подвесили между двух опор на одной высоте. Расстояние между точками крепления цепи задано (L). Известна линейная масса цепи (m_c). Определить длину цепи, при которой удерживающие ее силы в точках крепления (F) будут минимальны. Если длину цепи увеличивать, то эта сила будет расти за счет веса цепи (за счет вертикальной составляющей силы разрыва F_y). Если же цепь укорачивать, то эта сила будет расти за счет силы натяжения цепи (за счет горизонтальной составляющей силы разрыва F_x).

Для решения этой задачи нужно связать силу F и длину цепи S, т. е. создать функцию F(S). Описание решения этой задачи для более сложного случая (цепь с грузом, подвешенная в точках на разных уровнях от земли) будет дано в этюде 7. На рисунке 4.49 задаются три функции пользователя: цепная функция, ее производная[8] по x и функция длины цепи в зависимости от x. Величина x — это горизонтальное расстояние от левой опоры. Задача сводится к решению системы двух нелинейных алгебраических уравнений (равенство высот точек крепления цепи и длина цепи) с использованием блока Given-Find, где функция Find возвращает не два конкретных числа (численное решение системы двух уравнений), а формирует функцию пользователя с именем Ans. Затем эта функция-вектор (вектор потому, что функция возвращает два числа) раскладывается на две отдельные функции a(S) и h_c(S). Далее формируется целевая функция оптимизации F(S), по которой строится график и у которой уточняется минимум — точка, где значение производной равно нулю.

Оптимизировать можно не только конкретное физическое явление — провисание цепи, например, но и... компьютерное решение задачи. Если говорить о решении, показанном на рис. 4.49, то можно сообразить, что высота подвеса цепи h тут лишняя. А сама задача может включать в себя решение не системы двух уравнений, а одного уравнения. Этот подход отображен на рис. 4.50.

[8] Тут правильнее было бы написать «частная производная», т. к. функция y имеет не один, а целых четыре аргумента. Это человек считает, что функция y имеет один аргумент x (расстояние по горизонтали от левого подвеса цепи) и три параметра L (расстояние между точками подвеса цепи), h_c (минимальное значение y) и a («крутизна» цепной линии). Для пакета Mathcad все эти переменные также являются аргументами функции. Если, например, спросить у студентов, чему равна производная от функции xn, то 99 студентов из ста скажут, что не знают, или назовут выражение nx^{n-1}, и только один сообразит, что нужно уточнить, по какой переменной ведется дифференцирование — по x (степенная функция) или по n (показательная функция).

$$y(x,L,h_c,a) := h_c + a\cdot\left(\cosh\left(\frac{x-\frac{L}{2}}{a}\right)-1\right)$$ Цепная функция (см. этюд 7)

$$y'(x,L,a) := \frac{d}{dx}y(x,L,h_c,a) \rightarrow \sinh\left(\frac{x-\frac{L}{2}}{a}\right)$$ Производная цепной функции

$$S_c(L,a) := \int_{0m}^{L}\sqrt{1+y'(x,L,a)^2}\,dx$$ Длина цепной линии

$h := 1m \quad L := 1m \quad m_c := 1\frac{kg}{m}$ Исходные данные

$$F_y(S) := \frac{m_c\cdot S\cdot g}{2}$$ Вертикальная сила, действующая на цепь в точках подвеса (вес цепи равномерно распределяется на две точки подвеса)

Given $\quad a := 1m \quad h_c := 0.2m$ Первое приближение к решению

$h = y(0m,L,h_c,a)$ Слева цепь подвешена на высоте h)

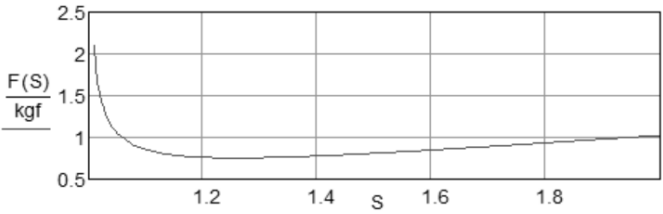
$S = S_c(L,a)$ Длина цепи $\quad a > 0m$ Это цепь, а не арка

$Ans(S) := Find(a,h_c) \quad a(S) := Ans(S)_0 \quad h_c(S) := Ans(S)_1$

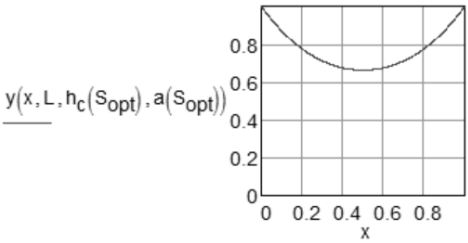
$\alpha(S) := atan(y'(L,L,a(S)))$ Угол наклона цепи у правого подвеса

$$F(S) := \frac{F_y(S)}{\sin(\alpha(S))}$$ Сила растяжения цепи в точках подвеса

$$S := L + \frac{L}{100}, L + \frac{L}{100} + \frac{L}{300} .. 30L$$ Дискретные значения S для построения графика



$$S_{opt} := root\left(\frac{d}{dS}F(S), S, L+\frac{L}{1000}, 2L\right) = 1.258m$$



$h_c(S_{opt}) = 0.662m$

$\frac{S_{opt}}{L} = 1.258$

$F(S_{opt}) = 0.754\cdot kgf$

Рис. 4.49
Создание функции пользователя с опорой на функцию Find

$$y(x,L,a) := a\cdot\left(\cosh\left(\frac{x-\frac{L}{2}}{a}\right)-1\right)$$ Цепная функция (см. этюд 7)

$$y'(x,L,a) := \frac{d}{dx}y(x,L,a) \rightarrow \sinh\left(\frac{x-\frac{L}{2}}{a}\right)$$ Производная цепной функции

$$S_C(L,a) := \int_{0m}^{L}\sqrt{1+y'(x,L,a)^2}\,dx$$ Длина цепной линии

$$L := 1m \qquad m_C := 1\frac{kg}{m}$$ Исходные данные

$$F_y(S) := \frac{m_C\cdot S\cdot g}{2}$$ Вертикальная сила, действующая на цепь в точках подвеса (вес цепи равномерно распределяется на две точки подвеса)

$$a := 1m$$ Первое приближение к решению

$$a(S) := \mathrm{root}\left(S - S_C(L,a),a\right)$$ Решение задачи

$$\alpha(S) := \mathrm{atan}(y'(L,L,a(S)))$$ Угол наклона цепи у правого подвеса

$$F(S) := \frac{F_y(S)}{\sin(\alpha(S))}$$ Сила растяжения цепи в точках подвеса

$$S := L+\frac{L}{100},L+\frac{L}{100}+\frac{L}{300}..2L$$ Дискретные значения S для построения графика

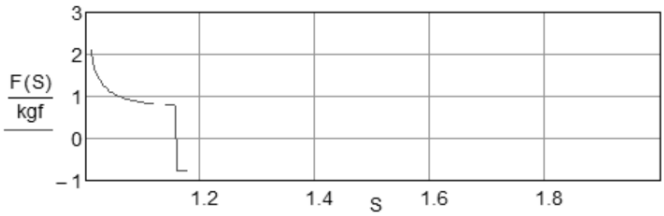


Рис. 4.50

Упрощение задачи об оптимальной форме цепи (неудачное использование функции root)

Но решение, показанное на рис. 4.50, оказалось неудачным: встроенная в Mathcad функция `root` в ее двухаргументном варианте (поиск нуля методом секущих) сначала выдавала правильный ответ (см. левую часть графика на рис. 4.50), потом перестала выдавать какой-либо ответ (пробел на графике), а затем стала выдавать неверный ответ — обсчитывать не цепь (`a > 0`), а арку (`a < 0`). При использовании блока `Given-Find` эта проблема решалась вставкой в систему двух уравнений ограничения `a > 0`. Это ограничение будет также учитываться, если функцию `root` задействовать в ее четырехаргументном варианте (поиск нуля методом половинного деления) — см. рис. 4.51.

Подходы к аналитическому (символьному) решению задачи об оптимальной цепи изложены на сайте **https://www.ptcusercommunity.com/thread/126010**.

Число 1.258 (отношение длины оптимальной цепи к расстоянию, между точками ее крепления на одном уровне) можно считать некой новой математической

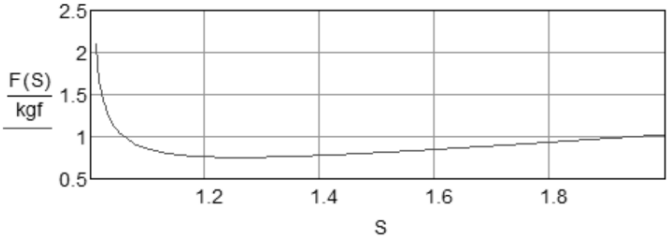
константой[9] и дать ей имя π_c (цепное (a **c**atenary, a **c**hain) число π). Можно начать гонку — находить все большее и большее количество знаков в этой константе наподобие того, как это делается в отношении «кругового» числа π: 3.142 (отношение длины окружности к ее диаметру).

$$a(S) := \mathrm{root}\left(S - S_C(L, a), a, 0.1\mathrm{m}, 10\mathrm{m}\right)$$ Решение задачи

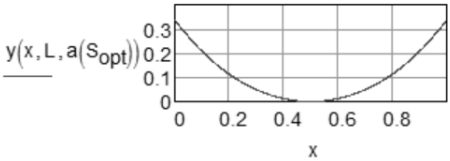
$$\alpha(S) := \mathrm{atan}(y'(L, L, a(S)))$$ Угол наклона цепи у правого подвеса

$$F(S) := \frac{F_y(S)}{\sin(\alpha(S))}$$ Сила растяжения цепи в точках подвеса

$$S := L + \frac{L}{100}, L + \frac{L}{100} + \frac{L}{300} .. 2L$$ Дискретные значения S для построения графика



$$S_{opt} := \mathrm{root}\left(\frac{d}{dS}F(S), S, L + \frac{L}{1000}, 2L\right) = 1.258\mathrm{m}$$



$$\frac{S_{opt}}{L} = 1.258$$

Рис. 4.51

Упрощение задачи об оптимальной форме цепи (удачное использование функции root)

Если требуется повесить цепь, веревку или трос между двумя столбами или стенами, то при отсутствии каких-то особых требований достаточно будет выбрать длину цепи, большей на 26% расстояния между точками крепления, находящимися, естественно, на одном уровне. В этом случае... см. выше. Обоснованность этого выбора мы еще обсудим в этюде 12.

Примечание. После написания этого дивертисмента и возникновения надежды на открытие новой математической константы[10] был сделан поиск в интернете по ключу «1.258 catenary», который привел к публикации в European Journal of Physics (2015; 36(2)) под названием «The optimum spanning catenary cable» (Оптимальная форма провисающего кабеля), в которой фигурировало число 1.258. Так что мы немного опоздали с приоритетом на эту константу. В другой публикации этого же автора

[9] Таких констант можно напридумывать множество. Еще один пример — числа на рис. 9.3 на стр. 200, задающие особые траектории движения трех планет. Но это скорее физические, а не математические константы. Есть сайты в Интернете, где собраны подобные константы. Адрес одного из них **http://www.people.fas.harvard.edu/~sfinch**.

[10] На циферблате математических часов, описанных в этюде 15 (см. рис. 15.1 на стр. 337), присутствуют три математические константы: число π, число e и мнимая единица i. Все эти три константы сгруппировались в операторе, отмечающем шесть часов.

(C.Y. Wang) обсуждалась оптимальная форма купола в виде поверхности, полученной вращением цепной линии вокруг оси Y (Journal of Engineering Mechanics. — 2015; 141(2))[11].

Задание читателям

1. Подобрать задачу по физике, к которой можно было применить большинство функций, описанных в данном этюде.

2. Запрограммировать методы численного решения алгебраических уравнений.

3. Решить с помощью сайта, показанного на рис. 4.22, задачи этого этюда.

4. Пловец, скорость которого v_1 в два раза меньше скорости течения реки v_2, хочет переплыть реку так, чтобы его снесло как можно меньше. Под каким углом к берегу он должен плыть?

5. Испытание осколочной гранаты производится в центре дна цилиндрической ямы глубиной 20 м. Образующиеся при взрыве осколки, скорость которых не превышает 300 м/с, не должны попадать даже на край ямы. Каким должен быть минимальный диаметр ямы?

6. Сделать такую простую в реализации, но интересную лабораторную работу. Подвешивается цепь в двух точках на одной высоте (см. последний график, показанный на предыдущей странице). Один конец цепи крепится на динамометре (безмене) так, чтобы можно было замерить силу натяжения цепи. Второй же свободный конец цепи перекидывается через блок. Длину провисающей части цепи можно менять, записывая при этом силу, которую показывает динамометр. Полученные точки в координатах «сила-длина цепи» нанести на первый график, показанный на предыдущей странице. Так мы сможем проверить теорию практикой.

[11] Такую поверхность называют катеноидой. Но во всех справочниках катеноида — это поверхность, образованную вращением цепной линии вокруг оси X. Такая поверхность образуется, если, например, между двумя кольцами, плоскости которых перпендикулярны линии, соединяющей их центры, натянуть мыльную пленку. А если цепную линию вращать вокруг оси Y? Если взять одно горизонтальное кольцо, поместить в него мыльную пленку, то ее провисание будет образовывать некую катеноиду?

Этюд 5

УГАДАЙ ОБРАЗ, ИЛИ ИГРА В МАТРИЦУ

В этюде будет рассказано, как по изображению (графическому файлу) можно получить матрицу с числами и что с этой матрицей можно потом сделать.

Математика: матрица, столбцы, строки, транспонирование матрицы, определитель, обратная матрица, целое число, эллипсоид, квадратное уравнение с тремя неизвестными, лемниската Бернулли, «объемная» лемниската Бернулли, треугольник, площадь треугольника, теорема косинусов, квадратное уравнение, теорема Пифагора, подобные треугольники.

Физика: цвет.

Информатика: поиск информации в Интернете, хранение цветного и черно-белого изображения в компьютере, преобразование изображение в матрицу и во вложенный массив данных, цикл с параметром, вложенный цикл с параметром, компьютерная анимация, форматы графических файлов, функции работы с изображением.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127825**

Есть такая телевизионная игра «Угадай мелодию». Проигрываются первые ноты какого-нибудь музыкального произведения, а выигрывает тот, кто первым это произведение назовет.

Такую игру можно перевести из audio в video — в визуальную плоскость и назвать «Угадай образ» или, если быть более точным, «Угадай человека или что-то еще по его неполному изображению». Портрет человека в этой игре нужно показывать на экране не весь сразу, а как мелодию — отдельными последовательными нотами, пардон, отдельными случайными точками (элементами растра), постепенно заполняющими экран. Чем больше точек будет показано, тем больше вероятность узнать образ. Ниже приведена и описана программа для Mathcad 15, которая проигрывает эту «портретную мелодию».

Портрет человека, отображаемый на мониторе компьютера[1], да и любое другое изображение — это, как правило, множество разноцветных точек или точек с оттенками серого. Такое изображение с точки зрения математики — это *матрица*, каждый элемент которой хранит целое число от 0 до 256, если говорить об изображении только с оттенками серого цвета. Цветное изображение — это уже три матрицы, хранящие значения интенсивности трех цветов (RGB: Red, Green, Blue — красный, зелёный, синий).

В последнее время бурно развивается направление в информационных технологиях, связанное с *распознаванием образов*. Два типичных примера практического использования достижений в этой науки.

Камера наблюдения фотографирует автомобиль, превысивший скорость на дороге или совершивший другое нарушение, а компьютер обрабатывает этот снимок, определяет номер госрегистрации автомобиля и выписывает штраф нарушителю.

При паспортном контроле на границе цифровая фотокамера фиксируют лицо и/или радужную оболочку глаза контролируемого, а компьютер сравнивает эти данные с теми, какие «вшиты» в чип предъявленного паспорта.

Недалеко то время, когда мы сможем подойти к банкомату и сказать: «Выдать сто рублей!» и это устройство выполнит нашу просьбу, узнав нас по внешнему виду и/или голосу.

Описываемая в данном этюде игра «Угадай образ» не только поможет нам развлечь друзей на вечеринке интересным конкурсом, но и приоткроет некоторые

[1] Или на экране телевизора. Но в настоящее время (эра цифрового телевидения) эти два устройства (компьютер и телевизор) все чаще и чаще оказываются в одном «ящике».

секреты современных технологий хранения и распознавания графических объектов.

На рисунке 5.1 можно видеть «разыгрывание» одного из самых узнаваемых графических объектов — портрета Че Гевары[2], смотрящего на нас с многочисленных настенных граффити, постеров и теннисок.

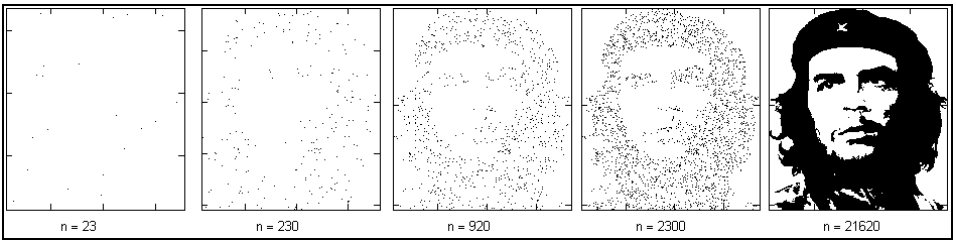


Рис. 5.1
Точечное рисование портрета Че Гевары

Портрет Че Гевары, показанный на правом изображении рисунка 5.1, состоит из 21 620 черных точек[3] на белом фоне. На рис. 5.1 он прорисован разным количеством точек (`n`): при `n = 230` нашего героя еще не узнать, а при `n = 920` можно уже поднимать руку и кричать: «Я узнал его!»

Компьютерный портрет рисуется так. В Интернете отыскивается изображение Че Гевары[4] и сохраняется на диске в черно-белом формате bmp-файла — см. рис. 5.2.

На рисунке 5.2 показан поиск с помощью Google портрета Че Гевары в Интернете и сохранение его на диске с помощью команды меню, появляющейся при нажатии правой кнопки мыши. Портреты можно не только выуживать из Интернета, но и делать самому — фотографировать друзей и близких на цифровую камеру, а файлы сохранять на диске для дальнейшей их обработки.

В среде математической программы Mathcad 15 есть команда меню вставки в расчет рисунка (рис. 5.3), применив которую и указав имя файла, можно нарисовать в Mathcad-документе выбранную картинку. Можно просто вставить рисунок из буфера обмена в расчет аккордом Ctrl+v, но в этом случае рисунок не будет привязан к выбранной переменной Mathcad. У нас на рисунке 5.3 она имеет имя `Che`.

Вставка картинки в Mathcad-документ — это сам по себе очень полезный инструмент. С помощью функции `if` (если) можно менять имя графического файла и подгружать ту или иную картинку в зависимости от расчетной ситуации. Так, например, делалось при показе формул пересчета видов концентрации — см. рис. 1.9 на стр. 24.

Черно-белую картинку, хранящуюся на диске в виде bmp-файла, можно превратить в *матрицу*, элементы которой будут фиксировать цвет точки изображения: 0 — черный и 256 — белый. Для этого в среде Mathcad есть функция READBMP — см. рис. 5.4.

[2] Че Гевара — очень противоречивая фигура. Одни считают его героем, борцом за свободу. Другие же называют чуть ли не террористом. Одним словом, все рисуют его черно-белыми красками. Мы тоже будем делать это, но в буквальном, а не переносном смысле — будем рисовать его портрет только черными и белыми точками.

[3] Есть такое направление в изобразительном искусстве пуантилизм (point, точка) — манера письма раздельными мазками правильной точечной (круглой) или прямоугольной формы.

[4] Там же в Интернете можно найти портреты и других узнаваемых людей (архитектурных силуэтов, промышленных объектов и т. д.) для нашей игры «Угадай образ».

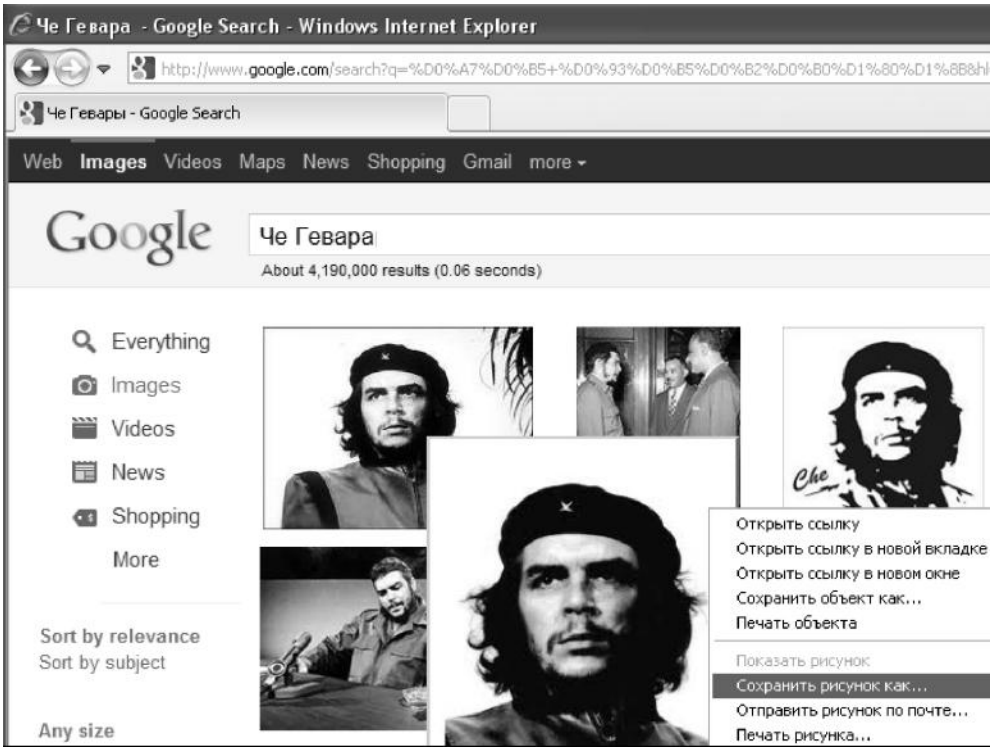


Рис. 5.2
Поиск портрета в Интернете и сохранение его на диске

Рис. 5.3
Вставка картинки в Mathcad-документ

Che := "Che.bmp"

Che

M := READBMP(Che) r := rows(M) − 1 = 206 c := cols(M) − 1 = 188

M =

	179	180	181	182	183	184	185	186	187	188
191	255	255	255	255	255	255	255	255	255	255
192	255	255	255	255	255	255	255	255	255	255
193	255	255	255	255	255	255	255	255	255	255
194	255	255	255	255	255	255	255	255	255	255
195	255	255	255	255	255	255	255	255	255	255
196	255	255	255	255	255	255	255	255	255	255
197	255	255	255	255	255	255	255	255	255	255
198	255	255	255	255	255	255	255	255	255	255
199	0	255	255	255	255	255	255	255	255	255
200	0	0	255	255	255	255	255	255	255	255
201	0	0	0	255	255	255	255	255	255	255
202	0	0	0	0	0	255	255	255	255	255
203	0	0	0	0	0	0	255	255	255	255
204	0	0	0	0	0	255	255	255	255	255
205	0	0	0	0	255	255	255	255	255	255
206	255	0	0	0	255	255	255	255	255	...

Рис. 5.4
Преобразование картинки в матрицу

На рис. 5.4 мы «распечатали» правый нижний угол матрицы `M`, чтобы показать, в каком виде будет храниться в матрице фрагмент левого плеча Че Гевары. В матрице `M` 207 строк (от нулевой до 206-й) и 189 столбцов (от нулевого до 188-го), хранящих информацию о 39 123 точках портрета, из которых 21 620 черных. На рисунке 5.4 показано, как с помощью встроенных в Mathcad функций `rows` и `cols` было подсчитано данное количество рядов и столбцов в матрице `M`. Эти числа нам будут нужны при дальнейшей обработке изображений.

Если у нас есть матрица — собрание чисел, сгруппированных по рядам[5] и столбцам, то с ней можно делать многое: класть на бок (в математике это называется транспонирование матрицы), переворачивать, сортировать и т. д. Если матрица квадратная, то у такой матрицы можно вычислить определитель, такую

[5] Правильнее говорить не «ряды», а «строки», как заведено в разделе математики, описывающем в том числе и матрицы (линейная алгебра). Но «ряды» начинаются с буквы «р», что делает более логичной работу с матричными функциями и операторами пакета Mathcad: c — column — столбец, r — row — ряд (а не строка).

матрицу с ненулевым определителем можно инвертировать (вычислять обратную[6] матрицу). Все эти инструменты есть в среде Mathcad.

Более того, пакет Mathcad оборудован специальной библиотекой обработки изображения — см. рис. 5.5.

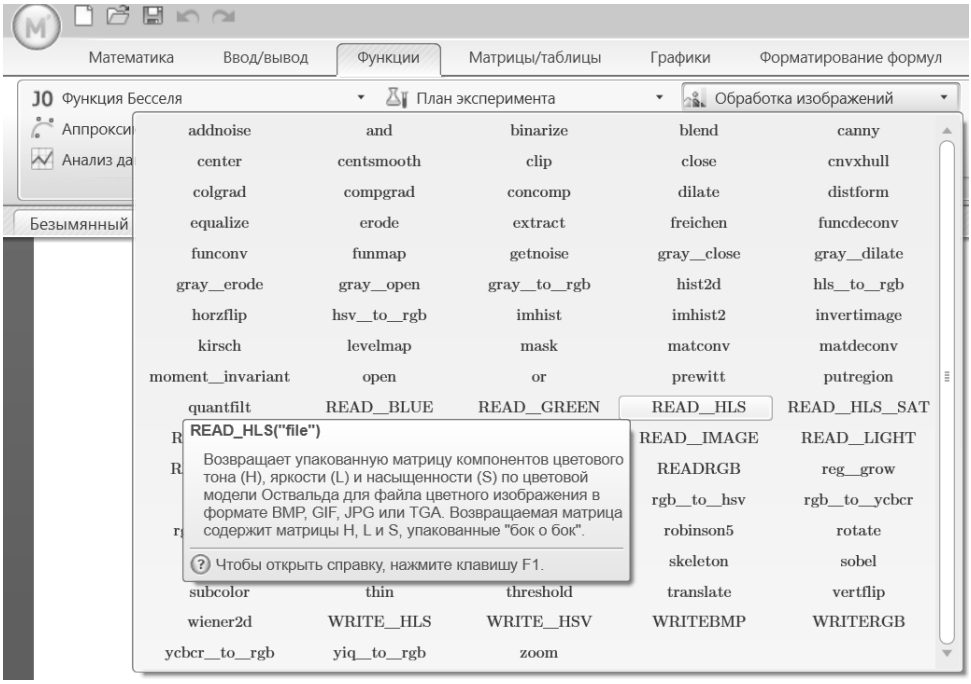


Рис. 5.5
Библиотека функций Mathcad работы с изображениями

Мы же нашу матрицу разложим на два вектора с именами `Ion` и `Iat`, которые будут хранить координаты черных точек нашего черно-белого портрета Че Гевары. Это делается в первой программе, показанной на рис. 5.6, двумя циклами `for`, перебирающими матрицу по столбцам и рядам. Если очередной элемент матрицы `M` будет равен нулю (черная точка портрета), то в вектор **X** будет записан номер данного столбца, а в вектор **Y** номер ряда.

Вторая программа, показанная на рис. 5.6, ядром которой является функция `rnd` (она будет главный действующим лицом этюда 16), возвращающая случайные числа, перемешивает элементы векторов `Ion` и `Iat`. После этого несложно средствами анимации Mathcad [44] поточечно рисовать нашего героя — см. его неполный портрет на рис. 5.6 и отдельные этапы такого «рисования» на рис. 5.1. Если же векторы `Ion` и `Iat` не перемешивать, то картинка будет выдаваться на экран компьютера построчно и можно четко видеть растры изображения. Так тоже можно вести игру «Угадай образ».

Программы и анимации, описанные в данном этюде [45], лежат на сайте владельца Mathcad — компании PTC (**http://communities.ptc.com/videos/1657**). В частности, на странице сайта **https://www.ptcusercommunity.com/message/263290** обсуждались детали ведения игры «Угадай образ» с цветными изображениями на примере знаменитого портрета Энди Уорхола «Лиз» — см. рис. 5.7. В левом верхнем углу этого рисунка показана распечатка матрицы `M`, хранящей портрет в трех векторах, каж-

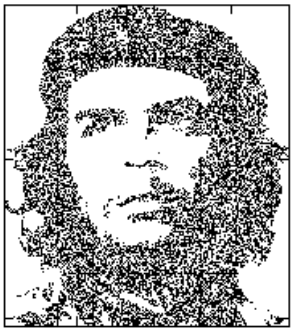
[6] Умножение обратной матрицы на исходную даёт единичную матрицу — матрицу с единицами на главной диагонали и с нулями в остальных местах.

дый из которых хранит матрицу интенсивности одного из трех цветов: RGB — Red, Green и Blue. С этими матрицами можно провести операции, показанные на рис. 5.6, и создать анимацию, кадры которой отображены на рис. 5.1, но в цветном исполнении.

```
(lon)   n ← 0
(lat) := for i ∈ 0 .. c
           for j ∈ 0 .. r
             if M[j,i] = 0
               X[n] ← i
               Y[n] ← −j
               n ← n + 1
         (X)
         (Y)

(lon)   n ← last(lon)
(lat) := for i ∈ 0 .. n − 1
           RN ← floor(rnd(n))
           while lon[RN] = "-"
             RN ← floor(rnd(n))
           (lon1[i]) ← (lon[RN])
           (lat1[i])   (lat[RN])
           lon[RN] ← "-"
         (lon1)
         (lat1)

FRAME := 500    n := ceil(last(lon) / 940) · FRAME    i := 0 .. n
```



lat_i

lon_i

n = 11500

Рис. 5.6

Преобразование матрицы в два вектора

Рис. 5.7
Угадай цветной образ

Примечание. Детство, отрочество и юность авторов книги прошло во время зарождения и расцвета не только космонавтики (см. этюд 9), но и аналогового телевидения. Они помнят, как родители копили деньги на первый телевизор КВН-49, как потом к нему прилаживалась линза (функция zoom — см. последнюю строку на рис. 5.5 и рис. 7.38 на стр. 174 [82]), как диктор объяснял, как можно перейти на вторую программу, переключив рычажок на задней панели телевизора, как в Политехническом музее в Москве появился первый цветной телевизор с тремя лучами в трубке для трех цветов, как появились плоские и цифровые телевизоры и т. д. Много воспоминаний навеивает этот этюд...

Но самый узнаваемый женский образ — это, конечно, образ Моны Лизы. С ним у первого автора книги случилась такая информационно-издательская история. Была у него другая книга с одним соавтором (см. **http://twt.mpei.ac.ru/ochkov/MC_PC/index.htm**). На обложку книги был помещен распечатанный крестиками и ноликами портрет Моны Лизы. Тогда такие распечатки принтеров были уже не в диковинку, но воспринимались с интересом. Соавтор (а он был фронтовиком, офицером-связистом) принес только что отпечатанную книгу домой и показал ее жене. Она спросила: «Кто это!?» Он ответил: «Как кто — Мона Лиза сиречь Джоконда». Жена грозно воскликнула: «Какая такая Джоконда!? Это Райка — твоя фронтовая радистка, с которой ты любовь крутил...» Скандал чуть ли не с рукоприкладством! Соавтор все это рассказал и показал Райкину фотографию — сходство полнейшее. Ну не с портретом из Лувра, а с картинкой на обложке книги. См. также рис. 15.7 на стр. 343.

ДИВЕРТИСМЕНТ 1. КОЛУМБОВО ЯЙЦО, ИЛИ ТОЧЕЧНАЯ СКУЛЬПТУРА

«Всеобщая история, обработанная «Сатириконом»[7]», так описывает событие, связанное с *колумбовым яйцом*: «...во время диспута с учеными Колумб, в доказательство шарообразной формы Земли раздавил на глазах присутствующих — без всяких приспособлений — куриное яйцо. Все ахнули и поверили Колумбу».

«Серьезная» же этимология выражения «колумбово яйцо» такая. Многие пытались вертикально поставить куриное яйцо, раскручивая его или подсовывая под него невидимые подпорки... Колумб же стукнул яйцом по столу, слегка надломил скорлупу и поставил яйцо вертикально. Так ли это было и было ли вообще, судить трудно, но выражение «колумбово яйцо» означает неординарное и несколько грубоватое решение какой-либо застарелой проблемы[8].

С колумбовым яйцом автор неожиданно столкнулся, работая в Интернете.

На форуме пользователей Mathcad PTC Community когда-то давно появился вопрос: «Как построить в среде Mathcad эллипсоид, описываемый уравнением $x^2/9+y^2/16+z^2/4 = 1$?» Тут же участники форума стали «раскручивать яйцо, подсовывать под него невидимые подпорки»... Каноническое («с подпорками») решение этой задачи выглядит так — см. рис. 5.8.

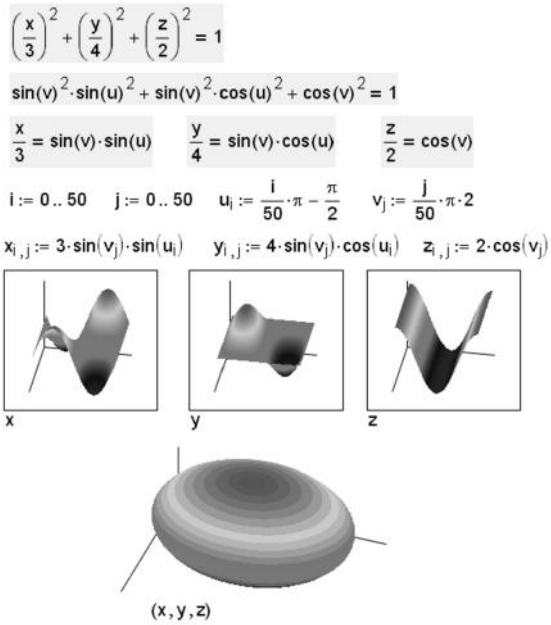


Рис. 5.8
«Параметрический» эллипсоид

Рассказывают такую историю. Воздушный шар вырвался из облаков. Летящие увидели на земле человека и закричали: «Где мы находимся?!» «Вы находитесь в корзине воздушного шара», — таков был ответ. Его дал *математик*. Только от

---

[7] Издание М. Г. Корнфельда. С.-Петербург, 1912. Сатирический пересказ истории древних, средних и новых времен, сделанный Тэффи, Осипом Дымовым и Аркадием Аверченко.

[8] Своеобразным «конкурентом» колумбова яйца можно считать Гордиев узел. Согласно древнегреческой легенде, фригийский царь Гордий особым (Гордиевым) узлом привязал ярмо к дышлу телеги. Предсказание оракула гласило, что тот, кто развяжет этот узел, получит господство над миром. По преданию, Александр Македонский в 331 г. до н. э. в ответ на предложение распутать узел разрубил его мечом.

математика можно услышать совершенно точный и совершенно бесполезный ответ[9].

Решение, показанное на рис. 5.8, также отличается абсолютной точностью и... абсолютной бесполезностью. Ведь спрашивающий о построении эллипсоида интересовался, конечно, не конкретным эллипсоидом, а общей методикой построения произвольной фигуры в среде Mathcad. Это типичная «математическая» ситуация: ответ на конкретный вопрос «затуманивается» хитрой методикой его поиска.

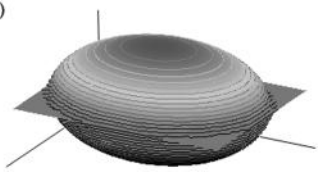
Яйцеобразная форма нашего эллипсоида заставляет вспомнить о лаврах Колумба, но не как первооткрывателя Америки[10], а как человека, решившего проблему одноименного яйца[11]. Посмотрим, как можно по-колумбовски упростить решение нашей задачи.

Первое «колумбово» решение задачи об эллипсоиде сводится к построению двух поверхностей, аналитическое описание которых получается при поиске корней уравнения эллипсоида относительно переменной z — см. рис. 5.9.

$$\left(\frac{x}{3}\right)^2+\left(\frac{y}{4}\right)^2+\left(\frac{z}{2}\right)^2=1 \text{ solve}, z \rightarrow \begin{bmatrix} \frac{1}{6}\cdot\left(-16\cdot x^2-9\cdot y^2+144\right)^{\frac{1}{2}} \\ \frac{-1}{6}\cdot\left(-16\cdot x^2-9\cdot y^2+144\right)^{\frac{1}{2}} \end{bmatrix}$$

$$f(x,y) := \text{if}\left[\left(\frac{x}{3}\right)^2+\left(\frac{y}{4}\right)^2<1, 2\cdot\sqrt{1-\left(\frac{x}{3}\right)^2-\left(\frac{y}{4}\right)^2}, 0\right]$$

$$g(x,y) := -f(x,y)$$



f, g

Рис. 5.9
Эллипсоид, склеенный из двух половинок

У эллипсоида на рис. 5.9 два недостатка — внешний и внутренний:

- внешний недостаток — на графике видно место «склейки» двух поверхностей, образующих эллипсоид;
- внутренний недостаток — данную методику не всегда удается перенести на замкнутую поверхность иной формы.

Второе «колумбово» решение задачи об эллипсоиде можно получить, если строить не сам эллипсоид (замкнутую поверхность 2-го порядка), а... геометрическое тело, ограниченное эллипсоидом.

На рисунке 5.10 показана совокупность точек с координатами X, Y и Z, которые оказались внутри эллипсоида. Остальные точки «упали на пол» — их Z-координата равна −2. Эллипсоид у нас получился как бы собранный их точек, но зато данная «точечная» методика, годится для любой замкнутой поверхности.

[9] Древнегреческий аналог этого анекдота таков. Похороны. Прохожий спрашивает: «Кого хоронят?» Ответ: «Вон того, кто в гробу лежит». А вот так звучит эта история, применительно к современному математику, преподающему в военном училище. Заблудившиеся спрашивают у прохожего: «Мы правильно идем к электричке?» «Нет, неправильно — не в ногу и не строем!»

[10] А этот факт до сих пор оспаривается многими.

[11] А приоритет Колумба в этом деле никем не ставится под сомнение.

$$ORIGIN := 1 \qquad N := 70000$$

$$X := runif(N, -3, 3) \quad Y := runif(N, -4, 4) \quad Z := runif(N, -2, 2)$$

$$i := ORIGIN .. N \quad Z_i := if\left[\left(\frac{X_i}{3}\right)^2 + \left(\frac{Y_i}{4}\right)^2 + \left(\frac{Z_i}{2}\right)^2 < 1, Z_i, -2\right]$$

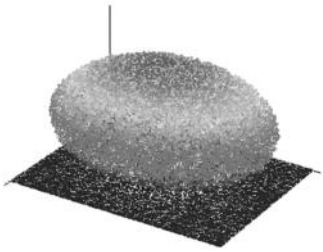


Рис. 5.10
«Точечный» эллипсоид

Так, на рисунке 5.11 можно увидеть, например, «объемную» лемнискату[12] Бернулли.

2. Volume lemniscate $(x^2+y^2)^2 - 7\cdot(x^2-z^2) + 5z^2 = 0$

$$N := 700000 \quad X := runif(N, -3, 3) \quad Y := runif(N, -1, 1) \quad Z := runif(N, -1, 1)$$

$$ORIGIN := 1 \quad i := ORIGIN .. N$$

$$Z_i := if\left[\left[(X_i)^2 + (Y_i)^2\right]^2 - 7\cdot\left[(X_i)^2 - (Z_i)^2\right] + 5\cdot(Z_i)^2 < 0, Z_i, 10\right]$$

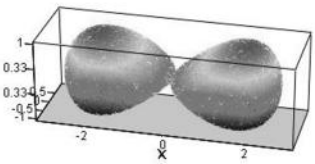


Рис. 5.11
«Объемная» лемниската Бернулли

На рисунке 5.11 лишние точки не упали на «на пол» (как на рисунке 5.10), а «ушли в небо»: их z-координата равна 10, а график отформатирован так (`z = -1...+1`), что точек не видно.

На рисунке 5.12 показано построение «объемной» лемнискаты Бернулли в среде Maple, где для такой работы есть соответствующая встроенная функция.

Заметим, что поверхность распалась на два несвязанных между собой фрагмента. Теоретическая же объемная лемниската состоит из двух одинаковых замк-

[12] Лемниска́та (от лат. lemniscatus — «украшенный лентами») — плоская алгебраическая кривая. Можно считать, что изначально название пришло из греческого языка, т. к. слово ληµνισχος (лемнискос) переводится как лента или повязка. Известно, что в Древней Греции «лемнискатой» называли бантик, с помощью которого прикрепляли венок к голове победителя на спортивных играх. Закон, по которому строится эта кривая, заключается в том, что произведение расстояний каждой точки лемнискаты до двух ее фокусов есть величина постоянная, равная квадрату половины междуфокального расстояния. Лемниската Бернулли названа в честь швейцарского математика Якоба Бернулли, положившего начало ее изучению. В технике лемниската применяется, в частности, в качестве переходной кривой на закруглениях малого радиуса, например, на железнодорожных линиях в горной местности или на трамвайных путях. Объемной лемнискатой называют поверхность, образованную вращением лемнискаты вокруг оси ox. Лемнискату мы уже видели на рис. 4.34 на стр. 91 и еще увидим на рис. 13.14 на стр. 303.

нутых поверхностей, соединяющихся в одной точке — в начале координат. На рисунке 5.13 можно видеть ваяние скульптуры из отдельных точек

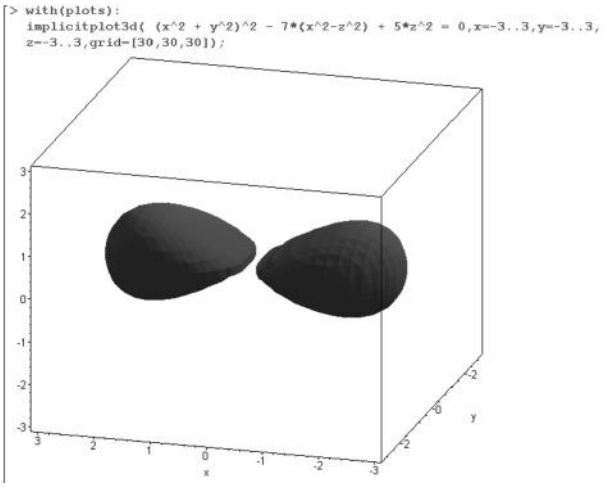


Рис. 5.12
«Объемная» лемниската Бернулли, построенная в среде Maple

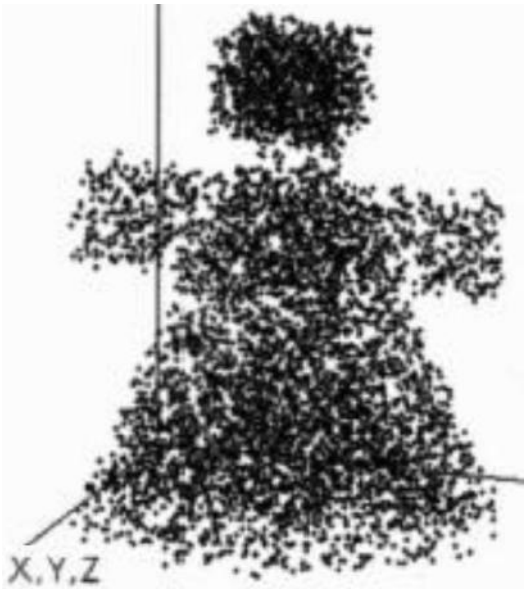


Рис. 5.13
«Точечная» скульптура в среде Mathcad

ДИВЕРТИСМЕНТ 2. ПАСЬЯНС

С матрицей (героиней этого этюда) мы имеем дело, раскладывая пасьянс, когда карты выкладываются на столе по рядам и столбцам. Поговорим о них!

Очень часто, решая ту или иную проблему, мы оказываемся в шкуре буриданова осла. Задача может иметь два альтернативных решения, как две охапки сена слева и справа от упомянутого животного. Все доводы «за» и «против» уравновешены. Как в этом случае поступить? Ехать или не ехать в командировку? Поку-

пать или не покупать еще один винчестер? Поистине гамлетовские вопросы задает нам жизнь!

Некоторые в таких ситуациях бросают монетку, другие загадывают мужчину или женщину и смотрят в окно, ожидая, кто первый появится. Но все это ненаучные методы. Монетка может куда-нибудь закатиться, а по улице как назло за целый час только кошка и пробежит...

Есть проверенный столетиями метод принятия подобных решений. Достаточно разложить пасьянс. Сошелся — решение принято и все сомнения прочь. Можно подыскать дополнительные доводы в его пользу и начать воплощать задуманное в жизнь. Удачно разложенный пасьянс психологически нас на это настраивает.

Но принять решение подобным образом иногда бывает трудно, так как не всегда под рукой есть колода карт, да и не совсем удобно раскладывать их на рабочем месте. Но это можно сделать и на экране дисплея. В среде Windows, а теперь и в Интернете есть игры-пасьянсы («Солитер» и «Косынка»), но мы придумаем что-нибудь новенькое, а главное, более занимательное и поучительное в духе этой книги — разложим в среде Mathcad старинный пасьянс «Турецкий платок». На это есть три причины.

1. Чтобы лучше освоить программную среду, нужно постараться решить в ней задачу, для этого крайне неподходящую. Никто еще не додумывался раскладывать пасьянсы в математических программах.

2. В детстве каждый нормальный человек, наигравшись, ломал игрушку, чтобы посмотреть, как она устроена. Посмотрим и мы, как тасуется колода и раскладывается пасьянс.

3. По традиции гадать на картах и раскладывать пасьянсы разрешается только в святки. Наш пасьянс можно считать числовой головоломкой, которую позволительно решать круглый год.

Mathcad-документ (см. рис. 5.14) позволяет разложить пасьянс «Турецкий платок» по следующим правилам. Из одной перетасованной колоды в 52 листа выкладывают картинкой вверх пять рядов по 10 карт в каждом. Последние две карты кладут в шестой (у нас он пятый[13]) неполный ряд на любое место, как правило, к первому (нулевому) и второму (второму) столбцам. Требуется распустить этот «платок», снимая из разных столбцов за один ход по две нижние одинаковые карты — тройки, дамы, тузы и т. д.

Для снятия карт нужно скопировать правую часть оператора `П = ...` (саму матрицу) на свободное место, подвести курсор к нужной текстовой константе (к карте: «2», «Д» — дама и т. д.), щелкнуть левой кнопкой мыши и нажать клавишу Delete. Вместо числа появится пустой квадратик. Саму же снятую карту можно оставлять под раскладкой пасьянса — см. правый нижний угол рис. 5.14.

Из разложенного пасьянса сняты две девятки. Потом можно снять две десятки, затем двух королей, затем две восьмерки и т. д. Если удастся попарно снять все 52 карты, то это означает, что в командировку все-таки ехать придется, а винчестер покупать надо.

Составление программы, формирующей матрицу с именем `П` (раскладка пасьянса) — прекрасное и, что не менее важно, занимательное средство изучения таких базовых понятий линейной алгебры, как вектор и матрица. Так, при формировании матрицы `П` транспонируется матрица `Масть` (она ставится «на попа» — матрица с одной строкой превращается в матрицу с одним столбцом, то есть в век-

[13] Можно было системной переменной ORIGIN присвоить значение единицы (см. этюд 2 с ремарками по этому поводу), чтобы не было такой раздвоенности. Но особой программистской нужды в этом нет. Рассказывают, что одного программиста забрали в армию, поставили в строй на первое место и скомандовали «По порядку рассчитайсь!» Программист сначала поинтересовался, по какой системе (двоичной, восьмеричной, десятичной или шестнадцатеричной) нужно рассчитаться, а потом сказал: «Нулевой!» Сейчас, слава богу, появились у нас специальные роты для решения задач кибервойн. Не все же в армии только муштрой заниматься...

тор). Далее с помощью функции `stack` составляется новая нерастасованная колода карт, где одна отсортированная масть идет за другой (вектор `Колода`). Затем в цикле с параметром (`for...`) с помощью цикла `while` и функции `rnd` идет формирование перетасованной колоды — вектора `Тасованная_колода`, который в конце программы двойным циклом с двумя параметрами (`for... for...`) складывается слоями в матрицу `П`.

```
П:= | "Пасьянс Турецкий платок"
    | Масть ← ["2" "3" "4" "5" "6" "7" "8" "9" "10" "В" "Д" "К" "Т"]^T
    | Колода ← (stack(stack(Масть, Масть), stack(Масть, Масть)))
    | for Карта ∈ 0..51
    |   while 1
    |     Случайное_число ← floor(rnd(52))
    |     if Колода[Случайное_число] ≠ "пусто"
    |       Тасованная_колода[Карта] ← Колода[Случайное_число]
    |       Колода[Случайное_число] ← "пусто"
    |       break
    | for Столбец ∈ 0..9
    |   for Ряд ∈ 0..5
    |     Карта ← 10·Ряд + Столбец
    |     Пасьянс[Ряд, Столбец] ← if(Карта ≤ 51, Тасованная_колода[Карта], "")
    | Пасьянс
```

Вывод на дисплей раскладки пасьянса

$$П=\begin{bmatrix} \text{"2"} & \text{"Д"} & \text{"6"} & \text{"7"} & \text{"5"} & \text{"В"} & \text{"Д"} & \text{"4"} & \text{"6"} & \text{"9"} \\ \text{"Т"} & \text{"8"} & \text{"3"} & \text{"7"} & \text{"10"} & \text{"Д"} & \text{"4"} & \text{"3"} & \text{"8"} & \text{"Д"} \\ \text{"4"} & \text{"К"} & \text{"7"} & \text{"2"} & \text{"6"} & \text{"8"} & \text{"В"} & \text{"3"} & \text{"Т"} & \text{"Т"} \\ \text{"5"} & \text{"В"} & \text{"2"} & \text{"2"} & \text{"К"} & \text{"К"} & \text{"3"} & \text{"10"} & \text{"7"} & \text{"5"} \\ \text{"4"} & \text{"8"} & \text{"В"} & \text{"9"} & \text{"К"} & \text{"10"} & \text{"Т"} & \text{"10"} & \text{"9"} & \text{"5"} \\ \text{"6"} & \text{"9"} & \text{""} & \text{""} & \text{""} & \text{""} & \text{""} & \text{""} & \text{""} & \text{""} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{"2"} & \text{"Д"} & \text{"6"} & \text{"7"} & \text{"5"} & \text{"В"} & \text{"Д"} & \text{"4"} & \text{"6"} & \text{"9"} \\ \text{"Т"} & \text{"8"} & \text{"3"} & \text{"7"} & \text{"10"} & \text{"Д"} & \text{"4"} & \text{"3"} & \text{"8"} & \text{"Д"} \\ \text{"4"} & \text{"К"} & \text{"7"} & \text{"2"} & \text{"6"} & \text{"8"} & \text{"В"} & \text{"3"} & \text{"Т"} & \text{"Т"} \\ \text{"5"} & \text{"В"} & \text{"2"} & \text{"2"} & \text{"К"} & \text{"К"} & \text{"3"} & \text{"10"} & \text{"7"} & \text{"5"} \\ \text{"4"} & \text{"8"} & \text{"В"} & & \text{"К"} & \text{"10"} & \text{"Т"} & \text{"10"} & \text{"9"} & \text{"5"} \\ \text{"6"} & \Downarrow & & \Downarrow & & & & & & \\ & \text{"9"} & & \text{"9"} & & & & & & \end{bmatrix}$$

Рис. 5.14
Пасьянс с Mathcad

Программу, формирующую матрицу `П`, можно развить: заставить ее отбраковывать явно нерешаемые раскладки — такие, например, где в одном столбце оказались три или даже четыре одинаковые карты. Еще одна тупиковая ситуация возникает тогда, когда две пары карт крест-накрест закрывают друг друга. Правильные ли мы сняли девятки при первом ходе — нужную ли пару из трех девяток? Это будет хорошим упражнением, закрепляющим навыки работы с программированием и «матричными» операторами и функциями в среде Mathcad. А вот более сложное задание читателю: доработать программу так, чтобы она сама раскладывала пасьянс либо на худой конец сообщала, что его можно решить, просто снимая снизу первые подвернувшиеся одинаковые открытые карты. Более умная стратегия, повторяем, подразумевает выбор двух одинаковых карт из трех или четырех открытых.

Расчет, показанный на рис. 5.14, при каждом открытии файла будет выдавать одну и ту же раскладку. Это связано с тем, что функция `rnd` генерирует ряд *псевдослучайных* чисел — повторяющихся в каждой серии вызовов. Чтобы выдавалась случайная раскладка, нужно до раскладки пасьянса выполнить функцию `rnd` случайное число раз, опираясь на случайный «хвостик» машинного времени Mathcad — см. рис. 5.15.

```
Область
t := time(0)        N := floor((t − round(t)) · 10000) = 3280        N := for i ∈ 0..N
                                                                        N ← rnd(1)
```

Рис. 5.15
Генерация истинно случайных чисел

Помимо линейной алгебры, наша программа затрагивает и другие интересные разделы математики — теорию вероятностей, статистику (см. функцию `rnd`, генерирующую псевдослучайные числа). Интересный вопрос: можно ли составить программу, высчитывающую вероятность сходимости того или иного пасьянса? Считается, что пасьянс «Солитер», входящий в стандартную поставку Windows, раскладывается при любых начальных раскладках. И если пасьянс не сошелся, то это вина человека. Но это только гипотеза, которую читатель при желании может доказать.

Часто бывает так, что человек прекрасно делает то, чему он учился играючи и с большим удовольствием: ходит, говорит, плавает, катается на велосипеде и т. д. и т. п. Автор, например, задолго до школы научился считать, играя в карты: валет — двойка, дама — тройка и т. д. И сейчас, обращаясь к своим студентам, не устает повторять: главное, что нужно получать от учебы, — это не знания, не практические навыки, а... удовольствие. Не получая радости от учебы, от повседневного труда, человек тратит свою жизнь впустую... Например, если тяжело идет освоение линейной алгебры, напишите программу раскладки пасьянса и постарайтесь получить удовольствие.

Начинать изучение векторов и матриц (массивов данных) в курсе программирования (информатики) можно со знакомства со встроенными функциями и операторами конкретной программной среды. А можно, раскладывая пасьянсы.

Одна из самых рутинных операций в машинной графике — узнать, находится ли заданная точка внутри какого-либо многоугольника. Это одна из задач, которую человек решает «сходу», а компьютеру нужна для этого специальная программа. Сходу — это с первого взгляда, если, конечно, точка не расположена слишком близко от одной из сторон многоугольника. А любая плоская фигура на экране дисплея — это многоугольник: точки растра дисплея, через которые проходит ломаная замкнутая линия, состоящая из отрезков прямых.

Есть множество алгоритмов решения этой задачи. Один из них состоит в разбивке многоугольника на отдельные треугольники и определении, находится ли точка в одном из них. А в случае треугольника (простейшего многоугольника) эта задача может быть решена так. Берем точку P и соединяем ее со всеми вершинами треугольника. Получаем три треугольника с вершинами в точке P и с основаниями — сторонами исходного треугольника. На рис. 5.16 показан один из таких вспомогательных треугольников. Вычисляем площадь исходного треугольника и площади трех построенных треугольников.

Точка P лежит внутри треугольника тогда и только тогда, когда площади всех трех вспомогательных треугольников меньше площади исходного. Как вычисляются площади треугольников по координатам их вершин — см. этюд 13. В программе, показанной на рис. 5.10, эти операторы скрыты в захлопнутой области Mathcad.

Иногда приходится делить на треугольники сам треугольник. Вот какая поучительная математическая история случилась с одним из авторов.

Однажды (первый автор переходит к рассказу от первого лица) я помогал внучке решать такую задачу по математике: дан треугольник, у которого одна сторона равна 12 см, а один из углов, примыкающий к этой стороне, равен 120°. Сторона, лежащая напротив этого угла, равна 28 см. Найти длину третьей стороны треугольника и высоту, проведенную от заданного угла.

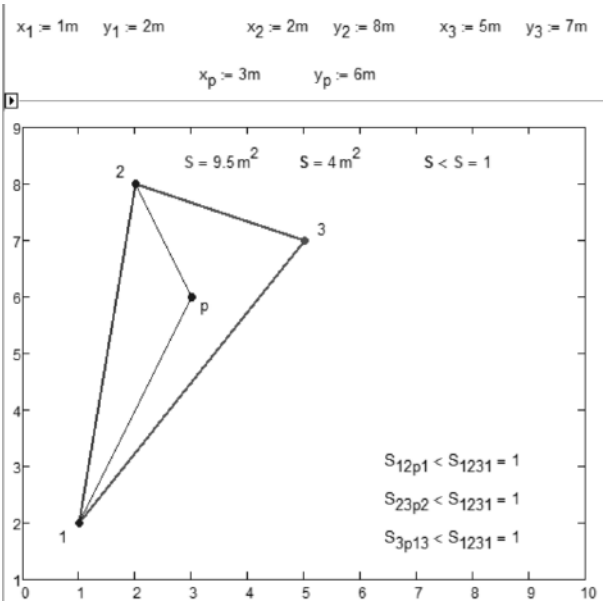


Рис. 5.16
Находится ли точка в треугольнике?

Я тут же подсел к компьютеру, составил систему шести алгебраических уравнений и без проблем решил ее в среде Mathcad с помощью решателя Solve — см. рис. 5.17 и 5.18.

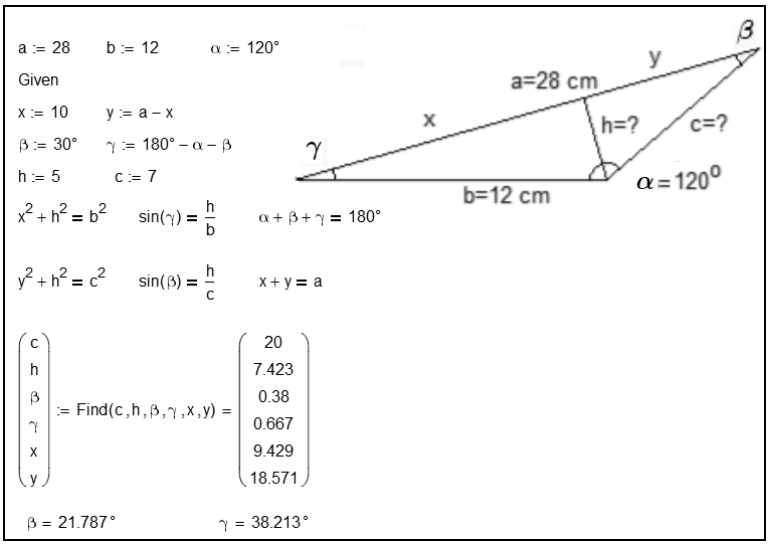


Рис. 5.17
Задача о треугольнике — решение системы уравнений в среде Mathcad 15

В маткадовском блоке `Given-Find` (`Solve` (Решить) — в Mathcad Prime) нужно задать начальное приближение к решению, записать ограничения (систему уравнений) и вызвать функцию `Find` (Найти), которая вернет значения неизвестных, превращающих уравнения в тождества.

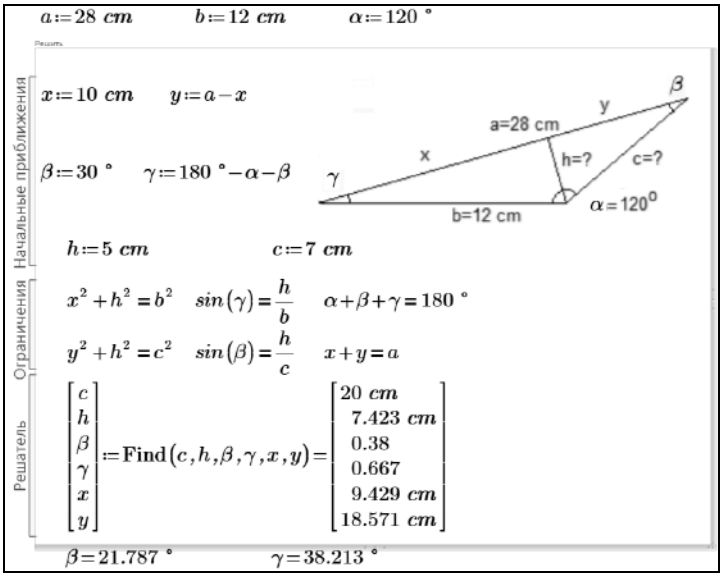


Рис. 5.18

Задача о треугольнике — решение системы уравнений в среде Mathcad Prime (b)

Когда я показал это решение внучке, она сказала, что так задачи они в школе не решают и что тут нужно применить теорему косинусов, которую они в школе изучают уже чуть ли не всю четверть.

Я вспомнил, что была такая теорема, но как она выглядит — забыл напрочь. Внучка мне подсказала, что это такое. Я дополнительно справился в Интернете об этой теореме и переписал решение задачи — см. рис. 5.19.

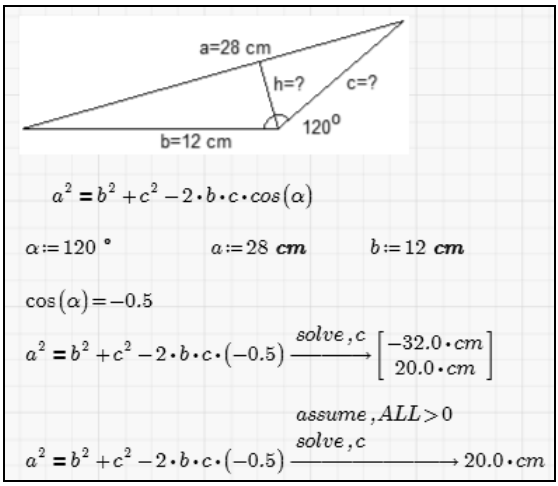


Рис. 5.19

Задача о треугольнике — решение квадратного уравнения

Я понял, что внучку и ее одноклассников учат решать квадратные уравнения (и мы это уже отмечали в этюде 4 — см. рис. 4.11 и 4.12 на стр. 70 и 71), но не разрешают при этом использовать компьютер (см. выше) или Интернет (а там есть интерактивные решатели для таких задач) для решения систем уравнений — линейных и нелинейных, алгебраических и дифференциальных. А ведь многие школьные и вузовские задачи по

математике, физике, химии и другим дисциплинам сводятся к решению систем уравнений. Школьнику или студенту достаточно понять суть задачи — ее «математику, физику, химию», составить систему уравнений, решить ее на компьютере и сделать проверку решения. Но нет! Школьников и студентов заставляют заучивать кучу правил и теорем, которые являются не чем иным, как готовыми решениями этих уравнений и систем. Нашу систему шести уравнений подстановками можно свести к одному квадратному уравнению, описывающему теорему косинусов, но можно этого и не делать, поручив эту работу компьютеру.

Задача на рис. 5.17 и 5.18 решена полностью — найдены длина третьей стороны треугольника и одна из его высот. Решение же, показанное на рис. 5.19, неполное — нужно будет еще искать высоту треугольника. В решении, показанном на рис. 5.17 и 5.18, тоже задействована теорема — теорема Пифагора, которую, в отличие от теоремы косинусов (теоремы Пифагора «с хвостиком»), знают все. Здесь нужно еще вспомнить, что такое синус — и все! Задача (рис. 5.17 и 5.18) решена, а ответ даже избыточен. Подход к решению, отображенный на рис. 5.17 и 5.18, существенно более универсален, чем подход, показанный на рис. 5.19. Любой треугольник или даже многогранник можно разбить на отдельные прямоугольные треугольники, составить несколько систем уравнений и решить их на компью-тере.

Да, многие школьные задачи с многоугольниками можно свести к решению систем уравнений. Многие, но далеко не все. Вот типичная такая задача. Треугольник 1–2–3 (см. рис. 5.20) разделен на две части отрезком 4–5, параллельным стороне треугольника 1–3, равной 10 см. Отрезок 4–5 делит треугольник 1–2–3 на две равные по площади части. Определить длину отрезка 4–5.

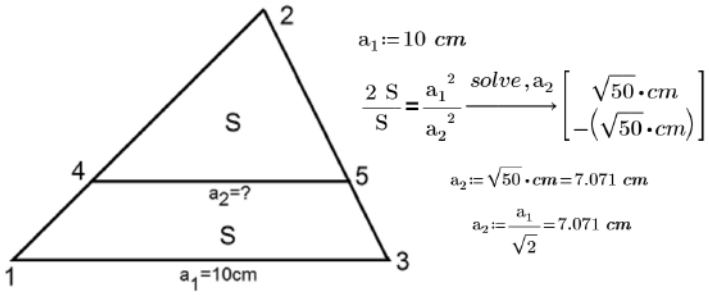


Рис. 5.20
Решение задачи о двух подобных треугольниках

Эту задачу уже не получится полностью переложить на плечи компьютера. Тут нужно будет самому сообразить, что треугольники 1–2–3 и 4–2–5 подобны, а площади подобных треугольников соотносятся друг к другу как квадраты их сторон. После такого уточнения решение задачи можно поручить компьютеру (см. оператор `solve` на рис. 5.20) или закончить самому, сообразив, что в соотношении сторон подобных треугольников, первый из которых больше по площади второго в два раза, заложен корень из двух.

А вот еще одна задача, с которой человек, знающий геометрию, справится намного лучше, чем компьютер. Внутри произвольного треугольника взята произвольная точка. Через эту точку проведены прямые, параллельные сторонам этого треугольника. Части этих прямых, а также сторон треугольника образуют внутри исходного треугольника три новых треугольника, площади которых S_1, S_2 и S_3 известны. Требуется найти площадь S исходного треугольника. Ответ довольно прост: $S=\sqrt{S_1+S_2+S_3}$. Аналитическое решение так же, как и в предыдущей задаче, основано на свойстве подобных треугольников, и еще на свойствах параллелограмма. А для получения численного решения потребуется преодолеть содержащуюся в условии неопределенность.

В принципе, все планиметрические задачи, приводимые в школьных задачниках, можно разделить на две группы: задачи, которые поддаются алгебраизации — сразу сводятся к решению систем уравнений, и задачи, перед решением которых с помощью компьютера нужно немного подумать самому и вспомнить некоторые теоремы. Возможно задачи первого типа целесообразно изъять из задачников (оставив лишь минимум шаблонных типовых), а второго типа — оставить, развить и усложнить так, чтобы школьник, решая их с помощью компьютера или без оного, мог мобилизовать и свои умственные способности.

ДИВЕРТИСМЕНТ 3. И ЕЩЕ ОБ УГАДЫВАНИИ МЕЛОДИИ

Первый автор книги когда-то давно предлагал озвучивать... начала и концы всех функциональных программных блоков по системе европейского звукоряда:

- функциональный блок — до;
- цикл «пока» — ре;
- альтернатива — ми;
- цикл «до» — фа;
- цикл с параметром — соль;
- множественное ветвление — ля;
- вызов процедуры — си.

При этом длительностью звука фиксируется глубина вхождения той или иной структуры. Прогонка программы на компьютере со звукогенератором и колонками рождает мелодию, прослушивая которую, можно проследить за ходом решения поставленной задачи, зафиксировать моменты зацикливания и пр. Можно также вести новую игру «Угадай программу» — проигрывается мелодия, созданная по вышеописанной схеме, а играющие должны угадывать, что это: сортировка массива, решение системы уравнений, численное интегрирование и т. д. — см. **http://twt.mpei.ac.ru/ochkov/128**.

Задания читателям

1. Составить программу и/или найти встроенную функцию, высчитывающую количество точек разного цвета в графическом файле.

2. Составить программу и/или найти встроенную функцию создания негативного или иного альтернативного изображения графического файла.

3. Подсчитать частоту встречи точек того или иного цвета в графическом файле и построить по этим данным гистограмму (см. рисунки 14.19 и 14.21 на стр. 323 и 324).

4. Давайте ради интереса и удовольствия заменим точки в матрице изображения шоколадными конфетами в ячейках кондитерских упаковок. Решите вот такую задачу. На кондитерской фабрике работает автомат по укладке шоколадных конфет. Он представляет собой механический манипулятор, способный перемещаться вдоль одной прямой перпендикулярно транспортеру, по которому движутся пустые коробки с ячейками для конфет. Конфета моментально попадает в ячейку, как только манипулятор окажется над ней. Рассмотрим движение автомата и коробок на плоскости `XOY`. Координаты ячеек `(x, y)` — это натуральные числа, причем в исходном положении $8 \le x \le 26$, $2 \le y \le 15$ (все значения координат заданы в дюймах). Лента транспортера начинает двигаться в направлении, противоположном оси `OX`, со скоростью `v = 1` дюйм/с. Одновременно из начала координат вдоль оси `OY` с постоянной скоростью без остановок начинает двигаться манипулятор. Какое максимальное количество конфет сможет уложить манипулятор за время однократного пересечения транспортера и с какой скоростью он должен двигаться?

5. Решите задачу 4 при условии, что манипулятор должен уложить конфеты в «подарочную коробку» непрямоугольной формы, т. е. количество конфет в каждом ряду подчиняется некоторой зависимости. Как в этом случае определить функцию скорости манипулятора?

Этюд 6

ЧТО ПОЧЕМ, ИЛИ СТАРАЯ ЗАДАЧА НА НОВЫЙ ЛАД

В этюде будет рассказано, как можно использовать компьютер для решения математических задач из художественной литературы.

Математика: метод конечного перебора, кубическое уравнение, мнимая единица, комплексное число, система линейных алгебраических уравнений (СЛАУ), уравнение в целых числах, факториал, объем цилиндра, объем конуса, недоопределенная система уравнений.

Физика: старинные и современные системы физических величин и единиц их измерения, площадь поверхности, объем, водоизмещение.

Информатика: компьютерные средства аналитических преобразований, компьютерные инструменты работы с физическими величинами, цикл с параметром, вложенный цикл, условный оператор, переменная диапазона, цикл с предпроверкой, графическое решение системы уравнений.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127824**

Была и есть очень интересная книга Сергея Александровича Рачинского **(http://ru.wikipedia.org/wiki/Рачинский,_Сергей_Александрович)** «1001 задача для умственного счета». В печатном виде она издавалась в XIX веке. В наше время она возродилась в виде сайта **www.1001task.ru** с удобными средствами поиска и комментирования задач. На этом же сайте есть ссылка на очень интересную дискуссию, которая развернулась в журнале «Наука и жизнь» по проблеме устного счета в современном образовательном процессе [46], —

http://www.nkj.ru/ archive/articles/6347.

Владельцы «яблочных» устройств iPhone и iPad могут бесплатно скачать задачник Рачинского из сетевого магазина AppStore и поместить его в библиотеку iBooks.

На сайте **1001task.ru** особо подчеркивается, что все задачи книги нужно решать сугубо в уме без привлечения каких-либо счетных устройств и приспособлений: бумаги и ручки, калькулятора и пр. С бумажным вариантом книги это требование выполнить несложно — ушел с задачником в глухой лес и решай себе в уме на здоровье. Но с электронными версиями книги это требование выполнить весьма сложно. Ведь второе, устаревшее название компьютера — это электронно-вычислительная машина (ЭВМ). Да и само слово компьютер (computer) означает вычислитель... На компьютере (даже если это не традиционный «вычислитель, а смартфон или «таблетка») есть под рукой счетные устройства. Это не только тривиальный калькулятор, приложенный к операционной системе, но и программы, специально предназначенные для решения сложных математических задач: Mathematica, Maple, Matlab, Mathcad, SMath и др. Вот примеры решения некоторых задач из книги Рачинского, выполненные в среде Mathcad, — см. рис. 6.1–6.5.

Грешно, конечно, смотреть свысока на автора задачника и его читателей. Вы, мол, ломали и ломаете головы, придумывая задачи[1] и решая их в уме, а мы разделываемся с ними буквально в два счета — раз-два и ответ готов...

Что можно сказать по этому поводу?! Зачем мы в наш компьютерный век снова обращаемся к старым задачам?!

Во-первых, они не просто «старые», а «старые добрые задачи».

[1] Интересно, составлял ли Рачинский свой задачник «сугубо в уме» или он все-таки пользовался ручкой, бумагой или счетами — компьютером XIX в.

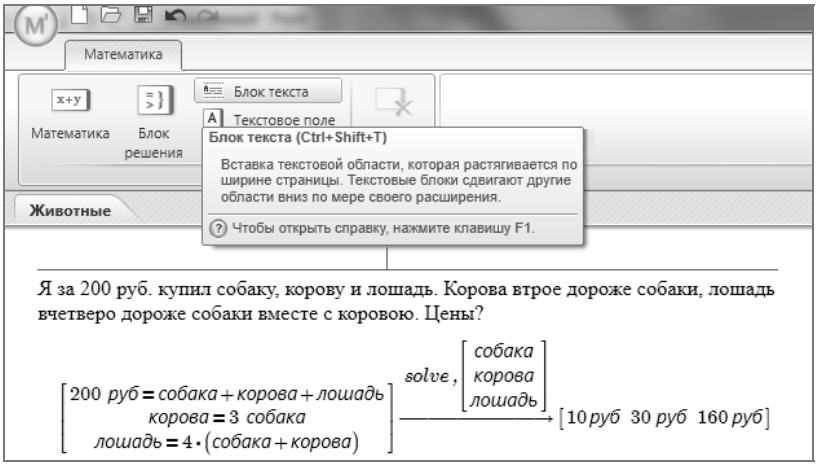


Рис. 6.1
Задача о собаке, корове и лошади

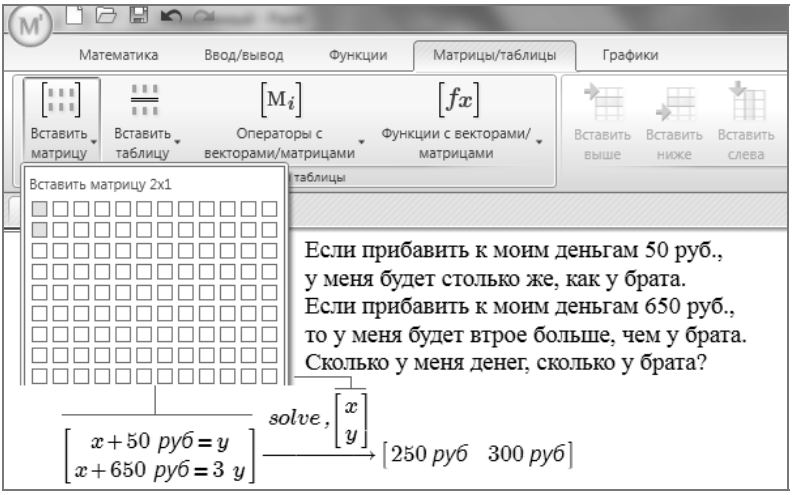


Рис. 6.2
Задача о деньгах двух братьев

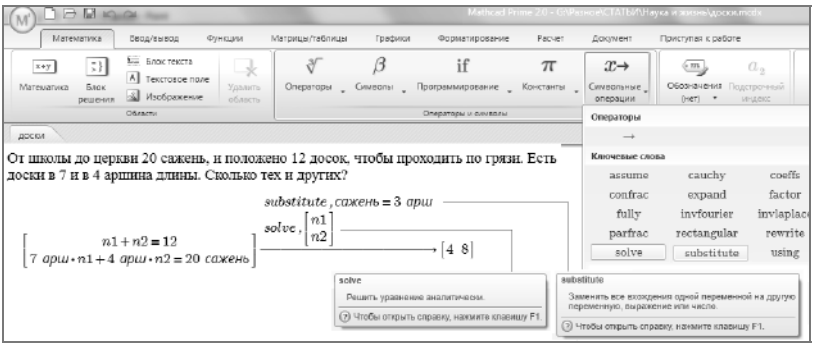


Рис. 6.3
Задача о дороге в церковь

1000 вершков — сколько саженей, аршин и вершков?

$$\frac{1000\ \textit{вершок}}{\textit{сажень}} \xrightarrow[\textit{substitute},\ \textit{арш} = 16\ \textit{вершок}]{\textit{substitute},\ \textit{сажень} = 3\ \textit{арш}} \frac{125}{6} = 20.833$$

$$\frac{1000\ \textit{вершок} - 20\ \textit{сажень}}{\textit{арш}} \xrightarrow[\textit{substitute},\ \textit{арш} = 16\ \textit{вершок}]{\textit{substitute},\ \textit{сажень} = 3\ \textit{арш}} \frac{5}{2} = 2.5$$

$$\frac{1000\ \textit{вершок} - 20\ \textit{сажень} - 2\ \textit{арш}}{\textit{вершок}} \xrightarrow[\textit{substitute},\ \textit{арш} = 16\ \textit{вершок}]{\textit{substitute},\ \textit{сажень} = 3\ \textit{арш}} 8$$

Ответ 1000 вершков = 20 саженей 2 аршина 8 вершков

Рис. 6.4
Разложение 1000 вершков на сажени, аршины и вершки

У медника было 8 кусков меди, весом каждый в 1 фунт 8 лотов. Из этой меди он сделал чайники, весом каждый 1 фунт 21 лот 1 золотник. Сколько чайников?

$$8 \cdot (\textit{фунт} + 8\ \textit{лот}) = n \cdot (\textit{фунт} + 21\ \textit{лот} + \textit{золотник}) \xrightarrow[\textit{substitute},\ \textit{лот} = 3\ \textit{золотник}\ \ \textit{solve},\ n]{\textit{substitute},\ \textit{фунт} = 32\ \textit{лот}} 6$$

Рис. 6.5
Задача о медных чайниках

Во-вторых, устный счет — это прекрасная гимнастика для ума[2]. И это мы отмечали во многих этюдах книги. И использование тут компьютера или калькулятора равносильно дооборудованию спортивного тренажера (современный гимнастический снаряд для тела[3])... гидроусилителями.

Решение задач Рачинского, в основном, сводится к поиску корней алгебраических уравнений и их систем (см. этюд 4). В среде Mathcad для решения таких задач предусмотрен оператор ***solve***. В некоторых задачах его нужно дополнить оператором ***substitute***, заменяющим одну переменную на другую или на выражение[4]. С помощью этого оператора мы подсказали компьютеру, что в сажени три аршина (рис. 6.3 и 6.4), в фунте 32 лота, а в лоте 3 золотника (рис. 6.5[5]), т. е. мы научили компьютер обращаться со старыми русскими мерами веса и длины, которые были в ходу в момент написания задачника. Задача на рис. 6.4 непосредственно учит нас работать со старыми недесятичными единицами длины[6]. Такая задача, кстати говоря, актуальна и в наши дни. Кто может сходу подсчитать в уме, сколько лет,

[2] В наше время *программирование* — лучшая гимнастика для ума. Но далеко не всем доступны эти «умственные физические упражнения». Некоторые форумы (форум математических пакетов **http://forum.exponenta.ru**, например) при входе требует сложить четыре числа. Многие, сидящие за компьютером, не могут это сделать в уме и прибегают к помощи калькулятора.

[3] Автор в Интернете наткнулся на такую фотографию. Пустая лестница. Справа и слева от нее эскалаторы, поднимающие и опускающие довольно тучных людей. Вверху, в конце лестницы здание с вывеской «Fitness club» — см. **http://www.goodexperience.com/broken/i/04/02/america-fitness-s.jpg**.

[4] На рис. 6.1–6.3 дополнительно показаны инструменты решения задач и оформления решения: операторы символьной математики Mathcad (рис. 6.3), команда меню вставки комментария (рис. 6.1) и команда меню вставки вектора с двумя элементами (матрица 2 на 1), куда записывается система уравнений и неизвестные уравнения (рис. 6.2).

[5] Современные единицы измерения встроены в Mathcad, но старых русских единиц в нем нет.

[6] Переход от недесятичной системы исчислений (в версте 500 саженей, в сажени 3 аршина, в аршине 16 вершков) к десятичной (в километре 1000 метров, в метре 100 см, в сантиметре 10 миллиметров и т. д. до набившим всем оскомину нанометров) — это первый удар по устному счету. Калькулятор с компьютером только добивают его...

месяцев, недель, дней, часов, минут и секунд в миллиарде секунд[7]?! На рис. 6.6 показано решение этой задачи в среде Mathcad[8].

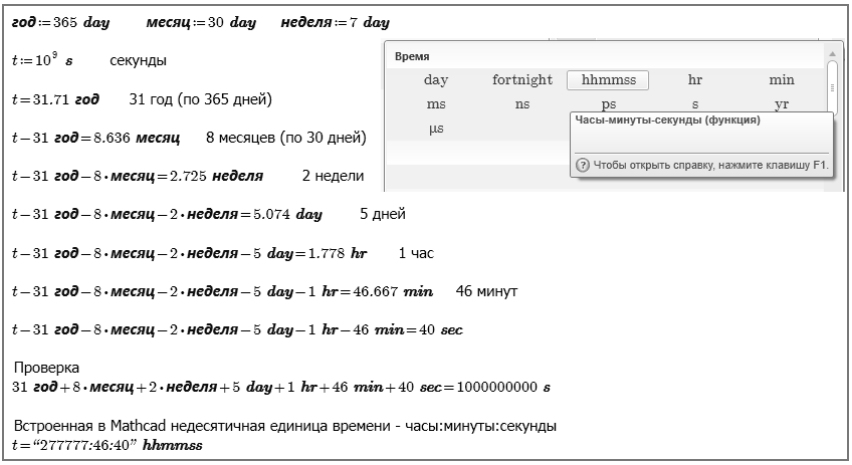


Рис. 6.6
Разложение миллиарда секунд

Так какой же «новый лад» можно «приладить» к старым арифметическим задачам!?

Во-первых, тексты задач из книги Рачинского заканчиваются численными ответами с единицами измерений или без оных[9]. Наши компьютерные решения задач (см. рис. 6.1–6.5) можно рассматривать как... ответы на задачи. Более того, в таких ответах записано и современное компьютерное решение задач. Задачник с такими ответами превращается в решебник, т. е. задачник с подробным описанием решений.

Во-вторых, авторы современных задачников по математике, предназначенных для устного счета и/или для решения с привлечением бумаги с ручкой и калькулятора, могут использовать не только «бумагу с ручкой и калькулятор», но и вышеупомянутые математические программы для составления задач и, главное, для проверки правильности и единственности решения — см. сноску 1 данного этюда. Учитель тут должен идти впереди учеников. У нас же часто получается так, что школьники прекрасно знают компьютер, а учитель даже боится подойти к нему. При этом данная «компьютеробоязнь» маскируется заявлениями, что «компьютер в процессе обучения приносит только вред» — мешает, например, освоению приемов устного счета.

В-третьих, задачи Рачинского и другие подобные можно модернизировать так, чтобы их без компьютера решить было бы довольно сложно. Пример на рис. 6.7.

[7] «Десятичность» не до конца охватила систему измерений. Кроме времени, тут можно упомянуть и угловые единицы с градусами, минутами и секундами...

[8] Сравнивая рисунок 6.4 с рисунком 6.6, где решаются одинаковые по сути задачи, можно отметить, что в первом случае задействована *символьная математика* пакета Mathcad, а во втором — *численная*.

[9] В современных задачниках ответы, как правило, помещают в конце книги.

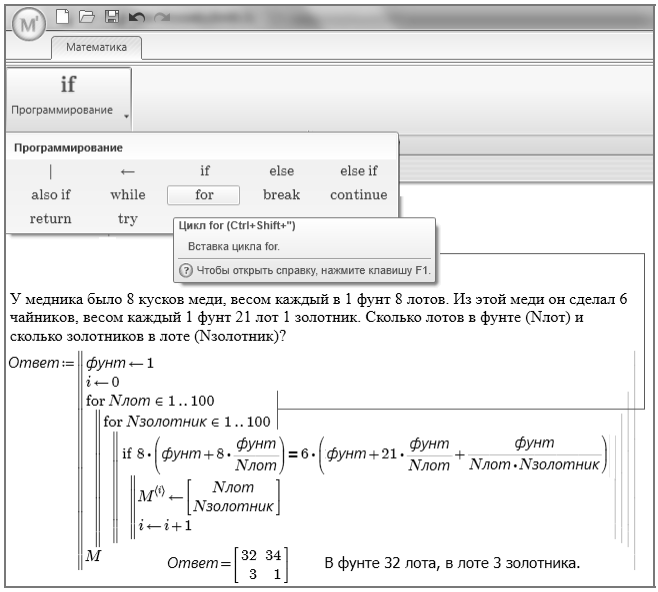


Рис. 6.7

Определение старых русских единиц массы (веса)

И гимназист, и неграмотный русский крестьянин XIX в. прекрасно знали, сколько фунтов в пуде, вершков в аршине, а аршин в сажени. Современный школьник за такой информацией полезет в справочник или Интернет. А если нет справочника под рукой, а Сеть временно недоступна[10]?! Не беда! В задачу о меднике (рис. 6.5) с ответом (6 чайников) «вшито» соотношение фунта, лота и золотника. Решая новую задачу (рис. 6.7), школьнику придется изощряться не в устном счете, а в умении составлять программу для компьютера. На рисунке 6.7 показано одно из возможных решений задачи, основанное на переборе всех возможных вариантов (см. про перебор и в этюде 17): в фунте может быть от 1 до 100 лотов, а в лоте — 1 до 100 золотников[11]. Кстати говоря, почти все задачи Рачинского подобраны так, что у них целочисленные ответы. При решении таких задач можно успешно использовать метод перебора, расцветший в эпоху компьютеров с их невероятным быстродействием [47]. Чему равен кубический корень из 1728? Перебрал числа в разумном интервале (от 1 до 100, например) и запомнил то число, куб которого равен 1728 (см. рис. 6.9). В уме этого, конечно, делать не будешь, а на компьютере — пожалуйста[12]. Пусть читатель попробует перебором вариантов решить задачу на рис. 6.3 без указания того, что от школы до церкви проложено именно 12 досок: «От школы до церкви 20 сажень, и положены доски, чтобы проходить по грязи[13]. Есть доски в 7 и в 4 аршина длины. Сколько тех и других? [Ответ: 4 и 8 или 8 и 1]» — см. рис. 6.8. А если короткая доска будет длиной в один аршин, то

[10] Многие не могут спокойно работать на компьютере с отключенным Интернетом, даже в том случае, когда Интернет не нужен и, более того, вреден — отвлекает от выполнения срочной работы. Возникает чувство, что тебя на время заперли в комнате... Выходить никуда не надо, но все равно неприятно...

[11] Да, именно так. Перебор нужно начинать не с двух, а с единицы. Ведь в задаче может быть подвох — используются одинаковые единицы измерения с разным названием: «Я заплатил за товар стольник. Сколько я заплатил рублей и целковых?»

[12] Кубический корень из 1728 — это не одно число, а три: двенадцать и еще два комплексных числа. В этом можно убедиться, если решить в Mathcad уравнение $x^3 = 1728$. Перебором тут можно получить только один ответ. Такой же неполный ответ дают и калькуляторы с компьютерами — см. рис. 6.9.

[13] Эта российская проблема не решена до сих пор. Вспомним о двух наших бедах — «о дураках и дорогах!» Авторы надеются, что та книга как-то исправит первую беду.

вариантов укладки досок будет уже восемь! Хотя правильный ответ будет 8 и одна доска — укладывают сначала длинные доски, а потом, когда очередная длинная не умещается, кладем на дороге короткие (короткую)... Тут уже к математике примешиваются другие научные дисциплины — экономика, например. Цена погонного метра длинной доски (руб./м) обычно меньше, чем у короткой: второй вариант ответа в задаче о досках (8 досок по 7 аршин и одна доска в один аршин) будет дешевле ещё и из-за того, что девять досок укладывать проще, чем двенадцать...

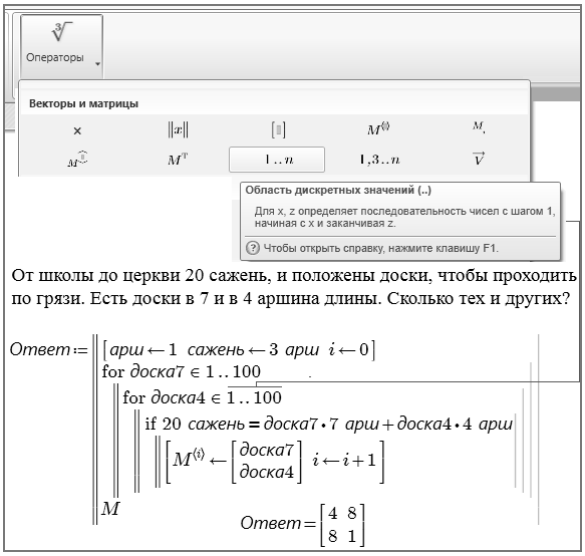


Рис. 6.8
Задача о дороге, вымощенной досками

Задачник Рачинского интересен и в том плане, что по нему можно узнать, как жили люди в позапрошлом веке, чем они занимались, чем торговали и какие были цены... В художественной литературе (а задачник Рачинского, помимо прочего, отличается и хорошим литературным стилем) часто можно встретить математические выкладки, касающиеся, в основном, финансовых проблем [48]. Вот три из них.

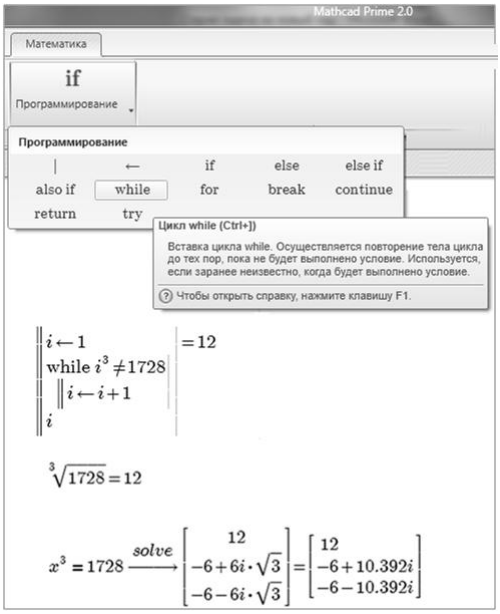


Рис. 6.9
Поиск кубического корня

Задача 1. Рассказ А. П. Чехова «Репетитор» начинается с такого диалога:

— Теперь по арифметике... Берите доску. Какая следующая задача?

Петя плюет на доску и стирает рукавом. Учитель берет задачник и диктует:
— «Купец купил 138 арш. черного и синего сукна за 540 руб. (аршин — это русский ярд — примечание авторов). *Спрашивается, сколько аршин купил он того и другого, если синее стоило 5 руб. за аршин, а черное 3 руб.?» Повторите задачу.*
Петя повторяет задачу и тотчас же, ни слова не говоря, начинает делить 540 на 138.
— Для чего же вы это делите? Постойте! Впрочем, так... продолжайте. Остаток получается? Здесь не может быть остатка. Дайте-ка я разделю!
Зиберов делит, получает 3 с остатком и быстро стирает.
«Странно... — думает он, ероша волосы и краснея. — Как же она решается? Гм!.. Это задача на неопределенные уравнения, а вовсе не арифметическая...»
Учитель глядит в ответы и видит 75 и 63.
«Гм!.. странно... Сложить 5 и 3, а потом делить 540 на 8? Так, что ли? Нет, не то».
— Решайте же! — говорит он Пете.
— Ну, чего думаешь? Задача-то ведь пустяковая! — говорит Удодов Пете. — Экий ты дурак, братец! Решите уж вы ему, Егор Алексеич.
Егор Алексеич берет в руки грифель и начинает решать. Он заикается, краснеет, бледнеет.
— Это задача, собственно говоря, алгебраическая, — говорит он. — Ее с иксом и игреком решить можно. Впрочем, можно и так решить. Я, вот, разделил... понимаете? Теперь, вот, надо вычесть... понимаете? Или, вот что... Решите мне эту задачу сами к завтраму... Подумайте...
Петя ехидно улыбается. Удодов тоже улыбается. Оба они понимают замешательство учителя. Ученик VII класса еще пуще конфузится, встает и начинает ходить из угла в угол.
— И без алгебры решить можно, — говорит Удодов, протягивая руку к счетам и вздыхая. — Вот извольте видеть...
Он щелкает на счетах, и у него получается 75 и 63, что и нужно было.
— Вот-с... по-нашему, по-неученому.

Такую систему двух линейных алгебраических линейных уравнений (СЛАУ) с двумя неизвестными (а именно она просматривается в приведенном диалоге: `x + y = 138` и `5x + 3y = 540`) можно решить без компьютера и даже без калькулятора, выполнив несложные арифметические действия на бумаге или просто в уме. На счетах (как отец Пети делал это) это сейчас, конечно, мало кто сделает. Мы эту задачу решим на компьютере (на современных «счетах»), воспользовавшись программой Mathcad — см. рис. 6.10.

$$\begin{bmatrix} x+y=138\ арш \\ 5\ \frac{руб}{арш}\cdot x+3\ \frac{руб}{арш}\cdot y=540\ руб \end{bmatrix} \xrightarrow{solve,\begin{bmatrix} x \\ y \end{bmatrix}} [63\,арш\ \ 75\ арш]$$

Рис. 6.10
Решение задачи о купце и сукне в среде Mathcad Prime

На рис. 6.10 показано, как в среде Mathcad решается задача о купце и сукне с помощью оператора ***solve***, предназначенного для поиска корней одиночных алгебраических уравнений и их систем, линейных и нелинейных: «Вот-с... по-нашему, *по-компьютерному*...»!

Мы можем найти на русской странице Википедии, посвященной описанию русских счёт с костяшками, такой текст, связанный с этим рассказом Чехова.

Само «неучёное» решение Чеховым в рассказе не приводится, но оно легко может быть реконструировано, поскольку задача имеет стандартное арифметическое решение, опирающееся на логику и состоящее в выполнении шести арифметических действий.

«Предположим, что всё купленное сукно было синее. Тогда партия в 138 аршин стоила бы $5 \cdot 138 = 690$ рублей. Но это на $690 - 540 = 150$ рублей больше того, что было заплачено в действительности. «Перерасход» в 150 рублей указывает на то, что в партии имелось более дешевое, чёрное, сукно — по 3 рубля за аршин. Этого сукна столько, что из двухрублёвой разницы ($5 - 3 = 2$ рубля) получается 150 «лишних» рублей. То есть $150 / 2 = 75$ аршин чёрного сукна. Отсюда $138 - 75 = 63$ аршин сукна синего».

«Щёлканье на счётах», выполненное Удодовым, выглядело следующим образом.

1. Прежде всего, Удодов-старший «набирает» число 138: одна косточка на первой проволоке, три — на второй, восемь — на третьей.

2. Затем он должен умножить 138 на 5. Для упрощения счёта вместо этого он сначала умножает 138 на 10, не делая никаких манипуляций, просто мысленно перенося все косточки одним рядом выше, после чего делит на 2: на каждой проволоке, начиная снизу, откидывает половину косточек. На третьей проволоке, где отложено восемь косточек, откидывает четыре; на средней проволоке из трех косточек откидывает две, при этом одну из них мысленно заменяет десятью нижними и делит их пополам, то есть добавляет пять косточек к тем, что находятся на следующей проволоке; на верхней проволоке убирает одну косточку, прибавляя пять к косточкам на второй проволоке. В результате на верхней проволоке косточек нет, на второй осталось шесть, на третьей — девять. Итого — 690.

3. Далее Удодову-старшему нужно из 690 вычесть 540: со второй проволоки убирается пять косточек, с третьей — четыре. Остается 150.

4. Теперь 150 нужно поделить пополам (см. выше) — получается 75.

5. Осталось из 138 вычесть 75. Повторно «набирается» 138, на второй проволоке нужно отбросить семь, но там только три. Не хватает четырёх, поэтому Удодов-старший оставляет на проволоке шесть косточек (если ему лень вычитать в уме четыре из десяти, он может перекинуть весь десяток на второй проволоке влево и отбросить от него «недовычтенные» четыре косточки), а с первой проволоки сни-

мает одну. Осталось на третьей проволоке из восьми косточек отбросить пять. Получается 63.

Мама первого автора этой книги могла высчитывать на счетах... квадратный корень. Кубический, правда, уже не могла — см. рис. 6.9.

В задаче о купце и сукне число уравнений *равно* числу неизвестных. Но в жизни и, соответственно, в художественной литературе так бывает далеко не всегда. Кроме того, сами задачи могут быть сформулированы не совсем четко, и «размыты» по всему тексту — см. вторую и третью задачу.

Задача 2. В романе другого русского писателя Ф. М. Достоевского «Игрок» можно найти[14] семь цитат, в которых зашифрованы курсы европейских валют во времена написания романа (вторая половина XIX в.):

Цитата 1

— Вы их немедленно получите, — ответил генерал, покраснев немного, порылся у себя в бюро, справился в книжке, и оказалось, что за ним моих денег около ***ста двадцати рублей****.*

— Как же мы сосчитаемся, — заговорил он, — надо переводить на талеры. Да вот возьмите ***сто талеров****, круглым счетом, — остальное, конечно, не пропадет.*

и далее

Вам следует дополучить с меня эти четыре ***фридрихсдора*** *и* ***три флорина*** *на здешний расчет.*

Цитата 2

Полина просто рассердилась, когда я передал ей всего только ***семьсот гульденов****.*

и далее

Слушайте и запомните: возьмите эти ***семьсот флоринов*** *и ступайте играть, выиграйте мне на рулетке сколько можете больше; мне деньги во что бы ни стало теперь нужны.*

Цитата 3

Я начал с того, что вынул ***пять фридрихсдоров****, то есть* ***пятьдесят гульденов****, и поставил их на четку.*

Цитата 4

— Да-с, вот взяла да и выиграла двенадцать тысяч флоринов! Какое двенадцать, а золото-то? С золотом почти что ***тринадцать*** *выйдет. Это сколько по-нашему? Тысяч шесть, что ли, будет?*

Я доложил, что и за ***семь*** *перевалило, а по теперешнему курсу, пожалуй, и до* ***восьми*** *дойдет.*

Цитата 5

— Oui, madame, — вежливо подтвердил крупер, — равно как всякая единичная ставка не должна превышать разом ***четырех тысяч флоринов****, по уставу, — прибавил он в пояснение.*

Я поставил самую большую позволенную ставку, в ***четыре тысячи гульденов****, и проиграл.*

Цитата 6

Ей проходилось получить ровно ***четыреста двадцать фридрихсдоров****, то есть* ***четыре тысячи флоринов*** *и* ***двадцать фридрихсдоров****.*

Цитата 7

— Полина, вот ***двадцать пять тысяч флоринов*** *— это* ***пятьдесят тысяч франков****, даже больше.*

Если в рассказе А. П. Чехова уравнения были прописаны в явном виде, то в «Игроке» Ф. М. Достоевского они зашифрованы в цитатах. Составив и решив соответствующую систему уравнений, можно узнать курсы валют (талер, фридрихсдор, флорин, гульден и франк) по отношению к рублю:

120 рублей = 100 талеров + 4 фридрихсдора + 3 флорина (цитата 1 — см. выше);

700 гульденов = 700 флоринов (цитата 2);

[14] А в эру Интернета и электронных библиотек это делать стало намного проще: задал ключевой слово поиска и считай, сколько раз оно встречается в тексте.

5 фридрихсдоров = 50 гульденов (цитата 3);
13 000 флоринов = 8 000 рублей (цитата 4);
4 000 флоринов = 4 000 гульденов (цитата 5);
420 фридрихсдора = 4 000 флоринов + 20 фридрихсдоров (цитата 6);
25 000 флоринов = 50 000 франков (цитата 7).

Если число уравнений (у нас их семь) превышает число неизвестных (у нас их пять), то такая система уравнений называется *переопределенной*.

Задачу «Игрока» также можно решить без компьютера, последовательно высчитывая курсы отдельных валют (гульден равен флорину, фридрихсдор равен десяти гульденам и т. д.), и определить их стоимость по отношению к рублю. Но мы, как и в случае с первой задачей, попросим это сделать компьютер. Пусть он сам подставляет и упрощает, что считает нужным — см. рис. 6.11.

$$\begin{bmatrix} 120\ руб = 100\ талер + 4\ фридрихсдор + 3\ флорин \\ 700\ гульден = 700\ флорин \\ 5\ фридрихсдор = 50\ гульден \\ 13000\ флорин = 8000\ руб \\ 4000\ флорин = 4000\ гульден \\ 420\ фридрихсдор = 4000\ флорин + 20\ фридрихсдор \\ 25000\ флорин = 50000\ франк \end{bmatrix} \xrightarrow{solve,\ \begin{bmatrix} талер \\ фридрихсдор \\ флорин \\ гульден \\ франк \end{bmatrix},\ float,\ 2} \begin{bmatrix} 0.94 \cdot руб & 6.2 \cdot руб & 0.62 \cdot руб & 0.62 \cdot руб & 0.31 \cdot руб \end{bmatrix}$$

Рис. 6.11
Решение задачи о курсе валют в среде Mathcad Prime

На рис. 6.11 показано решение в среде Mathcad задачи о курсах валют с использованием оператора ***solve***. Из решения, показанного на рис. 6.11, видно, что во второй половине XIX в. талер стоил 94 копейки, фридрихсдор 6 рублей 20 копеек, флорин и гульден по 62 копейки, а франк 31 копейку. При таком раскладе генерал заплатил учителю (Алексею Ивановичу) 117 руб. 73 коп. («около 120 руб.» — см. цитату 1 выше), а «бабуленька» выиграла на рулетке 7930 руб. («до восьми тысяч дойдет» — см. цитату 4).

Примечание. То, что флорин, гульден равны по стоимости, а фридрихсдор в десять раз дороже флорина или гульдена и франк в два раза дешевле флорина или гульдена, ясно и без компьютера. Но мы специально ввели в компьютер все семь цитат, чтобы получилась переопределенная система уравнений.

Задача 3. Хороший образец денежных расчетов можно найти в «Мещанине во дворянстве» Мольера (**http://lib.ru/MOLIER/meshanin.txt**).

Дорант. Я намерен с вами расквитаться. Давайте вместе подсчитаем, сколько я вам всего должен. <...> Вы хорошо помните, сколько вы мне ссудили?
Г-н Журден. По-моему, да. Я записал для памяти. Вот она, эта самая запись. В первый раз выдано вам двести луидоров.
Дорант. Верно.
Г-н Журден. Еще выдано вам сто двадцать.
Дорант. Так.
Г-н Журден. Еще выдано вам сто сорок.
Дорант. Вы правы.
Г-н Журден. Все вместе составляет четыреста шестьдесят луидоров, или пять тысяч шестьдесят ливров.
Дорант. Подсчет вполне верен. Пять тысяч шестьдесят ливров.
Г-н Журден. Тысячу восемьсот тридцать два ливра — вашему поставщику перьев для шляп.
Дорант. Совершенно точно.
Г-н Журден. Две тысячи семьсот восемьдесят ливров — вашему портному.
Дорант. Правильно.
Г-н Журден. Четыре тысячи триста семьдесят девять ливров двенадцать су восемь денье — вашему лавочнику.

Дорант. Отлично. Двенадцать су восемь денье — подсчет верен.
Г-н Журден. И еще тысячу семьсот сорок восемь ливров семь су четыре денье — вашему седельнику.
Дорант. Все это соответствует истине. Сколько же всего?
Г-н Журден. Итого пятнадцать тысяч восемьсот ливров.
Дорант. Итог верен. Пятнадцать тысяч восемьсот ливров. Дайте мне еще двести пистолей и прибавьте их к общей сумме: получится ровно восемнадцать тысяч франков, каковые я вам возвращу в самое ближайшее время.

В этом отрывке фигурируют такие единицы стоимости: луидор, ливр, су, денье, пистоль и франк. Соотношение между луидором и ливром лежит на «поверхности» текста: 460 луидоров = 5060 ливров, следовательно, луидор = 11 ливров. Поиск остальных соотношений требует уже некоторых «умственных напряжений» и средств Mathcad.

Легко сообразить, что 8 денье и 4 денье дают в сумме один су, т. е. 12 денье. Аналогичные выкладки, связанные с тем, что итоговая сумма — круглое число (сумма без французских «копеек»), можно приложить и к су (12 су + 8 су = ...), но стоп! Давайте перенесем решение в среду Mathcad (рис. 6.12).

$$\left[\begin{array}{l}\left(\begin{array}{l}200\cdot louis\\ +\,120\ louis\\ +\,140\ louis\\ +\,1832 livres\ ...\\ +\,2780 livres\ ...\\ +\,4379\ livres + 12\ sols + 8\ deniers\ ...\\ +\,1748\ livres + 7\ sols + 4\ deniers\end{array}\right) = 15800\cdot livres\\ (200 + 120 + 140)\cdot louis = 5060\cdot livres\\ 20\ sols = livre\end{array}\right] \begin{array}{l}\text{solve}, \begin{pmatrix} louis \\ livres \\ sols \end{pmatrix}\\ \rightarrow (2640 deniers \quad 240 deniers \quad 12 deniers)\end{array}$$

$$sol = 12\ deniers \qquad livres = 20\ sols \qquad louis = 11\ livres$$

Рис. 6.12
Первая задача Мольера

Из решения (рис. 6.12) видно, что су = 12 денье, ливр = 20 су (240 денье), а луидор = 11 ливрам (12 су + 8 су = 1 ливр — повторяем, эту задачу можно было решить и без компьютера.

Труднее обстоит дело с пистолями и франками. Эта задача до конца не определена.

$$\begin{pmatrix} 15800\ livres + 200\ pistoles = 18000\ francs \\ pistoles = 11\ livres \end{pmatrix} \begin{array}{l}\text{solve}, \begin{pmatrix} livres \\ pistoles \end{pmatrix}\\ \longrightarrow (francs \quad 11 francs)\end{array}$$

$$livres = francs \qquad pistoles = louis$$

Рис. 6.13
Вторая задача Мольера

На рис 6.13 найдено решение этой задачи в предположении, что ливр равнялся франку. Здесь речь может идти о двух разных названиях одинаковых денежных единиц. Лет через сто историки будут высчитывать соотношение между долларом и «баксом», забыв, что это одно и то же.

Если же число уравнений алгебраической системы *меньше* числа неизвестных, то такая система называется *недоопределенной*. Репетитор Зиберов из рассказа А. П. Чехова так ошибочно пытался определить задачу о купце и сукне. Но она была четко определена — в ней два алгебраических уравнения с двумя неизвестными; сама же система имеет одно единственное решение. Недоопределенную за-

дачу можно «увидеть» в описании подводной лодки «Наутилус» — «героине» романов Жуль Верна «Двадцать тысяч лье под водой» и «Таинственный остров». Вот что можно узнать из разговора капитана Немо с профессором Аронаксом о размерах подводной лодки: «Вот, господин Аронакс, чертежи судна, на котором вы находитесь. Судно представляет собой сильно удлиненный цилиндр с коническими концами. <...> Площадь его равняется одной тысяче одиннадцати и сорока пяти сотым квадратных метров, объем равен одной тысяче пятистам и двум десятым кубических метров; короче говоря, корабль, полностью погруженный в воду, вытесняет тысячу пятьсот и две десятых кубических метров, или тонн, воды». Примечание. Тут капитан Немо, вернее, сам автор романа — Жюль Верн немного переборщил, указав параметры подводной лодки с излишней точностью. Достаточно было бы сообщить, что площадь судна равняется 1010 м^2 или даже просто 1000 м^2, а объем (водоизмещение в подводном состоянии) — 1500 м^3. Лишние цифры после запятой можно наверно объяснить тем, что Жюль Верн был адвокатом, а не инженером по образованию. Для справки: водоизмещение в подводном состоянии современных атомных подводных лодок может достигать 50 000 тонн. И еще для справки, но уже не технической, а филологической. В нашей речи правило о том, что перестановка мест сомножителей не меняет произведение, не всегда работает. Одно дело сказать, что «тут двадцать метров», а другое — «тут метров двадцать». Первое означает, что это более-менее точно двадцать метров, а второе — приблизительно двадцать метров. Кстати, это правило не всегда действует и в математике, вернее — в численной математике.

$$n := 150$$

$$\prod_{i=1}^{n} i = 5.71338395644585 \cdot 10^{262}$$

$$\prod_{i=n}^{1} i = 5.71338395644586 \cdot 10^{262}$$

$$n! = 5.71338395644586 \cdot 10^{262}$$

$$n! \rightarrow 571338395644585459047 8^{\cdots}$$

Рис. 6.14
Перестановка местами сомножителей

Но вернемся к подводной лодке.

Если принять допущение, что под площадью подводной лодки понимается не ее «жилая» площадь (площадь ее помещений), а площадь поверхности и что центральная часть подводной лодки — это круглый прямой цилиндр, а ее нос и корма — это круглые прямые конусы одинаковой высоты (см. чертеж на рис. 6.15), то задача сводится к решению двух нелинейных алгебраических уравнений с тремя неизвестными. Две предыдущие задачи сводились к решению систем *линейных* алгебраических уравнений.

На рис. 6.15 показано общее аналитическое (символьное) и графическое решение задачи о размерах подводной лодки «Наутилус» в среде Mathcad с помощью оператора ***solve***. Но этот оператор в задаче о «Наутилусе» не возвращает конкретные числовые значения, как это было при решении предыдущих задач, а формирует четыре функции с аргументом `R`. Недоопределенные системы алгебраических уравнений могут иметь множество решений. В нашем случае размеры субмарины (длина носа и кормы `H` и длина центральной части подводной лодки `L`) при известном ее объеме и площади поверхности зависят от радиуса сечения цилиндрической части `R`. Система двух нелинейных уравнений, показанных на рис. 6.15, име-

ет два решения. Поэтому оператор ***solve*** вернул нам не вектор, а матрицу, состоящую из двух строк (длины `H` и `L`) и двух столбцов (два решения задачи) и хранящую выражения (на рис. 6.15 они из-за экономии места не показаны полностью) для функций с именами `H1`, `H2`, `L1` и `L2`. На рис. 6.15 эти функции отображены графически в области действительных положительных чисел (решений) для величин `H` и `L`. Аналитический вид этих функций на рис. 6.15 показан не полностью не только из-за экономии места, но скорее потому, что он мало что дает для анализа зависимостей. Иное дело графики. На рис. 6.16 эти графики показаны в области реальных размеров подводной лодки.

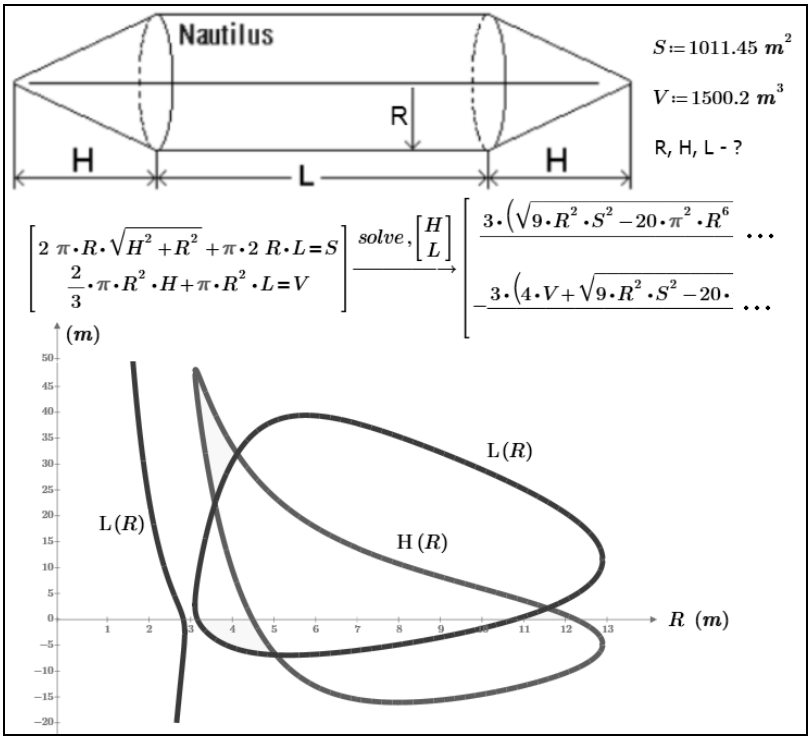


Рис. 6.15

Полное решение задачи о размерах подводной лодки «Наутилус» в среде Mathcad

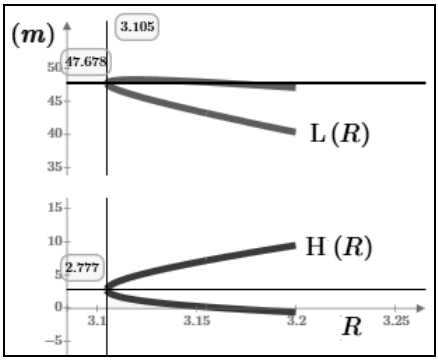


Рис. 6.16

Решение задачи о размерах подводной лодки «Наутилус» в среде Mathcad

Если допустить, что радиус R сечения цилиндрической части субмарины равен 3.105 м (см. вертикальный маркер на рис. 6.16), то длина ее конических частей (носа и кормы) H будет равна 2.777 м, а длина центральной цилиндрической части L — 47.698 м. Это, кстати, соответствует изображениям «Наутилуса» из иллюстраций книги Жюль Верна. Кстати, при заданных параметрах субмарины она может быть в форме цилиндра (точка пересечения кривой H с осью абсцисс при R ≈ 3.2 м — см. рис. 6.15) или составленная из двух конусов без центрального цилиндра (точка пересечения кривой L2 с осью абсцисс при R ≈ 4,5 м).

Авторы предлагают своим студентам (и читателям этой книги) продолжить решение задачи о размерах подводной лодки «Наутилус» (заодно и прочитать или перечитать романы о ней и ее капитане).

Задания читателям

1. Рассчитать минимальное значение R, при котором задача имеет решение (см. левые оконечности кривых не рис. 6.15 и 6.16).

2. Рассчитать значения R, при которых лодка превращается в цилиндр без конусов (H = 0) или в два конуса, составленных основаниями друг к другу (L = 0).

3. Рассчитать размеры лодки, если ее нос будет выполнен в виде полусферы, а корма в виде конуса.

4. Рассчитать размеры подводной лодки, при которых ее объем будет равен 1011,45 м^3, а площадь поверхности будет минимальной.

5. Рассчитать размеры подводной лодки, при которых ее площадь поверхности будет равна 1500.2 м^2, а объем будет максимальным.

6. Семиклассник Петя сделал модель корабля и стал испытывать её в цилиндрической бочке. Затем подошла Катя, младшая сестра Пети, посадила на корабль в качестве «пассажира» своего резинового ёжика и стала играть. Петя заметил, что при плавании корабля с ёжиком уровень воды в бочке выше на 1 см того уровня воды, который был в бочке изначально (без корабля и без ёжика). В результате неосторожности при игре корабль перевернулся и пошёл ко дну, при этом ёжик остался на плаву. Петя заметил, что уровень воды в бочке при этом понизился. Попробуйте рассчитать величину понижения уровня, если масса корабля в $n = 3/2$ раза больше массы ёжика, а плотность материала модели корабля в $k = 2$ больше плотности воды. Подсказка: понижение уровня, выраженное в миллиметрах, — целое число.

7. Приняв, что плотность воды равна 1 г/см^3, а подводной лодки с геометрией, показанной на рис. 135, — 0.8 г/см^3, рассчитать, насколько лодка будет подниматься над водой. Подходы к решению данной задачи даны в [85]. Затем эта подводная лодка заполняет свои специальные отсеки водой, увеличивая тем самым свою плотность до 1.2 г/см^3, и начинает опускаться под воду. Оценить динамику такого погружения, используя подход, отображенный на рис. 9.13 на стр. 212 (ныряние в воду) и 9.14 на стр. 213 (пуск ракеты из подводного положения).

Этюд 7

ЦЕПНАЯ ФУНКЦИЯ, ИЛИ ПЯТЫЙ ЭЛЕМЕНТ

В этюде будет рассказано, как на компьютере можно рассчитывать параметры подвесной канатной дороги и о многом другом интересном, связанным с цепной функцией. Кроме того, будет рассказано об электрических цепях, прожекторе и линзе.

Математика: экспонента, гиперболический синус и косинус, цепная функция, производная, парабола, катеноида, длина кривой, длина дуги окружности, разложение в ряд Тейлора, разложение в ряд Фурье, система линейных алгебраических уравнений (СЛАУ), ранг матрицы, теорема Кронекера — Капелли, предел, синусоида, основание натурального логарифма.

Физика: арка максимальной прочности, трение каната о кнехт, подвесной канатный мост, подвесная канатная дорога, баланс сил и моментов сил, центр масс, квадратное колесо, кинематика велосипеда, электрическая цепь, закон Ома, электродвижущая сила, закон отражения света, закон Снелла.

Информатика: численное и аналитическое решение алгебраического и дифференциального уравнения и их систем, плоская и трехмерная компьютерная графика, программистское уравнение, компьютерная анимация, программирование, цикл с параметром, отладка компьютерных расчетов.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127823**

Как уже отмечалось ранее, в настоящее время в образовательном процессе намечается тенденция одновременного чтения курса *информатики* и *математики*. В рамках этой учебной дисциплины (ее, повторяем, условное название Физико-математическая информатика, ИнфоМатика или МатИнформатика[1]) можно также затрагивать вопросы и учебных курсов *Физика* и/или *Введение в специальность*. Тем более есть прецеденты: в тридцатые годы прошлого столетия, например, выдающийся ученый Норберт Винер читал в университете годовой объединенный курс лекций по математике и электротех-нике.

На начальном (переходном) этапе можно так сразу кардинально не менять содержание лекций по математике и информатике, объединяя их в одну[2], а поступить по-иному. Допустимо на первой паре лекционных занятий прочесть студентам классическую математическую лекцию по анализу функции одного аргумента, например, а затем на второй паре — лекцию в рамках занятий по информатике о том, как Интернет и современные математические программы могут на примере решения конкретной инженерной задачи работать с производными, интегралами, рядами и прочими математическими понятиями, о которых было рассказано на предыдущей лекции. В такую связку можно вклинить и лекцию по физике (разделы: механика, электричество, оптика и пр.). Получится очень интересная образовательная междисциплинарная технология в рамках дисциплины с условным названием **физико-математическая информатика** (STEAM в западной терминологии — Science, Technology, Education, Art, Mathematic): рассматриваются теоре-

[1] Анекдот советских времен. В одном институте объединили кафедры кибернетики и математики. Получилась кафедра с несколько неприличным названием, которое можно найти на сайте книги.

[2] Когда авторы предложили одному своему коллеге-математику читать лекции в современной аудитории с компьютером, Интернетом, экраном, мультимедийным проектором и иллюстрировать лекционный материал «живыми» решениями задач в средах каких-либо математических программ, то он полушутя-полусерьезно ответил, что можно еще привлечь и «пританцовку», какая сопровождает певцов на эстраде. Доказал теорему — и тут на сцену, пардон, в аудиторию с визгом вбегают полуголые девицы и лихо отплясывают канкан... Можно лекцию читать также и под фонограмму, а под потолком аудитории устроить лазерное шоу — показывать, например, поверхность решения дифференциального уравнения в частных производных или настоящую 3D-анимацию...

тические аспекты некоторого процесса (математика), далее ставится эксперимент и дается его толкование (физика), а в завершении создается, реализуется (аналитически и/или численно) и анимируется математическая модель процесса.

Вот конкретное содержание подобной лекции или серии лекций, захватывающей математику, физику (теоретическую механику, теоретические основы электротехники) и информатику.

Студентам-электрикам на лекциях рассказывают такую «страшную» и поучительную историю. Зимой перекидывали через одну сибирскую реку воздушную линию электропередачи (ЛЭП). Все было вроде бы правильно рассчитано, спроектировано и сделано, и высоковольтные провода не должны были мешать судоходству[3]. Но весной, когда с реки сошел лед и началась навигация, один теплоход все-таки задел своей металлической мачтой провод под напряжением. А ведь опасно было просто проплывать вблизи высоковольтных проводов без касания — мог произойти электрический разряд! В чем причина такой инженерной ошибки? Весной потеплело, провисающие над рекой провода стали длиннее вследствие термического расширения металла и зазор между нижней точкой провода и рекой стал меньше того, какой был во время зимней прокладки воздушной линии электропередачи[4]. Кроме того, уровень воды в реке оказался аномально высоким. Что явилось причиной аварии — термическое расширение проводов (их еще и электрический ток немного нагрел), высокий уровень воды в реке и/или что-то еще? Ответить на этот вопрос нельзя без расчетов, приведенных ниже.

Если взять абсолютно гибкий провод (цепь) и подвесить его (ее) на двух опорах, то форму кривой, которую сформирует цепь, будет задавать функция, которую так и называют — *цепная функция*. Цепь можно порвать, но ее нельзя поломать изгибом. На звенья провисающей цепи действуют только силы растяжения и там нет изгибающих сил, какие имеют место, например, в жесткой балке. Если провисающей цепи сказать «замри» и перевернуть ее «вверх тормашками», то получится *арка*, которая также обладает подобным уникальным свойством — на элементы такой арки действуют только силы сжатия и там нет изгибающих сил. Это свойство арки априорно и интуитивно пытались использовать еще в древности: значение сопротивления сжатию многих строительных материалов намного выше значения сопротивления изгиба: кирпич, например, можно разломить на две половинки и руками[5], но практически невозможно раздавить вручную.

В монографиях [49] и справочниках, а также в Интернете[6] можно найти формулы для цепной функции, полученные в результате решения дифференциального уравнения баланса сил, действующих на элементарный элемент (пардон за тавтологию — на звено) цепи. В этих формулах присутствует самая, наверное, «стой-

---

[3] Инженеру, повторяем, приходится решать не просто инженерные задачи, а задачи с элементами экономики. В случае с ЛЭП можно было для страховки увеличить высоту двух прибрежных мачт или потуже натянуть провода, но это бы привело к удорожанию строительства и эксплуатации линии электропередачи.

[4] Есть такая задача — насколько рельсы Транссиба укорачиваются зимой. Тут можно задать две температуры, уточнить коэффициент линейного термического расширения рельсовой стали и решить задачу простым умножением. А можно смоделировать характер температурного поля рельсов зимой и летом и решить задачу через программирование (метод конечных элементов) или интегрирование. Кстати, на авторском сайте **http://twtmas.mpei.ac.ru/mas/Worksheets/HBPipePP/Pipe_Tab_5_1.mcd** можно найти значения удельных тепловых удлинений для некоторых сталей. Когда-то давно в старой России в качестве примера расширения тел при нагреве приводили тот факт, что «после бани сапоги трудно на ноги натягивать». Пример был явно неудачный, но закон расширения тел при нагреве он помогал запомнить.

[5] Или разбить о голову — забава многих действующих и бывших, трезвых и не совсем трезвых десантников.

[6] Сейчас, повторяем, при возникновении вопроса рука все чаще и чаще тянется за справкой не к полке с книгами, а к мышке компьютера.

кая» элементарная функция *экспонента*[7] — e^x, главное свойство (стойкость!) которой состоит в том, что ее можно сколь угодно раз дифференцировать (брать производную по x), но с неё «как с гуся вода» — она остается сама собой и не изменяет свой вид. Парабола же, похожая на графике на цепную линию (мы об этом еще упомянем ниже), превращается в нуль после всего лишь троекратного взятия производной: $ax^2 + bx + c \to 2ax + b \to 2a \to 0$.

Формулу цепной функции можно найти не только в книгах или в Интернете, но и на... архитектурных объектах, эксплуатирующих красоту и уникальное прочностное свойство перевернутой цепи, — см. на рисунке 7.1 знаменитую арку (Gateway Arch), возведенную в американском городе Сент-Луисе на западном берегу реки Миссисипи[8].

Рис. 7.1

*Арка в Сент-Луисе в виде перевернутой цепи (**https://en.wikipedia.org/wiki/Gateway_Arch**)*

У основания арки написана ее формула в футах:

$$y = 757.7' - 127.7' \cdot \mathrm{ch}(x / 127.7').$$

В формуле цепной функции фигурирует так называемый *гиперболический косинус* ch, состоящий их двух экспонент: $\mathrm{ch}(x) = (e^x + e^{-x})/2$. В нотации пакета Mathcad он обозначен как cosh. Если в этой формуле плюс заменить на минус, то мы получим *гиперболический синус* (sinh или sh — см. рис. 7.9 ниже), присутствующий в формуле производной цепной функции. Во второй же производной снова появится гиперболический косинус. Эта особенность (cos-sin-cos-sin...) и другое роднит обычные тригонометрические функции с их гиперболическими тезками.

На рисунке 7.2 показан расчет размеров арки города Сент-Луис в более привычных для нас метрах с опорой на вышеприведенную формулу в футах.

---

[7] В России сайт, поддерживающий математические пакеты, включая и Mathcad, имеет имя **exponenta.ru**. Там, кстати, можно найти строгий математический вывод уравнения цепной линии — см. **http://www.exponenta.ru/educat/class/test/hyperb/10.asp**.

[8] Если смотреть на эту арку с другого, восточного берега реки, то на водной поверхности можно видеть... провисающую цепь — отражение арки. Сам же город, кстати, находится в штате... Миссури. Это сделано, наверно, для того, чтобы школьникам география медом не казалась.

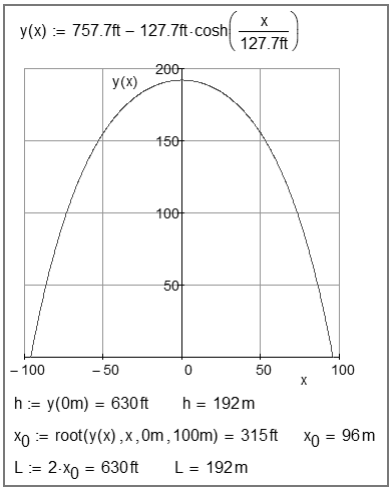


Рис. 7.2
Расчет геометрии арки города Сент-Луиса в среде Mathcad

На рисунке 7.2 не только построен график функции, прописанной на арке Сент-Луиса, но и определена ее высота (значение функции `y(x)` при `x = 0`) и «шаг» арки — расстояние `L` между ее опорами на поверхности земли. Арка вписывается в квадрат со стороной 630 футов[9] (210 ярдов или 192 метра: ярдов точно, а метров — приблизительно).

Традиционная (каноническая[10]) форма записи цепной функции, приводящаяся в всех справочниках, такая `y = a · ch(x / a)`. Она лаконична, но не совсем удобна в практических расчетах, которые будут описаны ниже. Поэтому мы будем работать с другой формулой цепной функции, имеющей один аргумент `x` и три параметра, задающих не только ее форму, но и ее положение относительно выбранной системы координат. Мы уже не удержались и в этюде 4 поработали с этой формулой — см. рис. 4.49 на стр. 101.

На авторском сайте по адресу **http://twt.mpei.ac.ru/MCS/Worksheets/chain.xmcd** размещен сетевой интерактивный открытый расчет провисания цепи, отображенный на рис. 7.3. При создании этого расчета была использована цепная функция вида $y = h + a \cdot ch((x - x_0)/a) - 1)$, в которой x_0 — это значение аргумента, при котором цепная функция имеет либо минимум (провисающая цепь — см. рис. 7.3), либо максимум (арка[11] — см. рис. 7.1 и 7.2); `h` — это значение цепной функции при x_0 (минимальное или максимальное ее значение — высота арки, например). Параметр же a характеризует вид цепной линии: при $a < 0$ — это провисающая цепь,

[9] 630 футов — это некое магическое число. Есть проект возведения в Нью-Йорке колеса обозрения с таким диаметром. Многие небоскрёбы в Америке имеют именно такую высоту. Попытки авторов разгадать с помощью Интернета загадку 630 футов не увенчались успехом. Может быть, читателям это удастся.

[10] Каноническая в том смысле, что выбрана такая система координат, при которой в формуле остается всего лишь один параметр `a`. Остальные два параметра при этом становятся равными нулю. В формуле арки города Сент-Луиса оставлены два параметра, прописанные в виде констант: `757.7` и `127.7` футов. В этюде 9 мы попытаемся гиперболу привести к каноническому виду: обсудим технологию трансформации плоской кривой второго порядка, имеющей пять параметров, к каноническому виду, имеющему два параметра. Каноническая запись параболы также имеет один параметр `p`: $y = 2p \cdot x^2$. Квадратное же уравнение имеет три параметра.

[11] На арке город Сент-Луис лучше было бы прописать нашу формулу, из которой четко была бы видна высота арки и ее ширина (полуширина) у основания. У левого основания балки можно было бы двумя стрелками (горизонтальной и вертикальной) зафиксировать начало координат. Сейчас же формула, написанная на арке, подразумевает, что начало координат находится на земле под точкой максимальной высоты арки (см. рис. 7.2).

при a > 0 — арка. Цепь будет стремиться к прямой лини, но никогда не станет ею, если цепь сильно натянуть с двух концов: вспомним гитарную струну. А еще можно вспомнить... кирпичную кладку, горизонтальные межкирпичные линии которой (зазоры с застывшим раствором) это строго говоря не отрезки прямых, а отрезки... цепной линии. При кладке кирпичной стены каменщики натягивают строго горизонтально бечеву (веревочку) для контроля ровности и горизонтальности укладки кирпичей. Но эта бечева все-таки слегка провисает. Натягивается же эта веревочка перекидыванием ее либо через блок, либо через гладкий цилиндрический стержень с подвешиванием груза. Этот горизонтальный конец веревочки служит заодно и для контроля вертикальности кладки кирпичной стены. Тут, кстати говоря, «всплывает» еще одна экспонента — сила трения каната о цилиндр, на который он накинут или намотан, пропорциональна экспоненте от угла намотки. Эту закономерность использовали еще в древности, пришвартовывая корабли к пристани. Формулу же с экспонентой вывел великий Эйлер.

Сайт, показанный на рис. 7.3, требует от посетителя ввести значения высот подвеса цепи на двух опорах (мачтах ЛЭП — см. начало этого этюда со «страшной электротехнической историей»): высоты h_1 и h_2, значение расстояния между опорами по горизонтали L и значение длины цепи S. По этим данным определяется геометрия цепи, в частности, значение h минимального зазора между цепью и нулевой отметкой (землей или водой в реке). Дополнительно рассчитываются значения углов α_1 и α_2, при которых наша цепи будет тянуть опоры. Эти углы нам пригодятся при решении задачи о цепи с подвешенным грузом (см. ниже). Задача сводится к решению системы трех уравнений с тремя неизвестными — параметрами x_0, h и a, формирующими цепную функцию. Первое уравнение — это абсцисса точки крепления цепи у левой опоры, второе уравнение — абсцисса точки крепления цепи у правой опоры и третье уравнение — это формула длины кривой[12].

В расчете, показанном на рисунке 7.3, был использован блок Given-Find (см. этюд 4). При этом не показано задание первого предположения — оно запрятано в скрытой области. Решение проверяется прикладыванием к кривой линии на графике, прорисованной в правильном масштабе, реальной металлической цепи. На рисунке 7.4 показана такая проверка — распечатка расчета, отображенного на рис. 7.3, вставка распечатки в деревянную рамку, к которой прибита контрольная цепь. Такое «произведение изобразительного искусства» висит в кабинете одного из авторов и привлекает внимание многих математиков и нематематиков. Этот сайт можно использовать для обсчета «страшной истории», приведенной в начале этюда. Для этого необходимо ввести в расчет «зимнее» и «летнее» значения длины провода и подсчитать «зимний» и «летний» зазоры над рекой. Но необходимо также учитывать и линейное удлинение провода за счет собственного веса, которое в свою очередь также зависит от температуры провода. Так что тут не так все просто.

Второе авторское «произведение изобразительного искусства», также связанное с цепной линией, показано на рис. 7.6. Этот цепной (висячий) мост был нарисован так [50]. Когда появилась версия Mathcad со встроенной функцией odesolve (см. этюд 9), то авторы сразу попытались протестировать ее на такой простой краевой задаче: даны четыре точки крепления цепи моста (0–0 — земля, 2–5 — первый пилон, 9–5 — второй пилон и 11–0 — земля на другом берегу реки), необходимо определить форму цепи, отвечающей такому ее свойству: вторая произ-

[12] Это уравнение и его вывод также несложно найти в справочниках или в Интернете. В подынтегральном выражении ясно просматривается теорема Пифагора. В случае с цепной линией его можно упростить, избавившись от определенного интеграла на отрезке от 0 до L. Но мы не будем этого делать, оставляя «первородный» вид выражения. Так будет лучше в образовательных целях, сообразуясь с идеями, высказанными в первом этюде книги.

водная равна самой функции (вторая производная гиперболического косинуса равна гиперболическому косинусу).

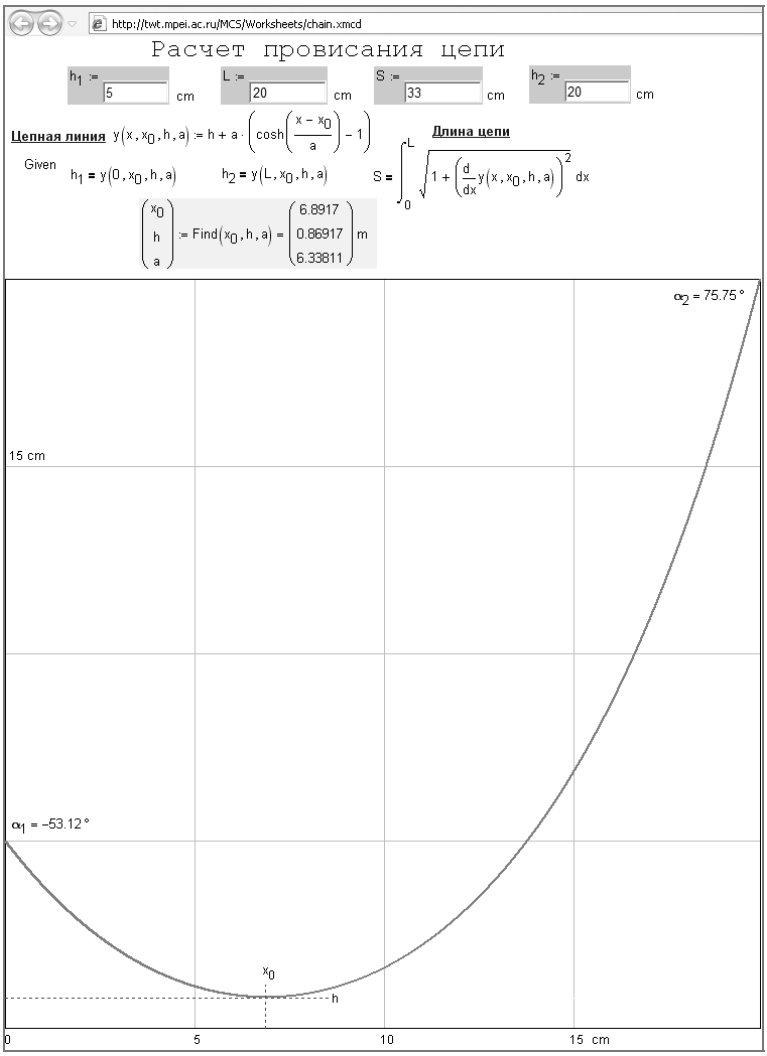


Рис. 7.3

Сайт для расчета провисания цепи

Решение этой задачи (a — численное и b — аналитическое) показано на рис. 7.5 с графическим отображением на плоскости, а на рис. 7.6 в объеме. На рисунке 7.6 прорисованы «две цепи» с четырьмя точками крепления у каждой с добавлением тумана[13], перспективы, полотна дороги с разделительной линией, узлов крепления вертикальных канатов и... парящего орла. Сами вертикальные канаты не нарисованы и вот почему. Цепь (канат) подвесного моста имеет форму не цепной функции, а уже упоминавшейся нами параболы. В этом случае нагрузки от полотна дороги на вертикальные канаты при их равномерном расположении вдоль моста будут одинаковыми.

[13] Туман, перспектива и освещение — это основные инструменты повышения объемности трехмерных объектов на плоском дисплее компьютера или на листе бумаги.

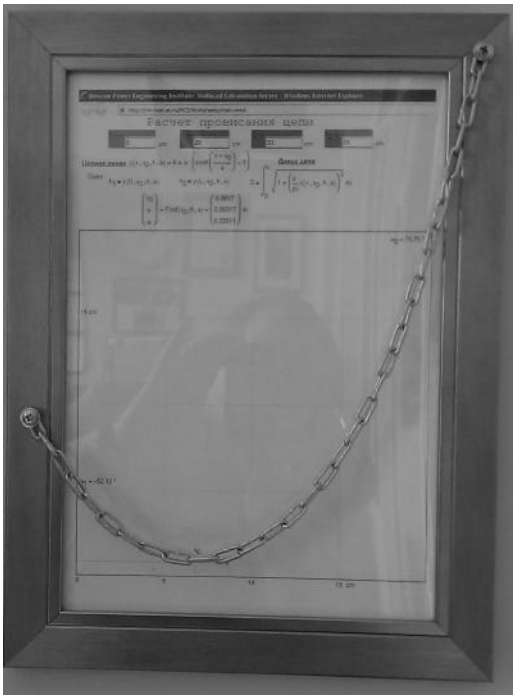


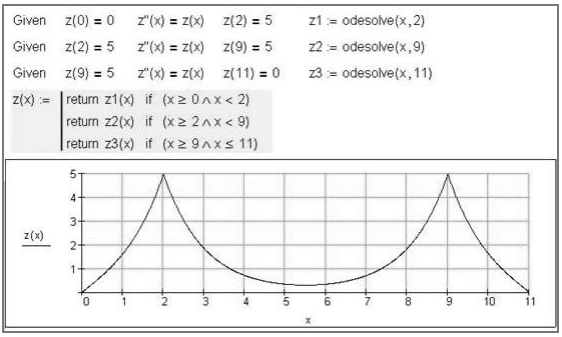
Рис. 7.4
Реальная и виртуальная цепи

Так что на рисунке 7.6 мы видим пока только строящийся мост, к которому на понтонах подвели полотно проезжей части и к которому еще только собираются крепить вертикальные канаты определенной длины так, чтобы цепь изменила свою форму с цепной функции на параболу[14]. Строго говоря, цепная функция и парабола — это две различные кривые, графики которых никогда не совпадут. Но, как известно, все бесконечно дифференцируемые функции (а к ним относится и цепная функция) могут быть представлены в виде бесконечной суммы степенных функций. Такое представление называется рядом Тейлора. Таким образом, если в разложении $a \cdot ch(x/a) = a \cdot [1 + (1/2!)\cdot(x/a)^2 + (1/4!)\cdot(x/a)^4 + (1/6!)\cdot(x/a)^6 + \ldots]$ учитывать только первые два слагаемых, то с некоторой погрешностью цепная линия прибли-

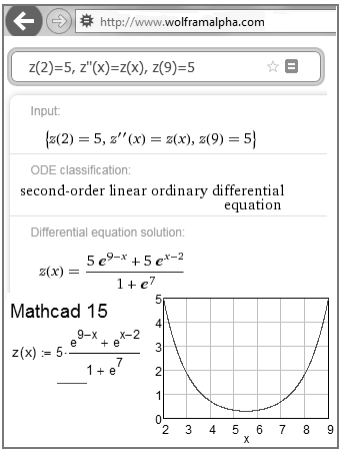
[14] *«...Другой способ начертить искомую параболу на призме состоит в следующем. Вобьём в стену два гвоздя на одинаковой высоте над горизонтом и на таком расстоянии друг от друга, чтобы оно равнялось двойной ширине прямоугольника, на котором желательно построить параболу; между одним и другим гвоздём подвесим тонкую цепочку, которая свешивалась бы вниз и была такой длины, чтобы самая низкая точка её находилась от уровня гвоздя на расстоянии, равном длине призмы. Цепочка эта, свисая, расположится в виде параболы, так что, отметив её след на стене пунктиром, мы получим полную параболу, рассекаемую пополам перпендикуляром, проведённым через середину линии, соединяющей оба гвоздя. Галилео Галилей. «Беседы и математические доказательства...» 1638 г.*

Однако мэтр ошибался. Между параболой и линией провисшей цепи будет небольшое различие. Лишь полвека спустя Иоганном Бернулли, Готфридом Лейбницем и Христианом Гюйгенсом было выведено уравнение цепной линии. В нём участвует параметр, изменяя который можно получать различные кривые провисания цепи. Возникновению самого названия «цепная линия» мы обязаны Гюйгенсу. (Читать полностью: **http://www.etudes.ru/ru/etudes/catenary.**) Кстати, если студентам на занятиях по математике, физике или информатике показать реальную провисающую цепь (авторы всегда делают это перед чтением данной лекции-этюда) и спросить их, какой формулой она описывается, то студенты, вторя Галилею, хором ответят, что это парабола. Еще один способ показать цепную линию — это натянуть мыльную пленку внутри проволочного кольца и разместить его горизонтально. Пленка провиснет и примет форму поверхности, образуемой вращением цепной линии вокруг вертикальной оси.

жается к параболе. Очевидно, что погрешность тем меньше, чем ближе точка x к нулю и чем больше по модулю параметр a. Указанные различия хорошо видны на рис. 7.30 (стр. 167). На рисунке 7.31 (стр. 168) показано, как пакет Mathcad работает с рядом Тейлора на примере цепной функции.



a



b

Рис. 7.5

Решение задачи о виртуальном висячем мосте:

a — численное решение в среде Mathcad 15 (три пролета); *b* — аналитическое решение на сайте Wolfram с построением графика в среде Mathcad 15 (центральный пролет).

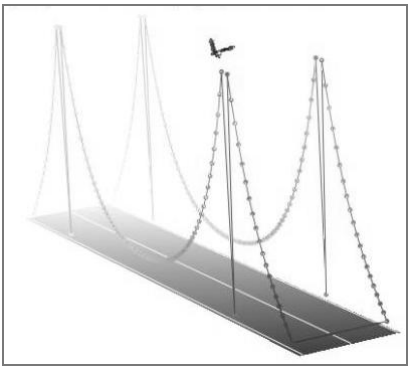


Рис. 7.6

Виртуальный висячий мост

Вот такие не только эстетические, но и «инженерно-математические» чувства возникают при любовании «произведениями изобразительного искусства», показанными на рис. 7.1, 7.4 и 7.6.

Провисающая цепь имеет еще одно интересное свойство — ее вид никак не зависит от массы цепи. Так что и тонкий шнур (бечева каменщика), и массивная цепь (ограждение памятника) будут одинаково провисать, если у них одинаковые длины и координаты точек крепления. В связи с этим на ум приходит еще один знаменитый физический закон с элементами парадоксальности — скорость падения тела в вакууме одинаковая и для перышка, и для кирпича (см. этюд 9, где это будет описано подробнее).

Задача о провисании цепи, которую мы только что описали, довольно тривиальна. А вот более сложная и более «свежая» задача. Она возникла во время прогулки одного из авторов по дачному участку, где висят не только пожарные ведра (см. этюд 12), но и электрические провода, натянутые между столбами. Так вот, на одном проводе был подвешен... кирпич для того, чтобы ветер сильно не раскачивал оголенный электрический провод, и он не задевал соседний. Мысли автора тут же переключились с кирпича на... канатную подвесную дорогу, и сразу возникла идея обсчитать все это и создать такую анимацию: на двух столбах высотой H_1 и H_2 подвешен абсолютно гибкий трос (канат, цепь) длиной S и удельной (линейной) массой g_c, вдоль которого скользит груз массой G. Как будет меняться форма этого каната и какие силы будут действовать на канат в точках опоры и в точке крепления груза? Мы в этюде 4 уже решали подобную упрощенную задачу: определяли длину цепи без груза, которая будет тянуть свои точки крепления, расположенные на одном уровне, с минимальной силой (см. рис. 4.49 и 4.51 на стр. 101 и 103).

Задачу о канатной подвесной дороге вывесил на форуме PTC Community один финский студент, дипломная работа которого была связана с ней. Это обычная современная практика — инженер, получив новое задание, сразу ищет аналогичные решения [51], учась не на своих, а на чужих ошибках. Студент же в такой ситуации ищет так называемую «рыбу» — уже выполненное кем-то учебное задание[15]. В эпоху Интернета это делать стало намного проще, прося о помощи аудиторию социальных и профессиональных сетей. В [51] сделана попытка не борьбы с такой формой списывания (это сизифов труд), а попытка перевести это явление в цивилизованное русло.

Ясно, что мы тут будем иметь дело уже с двумя цепными функциями[16], «подвешенными» между каждым из столбов и точкой подвеса груза. Если вес груза увеличивать, то эти кривые будут стремиться к прямым линиям и задача будет существенно упрощена. Если же вес груза уменьшать до нуля, то эти две цепные линии должны слиться в одну, которую мы обсчитали ранее (см. рис. 7.3 в этом этюде и рис. 4.49–4.51 в этюде 4). Задачу о подвесной канатной дороге с двумя отрезками цепной линии и грузом можно свести к решению системы девяти уравнений, четыре из которых учитывают геометрию, а пять — баланс сил и моментов сил (см. схему на рис. 7.7).

Вот эти уравнения.

Геометрия

1. Сумма длин двух ветвей (частей) цепи равна заданному значению S.

2. Левая ветвь цепи крепится у левого столба и заканчивается в точке подвеса груза.

---

[15] Текст-«рыба» (также текст-заполнитель или текст-манекен) — на жаргоне дизайнеров текст, вставляемый в макет и не несущий смысловой нагрузки (Википедия). В Сети можно найти и другие толкования этого термина. Так, композитор сочиняет музыку к песне, к которой еще нет текста, по «рыбе» — по тексту, составленному из названия рыб: «треска, селедка, окунь, сом...» Главное тут, чтобы ритм соблюдался. Кстати, в Киеве студенты на «рыбу» кажут «коза» — «У тебя есть коза курсовой?»

[16] Кому-то это ясно, а кому-то нет. На форуме **https://www.ptcusercommunity.com/thread/51002** возникла целая дискуссия на эту тему.

3. Правая ветвь цепи начинается в точке подвеса груза и крепится у правого столба.

4. Две ветви цепи пересекаются в точке подвеса груза.

Баланс сил

5. Сумма вертикальных составляющих реакции опор ($F_{y1} + F_{y2}$) равна сумме веса цепи и веса подвешенного груза. Сразу отметим, что две горизонтальные составляющие реакции цепи у опор равны ($F_{x1} = F_{x2} = F_x$), т. к. других горизонтальных сил в системе нет.

6 и 7. Если $F_{x1} = F_{x2} = F_x$ и известны значения производных двух цепных функции в точках опоры, то можно рассчитать значения F_{y1} и F_{y2}, принимая во внимание, что значение производной — это тангенс угла наклона касательной в данной точке (см. рис. 4.49 на стр. 101, где был задействован арктангенс для вычисления этого угла).

8. В точке пересечения ветвей цепи вес груза уравновешивается двумя вертикальными силами от левой и правой ветвей цепи. В этом уравнении тоже участвует значения производных, с помощью которых вертикальные составляющие сил выражаются через горизонтальные составляющие.

9. Равенство моментов сил, действующих на систему в точке пересечения ветвей цепи: три силы (F_{y1}, F_x у правой опоры и вес правой ветви цепи) разворачивают систему по часовой стрелке, а три другие силы (F_x у левой опоры, F_{y2} и вес левой ветви цепи) разворачивают систему против часовой стрелки. Тут сделано одно допущение, обоснованность которого мы еще обсудим в конце этюда: ординаты центров масс двух ветвей цепи находятся на серединах отрезков $0-x_0$ (левая ветвь цепи) и x_0-L (правая ветвь).

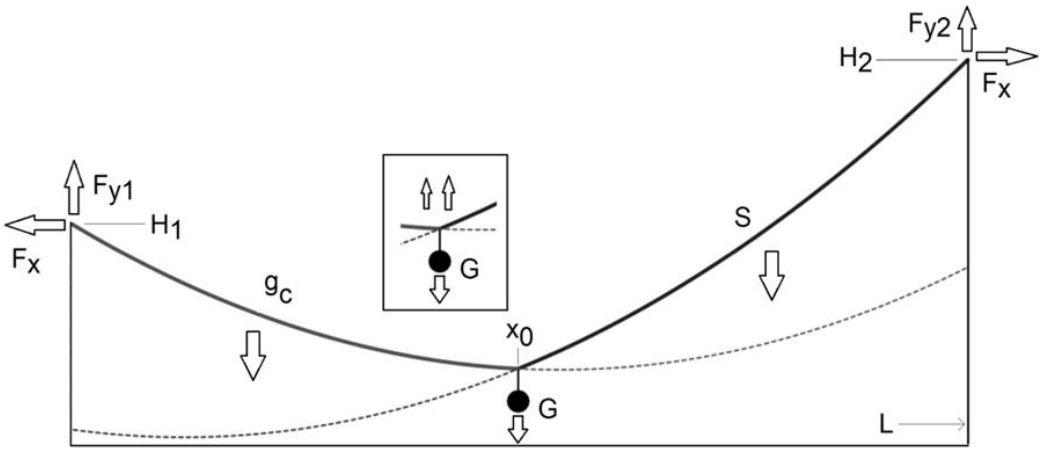


Рис. 7.7

Силы, действующие на подвешенную цепь с грузом

На рисунке 7.8 показано начало расчетного документа (Mathcad Prime 3) с таблицей исходных данных и двумя функциями пользователя — с самой цепной функцией (у нее один аргумент x и три параметра — x0, h и a — см. также рис. 7.3) и ее производной (один аргумент x и два параметра x0 и a). Параметры в уравнениях при их вызове будут иметь индексы 1 и 2, отмечающие левую или правую ветви цепи.

На рисунке 7.9 показаны девять уравнений, записанных в области ограничений блока **Решить** пакета Mathcad Prime (см. этюд 4). Не показана область ввода предположения по решению и область с функцией Find. Выбор начальных приближений (предположений) — это важный этап расчета. Рекомендуется после их

ввода нарисовать график, подобный тому, какой изображен на рис. 7.7. Первые приближения нужно давать такие, чтобы график был более-менее правдоподобен. Потом в работу вступает функция `Find`, которая должна решить задачу с заданной точностью. После решения системы уравнений стоит сделать проверку — вывести «на печать» левые и правые части уравнений системы и сравнить ответы. Они должны отличаться в пределах точности решения задачи численным методом.

H_1	H_2	L	S	G	g_c	x_0
(m)	(m)	(m)	(m)	(kg)	$\left(\frac{kg}{m}\right)$	(m)
7	12	120	121	50	1	60

$$y(x, x0, h, a) := h + a \cdot \left(\cosh\left(\frac{x - x0}{a}\right) - 1\right) \qquad y'(x, x0, a) := \sinh\left(\frac{x - x0}{a}\right)$$

Рис. 7.8

Исходные данные к задаче о провисающей цепи с грузом

Ограничения

$$S = \int_{0\,m}^{x_0} \sqrt{1 + \left(y'(x, x0_1, a_1)\right)^2}\, dx + \int_{x_0}^{L} \sqrt{1 + \left(y'(x, x0_2, a_2)\right)^2}\, dx$$ Длина двух частей цепи

$H_1 = y(0\ m, x0_1, h_1, a_1)$ Левая опора $\qquad H_2 = y(L, x0_2, h_2, a_2)$ Правая опора

$y(x_0, x0_1, h_1, a_1) = y(x_0, x0_2, h_2, a_2)$ Точка подвеса груза

$F_{y1} + F_{y2} = g \cdot (G + g_c \cdot S)$ Вес цепи с грузом

$F_{y1} = -F_x \cdot y'(0\ m, x0_1, a_1)$ Вертикальная реакция левой опоры

$F_{y2} = F_x \cdot y'(L, x0_2, a_2)$ Вертикальная реакция правой опоры

$G \cdot g = F_x \cdot \left(y'(x_0, x0_2, a_2) - y'(x_0, x0_1, a_1)\right)$ Баланс сил в точке подвеса груза

Баланс моментов сил относительно точки подвеса груза

$$F_{y1} \cdot x_0 + F_x \cdot \left(H_2 - y(x_0, x0_1, h_1, a_1)\right) + g_c \cdot g \cdot \int_{x_0}^{L} \sqrt{1 + \left(y'(x, x0_2, a_2)\right)^2}\, dx \cdot \frac{L - x_0}{2} =$$

$$= F_{y2} \cdot (L - x_0) + F_x \cdot \left(H_1 - y(x_0, x0_1, h_1, a_1)\right) + g_c \cdot g \cdot \int_{0\,m}^{x_0} \sqrt{1 + \left(y'(x, x0_1, a_1)\right)^2}\, dx \cdot \frac{x_0}{2}$$

Рис. 7.9

Система уравнений в задаче о провисающей цепи с грузом

На рисунке 7.10 показаны три графика, отображающие провисающую цепь при трех положениях груза: при x_0 = 5 м, x_0 = 60 м и x_0 = 115 м. Пунктиром отмечены продолжения двух ветвей цепной линии.

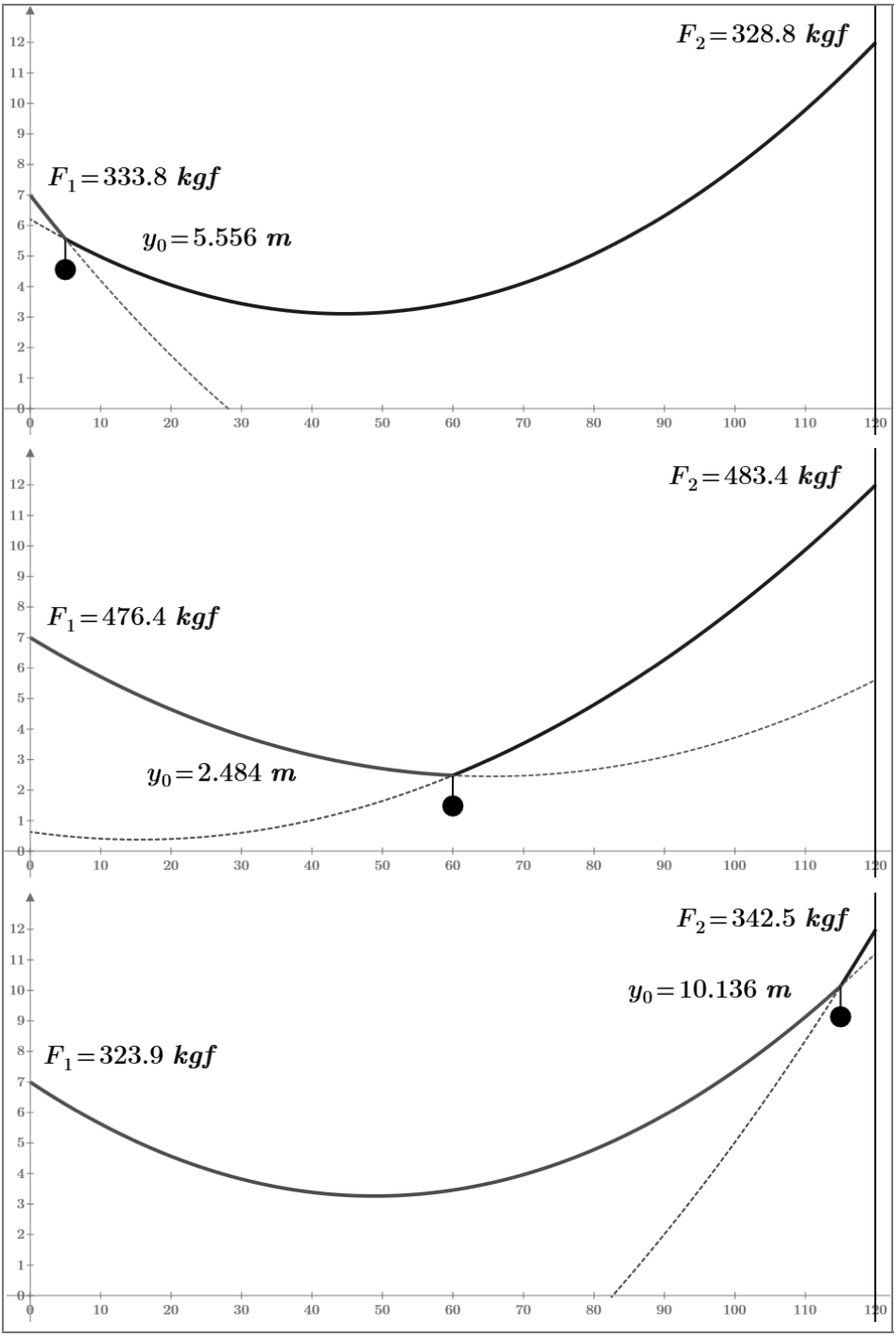


Рис. 7.10

Форма провисающей цепи при трех положениях груза (пропорции осей не соблюдены)

На рисунке 7.11 показан кадр анимации движения груза вдоль оси ординат.

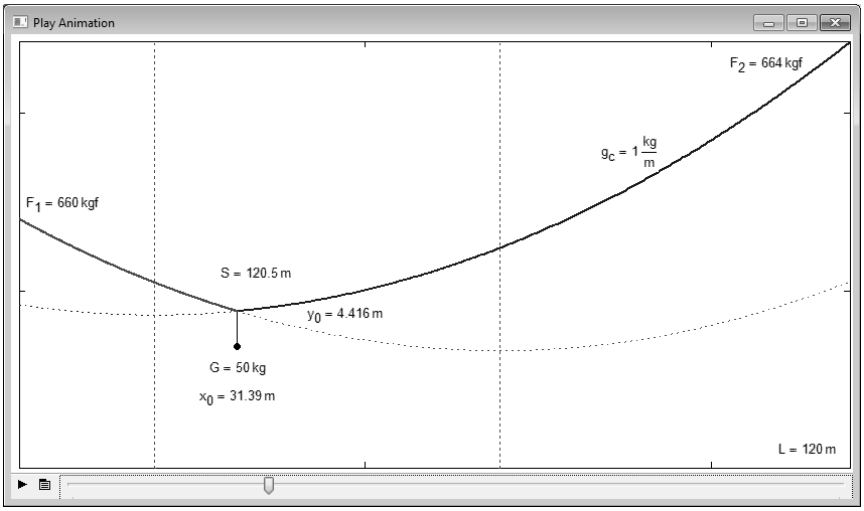


Рис. 7.11
Кадр анимации движения груза, подвешенного на цепи

Созданная математическая модель не будет работать при нулевой массе цепи ($g_c = 0$), т. к. в этом случае уравнения цепной функции (см. рис. 7.8) нужно будет менять на уравнения прямой линии. Да и такая невесомая цепь нереальна — она должна иметь нулевое сопротивление разрыву. Но модель должна работать при нулевой массе груза ($G = 0$) и выдавать результат, аналогичный тому, что дан на рис. 7.3. На рисунке 7.12 показан кадр анимации движения нулевого груза вдоль цепи, состоящей из двух слившихся ветвей цепи.

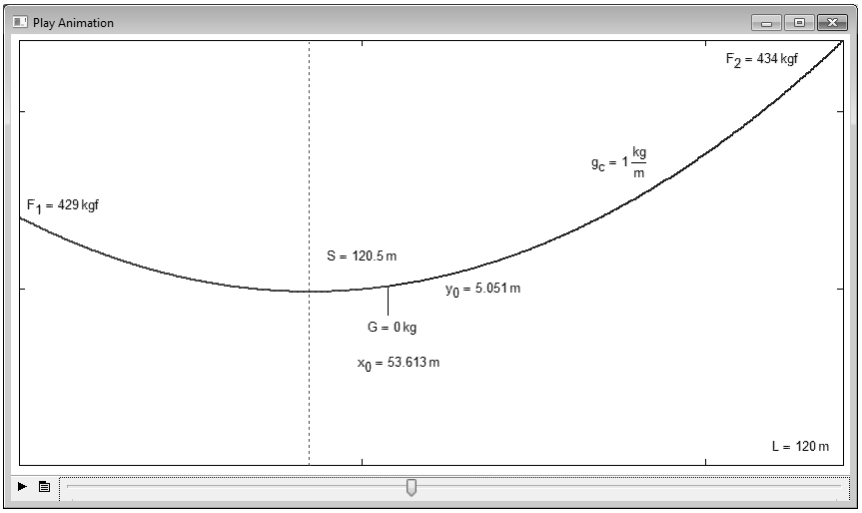


Рис. 7.12
Тестирование модели канатной дороги с нулевым грузом — все ОК

Но если груз оказывается вблизи левого или правого столба (рис. 7.13), то при нулевом весе груза две ветви цепи расходятся, что свидетельствует о дефекте в модели, что связано с допущением в девятом уравнении (баланс моментов сил относительно точки подвеса груза), отмеченным выше.

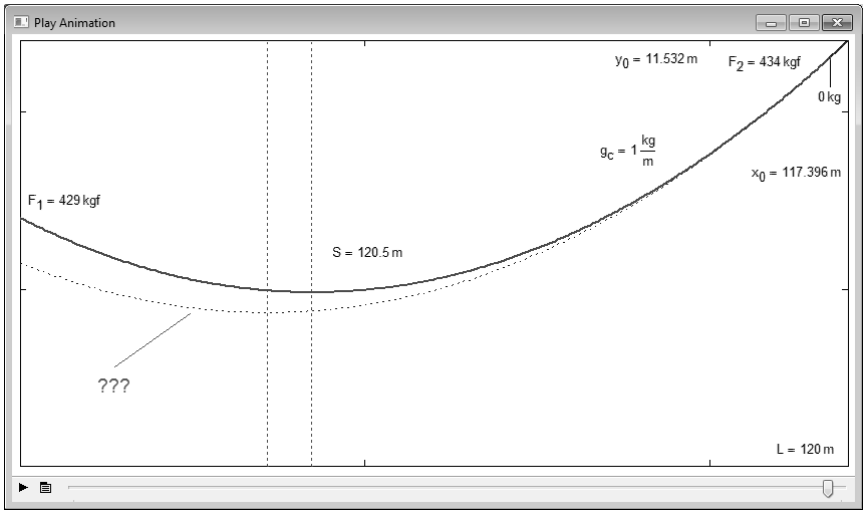


Рис. 7.13

Тестирование модели канатной дороги с нулевым грузом — ошибка у правого края

Рисунок 7.13 не давал покоя авторам. Вдруг кто-то спроектирует канатную подвесную дорогу с нашим допущением для пятого элемента[17], пардон, для девятого уравнения баланса моментов сил, и канат порвется. Не успокаивала и мысль о том, что максимальная нагрузка на канат возникает тогда, когда груз находится в середине каната, а не у его краев — см. цифры для F_1 и F_2 на рис. 7.10. Вспомнилась история, которую рассказывают об одном знаменитом русском механике. Он в студенческие годы устраивался подрабатывать в одно инженерное бюро. Ему в качестве испытания дали проект одного моста и попросили его проверить. Студент сел за работу, нашел в проекте ошибку и побежал в бюро, где закричал: «Этот мост строить нельзя — он разрушится!» Ему сказали, что этот мост был построен и он разрушился. Студент был принят на работу. Вспомнилось и то, что создатели мостов должны были стоять под ними во время первого испытания моста под максимальной нагрузкой либо ехать с этой нагрузкой по мосту первыми.

На рисунке 7.14 показано уточненное уравнение баланса моментов сил, в которое было вставлено определение момента сил от веса провисающих цепей, заданное с помощью цикла с параметром. Цепь разбивается на 1000 элементарных кусочков (элементов) со своим моментом сил, а затем производится суммирование.

Первый оператор на рис. 7.14 подсказывает, как можно выражение с тригонометрическими функциями заменить на более простое выражение с квадратом и квадратным корнем.

Как известно, уравнения бывают алгебраическими и неалгебраическими (трансцендентными — см. этюд 4), которые в свою очередь делятся на рациональные, иррациональные, логарифмические, показательные и прочие. Уравнения также бывают интегральными, дифференциальными (см. этюд 9). У нас же получилось некое *программистское* уравнение, если так можно выразиться. В таком уравнении как минимум один член представляет собой... программный код.

[17] «Пятый элемент» — фантастический боевик с элементами комедии. В основе истории лежит философская притча о любви. Она и есть тот единственный «пятый элемент», способный соединить четыре первичных материи, из которых состоит Вселенная. Наше пятое уравнение баланса моментов сил «соединило» в единую правильную математическую модель четыре разрозненных первичных уравнения баланса сил.

$$\frac{\Delta L}{\cos(\operatorname{atan}(x))} \rightarrow \Delta L \cdot \sqrt{x^2+1}$$

$$F_{y1} \cdot x_0 + F_x \cdot \left(H_2 - y\left(x_0, x0_1, h_1, a_1\right)\right) + \left| \begin{array}{l} M \leftarrow 0 \cdot kgf \cdot m \\ n \leftarrow 1000 \\ \Delta L \leftarrow \dfrac{L - x_0}{n} \\ \text{for } i \in 1..n \\ \quad \left| \begin{array}{l} l \leftarrow \dfrac{\Delta L \cdot i + \Delta L \cdot (i-1)}{2} \\ \Delta m \leftarrow g_c \cdot \Delta L \cdot \sqrt{\left(y'\left(\left(x_0 + l\right), x0_2, a_2\right)\right)^2 + 1} \\ M \leftarrow M + g \cdot \Delta m \cdot l \end{array} \right. \\ M \end{array} \right. =$$

$$= F_{y2} \cdot \left(L - x_0\right) + F_x \cdot \left(H_1 - y\left(x_0, x0_1, h_1, a_1\right)\right) + \left| \begin{array}{l} M \leftarrow 0 \cdot kgf \cdot m \\ n \leftarrow 1000 \\ \Delta L \leftarrow \dfrac{x_0}{n} \\ \text{for } i \in 1..n \\ \quad \left| \begin{array}{l} l \leftarrow \dfrac{\Delta L \cdot i + \Delta L \cdot (i-1)}{2} \\ \Delta m \leftarrow g_c \cdot \Delta L \cdot \sqrt{\left(y'\left(\left(x_0 - l\right), x0_1, a_1\right)\right)^2 + 1} \\ M \leftarrow M + g \cdot \Delta m \cdot l \end{array} \right. \\ M \end{array} \right.$$

Рис. 7.14

Уточненное девятое уравнение

Можно еще подумать и попытаться заменить суммирование в цикле на определенный интеграл[18] — см. рис. 7.15, где сделан контрольный расчет момента сил, действующего на правую ветвь цепи.

$$M := \left| \begin{array}{l} M \leftarrow 0\ kgf \cdot m \\ n \leftarrow 100000 \\ \Delta L \leftarrow \dfrac{L - x_0}{n} \\ \text{for } i \in 1..n \\ \quad \left| \begin{array}{l} l \leftarrow \dfrac{\Delta L \cdot i + \Delta L \cdot (i-1)}{2} \\ \Delta m \leftarrow g_c \cdot \Delta L \cdot \sqrt{y'\left(x_0 + l, x0_2, a_2\right)^2 + 1} \\ M \leftarrow M + g \cdot \Delta m \cdot l \end{array} \right. \\ M \end{array} \right. = 1817.49257527\ kgf \cdot m$$

$$M := g_c \cdot g \cdot \int_{x_0}^{L} \left(x - x_0\right) \cdot \sqrt{1 + \left(y'\left(x, x0_2, a_2\right)\right)^2}\, dx = 1817.49257527\ kgf \cdot m$$

Рис. 7.15

Замена суммы (программирования) на интеграл в девятом уравнении

Но суммирование в цикле (см. рис. 7.14 и первый оператор на рис. 7.15) предпочтительней интеграл (см. второй оператор на рис. 7.15), если оставаться в русле основной идеи первого этюда книги, утверждающей, что в формуле должна быть

[18] От интеграла же в ряде случаев можно избавиться, взяв его — найдя первообразную подынтегрального выражения: программистское уравнение можно в ряде случаев свести к интегральному, а интегральное к обычному алгебраическому или трансцендентному (неалгебраическому).

ясно видима ее физико-математическая суть. В сумме она сразу видна (суммирование моментов сил от элементарных участков цепи `M = M+g·Δm·l`), а в интеграле — нет.

Испытание с помощью анимации подправленной математической модели нашей подвесной канатной дороги показало, что две кривые цепной функции сливаются в одну и в том случае, когда груз нулевой массой находится вблизи опоры — сравните рис. 7.13 с рисунками 7.16 и 7.17.

H_1 (m)	H_2 (m)	L (m)	S (m)	G (kg)	g_c $\left(\frac{kg}{m}\right)$	x_0 (m)
7	12	120	121	0	1	0.5

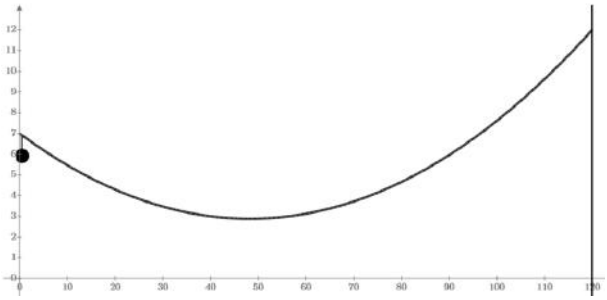


Рис. 7.16
Груз с нулевой массой у левой опоры подвесной канатной дороги

H_1 (m)	H_2 (m)	L (m)	S (m)	G (kg)	g_c $\left(\frac{kg}{m}\right)$	x_0 (m)
7	12	120	121	0	1	119.5

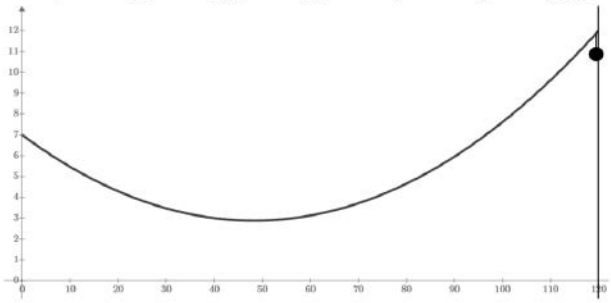


Рис. 7.17
Груз с нулевой массой у правой опоры подвесной канатной дороги (см. рис. 7.13)

Небольшое, но важное отступление от основной темы этюда.

На рисунках 7.14 и 7.15 показаны *программы*, созданные инструментами Mathcad. Не секрет, что во многих школах и вузах занятия по информатике — это фактически занятия по азам программирования, хотя информатика и программирование — это, как говорят математики, множества пересекающиеся, но не совпадающие. Нечеткие множества — см. этюд 12.

В среде Mathcad можно решать довольно сложные задачи, не прибегая к программированию и выполняя операторы в таком (естественном[19]) порядке — слева направо и/или сверху вниз. Но есть расчетные ситуации, когда этот порядок требуется изменить — выполнить, например, один блок операторов и не выполнять другой. Иногда требуется выполнить какую-то группу операторов несколько раз, чтобы автоматизировать, например, реализацию метода последовательных приближении, который часто используется в инженерной практике[20]. Или как в нашем случае с нагруженной цепью для того, чтобы рассчитать момент силы, приложенного к не совсем стандартному телу[21]. Первый атрибут (неотъемлемый признак) программы в среде Mathcad и не только — это *изменение естественного порядка выполнения операторов*.

На заре программирования этот порядок изменяли так — вставляли в программный код метки и операторы условного и безусловного перехода к меткам. В среде Mathcad, а также в других современных языках программирования это делается в основном[22] не через метки, а с помощью *структурных управляющих конструкций* — основных и вспомогательных[23]. При реализации этих конструкций расчетные операторы объединяются в *программные блоки* — второй атрибут программы. Такие блоки, имеющие, как правило, один вход и один выход, выполняются как единый оператор. В среде Mathcad программные блоки визуально отмечаются вертикальной чертой — сплошной одинарной (Mathcad 15) или двойной пунктирной (Mathcad Prime[24]). Эти линии появляются автоматически при вводе в блок нового оператора (через нажатие кнопки **Добавить строку — Add line** или | в панели программирования. В среде Mathcad Prime дополнительно справа от операторов блока проводится серая пунктирная линия. Это сделано для упрощения процедуры редактирования программы — для ввода в нее либо нового блока операторов (курсор находится правее серой вертикальной черты), либо для ввода нового оператора в блок (курсор находится левее серой вертикальной черты).

В среде языка программирования Pascal, который многие изучали и изучают в школе и который сейчас практически нигде не используется[25], начало и конец программного блока фиксируется ключевыми словами `begin` и `end` соответственно. В среде Mathcad блок операторов можно также сформировать, вставив операторы в матрицу с одной строкой — см., например, вторую строку в программе на

---

[19] Этот порядок (слева направо) можно назвать естественным, если не учитывать то, что есть письменности (иврит, арабский язык), где пишут справа налево.

[20] Метод последовательных приближении можно применить, если, например, нужно будет в задаче о подвесной канатной дороге рассчитать толщину используемого троса с известным удельным сопротивлением разрыву. Задается толщина троса, рассчитывается его сечение, по которому определяется сила, при которой трос оборвется. Далее рассчитывают максимальную силу, какая будет действовать на трос при его подвесе с грузом на канатной дороге. Если вторая сила окажется больше первой, то толщину троса увеличивают, а расчет повторяют. Если же вторая сила окажется существенно ниже первой, то толщину троса уменьшают. Такие последовательные приближения делают до тех пор, пока не получится приемлемый результат. Это можно делать, повторяя несколько раз данный фрагмент расчета, или вставив в расчет цикл.

[21] В среде Mathcad есть справочник с формулами момента инерции некоторых стандартных тел: цилиндра, конуса, пластины, диска и др.

[22] В среде Mathcad не в основном, а всегда.

[23] В среде Mathcad основной структурной управляющей конструкцией является цикл `while` (цикл с предпроверкой). Остальные инструменты программирования (операторы `if` и `for`) можно реализовать через оператор `while`. Операторы же `return`, `break` и `continue` введены в Mathcad относительно недавно. Они позволяют досрочно прервать выполнение операторов в программном блоке.

[24] При подготовке некоторых рисунков книги этот пунктир заменялся на сплошную линию. Кроме того, вторая линия иногда убиралась.

[25] Это к тезису об отрыве школы от жизни. История повторяется. В старой России в гимназиях учащиеся изучали латынь и древнегреческий. Но на этих языках никто уже не говорил и даже ученые разных стран перестали общаться (переписываться) на латыни, перейдя на немецкий или французский. В новой России продолжают изучать язык Паскаль, хотя его уже никто не использует в практических целях.

рис. 18.2 на стр. 372. Но это недокументированный прием, который, тем не менее, иногда используется для того, чтобы программа (или ее отдельные программные блоки) целиком уместилась на экране дисплея. Так ее лучше анализировать и редактировать.

В среде Mathcad Prime к такому «вертикальному сжатию» программы стали прибегать чаще из-за того, что стало возможным хранить в матрице операторы с разной размерностью (длина и масса, например), что было невозможно в среде традиционного Mathcad 15.

Третий атрибут программы — это *локальные переменные* — переменные, видимые только в самой программе. В среде Mathcad всех версий, начиная с Mathcad 5, где появилось программирование, локальная переменная задается оператором ← (левая стрелка), вид которого (но не его суть) в среде Mathcad 15 можно поменять на «=» или «:=». В среде Mathcad Prime такой опции пока нет.

Из вышеотмеченных трех атрибутов программирования, по правде говоря, атрибутом (неотъемлемым признаком) можно считать только первый — изменение естественного порядка выполнения операторов. Остальные два (блоки операторов и локальные переменные) в программе могут быть, а могут и не быть.

Одна из причин широкой популярности пакета Mathcad состоит в том, что с его помощью пользователи, решая свои рутинные задачи, могли плавно и незаметно входить в прекрасный мир программирования. Многие люди ошибочно считают, что они никогда не смогут написать на компьютере более-менее сложную программу и боятся иметь дело с языками программирования, ограничивая себя несложными расчетами с помощью, например, электронных таблиц. Языки программирования отталкивали от себя многих рядовых пользователей компьютера и тем, что в их среде до выполнения нужных расчетов необходимо было сделать какие-то дополнительные, часто непонятные операции. В среде языка Pascal, например, который, повторяем, до сих пор широко используется в школах и вузах для обучения программированию, нужно перед работой с переменными предварительно их задать и определить их тип: логическая (булева), целочисленная, вещественная, вещественная с двойной точностью и т. д. Эта операция была понятна и обоснована во времена, когда память компьютера была одним из лимитирующих параметров при решении задач. Сейчас же проще нарастить память компьютера, чем задавать тип переменных.

Работая в среде Mathcad, рядовой пользователь может решать довольно сложные задачи (решать алгебраические и дифференциальные уравнения (см. этюд 9), строить графики, создавать анимацию (см. этюд 8), вести сложный статистический анализ (см. этюд 14) и т. д. и т. п.), не прибегая к программированию. И мы уже это отметили. Но этот же пользователь незаметно для себя может постепенно начать использовать некоторые элементы программирования в своих расчетах, обогащая свой арсенал инструментальных средств. Некоторые пользователи, не зная о наличии в среде Mathcad тех или иных встроенных функций, начинают создавать их средствами программирования Mathcad. Так, например, если нужно найти минимальный или максимальный элемент вектора или матрицы, то можно с помощью цикла `for` с оператором `if` внутри перебрать все элементы массива и отобрать нужный. А можно вспомнить (или узнать), что в среде Mathcad для этого есть соответствующие встроенные функции `min` и `max`.

Важное замечание

Программирование нужно не только для того, чтобы с его помощью решать сложные задачи, для которых нет встроенных инструментов. Программирование — это очень мощное средство для развития «ума и сообразительности». Это своеобразная и очень эффективная «гимнастика» для ума. Обычно этот тезис относят к математике — элементарной (школьной) и вузовской (высшей). Изучив математику, человек, может быть, никогда и не будет решать уравнения и системы уравнений, брать интегралы, вычислять пределы, инвертировать матрицы и определять их ранги. Но, развив посредством математики свой интеллект, чело-

век сможет успешно решать свои житейские и производственные задачи. И, конечно, нужно помнить, что программист — это очень востребованная специальность. Особенно в тех случаях, когда человек приобрел и другую специальность — специальность теплотехника, например.

Многие на досуге любят решать различные головоломки: шарады, ребусы, кроссворды, судоку и т. д. и т. п. Так вот, составление и отладка программы, даже если она не представляет никакой практической ценности, — это хорошая головоломка в часы досуга, когда хочется занять свой ум и руки интересной задачей.

Но вернемся к канатной дороге.

Наши расчеты показывают, что мост или канатную дорогу нужно испытывать не только при максимальной, но и при минимальной (нулевой) нагрузке. И даже при отрицательной нагрузке — см. рис. 7.18.

H_1 (m)	H_2 (m)	L (m)	S (m)	G (kg)	g_c $\left(\frac{kg}{m}\right)$	x_0 (m)
7	12	120	121	−50	1	60

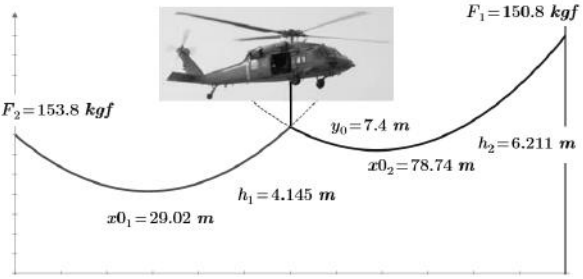


Рис. 7.18
Канатная дорога с грузом отрицательной массы

Мы уже подчеркивали особо, что инженер должен решать не просто технические, а технико-экономические задачи. Если, например, под нашей ЛЭП должен проплыть всего лишь раз аномально высокий объект, то можно не строить высокие мачты или не натягивать излишне провода, а... только приподнять их единожды с помощью, например, вертолета[26]. Кстати можно заметить, что у летящего с внешней подвеской вертолета трос тоже прогибается не по прямой, а по цепной линии. И это тоже можно обсчитать, используя подходы, описанные в этом этюде.

У этого «пятого элемента», пардон, девятого уравнения была и другая история. Сначала для решения этой задачи были составлены уравнения, учитывающие геометрию подвесной канатной дороги и балансы только сил (см. выше). Функция `Find` стала выдавать более-менее правдоподобные ответы, на которых решено было остановиться. Но при анимировании модели было замечено, что виртуальный канат как-то странно дергается, а при некоторых положениях груза превращался в... арку. Это насторожило и заставило тщательнее проанализировать математическую модель. Стало ясно, что в ней девять неизвестных и восемь уравнений — система была недоопределена (см. этюд 6) и имеет бесконечное множество решений, заставляющих дергаться анимацию. В этой ситуации функция `Find`, по идее, должна была возвращать текстовое сообщение о бесконечном числе решений, а не вектор одного из возможных решений, зависящих от первого приближения. Пришлось еще раз вспомнить институтский курс теоретической меха-

[26] Еще одно решение — если по реке плывет ретро-пароход с высокой трубой, то под мостом или линией ЛЭП эту трубу можно временно опускать.

ники и ввести в систему «пятый элемент» — девятое уравнение, учитывающее баланс моментов сил, с которым пришлось немного повозиться.

Наш расчет можно усложнять, подвешивая на цепь/канат второй, третий груз и ждать, что скорее порвется — реальная цепь или... функция `Find`, которая на каком-то этапе не сможет решить такую сложную систему уравнений. Усложнение можно вести и в других направлениях. Можно, например, учитывать упругость троса и изменение его длины под действием растягивающей силы, которая, кстати, имеет разное значение в разных точках троса. Можно решать дифференциальное уравнение колебания троса с грузом, вспомнив, с чего начался наш расчет — с дачного электрического провода с подвешенным кирпичом...

ДИВЕРТИСМЕНТ 1. КВАДРАТНОЕ КОЛЕСО

Арка, показанная на рисунке 7.1, установлена, как уже отмечено, на берегу реки Миссисипи[27]. Обычно в городах на набережных устраивают место для прогулок — пеших и велосипедных. Так вот, можно порекомендовать у подножья этой арки сделать велодорожку для езды на велосипедах, но не с обычными, а с... квадратными колесами. А чтобы не было тряски и велосипед перемещался сугубо по прямой, велодорожка должна походить на... стиральную доску, профиль которой состоит из отрезков... цепной линии — см. рис. 7.19.

Рис. 7.19
*Велосипед с квадратными колесами (источник **http://stanwagon.com**)*

Идея квадратных колес обсуждалась на сайте PTC Community — **https://www.ptcusercommunity.com/videos/2149**. Авторы написали письмо (естественно, электронное) человеку, изображенному на рис. 7.19, и попросили его прокомментировать идею дорожки для байка с квадратными колесами у подножия арки в Сент-Луисе. Этот человек, к удивлению, ответил, но в том плане, что он сейчас всецело поглощен новой кинематической идеей — идеей сверления... квадратных отверстий.

И еще одна неплохая идея. Велосипеды с квадратными колесами для езды по спецдорожке у подножия арки в Сент-Луисе можно хранить в круглом (цилиндрическом) помещении с крышей-куполом, сделанной в виде катеноиды — поверхности, полученной вращением цепной линии вокруг вертикальной оси, проходящей через точку максимума. При этом можно так оптимизировать поверхность, чтобы такая крыша давила на опорное кольцо с минимальной силой. В этюде 4 мы решали подобную плоскую задачу — см. рис. 4.49 и 4.51 на стр. 101 и 103.

[27] Нерадивый школьник в одном из рассказов Виктора Драгунского называет эту реку Миссиписи.

ПРОДОЛЖЕНИЕ ЭТЮДА... 7

Про цепь. Точнее, про велосипедную цепь, излишнее провисание которой может испортить велосипедную прогулку... у подножия арки в Сент-Луисе (см. рис. 7.1 и 7.19). Переключая скорости велосипеда, цепь перекидывают с шестеренки на шестеренку. Попробуем анимировать эту кинематическую модель.

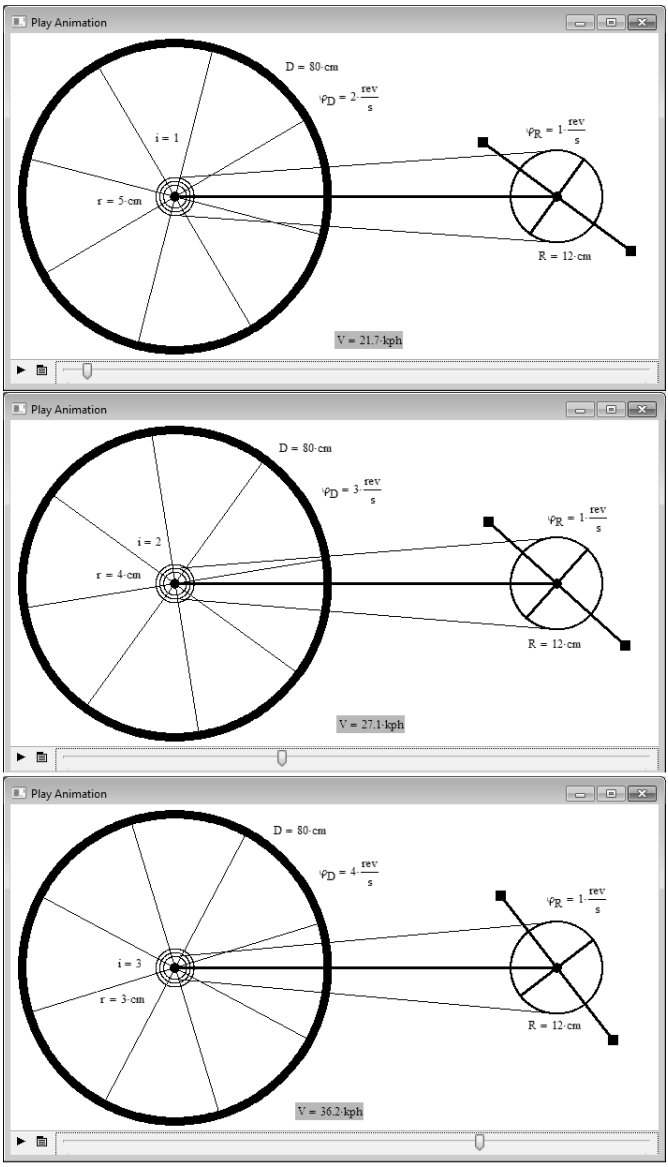


Рис. 7.20

Три кадра анимации трех скоростей велосипеда
(см. анимацию здесь ***https://www.ptcusercommunity.com/videos/2101****)*

Исходными данными для расчета, поддерживающего анимацию, показанную на рис. 7.20, является постоянная угловая скорость кручения педалей велосипеда (один оборот в секунду), а также геометрия велосипеда и его зубчатых колес. Рас-

считывается же линейная скорость велосипеда (21.7, 27.1 и 36.2 км/ч) при трех скоростях: `i = 1, 2` и `3` — трех шестеренках на втулке заднего колеса. А там может быть до девяти шестеренок. И еще до пяти шестеренок бывает на втулке педалей. Расчет их размеров и сочетаний перекидывания цепи — это интересная инженерная задача.

ПРОДОЛЖЕНИЕ ЭТЮДА 7. ТОЖЕ ПРО ЦЕПИ, НО ЭЛЕКТРИЧЕСКИЕ

Студентам технических вузов на первом курсе обычно читают два раздела высшей математики — математический анализ и линейную алгебру. Элементы матанализа (анализ функциональных зависимостей, дифференциальное и интегральное исчисление) мы затронули выше. Но если проследить, что будет с нашим электрическим током, перекинутым по цепеподобным проводам ЛЭП через реку, то можно описать и элементы линейной алгебры.

А что будет с током, «перекинутым по цепеподобным проводам» через реку? Напряжение тока будет понижено и он будет распределен по электрическим сетям, которые также иногда называют цепями.

Вот типовая задача электротехники — нахождение значений тока на отдельных участках электрической цепи в зависимости от заданных значений локальных сопротивлений и электродвижущей силы (ЭДС, EMF — an electromotive force). Она сводится к решению системы алгебраических уравнений, см. рис. 7.21.

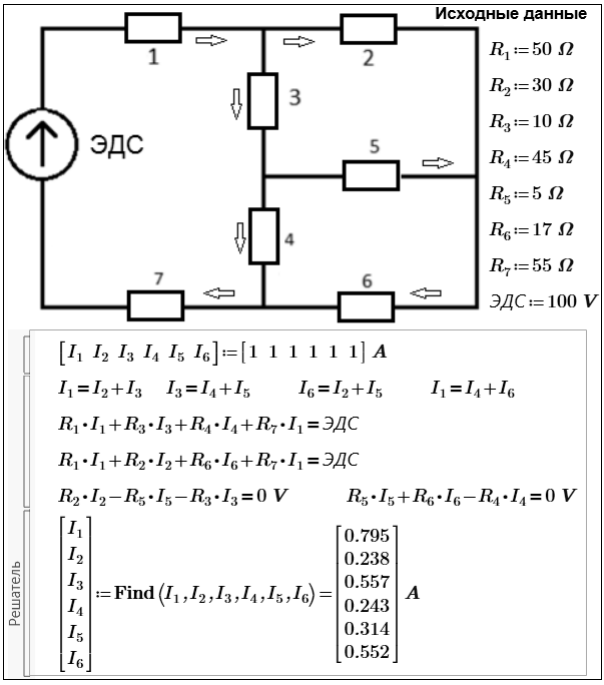


Рис. 7.21

Пример электротехнического расчета в среде Mathcad Prime

На рисунке 7.21 показано решение электротехнической задачи блоком **Решатель:** вводятся начальные приближения к решению: всем шести искомым значениям тока присваивается один ампер. Но можно давать и другие значения. Выбор этих значений — важный этап решения задачи. Пакет Mathcad иногда не может решить систему и просит изменить начальные приближения. Инженер, решающий на компьютере свою задачу, должен приблизительно знать, какое решение

ему нужно, и давать первое приближение вблизи этой точки. Далее в расчете на рис. 7.21 записываются уравнения баланса токов (первый закон Кирхгофа) и баланса «энергий» по отдельным замкнутым ветвям электрической сети (второй закон Кирхгофа). Функция `Find` возвратила значение шести токов, дающих приближенное решение исходной системы. У нас в задаче восемь уравнений и шесть неизвестных. Задача, следовательно, является *переопределенной*[28], т. к. в ней число уравнений превышает число неизвестных. В принципе, тут два уравнения уже лишние. Но можно добавить еще пару уравнений[29], применив второе правило Кирхгофа к замкнутым ветвям электрической цепи ЭДС–R_1–R_3–R_5–R_6–R_7 и ЭДС–R_1–R_2–R_5–R_4–R_7. Задача при этом будет еще более переопределена, но решение останется прежним. Такая ситуация говорит лишь о том, что некоторые уравнения линейно выражаются через другие и могут быть удалены из системы. Но заниматься анализом линейной зависимости и упрощением системы уравнений — задача сама по себе сложная. Поэтому в случае существования единственного решения системы (а мы имеем конкретный ток в цепи) можно решение переопределенной системы переложить на плечи компьютера.

Решение, показанное на рис. 7.21, вполне может устроить пользователя, если он будет уверен в единственности решения, т. е. в возможности искать это решение любыми доступными способами. Но и здесь требуется уточнить, о единственности какого решения идет речь: решения поставленной электротехнической задачи или решения конкретной системы линейных алгебраических уравнений, которая в общем случае может плохо описывать физическую модель. Кроме того, смущает то, что приходится задавать начальные приближения к решению[30]. Обычно это делается тогда, когда у системы несколько решений, а нужно найти только одно. Пользователь должен хотя бы примерно знать, сколько решений будет у поставленной реальной задачи, и какую тактику при решении конкретной задачи он должен использовать. Несколько решений бывают у нелинейных систем, а наша система, повторяем, линейная. У такой системы, моделирующей реальный физический процесс, обычно существует единственное решение, но есть случаи, когда решений нет или их бесконечное множество. Сейчас мы покажем, как можно в линейной системе алгебраических уравнений детерминировать эти три случая.

На рисунке 7.22 дано решение нашей задачи одним из инструментов Mathcad, предназначенным для решения систем *линейных* алгебраических уравнений — функцией `lsolve` (l — линейный, solve — решить).

В верхней части рис. 7.22 записаны наши уравнения, вытекающие из первого и второго законов (правил) Кирхгофа, дополненные «нулевыми» слагаемыми и коэффициентами так, чтобы получилась система линейных уравнений в ее классическом написании: коэффициенты при неизвестных, сами неизвестные и правые части уравнений. После этого можно сформировать матрицу коэффициентов при неизвестных `M` и вектор свободных членов `v`. Функция `lsolve` вернула нам ре-

[28] Не всякая задача, в которой число уравнений больше числа неизвестных, является переопределенной. Например, если уравнения линейно зависимы, то описанная система может быть как определенной (фактически это и показано на примере расчета), так и неопределенной. Соотношение числа уравнений и числа неизвестных не является критерием переопределенности. Возможно, вообще не следует уделять столько внимания этому тонкому вопросу. Может быть, лучше сказать, что с системами бывает всякое, а вот пакет сам умеет (или не умеет!) в этом разобраться.

[29] Если задача переопределена, то зачем же еще добавлять уравнения? Или почему их можно не добавлять, если они физически необходимы? Что-то следует обосновать. Наверное, сослаться на теорию расчета цепей или на что-то другое. В любом случае перед тем, как рассматривать вопрос численного решения системы, следует обосновать адекватность выбранной математической модели (имеющей вид системы линейных алгебраических уравнений) изучаемой физической модели электрической цепи.

[30] Почему же? Всегда ли? Для каких систем? Кто или что требует их задавать в рассматриваемом случае? Нужно знать специфику численного метода, зашитого в пакет.

шение, повторяющее решение на рис. 7.21, но без всяких «начальных приближений», которые нас смущали.

$$
\begin{aligned}
&1\cdot I_1-1\cdot I_2-1\cdot I_3+0\cdot I_4+0\cdot I_5+0\cdot I_6=0\ \boldsymbol{A}\\
&0\cdot I_1+0\cdot I_2+1\cdot I_3-1\cdot I_4-1\cdot I_5+0\cdot I_6=0\ \boldsymbol{A}\\
&0\cdot I_1-1\cdot I_2+0\cdot I_3+0\cdot I_4-1\cdot I_5+1\cdot I_6=0\ \boldsymbol{A}\\
&1\cdot I_1+0\cdot I_2+0\cdot I_3-1\cdot I_4+0\cdot I_5-1\cdot I_6=0\ \boldsymbol{A}\\
&\left(R_1+R_7\right)\cdot I_1+0\ \boldsymbol{\Omega}\cdot I_2+R_3\cdot I_3+R_4\cdot I_4+0\ \boldsymbol{\Omega}\cdot I_5+0\ \boldsymbol{\Omega}\cdot I_6=\text{ЭДС}\\
&\left(R_1+R_7\right)\cdot I_1+R_2\cdot I_2+0\ \boldsymbol{\Omega}\cdot I_3+0\ \boldsymbol{\Omega}\cdot I_4+0\ \boldsymbol{\Omega}\cdot I_5+R_6\cdot I_6=\text{ЭДС}\\
&0\ \boldsymbol{\Omega}\cdot I_1+R_2\cdot I_2-R_3\cdot I_3+0\ \boldsymbol{\Omega}\cdot I_4-R_5\cdot I_5+0\ \boldsymbol{\Omega}\cdot I_6=0\ \boldsymbol{V}\\
&0\ \boldsymbol{\Omega}\cdot I_1+0\ \boldsymbol{\Omega}\cdot I_2+0\ \boldsymbol{\Omega}\cdot I_3-R_4\cdot I_4+R_5\cdot I_5+R_6\cdot I_6=0\ \boldsymbol{V}
\end{aligned}
$$

$$
M:=\begin{bmatrix}
1 & -1 & -1 & 0 & 0 & 0\\
0 & 0 & 1 & -1 & -1 & 0\\
0 & -1 & 0 & 0 & -1 & 1\\
1 & 0 & 0 & -1 & 0 & -1\\
R_1+R_7 & 0\ \boldsymbol{\Omega} & R_3 & R_4 & 0\ \boldsymbol{\Omega} & 0\ \boldsymbol{\Omega}\\
R_1+R_7 & R_2 & 0\ \boldsymbol{\Omega} & 0\ \boldsymbol{\Omega} & 0\ \boldsymbol{\Omega} & R_6\\
0\ \boldsymbol{\Omega} & R_2 & -R_3 & 0\ \boldsymbol{\Omega} & -R_5 & 0\ \boldsymbol{\Omega}\\
0\ \boldsymbol{\Omega} & 0\ \boldsymbol{\Omega} & 0\ \boldsymbol{\Omega} & -R_4 & R_5 & R_6
\end{bmatrix}
\quad
v:=\begin{bmatrix}
0\ \boldsymbol{A}\\ 0\ \boldsymbol{A}\\ 0\ \boldsymbol{A}\\ 0\ \boldsymbol{A}\\ \text{ЭДС}\\ \text{ЭДС}\\ 0\ \boldsymbol{V}\\ 0\ \boldsymbol{V}
\end{bmatrix}
\quad
\mathrm{lsolve}(M,v)=\begin{bmatrix}
0.795\\ 0.238\\ 0.557\\ 0.243\\ 0.314\\ 0.552
\end{bmatrix}\boldsymbol{A}
$$

Рис. 7.22

Решение переопределенной системы линейных алгебраических уравнений в среде Mathcad Prime

Из нашей системы можно убрать два лишних уравнения[31], как мы уже отметили ранее. После этого матрица `M` станет квадратной, и у нее можно будет подсчитать детерминант (определитель), а затем инвертировать ее (найти обратную матрицу). Обратную матрицу можно умножить на вектор свободных членов и... получить решение задачи. Вот тут-то и пригодятся нам знания линейной алгебры, которые мы приобрели в институте! На рис. 7.23 решается «классическая» система шести линейных алгебраических уравнений с шестью неизвестными «классическим» способом — векторным умножением инвертированной (обратной) матрицы коэффициентов при неизвестных на вектор свободных членов. Убедиться в том, что мы имеем единственное решение, помогает функция `rank`, возвращающая ранг квадратной матрицы `M` и расширенной матрицы `M1`. Теорема Кронекера — Капелли гласит, что «система линейных алгебраических уравнений совместна тогда и только тогда, когда ранг ее основной матрицы равен рангу ее расширенной матрицы, причем система имеет единственное решение, если ранг равен числу неизвестных, и бесконечное множество решений, если ранг меньше числа неизвестных».

В нашей системе шести линейных алгебраических уравнений с шестью неизвестными (см. рис. 7.23) записаны три первых правила Кирхгофа и три вторых правила Кирхгофа. Если эту пропорцию нарушить, записать, например, два первых правила Кирхгофа для двух «тройников» и четыре вторых правила Кирхгофа для четырех замкнутых петель, то можно получить неверное решение[32]. И самое неприятное будет то, что это решение будет очень похоже на истинное[33]. Если нашу модель изменить (записать два, а не три первых правила Кирхгофа и четыре, а не три вторых правила Кирхгофа), то ранги новой основной и новой расширенной матриц при этом станут равны 5 (см. рис. 7.24), а не 6 (см. рис. 7.23), что будет сигналом об

[31] Почему они лишние, почему лишние именно эти, а не другие? С точки зрения теории систем линейных алгебраических уравнений лишние уравнения обнаруживаются, причем не единственным образом, только после нахождения ранга расширенной матрицы. Здесь, вероятно, уместнее, не вдаваясь в математические подробности, дать физическое или техническое обоснование выделения лишних уравнений.

[32] А можно и не получить? Как это определить?

[33] Что означает «похоже»? Если решение единственно и приблизительно (численно) совпадает с искомым, то для технических расчетов это не неприятность, а норма. Если же математическая теория доказывает отсутствие решения, а оно присутствует (т. е. вычисляется), либо существование бесконечного числа решений (неизвестных больше, чем ранг системы), а вычисляется какое-то одно, то использована некорректная модель. И это уже очень большая неприятность.

ошибке в решении, т. к. наша система имеет 6 неизвестных (см. выше теорему Кронекера — Капелли). Вывод такой: для решаемой задачи математическая модель, адекватная физической, содержит ровно шесть линейно независимых уравнений, которые можно правильно выделить из бóльшего (избыточного) числа непротиворечивых уравнений, используя либо математический подход (вычисление ранга матрицы и выделение базисного минора), либо физический подход (определение физической значимости каждого из уравнений). Первый требует хороших знаний общей теории решения систем линейных алгебраических уравнений, второй помимо знания теории цепей требует хорошо развитой инженерной интуиции и навыка в решении подобных задач. Но если не ставить своей целью уменьшение количества уравнений, то решение переопределенной алгебраической системы можно просто поручить компьютеру.

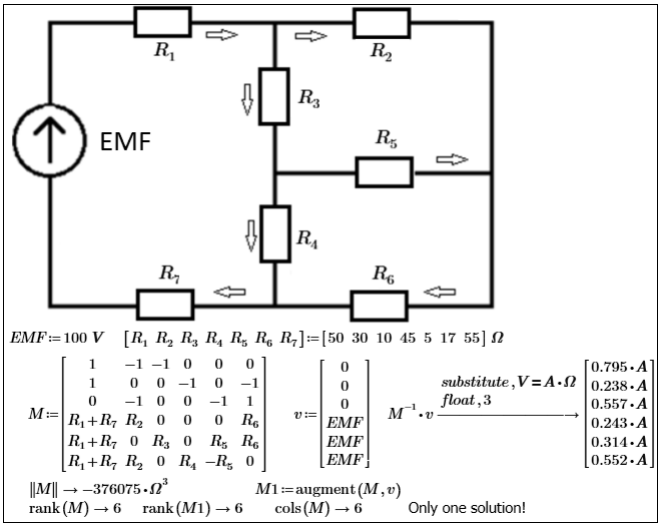


Рис. 7.23

Анализ и решение «классической» системы линейных алгебраических уравнений в среде Mathcad Prime с привлечением азов линейной алгебры

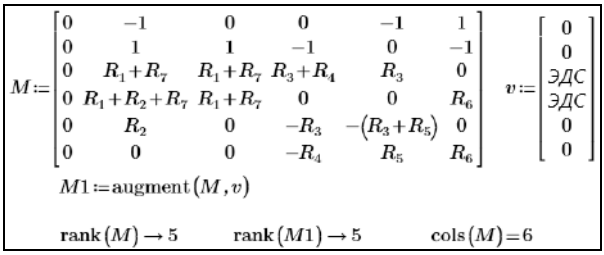


Рис. 7.24

Анализ неправильно составленной системы линейных алгебраических уравнений

На одной консультации перед экзаменом по информатике студенты одного из авторов книги признались, что они очень плохо сдали последний экзамен — экзамен по линейной алгебре, по учебному курсу, который студенты изучают параллельно курсам математического анализа и информатики. Ядром курса матанализа является изучение функциональных зависимостей. Линейная алгебра со своими векторами и матрицами нацелена на решение *систем линейных алгебраических уравнений* (СЛАУ). Анализируя провалы на экзамене (трудности понимания) студентов в линейной алгебре, этот автор, используя инструменты Mathcad, исследо-

вал систему трех линейных уравнений и привел на консультации для каждого случая геометрическую интерпретацию — см. рис. 7.25, 7.26 и 7.27.

Система линейных алгебраических уравнений A x= B совместна тогда и только тогда, когда ранг её основной матрицы A равен рангу её расширенной матрицы A1, причём система имеет единственное решение, если ранг равен числу неизвестных, и бесконечное множество решений, если ранг меньше числа неизвестных.

Случай 1

$$A \cdot x = B \quad A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad B := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad A1 := \mathrm{augment}(A, B) = \begin{pmatrix} 1 & 2 & 3 & 1 \\ 4 & 5 & 6 & 2 \\ 7 & 8 & 9 & 3 \end{pmatrix}$$

Ранг основной матрицы	$\mathrm{rank}(A) = 2$	Бесконечное множество решений
Ранг расширенной матрицы	$\mathrm{rank}(A1) = 2$	$\mathrm{rank}(A) = \mathrm{rank}(A1) = 1$
Число неизвестных	$\mathrm{cols}(A) = 3$	$\mathrm{rank}(A) < \mathrm{cols}(A) = 1$

Графическая иллюстрация - три плоскости пересекаются в одной линии

$$z1(x, y) := A_{0,0}x + A_{0,1}y + A_{0,2}z = B_0 \ \mathrm{solve}, z \ \rightarrow \frac{1}{3} - \frac{2 \cdot y}{3} - \frac{x}{3}$$

$$z2(x, y) := A_{1,0}x + A_{1,1}y + A_{1,2}z = B_1 \ \mathrm{solve}, z \ \rightarrow \frac{1}{3} - \frac{5 \cdot y}{6} - \frac{2 \cdot x}{3}$$

$$z3(x, y) := A_{2,0}x + A_{2,1}y + A_{2,2}z = B_2 \ \mathrm{solve}, z \ \rightarrow \frac{1}{3} - \frac{8 \cdot y}{9} - \frac{7 \cdot x}{9}$$

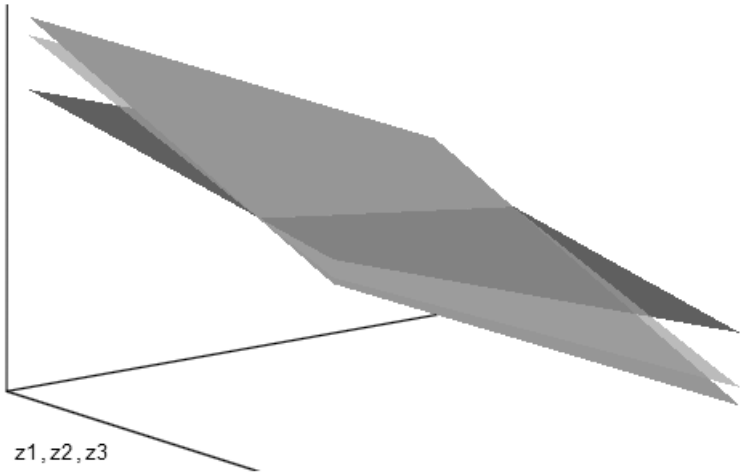


Рис. 7.25

Решение СЛАУ в среде Mathcad — бесконечное множество решений

Примечание. Выражение `rank(A) = rank(A1) = 1` означает не то, что ранги матриц `A` и `A1` равны единице, а то, что ранги этих матриц равны. В среде Mathcad единица (не ноль) в булевых выражениях это истина, а ноль — ложь (см. этюд 11).

Случай 2

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad B := \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \quad A1 := \mathrm{augment}(A, B) = \begin{pmatrix} 1 & 2 & 3 & 1 \\ 4 & 5 & 6 & 2 \\ 7 & 8 & 9 & 1 \end{pmatrix}$$

Ранг основной матрицы $\mathrm{rank}(A) = 2$

Ранг расширенной матрицы $\mathrm{rank}(A1) = 3$

Нет решения

$\mathrm{rank}(A) = \mathrm{rank}(A1) = 0$

Графическая иллюстрация - три плоскости попарно пересекаются в двух параллельных линиях (общей точки у плоскостей нет)

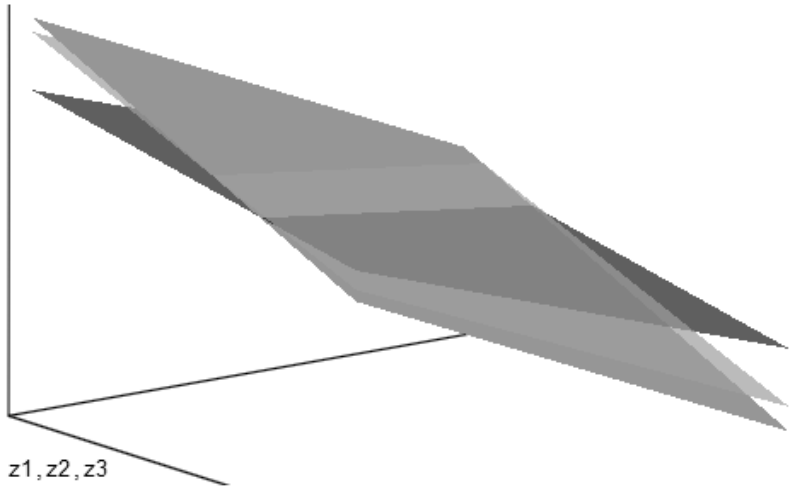


Рис. 7.26
Решение СЛАУ в среде Mathcad — нет решения

На рис. 7.25, 7.26 и 7.27 средствами Mathcad показаны три случая, возникающих при решении систем трех линейных алгебраических уравнений: бесконечное число решений (три плоскости пересекаются не в точке, а на прямой линии — рис. 7.25), отсутствие решения (три плоскости попарно пересекаются на двух разных параллельных прямых линиях — рис. 7.26) и единственное решение (три плоскости пересекаются в одной точке рис. 7.27). После такого разбора задачи с ее графической интерпретацией студенты сказали автору, что если бы им все это показали на занятиях по линейной алгебре, то они бы сдали экзамен намного лучше: не было бы простого зазубривания теорем, а было бы ясное понимание, если бы не сути задачи, то как минимум ее постановки.

Здесь тоже представляется весьма желательным и целесообразным решать задачи линейной алгебры на семинарах не «ручкой на бумаге», а «мышкой компьютера по его дисплею» или даже по большому экрану, висящему в аудитории, где читается курс линейной алгебры. Есть лазерные устройства, которые позволяют в аудитории над головами студентов построить плоскости, показанные на рис. 7.25, 7.26 и 7.27, и даже вращать их в пространстве.

Случай 3

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 4 \\ 3 & 2 & 1 \end{pmatrix} \quad B := \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \quad A1 := \mathrm{augment}(A, B) = \begin{pmatrix} 1 & 2 & 3 & 1 \\ 4 & 5 & 4 & 2 \\ 3 & 2 & 1 & 1 \end{pmatrix}$$

Ранг основной матрицы $\mathrm{rank}(A) = 3$

Ранг расширенной матрицы $\mathrm{rank}(A1) = 3$

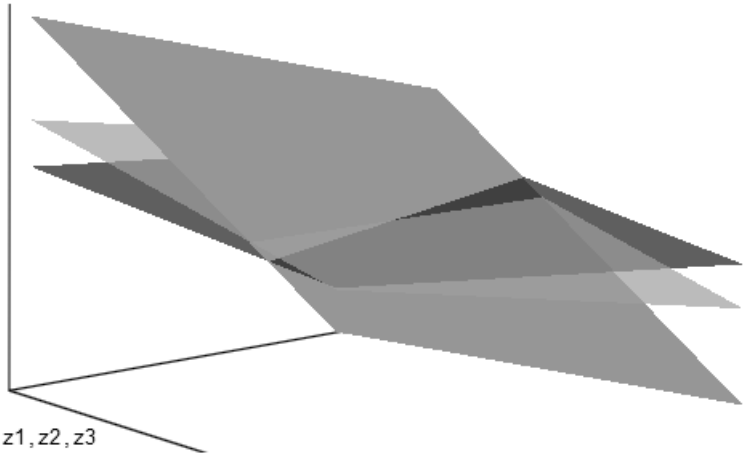
Число неизвестных $\mathrm{cols}(A) = 3$

Единственное решение

$\mathrm{rank}(A) = \mathrm{rank}(A1) = 1$

$\mathrm{rank}(A) = \mathrm{cols}(A) = 1$

Графическая иллюстрация - три плоскости пересекаются в одной точке



Решение

Первое приближение

Способ 1 $\quad x := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ Given $\quad A \cdot x = B \quad x := \mathrm{Find}(x) = \begin{pmatrix} 0.25 \\ 0 \\ 0.25 \end{pmatrix}$

Проверка

$A \cdot x = B = 1$

Способ 2 $\quad x := A^{-1} \cdot B = \begin{pmatrix} 0.25 \\ 0 \\ 0.25 \end{pmatrix}$

Способ 3 $\quad x := \mathrm{lsolve}(A, B) = \begin{pmatrix} 0.25 \\ 0 \\ 0.25 \end{pmatrix}$

Рис. 7.27

Решение СЛАУ в среде Mathcad — единственное решение

Комментарий электротехника

В классической электротехнике базовая модель электрической цепи представляется в виде системы уравнений для токов с использованием законов Кирхгофа следующим образом.

Сначала составляется схема замещения цепи. Для этого используются топологические понятия:

- ветвь — участок цепи с одним значением тока;
- узел — место соединения ветвей;
- контур — замкнутая петля, проходящая по узлам и включающая в себя ветви при последовательном обходе.

Пример на рисунке 7.28.

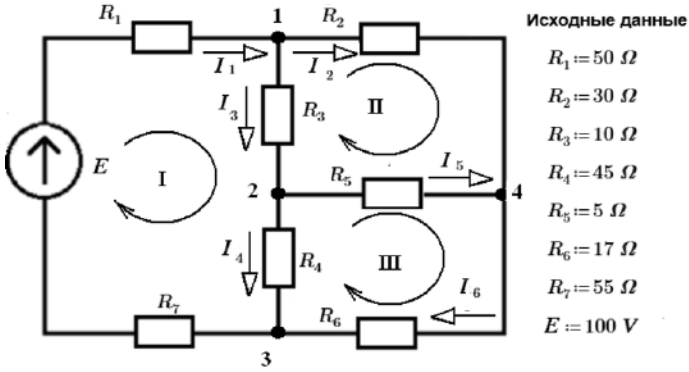


Рис. 7.28

Начало классического решения классической электротехнической задачи

В этой схеме:

- нумеруются элементы цепи (E, $R1 \ldots R7$);
- нумеруются ветви ($I1$, ..., $I6$) и помещаются стрелки условно-положительных направлений токов;
- нумеруются узлы **(1...4)**;
- нумеруются контуры **(I, II** и **III)**.

Далее определяются количество искомых токов `NX = nB`, количество узлов `nУ = 3`.

Потом составляется часть системы уравнений по первому закону Кирхгофа (`NУ` узловых уравнений). Для этого необходимо и достаточно выбрать `NУ = nУ - 1` узлов. При этом всегда получатся линейно независимые уравнения. К примеру `NУ = 3`. Для узлов **1...3** следует:

```
-I1 + I2 + I3 = 0,

-I3 + I5 + I4 = 0,

I1 — I4 - I6 = 0.
```

При записи уравнений можно использовать правило знаков как в теореме Остроградского — Гаусса (со знаком «−» токи, направленные к узлу), поскольку первый закон Кирхгофа является прямым следствием `div J = 0`[34]. Если после расчетов получится `I < 0`, то ток имеет другое направление.

Для остальных контурных уравнений `Nk = Nx - Ny` можно обеспечить линейную независимость, если каждый контур будет содержать хотя бы один элемент, не входящий в другие контуры. В примере выбраны контуры, содержащие меньшее количество элементов (`I, II, III`):

```
(R1 + R7) I1 + R3I3 + R4I4 = E,

R2I2 - R5I5 - R3I3 = 0,

R5I5 + R6I6 - R4I4 = 0.
```

Здесь для ЭДС E и токов I правило знаков определяет «+» при совпадении направления обхода контура и ЭДС или тока.

[34] Это дивергенция, которая вместе с ротором видится нерадивым студентам в страшных снах перед экзаменом по математике. Жалко, что мы в этой книге не уделили должного внимания этим важным понятиям математики.

Таким образом, проблема формирования системы уравнений для электрических цепей решается поэтапно с выбором узлов и контуров, обеспечивающих полноту системы уравнений и линейную независимость уравнений в системе.

Решение по вышеописанному подходу с использованием блока **Решить** пакета Mathcad Prime 3 показано на рис. 7.29.

Начальные приближения

$$[I_1\ I_2\ I_3\ I_4\ I_5\ I_6] := [0\ 0\ 0\ 0\ 0\ 0]\ A$$

Ограничения

$I_1 = I_2 + I_3$ Узел 1

$I_3 = I_5 + I_4$ Узел 2

$I_1 = I_4 + I_6$ Узел 3

$R_1 \cdot I_1 + R_3 \cdot I_3 + R_4 \cdot I_4 + R_7 \cdot I_1 = E$ Контур I

$R_2 \cdot I_2 - R_5 \cdot I_5 - R_3 \cdot I_3 = 0$ Контур II

$R_5 \cdot I_5 + R_6 \cdot I_6 - R_4 \cdot I_4 = 0$ Контур III

Решатель

$$\mathbf{Find}(I_1, I_2, I_3, I_4, I_5, I_6) = \begin{bmatrix} 0.795 \\ 0.238 \\ 0.557 \\ 0.243 \\ 0.314 \\ 0.552 \end{bmatrix} A$$

Рис. 7.29

«Электротехнический» расчет электрической цепи

Примечание. В программах схемотехнического моделирования матричный метод решения задачи анализа цепей не используется. Неоднородность матрицы коэффициентов системы затрудняет ее формирование из графического образа схемы. Применяют метод «узловых потенциалов». В этом методе искомыми величинами являются потенциалы узлов схемы. Одному узлу в схеме присваивается нулевое значение. Для него уравнение не составляется. Матрица коэффициентов содержит только междуузловые проводимости. Одновременно размерность системы уравнений значительно уменьшается. Так, в рассмотренном выше примере остаются только 3 уравнения. Токи ветвей вычисляется на втором этапе по найденным потенциалам узлов.

И последнее

С формулой, прописанной на арке в Сент-Луисе (рис. 7.1), не так все просто. Во-первых, эта арка — далеко не одномерная линия, а криволинейная полая балка с реальными довольно большими размерами в сечении (внутри арки люди могут подняться на ее вершину). Можно предположить, что эта формула относится к верхней или нижней образующей арки или к ее какой-то серединной линии. Во-вторых, из курса строительной механики [52] следует, что арочная конструкция имеет наибольшую прочность, если она выполнена в виде параболы, а не в виде цепной функции. Вернее, так. Если арка имеет нулевую толщину (перевернутая тонкая цепь), то тут более подходит цепная функция. Если же толщину арки увеличивать, т. е. сооружать реальную, а не виртуальную арку, то цепная линия должна постепенно переходить в параболу. Если же на эту арку насыпать грунт и проложить по этой конструкции прямую дорогу, построив арочный мост, например, то формула арки должна вернуться к цепной функции. Отсылаем читателя к видеокурсу профессора М. Н. Кирсанова [53], где все это довольно хорошо истол-

ковано. Мы же можем владельцам арки в Сент-Луисе дать еще одну хорошую идею: прочертить на арке с помощью, например, лазера параболу, подчеркивая тем самым сложность и красоту соответствующих расчетов. Хватило бы только толщины арки для этого шоу! На рисунке 7.30 показано начало этого расчета — вписывание в квадрат со стороной 630 футов цепной линии (y(x) — верхняя кривая) и параболы (p(x, a) — средняя кривая). Кроме того, нарисована еще одна цепная линия ($y_1(x)$ — нижняя кривая), у которой так вручную подобраны параметры, чтобы она касалась параболы в двух точках и имитировала нижнюю образующую арки в Сент-Луисе.

$$y(x) := 757.7' - 127.7'\cdot ch\left(\frac{x}{127.7'}\right) \quad y_1(x) := 715' - 108'\cdot ch\left(\frac{x}{107.7'}\right)$$

$$h := y(0m) = 630\cdot ft \qquad h = 192m$$

$$x_0 := root(y(x), x, 0m, 100m) = 315\cdot ft \qquad x_0 = 96m$$

$$L := 2\cdot x_0 = 630\cdot ft \qquad L = 192m$$

$$p(x, a) := h + a\cdot x^2$$

$$a := 0m = p(x_0, a) \text{ solve}, a \rightarrow -\frac{0.020833194002156725894}{m}$$

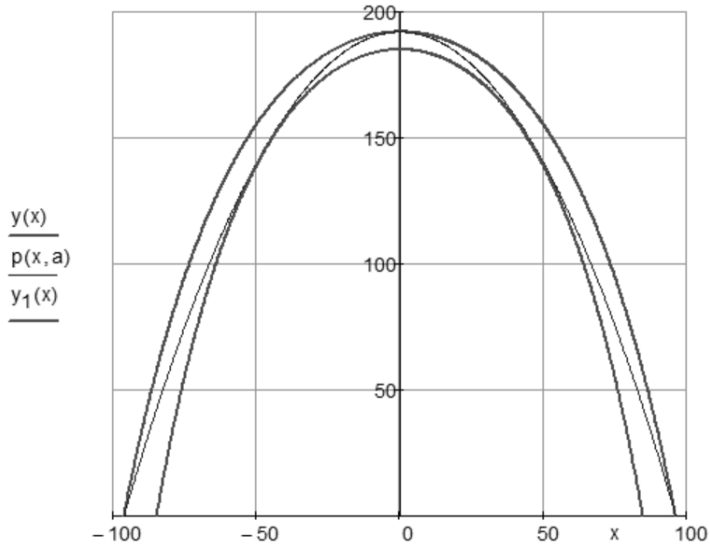


Рис. 7.30
Парабола, зажатая между двух цепных линий

Парабола, кстати, окажется короче цепной линии почти на 10 метров при той же высоте и размахе основания (квадрат со стороной в 630 футов — см. две верхние кривые на рис. 7.30). Строительного материала на сооружение параболической арки пойдет меньше, но будет ли она красивее арки в виде цепной линии? На этот вопрос простыми вычислениями не ответить. Тут нужна будет теория нечетких множеств, оперирующая лингвистическими переменными. Азы этой теории мы рассмотрим в этюде 12. Кроме того, нужно принимать во внимание, что в древности люди не знали ни экспоненты, ни цепной функции, но хорошо были знакомы со степенными функциями, с параболой, в частности. На рисунке 7.31 показано, как в среде Mathcad можно «помочь древним людям» — разложить

цепную функцию в ряд Тейлора (в степенной ряд), о котором мы упоминали ранее и который в нашем случае будет состоять из четных степеней 2, 4, 6, 8, 10 и т. д.

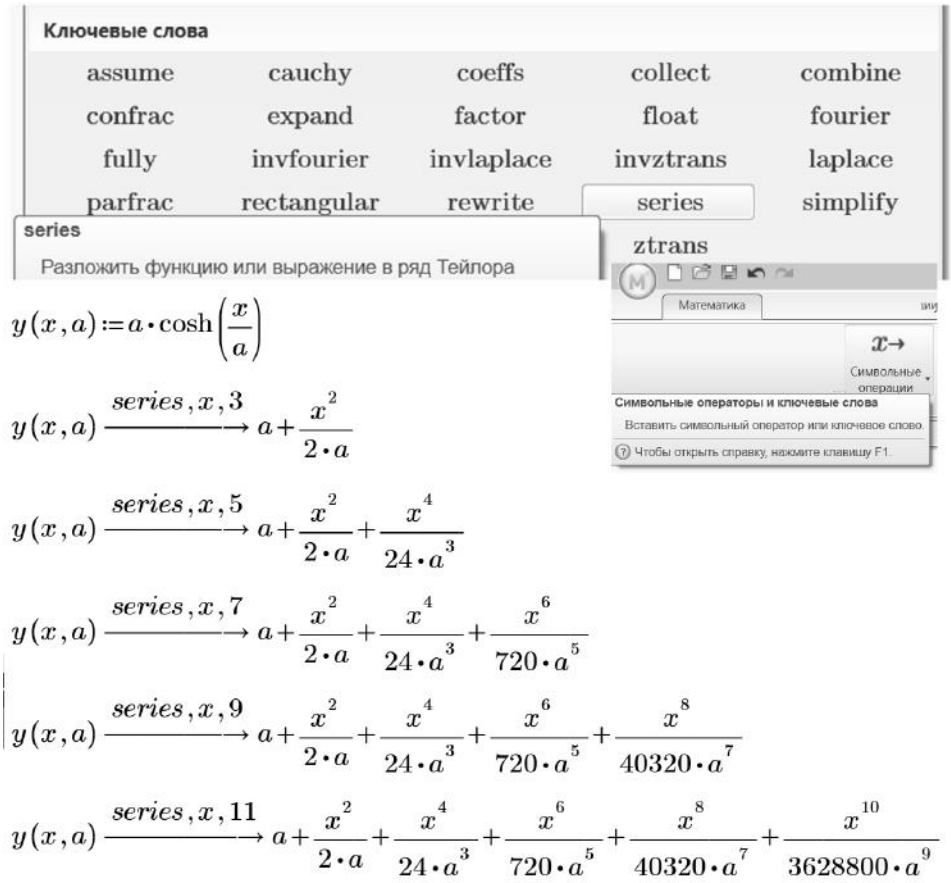


Рис. 7.31
Разложение цепной функции в ряд Тейлора в среде Mathcad

Несложно увидеть в числах 2, 24, 720, 40320, 362880 факториалы четных чисел 2, 4, 6, 8, 10 и вывести общую формулу разложения, включающего оператор суммы. Предлагаем это сделать читателям. Кстати, почти во всех справочных таблицах по ряду Тейлора на первом месте стоит именно экспонента, формирующая цепную линию. Одно из определений числа *e* ведется через сумму бесконечного ряда. Вот как в среде Mathcad вычислить это число — см. рис. 7.32.

Предлагаем также читателям создать такую анимацию: кадры анимации — это показатель степенной функции, которая приближается к заданной цепной функции по мере увеличения этого показателя степени. Эту анимацию (саму кривую и степенной ряд) можно проецировать у арки в Сент-Луисе, приобщая ее посетителей к математике. Еще одна задача: найти параметры квадратного уравнения, при котором оно наилучшим образом будет приближаться к уравнению цепной линии. Это можно сделать через табулирование двух функций и использования метода наименьших квадратов, описанного в этюде 14.

$$\sum_{n=0}^{\infty}\frac{1}{n!}\rightarrow e$$

$$\lim_{x\rightarrow\infty}\left(1+\frac{1}{x}\right)^{x}\rightarrow e$$

$$\int_{1}^{a}\frac{1}{t}\,\mathrm{d}t=1\xrightarrow{solve,a}e$$

$$\frac{\mathrm{d}}{\mathrm{d}t}a^{t}=a^{t}\xrightarrow{solve,a}\left|\begin{array}{l}\text{if } t\le 0\le \mathrm{Re}(t)\vee t\le 0\wedge \mathrm{Re}(t)<0\vee 0\le \mathrm{Re}(t)\wedge 0<t\vee \mathrm{Re}(t)<0<t\\ \quad e\\ \text{else}\\ \quad undefined\end{array}\right.$$

Рис. 7.32
Четыре способа вычислить в среде Mathcad число e

Читатель может зайти на авторский сайт по адресу **http://twtmas.mpei.ac.ru/mas/Worksheets/Symbolic_Series.mcd**, ввести свою функцию и посмотреть, как она раскладывается в степенной ряд вблизи заданной точки.

Раз мы упомянули ряд Тейлора (сказали А), то нужно вспомнить и ряды Фурье (сказать Б) — какой же матанализ может обойтись без двух этих рядов. Пакет Mathcad может также раскладывать функции и в тригонометрический ряд Фурье, члены которого содержат синусы и/или косинусы. Наш расчет электрической цепи (рис. 7.21–7.24, 7.28 и 7.29) подразумевал, что ток в цепи постоянный. Но в реальных бытовых и промышленных электрических цепях ток переменный и далеко не всегда отвечающий по форме именно синусоиде. Переменный ток генерируется в генераторах с вращающимся ротором. Таких генераторов с разными значениями частоты вращения, сдвига по фазе и амплитуды можно параллельно поставить несколько и генерировать переменный ток практически любой формы. В этом состоит основной фокус ряда Фурье, который демонстрировали многим поколениям студентов-электриков. На рисунке 7.33 показано, как установив в сети три (рис. 7.33*a*) или шесть генераторов (рис. 7.33*b*) со своими собственными синусоидами, можно вырабатывать электрический ток, форма которого будет приближается к пилообразной при увеличении числа генераторов синусоидального тока.

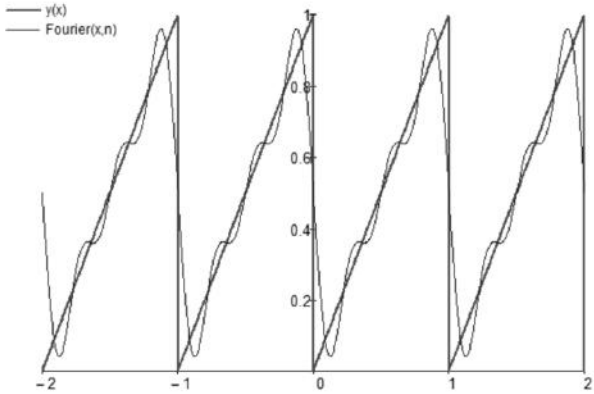
Читатель может зайти на авторский сайт по адресу **http://twt.mpei.ac.ru/MCS/Worksheets/Fourier.xmcd**, ввести свою функцию и посмотреть, какими гармониками она может быть описана. На сайте **https://www.ptcusercommunity.com/videos/1416** расположена авторская анимация гармонического ряда Фурье. Один из авторов книги в студенческие годы выполнял такую лабораторную работу по электротехнике. Или эта работа была фантазией автора — следствием изучения рядов Фурье на лекциях и семинарах по математике? Сейчас уже и не припомнишь! Суть работы. Берутся несколько генераторов синусоидального тока с разными частотами, амплитудами и сдвигами. Эти генераторы параллельно питают сеть, напряжение в которой анализируется осциллографом. Получается картинка, которую можно видеть на рис. 7.33. Сейчас такую лабораторную работу несложно сделать виртуальной на компьютере. Но лучше всего это сделать и реально и виртуально: снимать данные с осциллографа, подключенного к электрической сети и сравнивать их с данными, сгенерированными компьютером. Эта модель вполне реальна: наши электрические сети питают генераторы различных электростанций. Важная задача — синхронизация работы подключаемого генератора с электрической сетью.

$y(x) := x - \mathrm{floor}(x)$

$x_1 := -2$ $\quad x_2 := 2$ $\quad n := 14$ $\quad$ Recalculate

$i := 0..n$

$$b_i := \frac{2}{x_2 - x_1} \cdot \int_{x_1}^{x_2} y(t) \cdot \sin\left(\frac{2 \cdot \pi \cdot i}{x_2 - x_1} \cdot t\right) dt \qquad \mathrm{Fourier}(x) := \frac{1}{2} + \sum_{i=1}^{n} \left(b_i \cdot \sin\left(\frac{2 \cdot \pi \cdot i}{x_2 - x_1} \cdot x\right)\right)$$



$$b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -0.31831 \\ 0 \\ 0 \\ 0 \\ -0.15915 \\ 0 \\ 0 \\ 0 \\ -0.1061 \\ 0 \\ 0 \end{pmatrix}$$

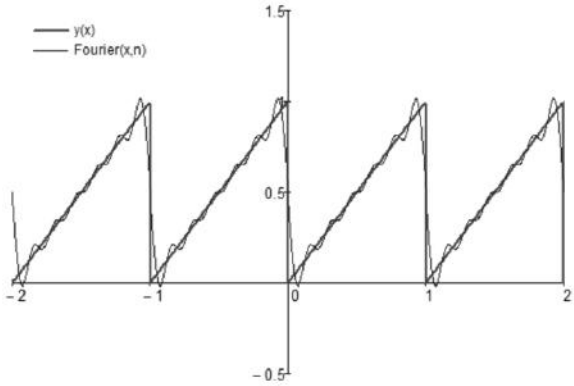
a

$y(x) := x - \mathrm{floor}(x)$

$x_1 := -2$ $\quad x_2 := 2$ $\quad n := 24$ $\quad$ Recalculate

$i := 0..n$

$$b_i := \frac{2}{x_2 - x_1} \cdot \int_{x_1}^{x_2} y(t) \cdot \sin\left(\frac{2 \cdot \pi \cdot i}{x_2 - x_1} \cdot t\right) dt \qquad \mathrm{Fourier}(x) := \frac{1}{2} + \sum_{i=1}^{n} \left(b_i \cdot \sin\left(\frac{2 \cdot \pi \cdot i}{x_2 - x_1} \cdot x\right)\right)$$



$$b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -0.31831 \\ 0 \\ 0 \\ 0 \\ -0.15915 \\ 0 \\ 0 \\ 0 \\ -0.1061 \\ 0 \\ 0 \\ 0 \\ -0.07958 \\ 0 \\ 0 \\ 0 \\ -0.06366 \\ 0 \\ 0 \\ 0 \\ -0.05305 \end{pmatrix}$$

b

Рис. 7.33
Разложение в ряд Фурье в среде Mathcad

ДИВЕРТИСМЕНТ 2. РЕАБИЛИТАЦИЯ ПАРАБОЛЫ

Рассматривая в этом этюде провисающую цепь, мы упомянули тот факт, что большинство людей, если их спросить о форме цепи, отвечают, что это парабола. А вот ситуация, когда ошибочно вместо параболы называют другую кривую.

Задача: по какой поверхности надо отшлифовать зеркало прожектора, чтобы все лучи, исходящие из источника света, помещенного в точке 0 на оси вращения, отражались бы зеркалом параллельно этой оси (рис. 7.34).

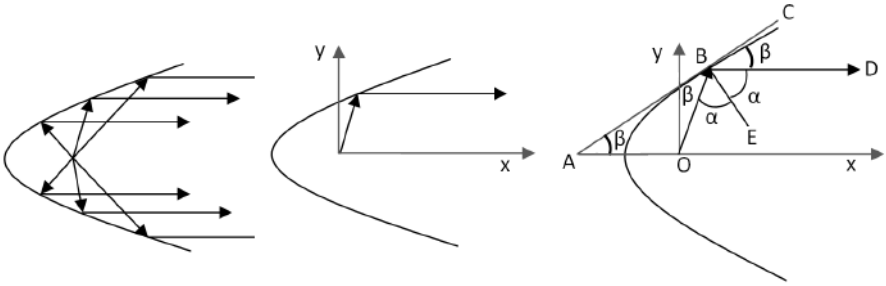


Рис. 7.34
Схема зеркала прожектора

Большинство людей тут отвечают, что эта поверхность должна быть гиперболоидом, опираясь не на физику задачи, а на... литературу и вспоминая знаменитый роман А. Н. Толстого «Гиперболоид инженера Гарина». На самом деле тут должен быть *параболоид*, а гиперболоид использован в романе из-за «звучности» этого слова: «Гипер!» Покажем это [82]. Правда, в последнее время многие тут вспоминают не только «лирику», но «физику» — *параболические* антенны для приема спутникового сигнала.

В приближении геометрической оптики, т. е. пренебрегая всеми волновыми световыми явлениями и заменяя световой поток большим количеством лучей, выведем форму отражающей поверхности зеркала, при которой лучи исходящие из одной точки после отражения от зеркала превратятся в параллельный поток лучей. Все лучи, исходящие из одной точки (из *фокуса*), после отражения преобразуются в параллельный поток лучей (левая схема на рис. 7.34). Рассмотрим один из таких лучей (средняя схема). Введем оси координат x, y. Установим, какую форму $y(x)$ должна иметь поверхность зеркала в указанных осях координат. При отражении от зеркала выполняется закон отражения (см. правую схему на рис. 7.34): углы $OBE = DBE = \alpha$. Поэтому углы $ABO = CBD = \beta$. Углы CBD и BAO равны как соответственные. Треугольник ABO равнобедренный и $|AO| = |OB|$. Поскольку точка B имеет координаты x и y, получаем $|AO| = y \cdot \mathrm{ctg}\,\beta - x = \dfrac{y}{\mathrm{tg}\,\beta} - x$, но $\mathrm{tg}\,\beta = \dfrac{dy}{dx}$, поэтому $|AO| = \dfrac{y}{\dfrac{dy}{dx}} - x$. С другой стороны, $|OB| = \sqrt{y^2 + x^2}$. Получаем дифференциальное уравнение: $\dfrac{y}{\dfrac{dy}{dx}} - x = \sqrt{y^2 + x^2}$. Введем дополнительное обозначение для значения x при y равном 0 $y(-F) = 0$. Решим полученное дифференциальное уравнение аналитически (рис. 7.35) и численно (рис. 7.36).

На графике рисунка 7.36 прорисованы две совпадающие кривые квадратного корня от x: наше зеркало — это параболоид вращения! Он образуется вращением параболы $2Fx = 2F^2 - y^2$ вокруг оси ox.

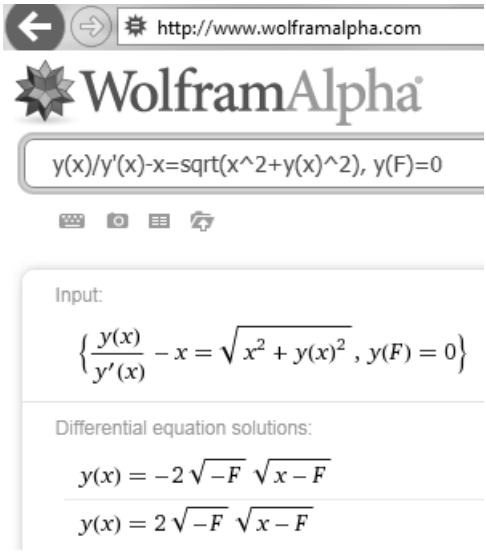


Рис. 7.35
Аналитическое решение уравнения прожектора

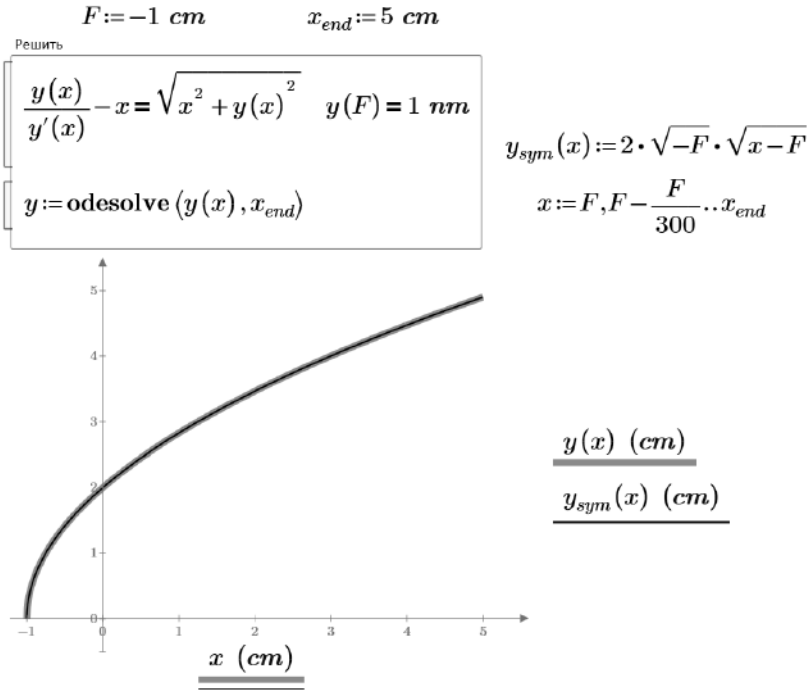


Рис. 7.36
Численное решение уравнения прожектора и сравнение его с аналитическим

Примечание. Начальным условием в задаче на рис. 7.36 стоит не `y(F) = 0`, а `y(F) = ...` одному нанометру. Если бы там стоял нуль, то задача не решалась бы числено из-за того, что производная в этой точке равна бесконечности. Пришлось чуть-чуть схитрить и обмануть компьютер. Можно сказать, что нанотехнологии помогли решить нам эту задачу!

А вот второе заблуждение, второй, так сказать, «оптический обман». Считается, что поверхности разного рода линз, фокусирующих лучи света, должны быть сделаны в виде части сферы. Так ли это, покажет следующая задача, сводящаяся также к решению дифференциального уравнения.

На рисунке 7.37 показана схема задачи о линзе, выполненной из материала с коэффициентом преломления `n`. Спрашивается, какой формы должна быть нижняя поверхность линзы (верхняя ее поверхность плоская), чтобы параллельный пучок света сошелся в фокусе, отстоящем от начала координат на расстоянии `F` (фокусное расстояние).

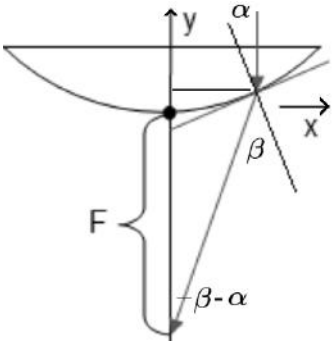


Рис. 7.37
Схема задачи о линзе

Мы намерено повернули луч света вниз, чтобы упростить задачу. И линзу для этого же перевернули выпуклой стороной вниз. Но обычно все оптические расчеты иллюстрируются схемами с горизонтальной оптической осью (см., например, рисунки 7.34 и 7.36). И это неспроста. Оптик-экспериментатор собирает оптическую систему на оптической скамье именно в горизонтальном положении, поэтому гораздо легче сопоставить модель и установку при таком изображении.

Решение данной задачи показано на рисунке 7.38. Она сводится к системе одного дифференциального уравнения (производная равна тангенсу угла наклона касательной) и двух алгебраических уравнений. Первое алгебраическое уравнение — это математическая запись закона Снелла (Снелля, Снеллиуса) с коэффициентом преломления. Второе выражает тангенс угла в виде отношения длины противолежащего катета `x` к длине прилежащего катета `F + y(x)` — см. треугольник на рис. 7.37. Так как в системе содержится одно дифференциальное уравнение, то при решении используется встроенная в Mathcad функция `Odesolve`.

На рисунке 7.39 кривая, являющаяся границей осевого сечения линзы и полученная при решении системы уравнений, дополнена пунктирной дугой окружности, проходящей через три точки: начало координат и два края линзы. Вывод: линза, которая должна сфокусировать параллельный пучок света в точке, не имеет сферическую форму. А какая это форма — этот вопрос мы задаем читателю!

Если при решении оптических задач допустить, что синус и тангенс угла равны самому углу при малом его значении (а такое допущение мы будем делать при рассмотрении математического маятника в этюде 9 — см. рис. 9.22 и 9.23 на стр. 223), то решение многих задач геометрической оптики существенно упрощается (линеаризируется) за счет, например, использования *матричных операторов*. Эти примеры мы поместили на сайте этюда.

$n := 2.417 \qquad F := 1\ cm \qquad x_{end} := 1\ cm$

Solve

$$y(0\ cm) = 0\ cm \qquad \alpha(0\ cm) = 0° \qquad \beta(0\ cm) = 0°$$

$$y'(x) = tan(\alpha(x)) \qquad n \cdot sin(\alpha(x)) = sin(\beta(x))$$

$$tan(\beta(x) - \alpha(x)) = \frac{x}{F + y(x)}$$

$$\begin{bmatrix} y \\ \alpha \\ \beta \end{bmatrix} := \mathrm{Odesolve}\left(\begin{bmatrix} y(x) \\ \alpha(x) \\ \beta(x) \end{bmatrix}, x_{end}\right)$$

$$x := -x_{end}, -x_{end} + \frac{x_{end}}{1000} .. x_{end}$$

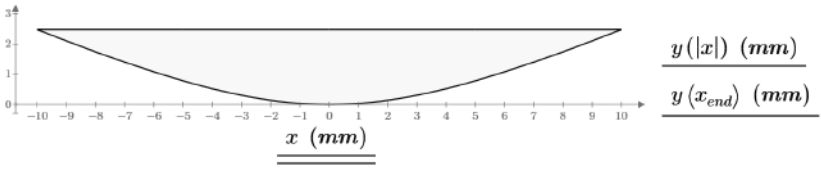


Рис. 7.38
Решение задачи о линзе

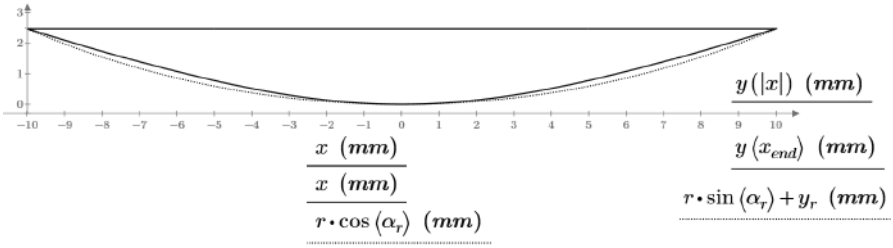


Рис. 7.39
Сферическая (пунктир) и «правильная» линза

Если наше тригонометрическое упрощение приложить к одному из основных оптических законов — *закону Снелла* (закону преломления света), то можно сказать, что в этом случае произведение показателя преломления первой среды на угол падения равно произведению показателя преломления второй среды на угол преломления.

Кроме тригонометрического в наших расчетах присутствовало и другое важное допущение: коэффициент преломления n света является константой и не зависит ни от длины волны луча света, ни от интенсивности светового потока, ни от положения луча в веществе. Но это далеко не так. Вспомним стеклянную призму, которая расщепляет белый свет на цветовые составляющие и помогает, например, определять состав вещества (спектроскопия), находить, например, гелий на Солнце. Или распространенные сувениры, где изображение создается лазером внутри остающейся целой стеклянной фигурки; так проявляется самофокусировка, когда под воздействием луча меняется показатель преломления среды, заставляя луч сжиматься в микроскопическую точку, при этом в веществе происходит микровзрыв, оставляющий белый след внутри стекла. Или известное явление миража, вызванное тем, что в неравномерно прогретом воздухе показатель преломления меняется с высотой и лучи света изгибаются над нагретой поверхностью песка или асфальта, создавая причудливые изображения предметов, находящихся вдали.

Современные компьютерные средства позволяют отказаться от многих допущений и более точно рассчитывать оптические приборы.

«Оптическая» добавка в данный этюд была сделана вместе с А. В. Соколовым — учителем физики московского лицея 1502 при Московском энергетическом институте, о котором будет рассказано в конце этюда 8. Кроме того, задачу о линзе помогли решить форумчане сайта PTC Community — см. **https://www.ptcusercommunity.com/message/423750**. Без оптики наши физико-математические этюды были бы неполными. Не хватает тут еще и этюдов, связанных с теплотой: с термодинамикой и теплопередачей. Но этому важному разделу физики посвящена отдельная «этюдная» книга одного из авторов [2]. Нет также этюдов по ядерной физике, теории остойчивости судов [85]... Но нельзя объять необъятное. Ни по физике, ни по математике, ни по информатике... Но попытка такого «объятия» будет предпринята в очередных изданиях книги. Вернее, так. Мы ожидаем новых решений от читателей, в том числе и по новым разделам физики, химии, математики, которые они могут размещать сами на сайте книги. Наиболее интересные решения будут переходить в бумажную книгу, если она, конечно, выживет — см. Предисловие.

Задания читателям

1. Вывести с помощью пакета Mathcad или Интернета формулу для цепной функции, полученную как результат решения дифференциального уравнения баланса сил, действующих на элементарный элемент (звено) цепи.

2. Решить задачу о форме цепи, подвешенной на двух опорах, к которой подвесили *два* или даже *три* груза.

3. Решить задачу о форме цепи, которая подвешена на двух опорах и которую снизу подпирают подвижным или неподвижным цилиндром (катком). Цепь катится по этому катку в определенное равновесное состояние. При неподвижном катке нужно учитывать сопротивление трения каната о цилиндр по формуле Эйлера, в котором присутствует экспонента.

4. Определить провисание цепи, составленной из двух цепей с различной линейной массой. Пример такого расчета можно видеть на рисунке 7.40. Что нужно записать в скрытой области?

5. Ювелирную цепочку соединяют в замкнутую цепь и растягивают ее в горизонтальных направлениях, ухватившись за две точки, — пытаются надеть на шею, например. Определить точки захвата (угол сектора цепочки, выложенной на столе в окружность), при которых... Сформулируйте сами ограничения в этой задаче. Одно из возможных ограничений см. в конце этюда 4 (минимум усилия на разрыв цепи в точках крепления).

6. Рассчитать провисание горизонтальной бечевы (веревочки), которую каменщики натягивают при кладке кирпичной стены для контроля ровности укладки кирпичей (вариант задачи 3).

7. Ввести в вышеприведенные расчеты *растяжение* цепи, приняв, что удлинение элементарного участка пропорционально приложенной силе (закон Гука). Коэффициент пропорциональности может быть довольно большим — подвесить на двух столбах можно, например, резинку и пружину. На авторском сайте **https://www.ptcusercommunity.com/videos/1357**, кстати, можно видеть траектории движения маятника в трех случаях: колебание на нерастяжимой нити без учета сил трения, колебание на растяжимой нити без учета сил трения и колебание на растяжимой нити с учетом силы трения.

8. Найти форму цепи (каната), подвешенной с одного конца и обдуваемой (омываемой) потоком воздуха (воды): летит, например, вертолет, а с него свисает фал для подвеса груза; или плывет корабль с болтающимся под водой обрывком якорной цепи. Силы трения каната или цепи о воздух или воду (силы, отклоняющие цепи или канат от вертикали) можно смоделировать, приняв допущения и

ограничения, описанные в этюде 9 (прыжки парашютиста и дайвера). Кроме того, нужно будет учитывать эффект планирования цепи или каната в обтекающей среде.

H_1 (m)	H_2 (m)	L (m)	S_1 (m)	S_2 (m)	g_{c1} $\left(\frac{kg}{m}\right)$	g_{c2} $\left(\frac{kg}{m}\right)$
7	12	12	8	12	10	3

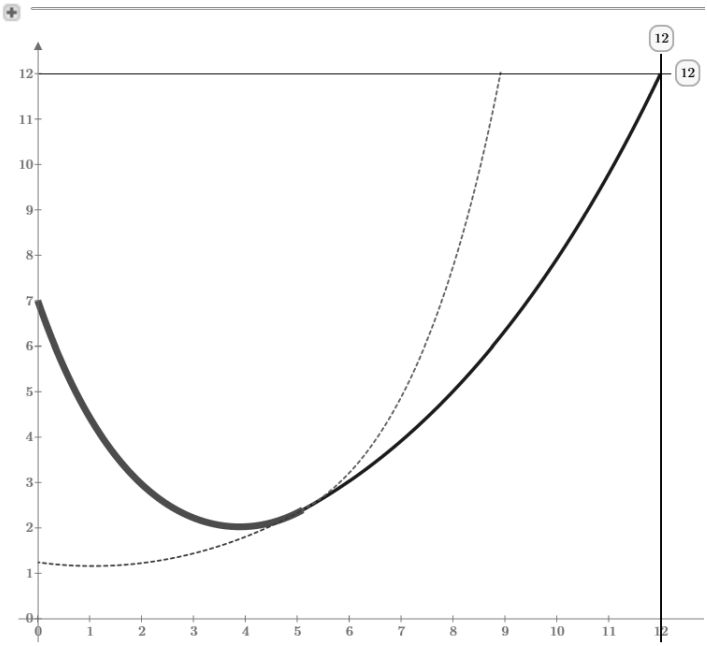


Рис. 7.40
Провисание составной цепи

9. Нарисовать в среде Mathcad помещение (для хранения велосипедов с квадратными колесами в городе Сент-Луисе) в виде цилиндра-стен и катеноиды-крыши. Параметры крыши при этом должны быть оптимальными в смысле ее нагрузки на стены.

10. Математический маятник, как известно, — это тонкая нерастяжимая и невесомая нить, к которой подвешен точечный груз. А давайте эту нить сделаем весомой, уберем подвешенный груз и, забегая вперед к этюду 9, смоделируем колебание такого маятника — обрыв цепи (череду связанных маятников).

11. Попытаться решить задачу об оптимальной длине цепи подвесной канатной дороги (см. рис. 7.7), при которой максимально возможная сила разрыва цепи будет минимальна.

12. Цепь подвешена на очень высоких башнях, расположенных друг от друга довольно далеко. На таких высоких, что приходится учитывать зависимость ускорения свободного падения от высоты и непараллельность векторов силы тяжести. Вывести формулу цепной линии для такого общего случая. Кстати, на авторском сайте **http://twt.mpei.ac.ru/MCS/Worksheets/g-h-psi.xmcd** выложен интерактивный расчет значения ускорения свободного падения от высоты над уровнем океана и географической широты местности.

13. Имеется гладкий жёсткий конус с площадью боковой поверхности $S = \pi l^2/3$, где l — образующая конуса. Из гибкой нерастяжимой нити длиной $l_0 = 34$ см $< l$ сделали кольцо, и одели его на конус. Затем к одной из точек кольца приложили силу F, направленную вдоль образующей конуса в сторону, противоположную вершине. На каком минимальном расстоянии x от вершины конуса будет проходить нить, когда она полностью натянется?

14. Как будет распространятся свет из точки, если зеркало сделать не в виде параболоида (см. рис. 7.36), а в виде... катеноиды.

15. Составьте и решите систему дифференциально-алгебраических уравнений для линзы с выпуклой стороной не вниз (см. рис. 7.37), а вверх.

16. «Доработать» гиперболоид инженера Гарина. Рассчитать схему «параболоида инженера Гарина»: выяснить какую форму должно иметь второе дополнительного зеркало и где оно должно быть расположено.

17. Сделать такую простую в реализации, но интересную лабораторную работу. Подвешивается в двух точках цепь (однородная (см. рис. 7.4 на стр. 143) или состоящая из частей с разной линейной массой — см. рис. 7.40), к которой крепится точечный груз (см. рис. 7.7 на стр. 146). Все это фотографируется на камеру, а файл снимка провисающей цепи передается на компьютер, где автоматически или вручную «оцифровывается»: генерируются два вектора значений абсцисс и ординат отдельных точек (звеньев) цепи. Затем создается математическая модель: система дифференциальных и алгебраических уравнений, описывающих провисание цепи, которая решается на компьютере численно и/или аналитически. Завершается работа сравнением данных, снятых с реальной цепи, и данных, полученных при реализации ее математической модели на компьютере: на экране дисплея ставятся точки по двум векторам, полученным со снимка, и прорисовывается кривая, полученная в результате решения системы уравнений. В завершении такой лабораторной работы ведется статистическая обработка решения, даются оценки погрешностей и прочее, и прочее... Тут можно перейти к эластичным и прочностным характеристикам материала цепи или каната (сталь, пластик), порассуждать об инженерных аспектах и красоте висячих мостов, воздушных линий электропередач, подвесных канатных дорог...

18. Проверить правильность решения задачи о провисании каната с грузом (см. рис. 7.10, стр. 148) можно на такой модели цепной функции. Цепь — это связанные материальные точки, располагающиеся в пространстве так, чтобы их потенциальная энергия была минимальна. Такой подход к анализу цепной функции (решение задачи оптимизации с ограничениями) будет изложен в этюде 21.

Этюд 8

ЖИВЫЕ КИНЕМАТИЧЕСКИЕ СХЕМЫ, ИЛИ РУКА ГЕЛЛЫ

В этюде будет рассказано, как можно создавать анимационные клипы на основе математических моделей различных механизмов и о том, как использовать сайт владельца пакета Mathcad для учебной работы.

Математика: уравнение окружности, уравнение прямой, система линейных алгебраических уравнений.

Физика: паровая машина, дворники автомобиля, кулисный механизм, стопоходящая машина, амортизатор.

Информатика: компьютерные инструменты аналитических преобразований, переменная диапазона, анимация, приближенное решение системы уравнений, сочетание символьного и численного решения радач, форум пользователей Mathcad, форум факультатива школьников.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127822**

Английский кузнец Томас Ньюкомен в 1705 г. построил паровую машину, которая использовалась для откачки воды из шахт[1]. Это была так называемая паровая машина атмосферного типа. В вертикальный цилиндр подавался насыщенный водяной пар из котла. Когда поршень достигал верхней предельной точки, клапан подачи пара перекрывался и открывался другой клапан, подающий холодную воду в цилиндр. Пар конденсировался, и поршень давал обратный ход. Далее воду из поршня сливали, подавали свежий пар из котла и все повторялось. На сайте **https://www.ptcusercommunity.com/docs/DOC-6643** помещена авторская анимация этой паровой машины, два кадра которой показаны на рис. 8.1.

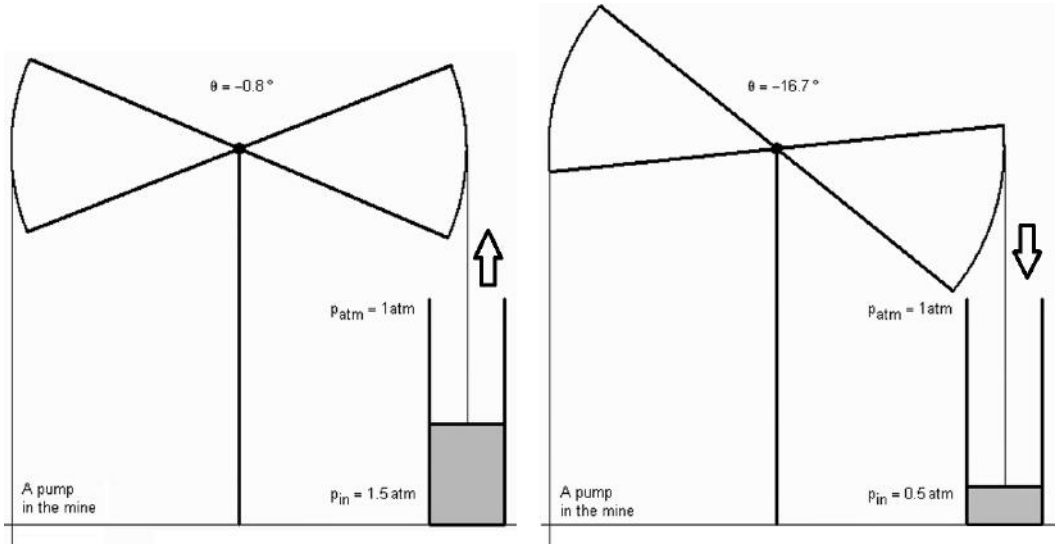


Рис. 8.1
Два кадра анимации атмосферной паровой машины

[1] Вода будет скапливаться и в прямолинейном туннеле для гравитационного поезда, о котором будет рассказано в этюде 9, и ее тоже нужно будет откачивать.

С этой машиной связана одна легенда. Мальчик Хэмфри Поттер в 1712 г. устроился учеником оператора такого парового насоса. Для осуществления рабочего цикла он должен был открывать клапан подачи пара в нижнем положении поршня и клапан подачи воды при полном его поднятии. Но этот сообразительный мальчик придумал, как избавить себя от монотонной работы. Он с помощью палочек и веревочек соединил ручки кранов и шток паровой машины. Получилось некое подобие кулисного механизма. Теперь кранами управлял сам поршень, а мальчик мог играть с товарищами во дворе. Так был изобретен *золотник* — устройство автоматического парораспределения паровой машины[2].

Паровую машину «в железе» сейчас можно увидеть только в музее. В Интернете размещено много анимаций паровой машины самой по себе или установленной на паровозе или пароходе. Создать анимацию паровой машины вполне реально с помощью математической программы Mathcad. Об этом пойдет речь в данном этюде.

На рис. 8.2 показан Mathcad-документ, рисующий, а затем анимирующий горизонтальную паровую машину, вернее, ее центральную часть, состоящую (см. слева направо) из маховика с радиусом R, шатуна (стержень 1–2) длиной l_1, штока (стержень 2–3) длиной l_2, цилиндра и поршня.

В начале расчета с помощью оператора присваивания (: =) задаются исходные данные: радиус маховика R, радиус окружности r, по которой будет вращаться левый конец шатуна (точка 1), длина шатуна l_1, длина штока l_2 и диаметр поршня D. После этого устанавливается угол поворота маховика θ. На рис. 8.2 эта величина задается двумя операторами θ : = FRAME° и θ : = 40°. При отладке программы будет работать второй оператор, который позволит пользователю вручную менять угол поворота маховика в диапазоне от 0° до 360° и видеть, как меняется взаиморасположение частей паровой машины. При создании анимации (об этом будет подробно сказано ниже) оператор θ : = 40° нужно будет отключить или стереть совсем, чтобы заработал первый оператор θ : = FRAME°. Отключается оператор с помощью специального переключателя, появляющегося после нажатия правой кнопки мыши, когда курсор находится на данном операторе. У выключенного из расчета оператора справа появляется метка — черный квадратик с правой стороны. Системная переменная FRAME по умолчанию равна нулю, следовательно, угол θ будет иметь нулевое значение. Но при создании анимации пакет Mathcad будет сам менять значение переменной FRAME (см. ниже), устанавливая разный угол поворота маховика: 0°, 1°, 2° и т. д. до 360°.

У нашей паровой машины четыре узловые точки, которые мы пронумеровали числами 0, 1, 2 и 3, первая из которых неподвижна. У точек 2 и 3 будут неизменными вертикальные координаты. В нулевую точку мы поместим начало координат: y_0: = 0m x_0: = 0m. (Переменные y_0 и x_0 — это не нулевые значения векторов y и x, а переменные с нижним индексом — переменные с именами y0 и x0, где нули (индексы) опущены чуть вниз. В среде Mathcad нижние индексы переменных вводятся с помощью точки в имени переменной — y.0 и x.0. При уводе курсора от такого имени переменной точка пропадает, а индекс смещается вниз.) Система координат у нас будет декартова. Но некоторые другие механические схемы (модель колебания маятника, например [54]) лучше «помещать» в полярную систему координат, которая также поддерживается в среде Mathcad. Правильный выбор начала и системы координат может существенно упростить решение задачи. Также очень важно определить узловые точки кинематической схемы и их нумерацию. Наша задача плоская, но в среде Mathcad есть и трехмерная графика, что позволяет реализовывать и объемные кинематические схемы с третьей координатой z.

[2] Золотник — это еще и старая русская единица массы — см. рис. 6.5 и 6.7 на стр. 125 и 127.

Простейшая паровая машина

Геометрия паровой машины (см. ниже)

$R := 1500mm$ $r := 700mm$ $l_1 := 2300mm$ $l_2 := 1600mm$ $D := 500mm$

Угол, на который повернулся маховик $\theta := FRAME°$ $\theta := 40°$

Решение

Точка 0 $y_0 := 0m$ $x_0 := 0m$

Точка 1 $y_1 := r\sin(\theta)$ $x_1 := r\cos(\theta)$

Точка 2 $y_2 := 0m$

$l_1 := l_1$ $y_1 := y_1$ $x_1 := x_1$ $y_2 := y_2$

$$l_1^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2 \text{ solve}, x_2 \rightarrow \begin{pmatrix} x_1 + \sqrt{l_1 + y_1 - y_2}\sqrt{l_1 - y_1 + y_2} \\ x_1 - \sqrt{l_1 + y_1 - y_2}\sqrt{l_1 - y_1 + y_2} \end{pmatrix}$$

$$x_2 := x_1 + \sqrt{l_1 + y_1 - y_2}\sqrt{l_1 - y_1 + y_2}$$

Точка 3: $y_3 := 0m$ $x_3 := x_2 + l_2$

Паровой цилиндр $\Delta := 50mm$

$$xc := \begin{pmatrix} r + l_1 + l_2 + \Delta \\ r + l_1 + l_2 + \Delta \\ -r + l_1 + l_2 - \Delta \\ -r + l_1 + l_2 - \Delta \\ r + l_1 + l_2 + \Delta \end{pmatrix} \qquad yc := \begin{pmatrix} \frac{D}{2} + \Delta \\ -\frac{D}{2} - \Delta \\ -\frac{D}{2} - \Delta \\ \frac{D}{2} + \Delta \\ \frac{D}{2} + \Delta \end{pmatrix}$$

$\theta 1 := 0°, 1° .. 360°$

Решение

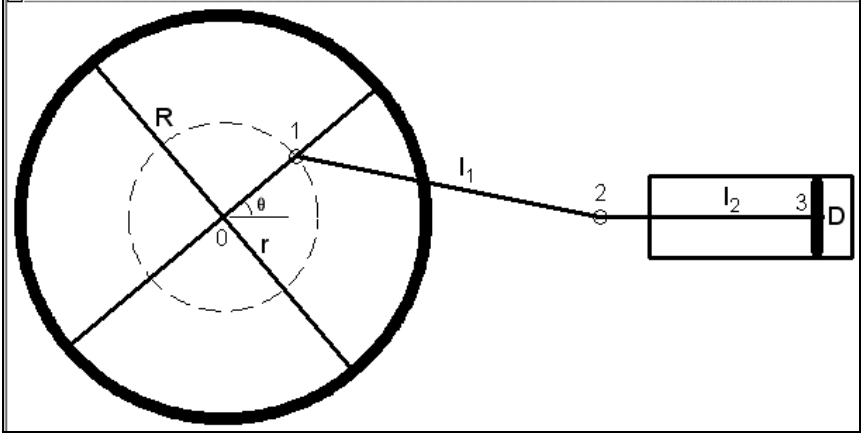


Рис. 8.2

Mathcad-документ, рисующий паровую машину

Чтобы получить изображение паровой машины, нужно рассчитать координаты точек 1, 2 и 3 в зависимости от значения угла θ. Это делается в области Mathcad-документа (рис. 8.2) с именем Решение. Операторы, задающие координаты точек, заключены в *область* для того, чтобы ее можно было сворачивать и видеть рисунок паровой машины сразу после операторов задания угла поворота маховика θ. Это упрощает процесс отладки программы: не нужно будет менять значение θ, а потом прокручивать экран дисплея по вертикали, чтобы увидеть рисунок. Свертывается область командой из списка, выпадающего при нажатии правой кнопки

мыши. Вставляется область в расчет командой из меню Вставка. Область можно также защитить (с паролем или без пароля) от редактирования.

Более-менее сложный расчет потребуется провести при определении ординаты точки 2 — значения переменной x_2: аналитически (символьно) решается уравнение длины шатуна. Это нелинейное уравнение, вытекающее из теоремы Пифагора, имеет два решения (корня) — шатун с заданными длиной l_1 и координатами точки крепления на маховике (x_1 и y_1) может находиться слева или справа от маховика. Оператор символьной математики solve, x_2 возвращает два решения в виде вектора. Первый элемент этого вектора мы возьмем в качестве решения нашей задачи — координаты x_2. Операторы $l_1 := l_1$, $y_1 := y_1$, $x_1 := x_1$ и $y_2 := y_2$, записанные перед оператором solve, x_2, «глушат» численные значения переменных l_1, y_1, x_1 и y_2 для того, чтобы оператор solve, x_2 вернул выражения, а не численные значения выражений. Операторы символьной математики (на рис. 8.2 они прописаны на сером фоне) после их использования можно убрать из расчета. Но мы их оставили, чтобы пояснить его ход. По правде говоря, уравнение положения штока можно решить и в уме без привлечения компьютерной математики аналитических (символьных) преобразований. Но мы эти инструменты задействовали для того, чтобы показать, как символьная математика помогает решать задачи в среде пакета Mathcad.

Паровой цилиндр — это прямоугольник, который мы будем рисовать по его четырем вершинам. Векторы xc и yc будут хранить координаты этих вершин. Переменная Δ — это значение «зазора» для цилиндра: на величину Δ цилиндр будет больше крайних точек хода поршня, а диаметр цилиндра будет больше диаметра поршня на величину 2Δ.

На рисунке 8.3 зафиксирован процесс вставки, заполнения и форматирования декартового графика, с помощью которого рисуется паровая машина. У графика семь кривых (а всего их может быть шестнадцать), описание которых (семь пар выражений) представлено на осях графика и перечислено в окне форматирования графика.

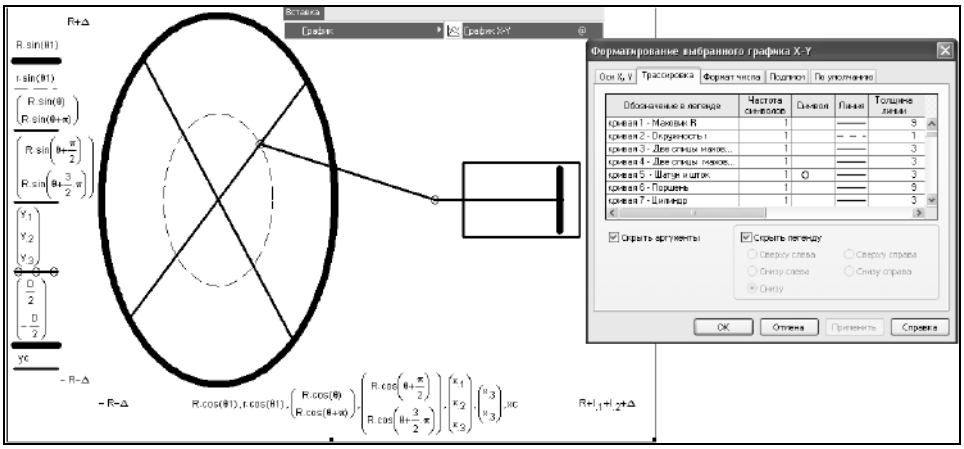


Рис. 8.3

Форматирование графика — схемы паровой машины

Кривая 1. Окружность маховика паровой машины рисуется так. Задается переменная θ1, хранящая не скалярное значение, а дискретные значения угла от 0° до 360° с шагом 1° (см. оператор в нижней части области Решение на рис. 8.2). На графике (рис. 8.3) строится так называемая параметрическая кривая с выражениями R·sin(θ1) на оси Y и R·cos(θ1) на оси X, что и позволяет нарисовать окружность (на рисунке это пока эллипс) с радиусом R. Если переменная θ1 будет хра-

нить дискретные данные не по всей окружности, а только по ее части ($\theta 1 := 40°$, $41° \ldots 90°$, например), то будет нарисована не вся окружность, а только ее часть — дуга. Такие пунктирные дуги можно видеть на рис. 8.6 ниже.

Кривая 2. Окружность движения точки крепления шатуна на маховике (точка 1) строится так же, как и окружность маховика с той лишь разницей, что эта окружность меньшего размера и прорисована не сплошной толстой линией, а тонким пунктиром. Центры наших двух окружностей с радиусами R и r находятся в начале координат. Но если к выражениям $R \cdot \sin(\theta 1)$ и $R \cdot \cos(\theta 1)$ прибавить константы, то окружность будет рисоваться со смещением относительно начала координат. Если же к выражениям $R \cdot \sin(\theta 1)$ и $R \cdot \cos(\theta 1)$ прибавлять не константы, а величины, зависящие от переменной θ (переменная анимации), то окружность будет двигаться. Так, например, можно анимировать катящееся колесо.

Кривые 3–7. Спицы маховика паровой машины рисуются для того, чтобы «закрепить» шатун на маховике и показать вращение маховика при анимации. Мы ограничились четырьмя спицами (двумя перекрещенными диаметрами), но спиц можно нарисовать и больше. В этюде 7 анимированное колесо велосипеда имеет 8 спиц. Спицы, как и другие отрезки прямых, а также ломаные линии, состоящие из отрезков прямых, рисуются заданием пары векторов, хранящих координаты точек: две точки для спиц и поршня, три точки для шатуна и штока поршня (точки 1, 2 и 3) и пять точек для парового цилиндра. Формирование пары векторов для цилиндра мы перенесли из области графика в область Решение, чтобы не загромождать график.

Размер графика нужно установить таким, чтобы маховик выглядел круглым (см. рис. 8.2; на рис. 8.3, повторяем, маховик выглядит овальным из-за того, что пропорции графика несколько искажаются при его форматировании). Изменение размера графика ведется протяжкой мышкой его рамки. Кроме того, при создании анимации нужно зафиксировать разброс графика по его осям. По умолчанию пакет Mathcad сам устанавливает этот разброс, ориентируясь на максимальные и минимальные значения величин по осям абсцисс и ординат, которые могут меняться. Но в процессе создания анимации этот разброс должен быть зафиксирован: на концах графика мы записали $-R - \Delta$ и $R + \Delta$ (ось X) и $-R - \Delta$ и $R + l_1 + l_2 + \Delta$ (ось Y), где Δ — это некий зазор не только для цилиндра паровой машины (см. выше), но и для графика. Если разброс графика не зафиксировать, то анимация будет «кривая»: окружность будет деформироваться в эллипс и т. д.

До эры кино и телевидения была популярна такая игрушка — блокнот (флипбук) с нарисованным на разных страницах человечком в разных позах. Если блокнот просто перелистывать, то движения человечка (анимации) не увидеть. Но если страницы блокнота отогнуть и быстро отпускать по одной, то возникнет анимация — нарисованный человечек начнет двигаться[3]. То же самое можно наблюдать и с нашим расчетом геометрических параметров паровой машины. Если вручную менять значение переменной θ и свернуть область Решение, то можно видеть, как меняется положение поршня, штока, шатуна и маховика (это делается, повторяем, при отладке программы), но анимации не получится. Анимация возникнет, когда скорость смены кадров будет, по крайней мере, равна десяти кадрам в секунду. «Перелистывать» наш «блокнот» в среде Mathcad с изображениями паровой машины в разных фазах движения призваны специальные *инструменты создания* анимации — см. рис. 8.4.

[3] Про эту игрушку мы еще расскажем в этюде 13.

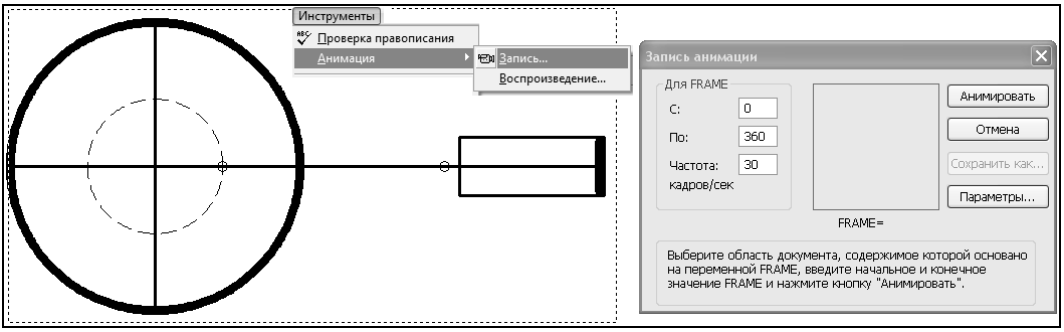


Рис. 8.4
Создание анимации паровой машины

Для создания анимации нужно, во-первых, передать управление независимой переменной анимации (у нас это переменная θ — угол поворота маховика) системной переменной FRAME (счетчик кадров анимации). Это делается, повторяем, отключением оператора θ : = 40°. Во-вторых, необходимо вызвать диалоговое окно создания анимации (рис. 8.4) и установить в этом окне нужные параметры анимации, а в-третьих, протяжкой мышкой определить область анимации. На рис. 8.4 эта область отмечена пунктиром. При создании анимации она будет отображаться в окне Запись анимации. Теперь после нажатия кнопки Анимировать (см. рис. 8.4) пакет Mathcad будет сам через системную переменную FRAME менять значение угла поворота маховика паровой машины θ и запоминать кадры анимации. По завершению создания анимации появится окно с анимацией — см. рис. 8.5. Эту анимацию можно запускать в среде Mathcad, а можно сохранить на диске в виде avi-файла и запускать уже без Mathcad в среде какого-либо проигрывателя (плеера) анимации Windows. Все! Паровая машина создана, а мы себя почувствовали неким «создателем», «творцом», который может вдохнуть жизнь в неподвижные виртуальные объекты.

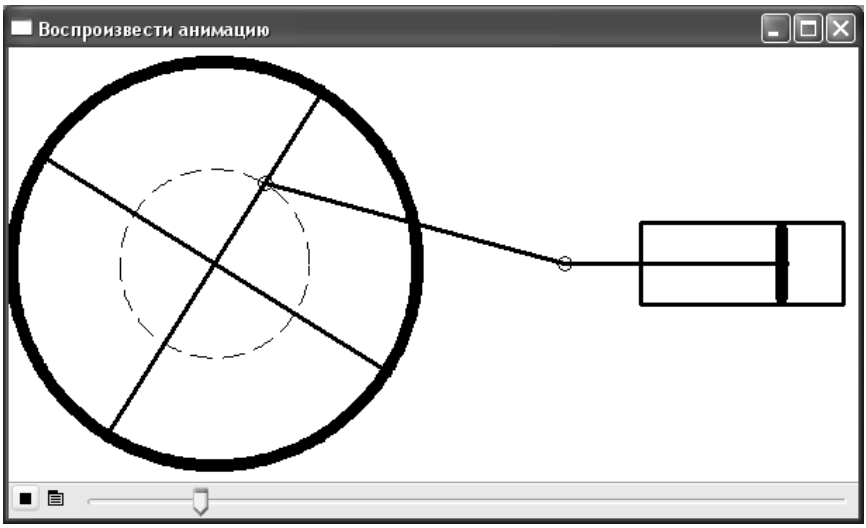


Рис. 8.5
Анимации паровой машины

Читатель может усложнить паровую машину — добавить к ней уже упоминавшийся легендарный золотник, который «мал да дорог»[4]. Но если это покажется слишком сложным, то можно сначала потренироваться на другой кинематической задаче — на задаче о движении дворников (стеклоочистителей) автомобиля. Паровую машину мы, повторяем, можем видеть только в музее или Интернете, но с автомобильными дворниками мы сталкиваемся чуть ли не каждый день.

Дворники автомобиля также окружены легендами. Рассказывают, как один дирижер во время дождя возвращался после концерта на такси домой. Шоферу приходилось часто останавливать машину, чтобы протереть ветровое стекло. А дирижер в это время еще «жил» своим концертом, отбивал в уме музыкальные такты и видел перед собой «метроном» — руку, смахивающую со стекла дождевые капли. На следующий день настоящий, а не воображаемый метроном с прикрепленной к нему щеткой был установлен на капоте машины. Так, говорят, были изобретены автомобильные дворники.

В эпоху паровых машин стояла задача преобразования поступательного движения во вращательное. Решая задачу о паровой машине (рис. 8.2–8.5) мы немного схитрили. Меняя угол поворота маховика, преобразовывали не поступательное движение поршня во вращательное движение маховика, а преобразовывали вращательное движение маховика в поступательное движение поршня. Мы фактически реализовали не паровую машину, а поршневой компрессор, приводимый в движение электромотором или двигателем внутреннего сгорания. Тем не менее, анимация у нас получилась, и выяснять, «кто кого движет», не так уж важно.

В настоящее время в связи с повсеместным распространением электромоторов с ротором и статором более актуальной стала проблема передачи вращательного движения в поступательное. Дворники автомобиля — это типичный пример такого преобразователя.

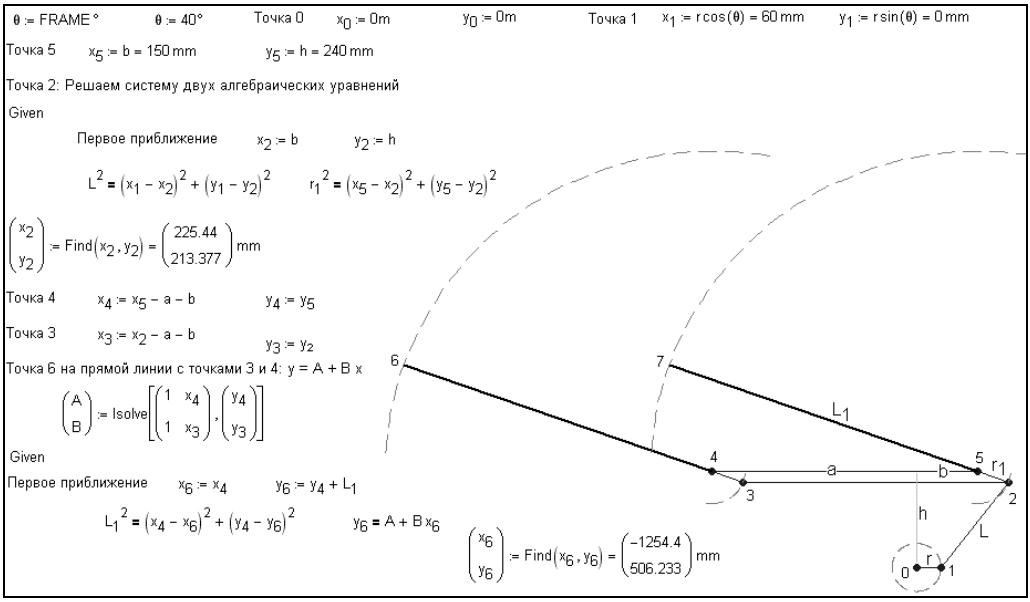


Рис. 8.6
Задача о дворниках автомобиля

[4] Не совсем ясна этимология этой поговорки. Золотник — это и парораспределительное устройство паровой машины, которое мало по сравнению с паровым цилиндром, например. Но золотник — это и, повторяем, старинная русская мера веса. Копейка рубль бережет, а золотник — пуд!

На рисунке 8.6 помещен расчет с помощью пакета Mathcad 15 узловых точек кинематической схемы автомобильных дворников (параллелограмм, одну из вершин которого «шевелит» кривошип) в зависимости от угла поворота электродвигателя. Сердцевиной расчета является определение координат точки 2. Здесь уже нужно будет решать не одно уравнение, как это было в случае с паровой машиной (рис. 8.2), а *систему* двух нелинейных алгебраических уравнений. И решать эту систему мы будем не аналитически (символьно; можно попытаться решить и аналитически, но решение будет уж очень громоздким), а численно через блок Given-Find (Дано-Найти), встроенный в Mathcad (см. рис. 8.6). Численное решение системы уравнений требует первого приближения к решению, которое задается операторами $x_2 := b$ и $y_2 := h$. Тут очень важно задать правильное первое приближение, чтобы блок Given-Find выдавал «правильное» решение во всем диапазоне изменения угла θ от 0° до 360°. Ведь опять же «шатун» дворника (линия 1–2) при ошибочном первом приближении может оказаться на «ошибочной» стороне от точки 5. В этом случае движение автомобильных дворников может сопровождаться «судорогами» — шатун будет появляться то справа, то слева от точки крепления. В принципе первое приближение к решению должно быть не константой, а переменной величиной, зависящей от переменной анимации θ.

Точка 6 лежит на прямой линии с точками 3 и 4. Тут приходится сначала с помощью функции lsolve решать систему двух линейных алгебраических уравнений

$$y_3 = A + B \cdot x_3$$

$$y_4 = A + B \cdot x_4$$

и определять значения коэффициентов A и B[5], а затем решить еще одну систему двух уравнений, одно из которых нелинейно, и находить координаты точки 6. Когда координаты всех точек определены, легко создать анимацию движения дворников автомобиля по методике, описанной выше для паровой машины — см. рис. 8.3 и 8.4.

В конце XIX в. на Всемирной выставке в Париже золотую медаль получила стопоходящая машина нашего соотечественника математика П. Л. Чебышёва[6]. В этой машине вращательное движение превращалось в движение, похожее на движение ноги при ходьбе по ровной поверхности.

На рисунке 8.7 показана кинематическая схема одной «ноги» стопоходящей машины Чебышёва, а также траектория движения точки 5 и распечатка векторов x и y — координат шести узловых точек, две из которых (1 и 2) неподвижны относительно начала координат, расположенной во второй точке. Нумерация точек у нас опять же начинается с нуля и идет в таком порядке, чтобы «ногу» стопоходящей машины Чебышёва можно было нарисовать, не отрывая карандаша от бумаги и не проводя одну линию дважды. Это позволяет рисовать на графике ломаную кривую с помощью пары векторов x и y. Отрезок прямой 2–3 совершает круговое движение вокруг точки 2. Координаты точки 0 можно найти, решив систему двух алгебраических уравнений, как мы это делали в задаче о дворниках автомобиля (рис. 8.6). Предлагаем читателю самому написать Mathcad-документ создания анимации движения «ноги Чебышёва» по методикам, описанным выше. Расстояния между точками мы не указываем — их можно просто измерить, прикладывая линейку к рис. 8.7. Наиболее сложно будет заставить Mathcad рисовать траекторию движения точки 5 (или точки 4 — линия 4–5 строго вертикальна). Есть раз-

[5] Эту задачу в этюде 2 (см. рис. 2.4 на стр. 44) мы решали в среде Mathcad Prime, где полностью работает механизм размерных величин и массивы могут хранить разноразмерные величины — безразмерную величину и длину. На рис. 8.6 в задаче о дворниках приходилось отключать инструменты размерных величин. Этот нюанс мы еще обсудим в этюде 9 на примере решения дифференциальных уравнений.

[6] В среде Mathcad, кстати, есть встроенная функция, носящая имя этого знаменитого математика — Tcheb. Чебышёв, кстати, родился недалеко от дачи одного из авторов книги — см. стр. 380.

ные способы рисования этой замкнутой пунктирной кривой. Можно, например, вручную («рабоче-крестьянский подход») протабулировать эту «замкнутую» функцию: изменять значения угла θ от 0° до 360° с шагом 10°, например, записывать полученные значения координат точки в векторы с 37 элементами, а затем отобразить эти два вектора на графике, соединив их пунктиром[7]. Точка 4 движется синхронно с точкой 5, точка 0 описывает дугу окружности, точка 3 совершает круговое движение, а точки 1 и 2 неподвижны. В данной задаче главное так подобрать длины рычагов, чтобы траектория движения точек 4 и 5 на некотором участке была наиболее близка к горизонтальной линии, а вся стопоходящая машина (а у нее две пары таких ног, движущихся со сдвигом на 180°) передвигалась без ощутимых вертикальных колебаний.

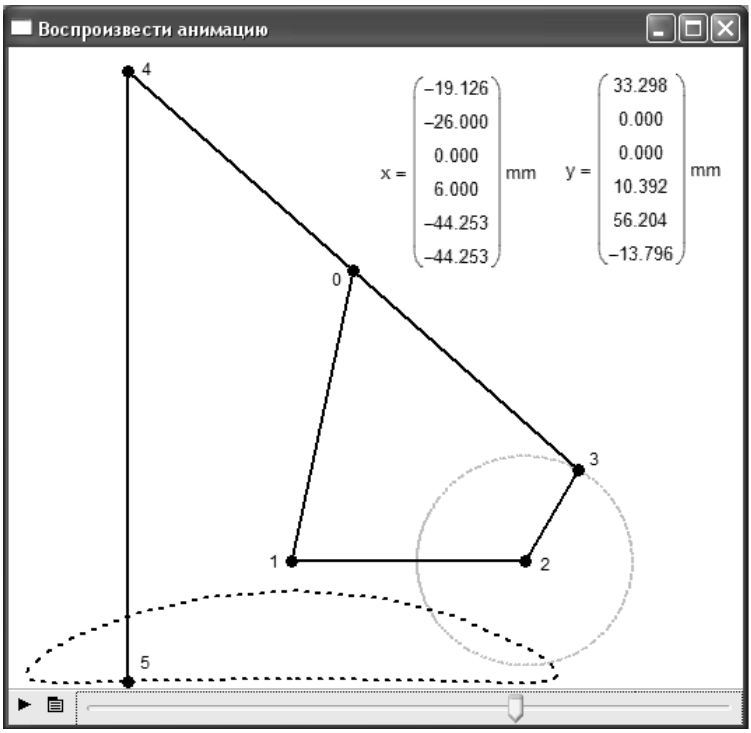


Рис. 8.7
Стопоходящая машина Чебышёва

У Чебышёва было много последователей в плане создания стопоходящих машин, переводящих вращательное движение в плавное горизонтальное. Наиболее знаменитым из них является датчанин Тео Янсен (Theo Jansen), который сконструировал и построил машину, схема которой показана на рис. 8.8. В этой машине так же можно менять длины между точками (суставами «ноги»), чтобы перемещение опорной точки «стопы» было по возможности горизонтальным. У стопоходящей машины Янсена только точки 0 и 1 неподвижны, остальные находятся либо в круговом, либо в более сложном (см. пунктирные линии на рис. 8.8) движении. У Чебышёва было также много и предшественников в плане создания стопоходящих машин, но П. Л. Чебышёв был первым, кто при конст-

[7] «Интеллигентский подход» состоит в том, чтобы на каждом шаге решения задачи формировать функцию пользователя, по которой и строится замкнутая пунктирная кривая траектории движения точки. Такой подход реализован в авторской анимации паровой машины с качающимся цилиндром — см. **https://www.ptcusercommunity.com/docs/DOC-6642**.

руировании таких механизмов стал использовать сложный математический аппарат. До Чебышёва такие устройства делали по наитию — подгоняя длину стержней. На рис. 8.7 показано устройство, «рисующее» некий сплюснутый с одного бока овал. Но математики доказали, что можно создать более сложное подобное механическое устройство, которое, например, может воспроизвести подпись любого человека...

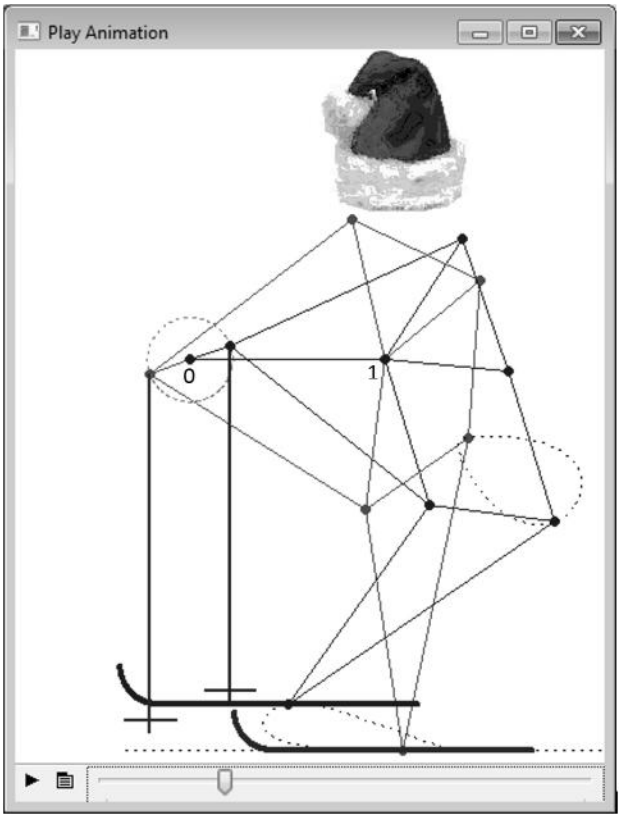


Рис. 8.8
Стопоходящая машина Янсена

На рисунке 8.8 показаны две «ноги» Янсена, одна из которых двигается с запаздыванием на 180°. Тео Янсен построил из дерева такую машину с десятком ног, которые приводятся в движение ветряным двигателем и «со скрипом», но двигаются по пляжу. Это видео можно найти на Youtube. Мы к этой анимации приделали шапочку, лыжи и лыжные палки. Получилась неплохая анимированная новогодняя открытка.

Данные и другие кинематические схемы собраны на сайте **http://communities.ptc.com/groups/kinematic-models-in-mathcad**.

Несколько слов об этом сайте. Этот сайт создан и поддерживается компанией PTC (www.ptc.com) — владельцем пакета Mathcad. Часть этого сайта отдана сообществу (communities) пользователей пакета Mathcad для того, чтобы они могли общаться между собой, задавать вопросы, вести блоги, обмениваться полезной информацией и т. д. На этом сайте можно создавать так называемые группы (groups) по отдельным темам. Одна из таких «групп» имеет название, совпадающее с названием данного этюда. С этого же сайта можно скачать бесплатную пробную версию пакета Mathcad.

Кроме того, как сказано в Предисловии, на сайте этого сообщества открыт сайт данной книги — **https:// www.ptcusercommunity.com/groups/etudes**. Мы еще вернёмся к этому сайту в конце этюда.

Читатель может сам по ключевым словам «паровая машина», «стопоходящая машина Чебышёва», «Theo Jansen» и др. найти в Интернете анимации, описанные в этюде. Но одно дело искать и запускать анимации Интернета, а другое дело создавать их самому в виртуальном или реальном виде!

При создании анимации в среде Mathcad с его богатым математическим аппаратом можно решать не только алгебраические уравнения (об этом было рассказано в этюде 4), но и *дифференциальные уравнения и системы* (этюд 9). В этом случае допустимо учитывать не только *кинематику* моделей, но и их *динамику*. Узловые точки таких моделей будут двигаться не только вследствие жестких связей, показанных на рисунках этюда, но и под действием силы тяжести, силы инерции и других реальных физических сил. Так, в статье [54] и в этюде 9 описано, как можно моделировать и анимировать в среде Mathcad движение планет и их спутников под действием силы инерции и силы всемирного тяготения, а в статье [55] — движение маятника под действием той же силы инерции, силы земного притяжения, силы натяжения нити (жесткой или эластичной) и силы сопротивления вязкой среды, где маятник качается.

Еще одно полезное применение анимации — создание клипов, поясняющих суть решаемых задач, — см. рис. 10.7 на стр. 247.

ДИВЕРТИСМЕНТ 1. РУКА ГЕЛЛЫ

«Та заспешила, всунула рыжую голову в форточку,
вытянула сколько могла руку, ногтями начала царапать
нижний шпингалет и потрясать раму. Рука ее стала удлиняться,
как резиновая, и покрылась трупной зеленью.
Наконец зеленые пальцы мертвой обхватили головку шпингалета,
повернули ее, и рама стала открываться».

М. А. Булгаков «Мастер и Маргарита»

На рисунке 8.9 показан кадр анимации так называемого правила (ударение на втором слоге) Чебышёва — устройства с жесткими связями, превращающего круговое движение точки 1 в почти прямолинейное колебательное движение точки 4 (см. анимацию этого устройства здесь **https://www.ptcusercommunity.com/thread/60298**). При реализации этой модели в среде Mathcad 15 используется встроенная функция `Find` для решения систем нелинейных алгебраических уравнений, описывающих связи в механизме. Длины стержней нужно подбирать или рассчитывать так, чтобы наблюдалась максимальная прямолинейность движения точки 4. Если при каком-то угле поворота рычага 0–1 длины стержня 2–3 не будет хватать, то реальный механизм застопорится, а в математической модели функция `Find` вернет сообщение об ошибке в том плане, что нет решения. Но если функцию `Find` заменить на функцию `MinErr`, то застопоривания в механизме и ошибки в расчете не будет, а будет наблюдаться... удлинение стержня 2–3 (с 30 до 35.8 см), как... руки Геллы из булгаковского романа (см. рис. 8.10 и эпиграф к дивертисменту). Про функцию `MinErr` рассказано в этюде 4 (рис. 4.20 на стр. 77).

Удлинение или укорачивание стержня в кинематической схеме (использование «мягкой», а не жесткой связи) может существенно сгладить ход тех же стопоходящих механизмов, которым не всегда приходится ходить по сугубо прямолинейной поверхности. Это, кстати, уже давно делается в «самобеглых колясках» — автомобилях и мотоциклах, снабженных рессорами, пружинами и амортизаторами.

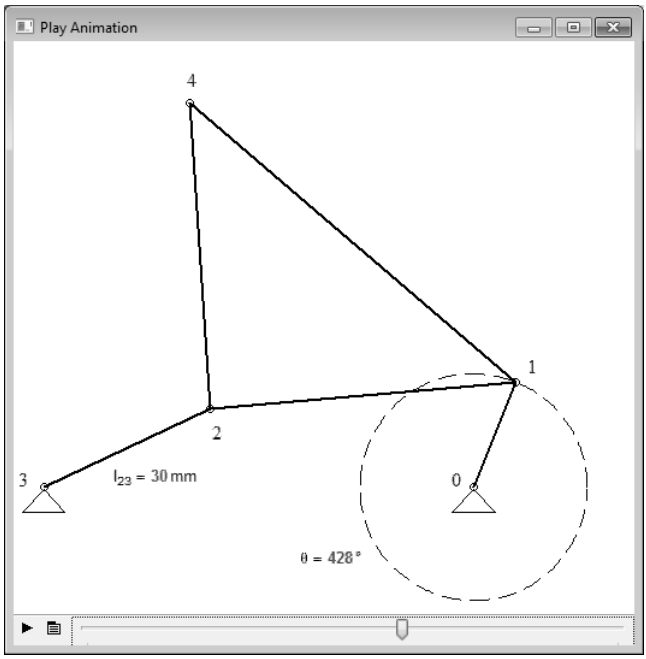


Рис. 8.9
Нормальная длина стержня 2–3 правила Чебышёва

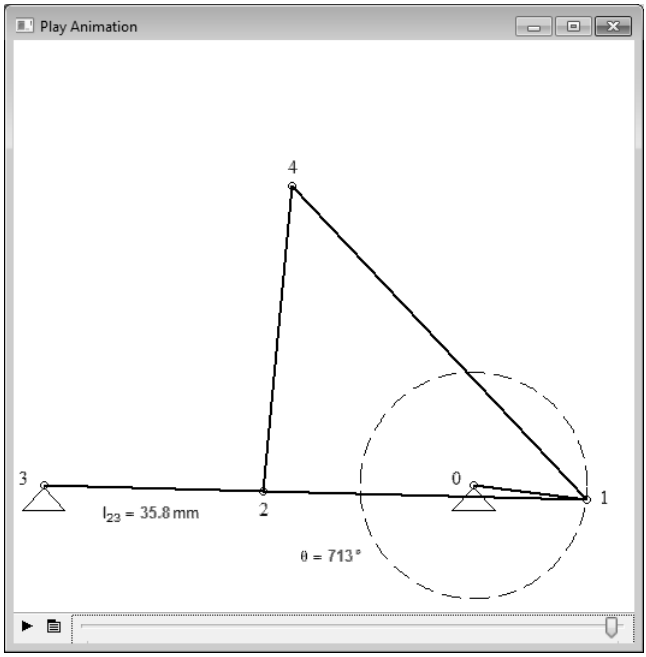


Рис. 8.10
Удлинение стержня 2–3 правила Чебышёва

ДИВЕРТИСМЕНТ 2. ДВЕРЬ В МИР МАТЕМАТИКИ И ФИЗИКИ

Через анимацию легко заинтересовать школьников математикой. Ведь создание мало-мальски сложной реальной анимации требует знания азов геометрии и алгебры. Поэтому-то один из авторов этой книги, когда ему предложили вести занятия со школьниками в одном продвинутом московском лицее, начал *факультатив* по применению математических пакетов при решении задач математики и физики с... создания анимации. Но сначала был создан сайт факультатива — см. рис. 8.11.

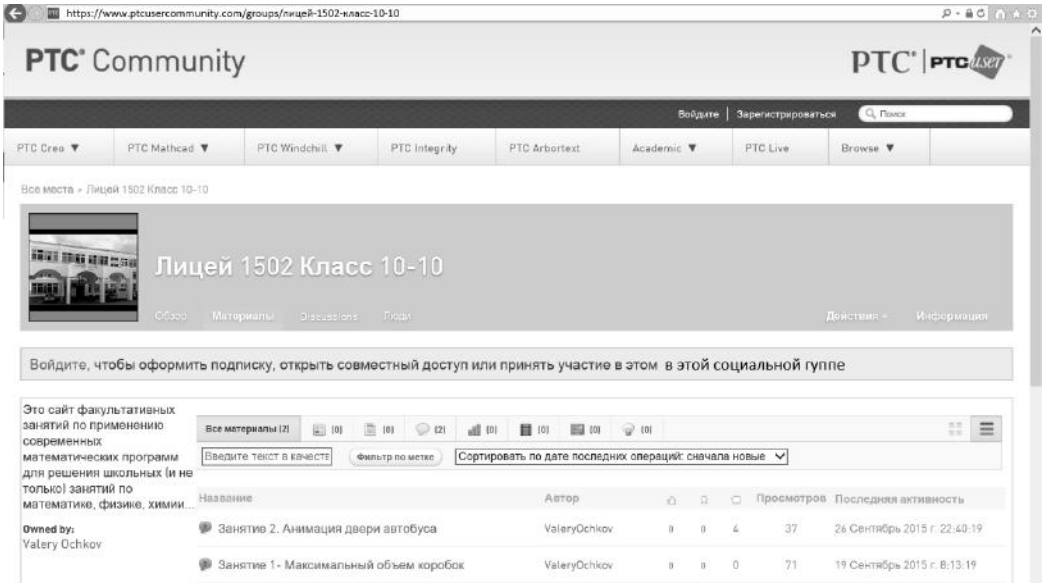


Рис. 8.11
Сайт факультативных занятий

Мы уже не раз упоминали в книге о PTC Community — о сообществе людей, использующих программы компании PTC, в частности, Mathcad в работе, при обучении, для развлечения. Так вот, на этом сайте можно создать группу (Group), в которой могут работать все или особо приглашенные и/или напросившиеся люди. В этой группе можно размещать все материалы факультатива: тексты, рисунки, анимации, Mathcad-документы, фотографии, голосования[8] и др., дополнять их комментариями, заданиями, обсуждениями, новыми более интересными и оптимальными решениями. Два первых занятия факультатива (см. рис. 8.11) касались анимации — первое косвенно (создавалась анимация складывания коробки из квадратной заготовки, поясняющая суть задачи, — см. рис. 10.7 на стр. 247), а второе — непосредственно: создавалась анимация движения автобусной двери при ее открытии.

Коробки мы складываем нечасто, а вот в общественном транспорте ездим почти каждый день и видим, как поворачивающийся стержень разворачивает рычаг, который открывает дверь автобуса (трамвая, троллейбуса), один край которой скользит в прямоугольном пазе — см. два фото на рис. 8.12. В старых автобусах две половинки двери состояли из своих собственных двух половинок, которые

[8] Например, по вопросу кто лучший ученик в классе. Совпадут ли тут мнения учителя и учеников?!

складывались гармошкой и открывали дверь. Конструкция была ненадежной[9] и часто ломалась. В современных же автобусах все сделано иначе.

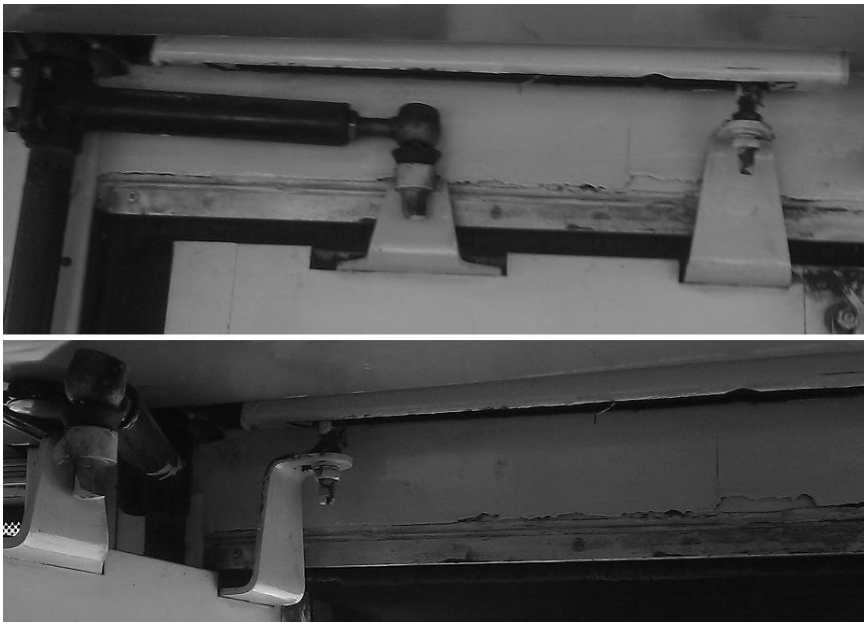


Рис. 8.12
Фото автобусной двери — закрытой и открытой

Начинают создание подобных анимаций с эскиза — с рисования рычагов и ввода обозначения — см. схему на рис. 8.13.

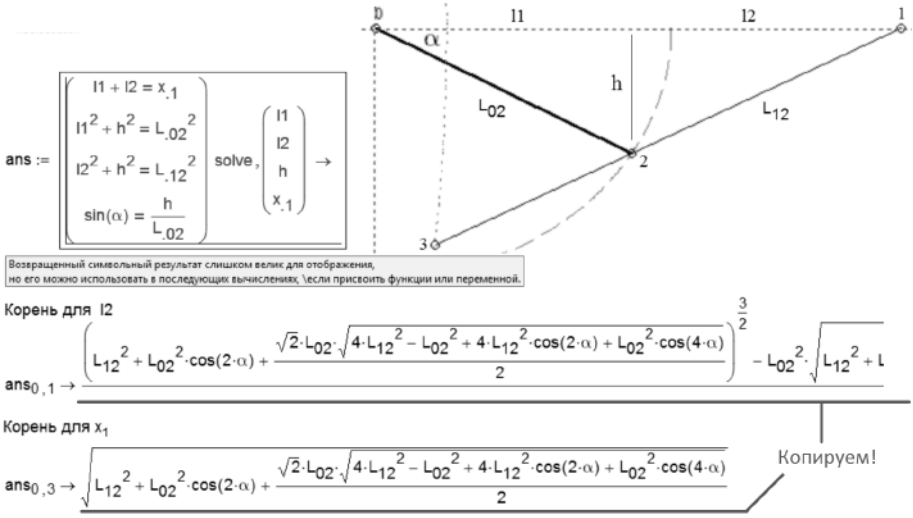


Рис. 8.13

Схема автобусной двери (ее вид сверху) сначала рисовалась от руки на бумаге (вернее, на экране планшета), но когда анимация была готова, один ее кадр был

[9] «…не дослушав полностью ответ девочек и получив портфелем с коньками по голове, мужчина сошел через закрытую дверь [автобуса]». Михаил Жванецкий.

доработан, дополнен линиями и обозначениями и вставлен в расчет, показанный на рис. 8.13. «Математика»: задачи тут сводится к решению системы четырех[10] алгебраических уравнений, основанных на простой «геометрии» задачи: треугольник, образуемый дверью автобуса (линия 1–2), открывающим рычагом (0–2) и пазом для двери (0–1), разбивается на два прямоугольных треугольника, к которым можно применить теорему Пифагора и то положение, что синус угла — это отношение длины противолежащего катета (h) к гипотенузе (L_{02} — длина рычага).

Систему уравнений мы решили аналитически, но пакет Mathcad 15 ответ не выдал по причине его больших размеров. Об этой особенности мы писали в этюде 4 (см. рис. 4.3 на стр. 65). Но мы смогли-таки выудить решения системы, выдав «на печать» отдельные элементы матрицы ans. Эти выражения будут использованы при формировании функции пользователя с именами xx_1 и ll_1 с аргументом α — углом поворота рычага автобусной двери — см. рис. 8.14.

Решение задачи

$$\alpha := \frac{\alpha_{end}}{n} FRAME = -30°$$

$$\alpha_{01} := 0, \frac{\alpha_{end}}{300} .. \alpha_{end}$$

$$x_2 := L_{02}\cos(\alpha) = 21.651\,cm \qquad y_2 := L_{02}\sin(\alpha) = -12.5\,cm \qquad y_1 := 0m$$

$$xx_1(\alpha) := \sqrt{L_{12}^2 + L_{02}^2\cos(2\alpha) + \frac{\sqrt{2}L_{02}\sqrt{4L_{12}^2 - L_{02}^2 + 4L_{12}^2\cos(2\alpha) + L_{02}^2\cos(4\alpha)}}{2}}$$

$$x_1 := xx_1(\alpha) = 44.449\,cm$$

$$ll2(\alpha) := \frac{\left(L_{12}^2 + L_{02}^2\cos(2\alpha) + \frac{\sqrt{2}L_{02}\sqrt{4L_{12}^2 - L_{02}^2 + 4L_{12}^2\cos(2\alpha) + L_{02}^2\cos(4\alpha)}}{2}\right)^{\frac{3}{2}} - L_{02}^2\sqrt{l}}{}$$

$$l2 := ll2(\alpha) = 22.798\,cm \qquad yy_3(\alpha) := L_{02}\sin(\alpha)\frac{L_{13}}{L_{12}} \qquad xx_3(\alpha) := xx_1(\alpha) - ll2(\alpha)\frac{L_{13}}{L_{12}}$$

$$y_3 := yy_3(\alpha) = -21.635\,cm \qquad x_3 := xx_3(\alpha) = 4.991\,cm$$

Решение задачи

Рис. 8.14
Формирование координат автобусной двери

Расчеты на рисунке 8.14 как, впрочем, и расчеты на рис. 8.13 помещены в области, которые можно свернуть при создании анимации, порядок которой описан выше. На рисунке 8.15 показаны установки X-Y-графика, по которому строится анимация открывающейся двери автобуса при изменении значений системной переменной FRAME от 0 до 300, которые меняют значения угла поворота двери с 0° до минус 90° с шагом минус 1°, которые в свою очередь меняют значения координат узловых точек…

Можно, конечно, возиться (копировать и вставлять) с выражениями для l2 и x_1 (см. рис. 8.13 и 8.14), а решить систему четырех уравнений не аналитически[11], а численно, задав первое приближение переменным l1, l2, h и x_1, но тогда бы не было функции $xx_1(\alpha)$ и мы бы не смогли построить траекторию движения свободного края двери. Так или иначе, анимация открывания двери автобуса была создана и размещена на сайте факультатива для всеобщего обозрения и комментирования — см. рис. 8.16.

[10] См. задание 4 в конце этюда.
[11] Тут, кстати, выдаются четыре корня, и нужно еще сообразить, какой корень пускать в дело.

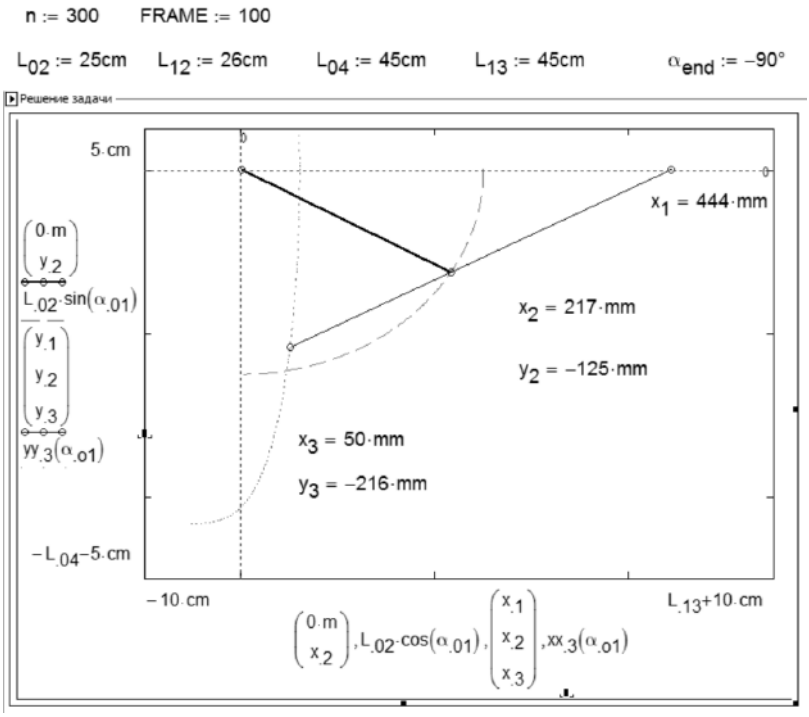


Рис. 8.15
Формирование графика движения автобусной двери

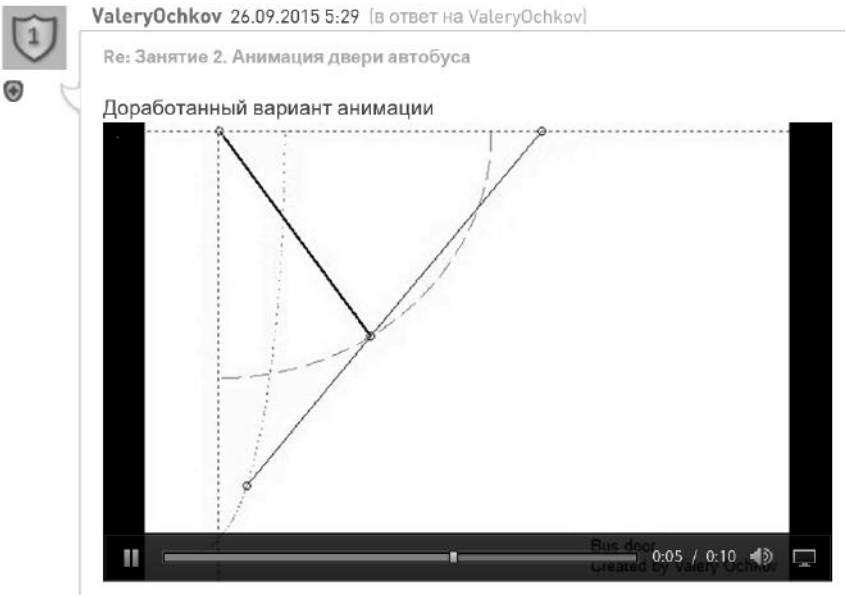


Рис. 8.16
Кадр анимации движения автобусной двери

Теперь можно менять геометрию двери — значения констант L_{02}, L_{12}, L_{04}, L_{13} (длины элементов двери), α_{end} (угол поворота открывающего рычага) и FRAME (номер кадра анимации — см. рис. 8.15) и видеть промежуточное положение двери. Если начальные установки будут несовместимы с геометрией двери, то численного решения не будет (функция Find будет краснеть), а выражения символьного решения будут возвращать комплексные числа с мнимой единицей. При этом график не будет строиться, а анимация будет иметь пробелы.

Пакетом Mathcad, как мы уже упоминали, владеет корпорация PTC. У этой фирмы главным продуктом является программа Creo (старое название Pro/Engineer[12]). Это САПР — система автоматизированного проектирования. Так вот, результаты расчета, сделанного в среде Mathcad, можно передать в среду Creo, где будет спроектирована реальная автобусная дверь. При этом можно ее размеры оптимизировать — сделать движение двери более плавным, учесть не только кинематику, но и динамику такого движения, уменьшить усилия на привод двери…

А вот еще одна анимация.

К лицею, где проводится факультатив, многие школьники, пардон, лицеисты добираются со всех концов Москвы так: сначала едут на метро до станции Новогиреево, а потом подъезжают на автобусе или троллейбусе (см. дверь выше) пару остановок. Но некоторые лицеисты едут до лицея на скутере, который везут с собой в метро. Скутер в вагоне метро можно сложить, но его колеса могут запачкать пассажиров. Так вот, на факультативе в лицее обсуждался «непачкающийся» скутер, который при складывании закрывает выдвижными крыльями свои колеса. И не только обсуждался — была создана анимация такого скутера — см. рис. 8.17 и **https://www.ptcusercommunity.com/thread/127948**. Хорошая идея — дополнять заявки на изобретения не только рисунками, но и анимациями.

И последнее. В среде Mathcad Prime 3 (а с этой версией авторы работали в момент написания книги) пока, увы, нет инструментов создания анимации. Но эту проблему можно решить так — создать несколько кадров-рисунков, по которым затем создать анимацию инструментами, которые есть в Интернете. Так, например, была создана авторская анимация провисания цепи (тема этюда 7), которая состоит из двух участков с разной удельной (линейной) массой и которую подтягивают за один конец — см. сайт **https://www.ptcusercommunity.com/message/418803**. Обсуждение этой технологии можно увидеть здесь **https://www.ptcusercommunity.com/thread/128651**.

Задания читателям

1. Создать анимацию паровой машины, показанной на рис. 8.1, и паровой машины, упоминаемой в сноске 7 на стр. 186. Мы с неким умыслом анимировали в этом этюде именно паровую машину. По-английски steam — это водяной пар. Паровые машины, пароходы и паровозы преобразовали мир в XIX в. В настоящее время технология обучения STEAM (Science, Technology, Education, Art and Mathematic — тема данной книги) также меняет мир.

2. Так изменить длину рычагов прямила Чебышёва, показанного на рис. 8.9, чтобы точка 4 двигалась как можно прямолинейней. Одно из решений — на предполагаемой прямой линии проставить несколько точек и свести решение к поиску корня системы уравнений. Такой подход к решению будет использован нами при определении коэффициентов уравнения кривой второго порядка, описывающей движение космического тела вблизи Земли (см. рис. 9.16–9.18, этюд 9).

[12] С этого программного продукта началась история фирмы PTC, которую, кстати говоря, основал выходец из Советского Союза. Мало кто знает, что два остальных «кита» автоматизированного проектирования — пакеты AutoCAD и SolidWorks были созданы также нашими соотечественниками. Истинно сказано: «Нет пророка в своем отечестве!»

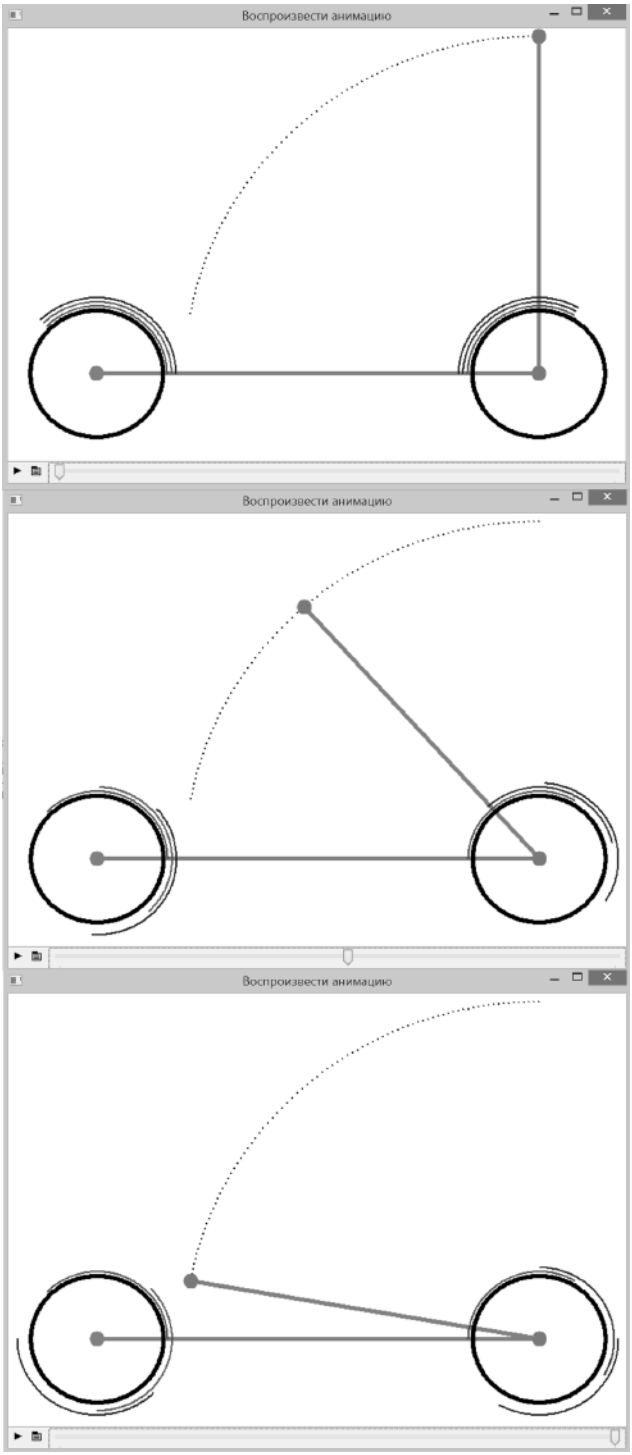
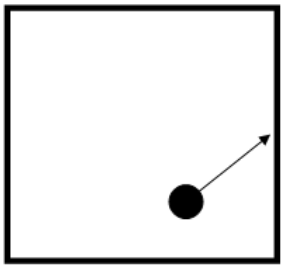


Рис. 8.17
Кадры анимации скутера с самозакрывающимися колесами

3. Мы допустили, что один край автобусной двери движется строго по прямой, совпадающей с линией закрытой двери — см. рис. 8.13. А можно ли и нужно ли этот паз сделать иначе: придать ему какой-то угол, сделать криволинейным?

4. Число уравнений в задаче об автобусной двери можно уменьшить до трех, приняв `l1 = x1 - l2`. Задача существенно упростится. Сделайте это. Мы же оставили 4 уравнения, руководствуясь лозунгом: «Тяжело в учении — легко в бою!»

5. Маленький шарик массой m, находящийся внутри жёсткой квадратной рамки (см. рисунок внизу), ударяется о середину одной из сторон рамки под углом 45°. Начальный импульс шарика равен p_0, сторона рамки равна a. Найдите величину и направление импульса рамки в момент времени $t = \frac{3}{2} ma / p_0$. За начало отсчёта времени выберите момент первого соударения шарика с рамкой. Удары шарика о стенки рамки абсолютно упругие; рамка имеет гладкие стенки и лежит на гладкой горизонтальной плоскости; масса рамки равна массе шарика; центр тяжести рамки совпадает с её геометрическим центром.



Этюд 9

НЕБЕСНАЯ МЕХАНИКА, ИЛИ ЧАСЫ КЕПЛЕРА

В этюде будет рассказано, как можно рассчитать и анимировать движение материальных точек (планет и спутников) под действием сил инерции, гравитации и сопротивления среды.

Математика: обыкновенное дифференциальное уравнение (ОДУ), система обыкновенных дифференциальных уравнений, начальные условия, краевая задача, интегральные кривые, определённый интеграл, производная, объем шара, площадь круга, кривая второго порядка, гипербола, эллипс, парабола, метод Эйлера решения систем дифференциальных уравнений, метод Рунге — Кутты, кривая Гаусса (нормального распределения), каноническое преобразование, аттрактор, система линейных однородных алгебраических уравнений, множество пропорциональных решений.

Физика: небесная механика, закон всемирного тяготения, второй закон Ньютона, второй закон Кеплера, часы Кеплера, гравитационная постоянная, ускорение свободного падения, сопротивление среды, хоровод планет, коррекция орбиты, перехват спутника, обмен спутниками, прыжок с парашютом, движение ракеты, барометрическая формула, спутник Земли, метеорит вблизи Земли, приземление космического аппарата, математический и физический маятник, гравитационный поезд, затухающие колебания, профиль трамвайных рельс, русские горки.

Информатика: численное и символьное решение дифференциального уравнения и их систем, методы решения дифференциального уравнения и их систем, компьютерная анимация, погрешность численного решения.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127821**

Диалог на занятии по компьютерному моделированию

Студент: «Смотрите, как у меня на экране два тела ловко по орбитам вертятся!»

Преподаватель, хмуро глядя на дисплей: «Тут у тебя не два тела вертятся, а целых три — Исаак Ньютон в гробу тоже переворачивается…»

Начнем с «потревоженного» Ньютона

Часто можно услышать такой риторический вопрос: «Как бы сейчас выглядело дифференциальное исчисление, если бы у Ньютона был компьютер?» У истории, как известно, не бывает сослагательного наклонения, но… Одни считают, что если бы у Ньютона все-таки был компьютер, то… никакого дифференциального исчисления не было бы. А было бы нагромождение решенных и якобы решенных задач, множество таблиц и кривых на дисплеях компьютеров, из которых какие-либо выводы, а тем более, обобщающие законы получить было бы чрезвычайно трудно — «голь (люди без компьютера) на выдумки хитра». Другие же высказываются в том плане, что «компьютеризированный Ньютон» помог бы нам избежать многих ошибок и заблуждений, уберег бы от поисков аналитических решений там, где их нет, и в конце концов, привел бы к более бурному развитию науки и техники в целом и математики в частности. Но правильный ответ, наверно, таков: «Если б у Ньютона был компьютер, то это бы означало, что… дифференциальное исчисление как наука уже существовала в течение трехсот лет до Ньютона, а сам Ньютон (один из создателей дифференциального исчисления) носил бы другое имя…»

Все сказанное о дифференциальном исчислении можно отнести и к закону Всемирного тяготения — второму великому детищу великого и «некомпьюризированного» Ньютона, которому по легенде этот закон «вошел в голову вместе с

упавшим яблоком». Давайте посмотрим, как современные компьютеры могут помочь или наоборот помешать нам разобраться в одной известной физической задаче.

С этим этюдом случилась такая история. Он был послан в виде статьи в один журнал, где в редколлегии состоит автор одной программы по расчету движения планет. Этот человек создавал ее в течение 30 лет, используя языки программирования. Теперь же математические пакеты, пришедшие на смену языкам программирования, позволяют подобные задачи обсчитывать намного проще и быстрее, с единицами измерения, с прекрасной графикой и анимацией, что будет показано в этом этюде.

На статью был получен отзыв с замечаниями «по делу» и с теми, где звучала некая скрытая обида. Мол, я на это чуть ли не всю свою жизнь положил, а теперь даже неспециалисты в данной области (небесная механика) могут все это делать полушутя. Замечания «по делу» были внесены в этюд. «Обидные» же замечания были просто приняты во внимание.

Этот эпизод вскрывает важную проблему. Нередко создают программы и пишут книги люди, сидящие, так сказать, на двух стульях. Они являются специалистами в некой области науки и техники (теплоэнергетики, например), но волею судеб вынужденные заниматься математикой и информатикой. Вследствие этого они нередко получают «щелчки» с той и другой стороны.

Итак, этюд по небесной механике

А суть дела такова, что не только специализированные пакеты, такие, например, что описаны в [56], но и универсальные математические программы позволяют решать задачи о движении материальных точек (планет, спутников, комет и т. д.) с заданной массой согласно второму закону Ньютона $m \cdot a = \sum F$ и закону всемирного тяготения $F = G \cdot m_1 \cdot m_2 / r^2$.

Так, например, в среде математической программы Mathcad есть функция `Odesolve`, численно решающая системы обыкновенных дифференциальных уравнений (ОДУ) [55], к которым сводится задача о движении материальных точек. Если использование этой функции дополнить средствами анимации (см. [44] и этюд 8), то можно получить довольно интересные и поучительные решения, связанные с темой этюда.

Давайте рассмотрим несколько примеров

Пример 1 (тестовый). Великолепная восьмерка. Движение трех материальных точек по орбите в виде знака бесконечности (или «великолепной восьмерки», по терминологии [56]).

В работе [57] дано аналитическое решение этого частного случая движения трех материальных точек[1]. Мы же сейчас решим эту задачу в среде Mathcad численно и сравним результаты численного и аналитического решений.

Для того чтобы в среде Mathcad 15 численно решить систему ОДУ, нужно ввести в расчет ключевое слово `Given` (`Дано`), записать саму систему уравнений, используя апостроф (он вводится аккордом Ctrl + F7) для обозначения первой производной или двойной апостроф для второй производной[2], а также ввести начальные условия — положение материальных точек и их скорости в начальный момент времени (на старте `t = 0`). В нашем случае дифференциальные уравнения движения материальных точек записываются парами — по горизонтальному (`x`) и вертикальному (`y`) направлениям.

[1] Для двух точек аналитические решения есть для всех случаев, сводящихся к эллипсу (окружности), параболе и гиперболе. Для трех и более точек аналитические решения есть для ограниченного числа случаев.

[2] Тут можно применять и оператор взятия производной `d/dt`.

На рисунке 9.1 можно видеть систему из шести ОДУ, записанную для трех материальных точек (черная, красная и синяя[3]): произведение массы на ускорение (левая часть уравнения) равно сумме двух гравитационных сил, действующих на рассматриваемую материальную точку от ее двух «соседок». Вернее, не самих ускорений и сил, а их проекций в горизонтальном[4] (x) и вертикальном (y) направлениях[5]. Каждое уравнение можно разделить на массу соответствующей планеты, оставив слева только вторую производную. Кроме того, можно убрать переменную G (гравитационная постоянная), т. к. в наших выкладках она принимается (пока — см. далее) за единицу. Это упрощает и ускоряет расчет. Но мы намеренно оставляем эти переменные в уравнениях, чтобы максимально сохранить физику задачи [31]: слева произведение массы точки на ее ускорение, справа силы, действующие на точку — планету или спутник.

Given

$$\mathbf{x}(0) = \mathbf{x}_0 \qquad \mathbf{x}'(0) = \mathbf{x}'_0 \qquad \mathbf{m}\cdot\mathbf{x}''(t) = \frac{\mathbf{m}\cdot G\cdot m\cdot(x(t)-\mathbf{x}(t))}{\left[\sqrt{(\mathbf{x}(t)-x(t))^2+(\mathbf{y}(t)-y(t))^2}\right]^3} + \frac{\mathbf{m}\cdot G\cdot \mathbf{m}\cdot(x(t)-\mathbf{x}(t))}{\left[\sqrt{(\mathbf{x}(t)-x(t))^2+(\mathbf{y}(t)-y(t))^2}\right]^3}$$

$$\mathbf{y}(0) = \mathbf{y}_0 \qquad \mathbf{y}'(0) = \mathbf{y}'_0 \qquad \mathbf{m}\cdot\mathbf{y}''(t) = \frac{\mathbf{m}\cdot G\cdot m\cdot(y(t)-\mathbf{y}(t))}{\left[\sqrt{(\mathbf{x}(t)-x(t))^2+(\mathbf{y}(t)-y(t))^2}\right]^3} + \frac{\mathbf{m}\cdot G\cdot \mathbf{m}\cdot(y(t)-\mathbf{y}(t))}{\left[\sqrt{(\mathbf{x}(t)-x(t))^2+(\mathbf{y}(t)-y(t))^2}\right]^3}$$

$$x(0) = x_0 \qquad x'(0) = x'_0 \qquad m\cdot x''(t) = \frac{m\cdot G\cdot \mathbf{m}\cdot(\mathbf{x}(t)-x(t))}{\left[\sqrt{(x(t)-\mathbf{x}(t))^2+(y(t)-\mathbf{y}(t))^2}\right]^3} + \frac{m\cdot G\cdot \mathbf{m}\cdot(\mathbf{x}(t)-x(t))}{\left[\sqrt{(x(t)-\mathbf{x}(t))^2+(y(t)-\mathbf{y}(t))^2}\right]^3}$$

$$y(0) = y_0 \qquad y'(0) = y'_0 \qquad m\cdot y''(t) = \frac{m\cdot G\cdot \mathbf{m}\cdot(\mathbf{y}(t)-y(t))}{\left[\sqrt{(x(t)-\mathbf{x}(t))^2+(y(t)-\mathbf{y}(t))^2}\right]^3} + \frac{m\cdot G\cdot \mathbf{m}\cdot(\mathbf{y}(t)-y(t))}{\left[\sqrt{(x(t)-\mathbf{x}(t))^2+(y(t)-\mathbf{y}(t))^2}\right]^3}$$

$$\mathbf{x}(0) = \mathbf{x}_0 \qquad \mathbf{x}'(0) = \mathbf{x}'_0 \qquad \mathbf{m}\cdot\mathbf{x}''(t) = \frac{\mathbf{m}\cdot G\cdot \mathbf{m}\cdot(\mathbf{x}(t)-x(t))}{\left[\sqrt{(\mathbf{x}(t)-\mathbf{x}(t))^2+(y(t)-\mathbf{y}(t))^2}\right]^3} + \frac{\mathbf{m}\cdot G\cdot m\cdot(\mathbf{x}(t)-x(t))}{\left[\sqrt{(\mathbf{x}(t)-x(t))^2+(y(t)-\mathbf{y}(t))^2}\right]^3}$$

$$\mathbf{y}(0) = \mathbf{y}_0 \qquad \mathbf{y}'(0) = \mathbf{y}'_0 \qquad \mathbf{m}\cdot\mathbf{y}''(t) = \frac{\mathbf{m}\cdot G\cdot \mathbf{m}\cdot(\mathbf{y}(t)-y(t))}{\left[\sqrt{(\mathbf{x}(t)-\mathbf{x}(t))^2+(y(t)-\mathbf{y}(t))^2}\right]^3} + \frac{\mathbf{m}\cdot G\cdot m\cdot(y(t)-\mathbf{y}(t))}{\left[\sqrt{(\mathbf{x}(t)-x(t))^2+(y(t)-\mathbf{y}(t))^2}\right]^3}$$

Рис. 9.1
Запись системы ОДУ в среде Mathcad 15

Блок решения системы ОДУ в среде Mathcad 15 заканчивается вызовом функции `Odesolve`[6] (рис. 9.2), имеющей следующие аргументы:

---

[3] В среде Mathcad 15 есть возможность окрашивать переменные в разные цвета, чем мы и воспользовались в расчете вместо того, чтобы приписывать переменным индексы: 1, 2, 3 и т. д. С другой стороны, все это может быть опубликовано и в черно-белом издании, где цвет у переменных только черный. Да и цветное издание может попасть в руки дальтоника, что тоже будет не очень хорошо. Когда-то давно первый автор послал статью о цвете в программах в один журнал. Статью не опубликовали — рецензент сказал, что цветные дисплеи и принтеры еще не скоро появятся и появятся ли вообще. А сам он (рецензент) — дальтоник. Через пять лет после попытки публикации этой статьи появился язык Visual Pascal, где цвет впервые использовался для выделения программистских конструкций: встроенных, пользовательских операторов, ремарок, сообщений об ошибках и т. д. (см. рис. 4.47 на стр. 99).

[4] В космосе, конечно, нет горизонтального и вертикального направлений в физическом смысле. Они появятся, когда мы позже будем рассматривать подобные задачи вблизи Земли.

[5] Некоторые читатели, увидев куб, а не квадрат в знаменателе дробей на рис. 9.1 и не видя числитель, могут тут вспомнить старый анекдот: «Если бы на голову Ньютона упало не яблоко, а более весомый предмет (кокос, например), то в законе всемирного тяготения расстояние возводилось бы не во вторую, а в третью степень».

[6] В среде Mathcad есть и другие инструменты численного решения дифференциальных уравнений (обыкновенных и в частных производных), но здесь они не рассматриваются.

- вектор имен функций, являющихся решением системы ОДУ (в среде Mathcad Prime за именами функции нужно в скобках указать ее аргумент — см. ниже);
- переменная, по которой ведется численное интегрирование (аргумент искомых функций; в среде Mathcad Prime эту переменную не указывают);
- конечное (правое) значение интервала интегрирования («левое» значение — это нуль);
- число точек, по которому ведется табулирование искомых функций (необязательный параметр — по умолчанию имеем 1000 точек).

Кроме того, нажатием правой кнопки мыши можно выбрать для функции `Odesolve` один из четырех возможных методов численного решения задачи:

- `Adams/BDF` — смешанный алгоритм Адамса и обратной формулы дифференцирования;
- `Fixed` (интегрирование методом Рунге — Кутты с фиксированным шагом — на сайте **http://twtmas.mpei.ac.ru/mas/worksheets/Euler.mcd** можно увидеть сравнение этого метода с методом Эйлера; см. также ниже, где эти методы описываются и сравниваются);
- `Adaptive` (интегрирование методом Рунге — Кутты с переменным шагом — см. **http://twtmas.mpei.ac.ru/mas/worksheets/rkadapt.mcd**);
- `Radau` — алгоритм RADAUS для жестких систем ОДУ.

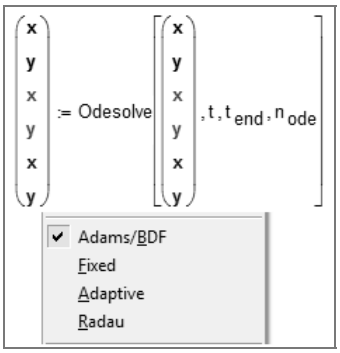


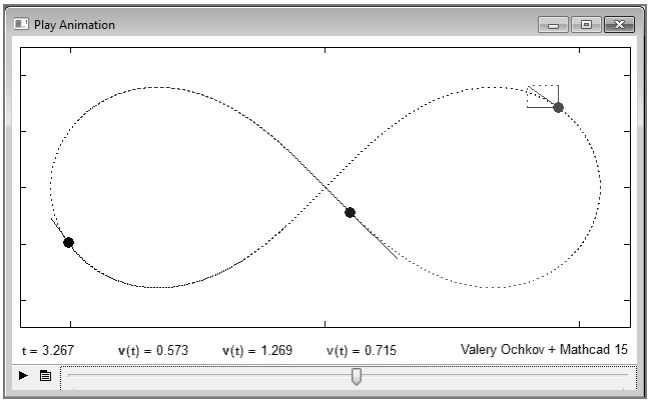
Рис. 9.2
Настройка функции Odesolve на нужный метод решения систем обыкновенных дифференциальных уравнений

В среде Mathcad Prime возможности смены метода решения ОДУ нет. Там используется только метод `Adams/BDF`.

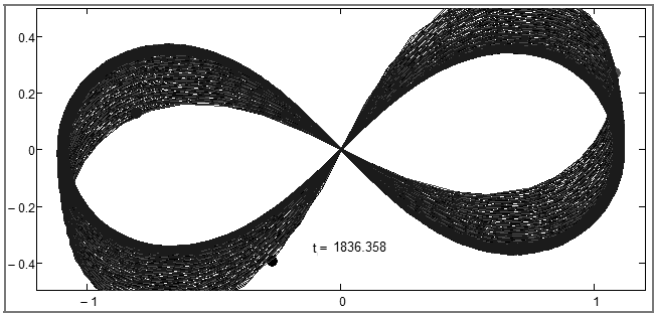
Если трем материальным точкам задать начальные условия, взятые из [57] и показанные на рисунке 9.3, то решением системы ОДУ (рис. 9.1) будут функции, графическое отображение которых (след-в-след) показано на рис. 9.4*а*.

$$\begin{pmatrix} m & x_0 & y_0 & x'_0 & y'_0 \\ m & x_0 & y_0 & x'_0 & y'_0 \\ m & x_0 & y_0 & x'_0 & y'_0 \end{pmatrix} := \begin{pmatrix} 1 & 0.97000436 & -0.24308753 & \frac{0.93240737}{2} & \frac{0.86473146}{2} \\ 1 & -0.97000436 & 0.24308753 & \frac{0.93240737}{2} & \frac{0.86473146}{2} \\ 1 & 0 & 0 & -0.93240737 & -0.86473146 \end{pmatrix}$$

Рис. 9.3
Начальные условия для задачи о «великолепной восьмерке» при движении трех планет



a



b

Рис. 9.4

a — кадр анимации движения трех материальных точек (планет) по орбите в форме знака бесконечности (см. саму анимацию здесь **https://www.ptcusercommunity.com/videos/5777**); показаны векторы скорости, который у красной планеты дополнительно разложен по осям *c* и *y*; *b* — графическое отображение потери точности численного решения системы ОДУ

Сравнение «телеметрических данных, снятых с орбиты» и показанных на рис. 9.4*a*, с данными из [57] дает полное совпадение результатов, что может свидетельствовать об адекватности нашей модели, зафиксированной в уравнениях рис. 9.1. Но если быть точными, то происходит следующее. На первых примерно ста «витках» орбиты не наблюдается отклонения численного решения от аналитического. Но при увеличении значения t происходит некое «размывание» орбиты вследствие накапливаемой ошибки численного решения задачи (рис. 9.4*b*). Эта особенность хорошо просматривается на втором примере, где в модель (см. рис. 9.1) добавлены новые уравнения и новые слагаемые у существующих для дополнительных двух точек — для коричневой и зеленой.

Пример 2 (тоже пока тестовый). Хоровод планет. На вершинах пятиконечной звезды помещаются пять материальных точек (планеты) с одинаковой массой. Начальные скорости точек равны по модулю и имеют направления по часовой стрелке по касательной к окружности, в которую вписана звезда (см. рис. 9.5*a*). Движение планет с такой стартовой позиции и с такими стартовыми скоростями, называемое хороводом, отображено на рис. 9.5*b* (в работе [56] рассматривается такое движение для трех точек — см. **https://www.ptcusercommunity.com/videos/5782**).

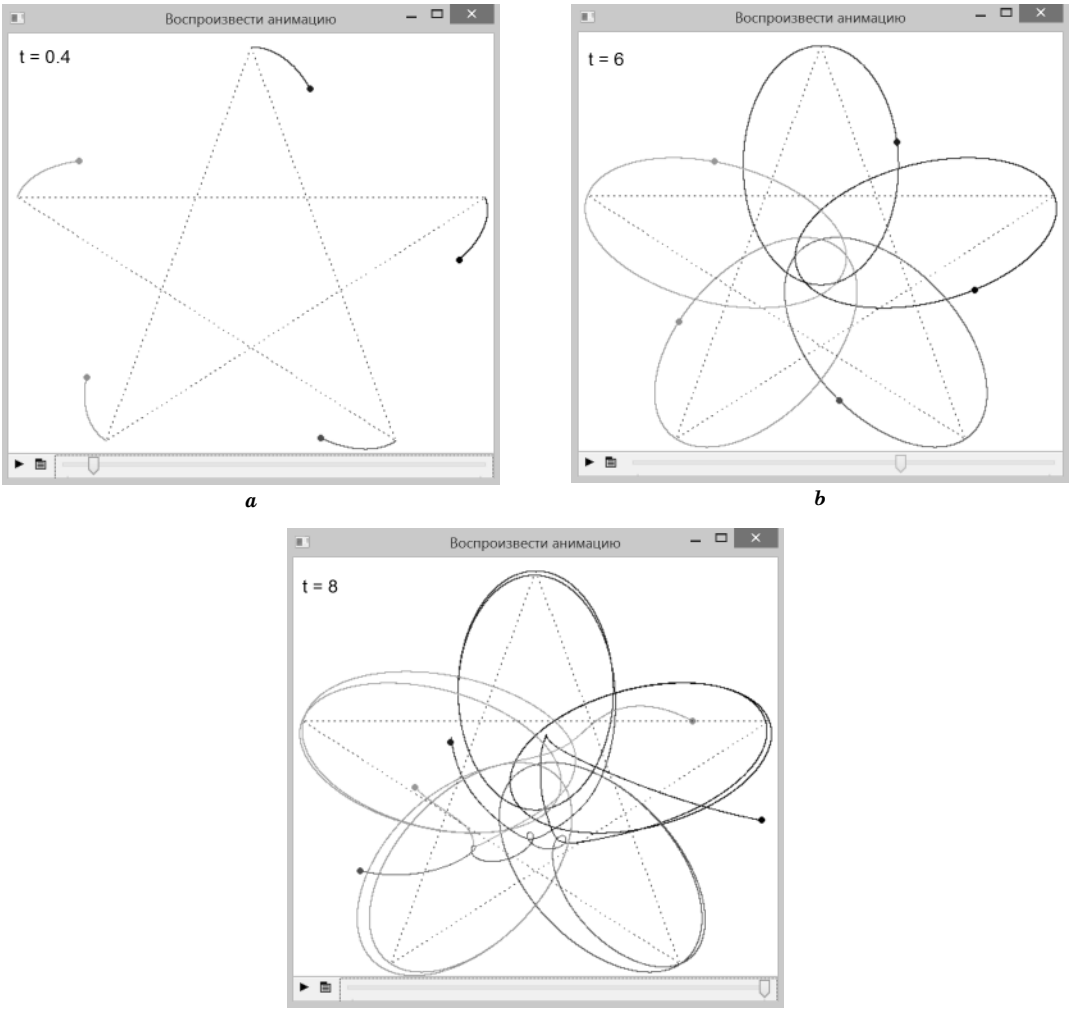


Рис. 9.5

*Кадры анимации хоровода пяти планет (**https://www.ptcusercommunity.com/message/301560**):*

a — старт; *b* — полное совпадение численного решения с аналитическим; *c* — сбой в численном решении.

«Космический хоровод», как и любой хоровод, не может длиться бесконечно долго — в какой-то момент он рассыпается, а его участники разбегаются в разные стороны. Что-то подобное можно наблюдать и в нашем хороводе планет при увеличении значения переменной численного интегрирования `t` — см. рис. 9.5*c*[7].

В задачу о движении материальных точек можно внести и третье измерение, переведя ее из плоскости в объем (в космическое пространство[8]). Для этого необхо-

[7] Такие «произведения изобразительного искусства» часто создаются детьми: сначала все идет более-менее хорошо, а потом ребенок устает рисовать правильно и малюет на почти готовом рисунке «каляку-маляку». Можно сказать, что Mathcad «устал рисовать нам красивый космический цветок» (рис. 9.5*b*) и закончил все «калякой-малякой» (рис. 9.5*c*).

[8] Мы говорим «полеты в межпланетном пространстве». А правильнее говорить «полеты на межпланетной плоскости». Большинство расчетов полетов естественных и искусственных космических тел ведется в двух измерениях с выбором правильной плоскости.

димо и достаточно записать еще одну тройку уравнений в систему, показанную на рисунке 9.1, и добавить начальных данных. На рисунке 9.6 показан «хоровод» трех планет в объеме. Третья ось направлена вверх (по ней задано одинаковое начальное равномерное движение планет вверх по оси z), а кривые являются графиками трех решений. Этот график можно также рассматривать как изменение местоположения точек на плоскости (см. рис. 9.4*а*) с течением времени (такие графики называются интегральными кривыми).

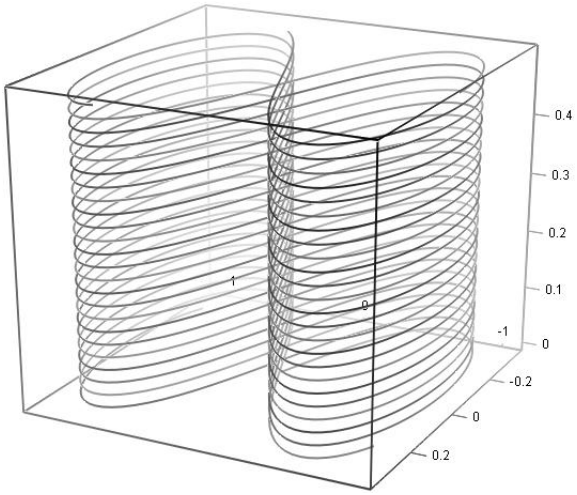


Рис. 9.6
Трехмерная задача о движении трех материальных точек
(https://www.ptcusercommunity.com/thread/60224)

На рисунке 9.4*а* фактически показана одна из трех проекций движения трех точек (рис. 9.6), когда ось z направлена на наблюдателя. Траектории, отображенные на рис. 9.6, — можно интерпретировать как заплетание трех прядей волос в косичку, что хорошо видно при анимации этого процесса. Если же начальные скорости для трех планет задавать разные, то можно получить очень запутанные траектории движения, из которых уже будет трудно выделить составляющие, связанные с самой физикой задачи и с потерей точности ее численного решения. Аналитическое (абсолютно точное) решение тут, повторяем, невозможно.

Замечание по 3D-графике

Объемные графики — это скорее некая рекламная штучка, чем инструмент визуализации результатов «серьезных» расчетов, расчетов, имеющих практическую, а не демонстрационную (презентационную, рекламную и др.) цель. Тем более, если такой результат нужно перенести на бумагу принтера, где нет возможности повернуть объемный график и рассмотреть его с другой точки, что допустимо на экране дисплея.

Поэтому, создавая Mathcad-документ, нужно сразу ориентироваться на простую, «ненакрученную» графику, помня, что экран дисплея и бумага принтера — это плоские объекты. Реальные же, серьезные задачи могут опираться на набор функции трех, четырех и более аргументов, графики которых нужно показывать в соответствующем четырех-, пяти- и более мерном пространстве. А у нас оно трехмерное: линия (1D) — плоскость (2D) — объем (3D). Не следует также забывать, что за неоправданно накрученной графикой очень часто скрывается довольно пустяковый расчет, маскирующий себя сложными объемными поверхностями с перспективой, подсветкой, «туманом» и другими «наворотами». Сложный же, имеющий явную практическую ценность расчет, как правило, иллюстрируется

несложными «плоскими» графиками, показывающими основные закономерности расчета. Такие расчеты, например, включают в себя не изображения поверхностей, которыми приятно любоваться, но с которыми трудно работать, а семейства кривых: см. пример на рис. 14.34 (стр. 333).

Если же 3D-графику необходимо использовать не для графического отображения функций двух аргументов, а для отображения неких абстрактных или реальных тел (неподвижных или движущихся), то для этого, конечно, лучше использовать не Mathcad, а специализированные средства.

Но вернемся к «плоским» космическим случаям, имея в виду, во-первых, то, что планеты Солнечной системы, например, движутся по орбитам, лежащим примерно в одной плоскости, и, во-вторых, то, что численным методам, встроенным в Mathcad, дай бог справиться с плоскими задачами…

Два наших предыдущих примера («восьмерка» и «хоровод») — это, повторяем, тесты для сравнения численных и аналитических решений, когда ошибка численных методов еще не достигла критического уровня. Адекватность решения при не слишком больших[9] значениях `t` позволяет нам применить данный способ решения (см. рисунки 9.1 и 9.2) к еще не исследованным случаям, для которых аналитические решения неизвестны и невозможны. Для этого, повторяем, необходимо (и достаточно ли — это отдельный вопрос) добавить новые уравнения с новыми слагаемыми в правой части, изменить начальные условия и выбрать подходящий метод численного решения.

Пример 3. Перехват спутника. Красная планета в паре с синим спутником движется в сторону черной планеты. У всех разные массы: 2, 0.01 и 20 единиц массы (позже мы перейдем к килограммам и тоннам[10]). Что произойдет с этими небесными телами в момент сближения? Для исследования этого случая мы проведем численные эксперименты и решим данную задачу, используя разные методы решения. На рис. 9.7 зафиксирован кадр анимации, который показывает, что в момент сближения планет спутник меняет своего «патрона» и начинает вращаться вокруг другой (черной) планеты. Красная же планета продолжает свой небесный путь в одиночестве.

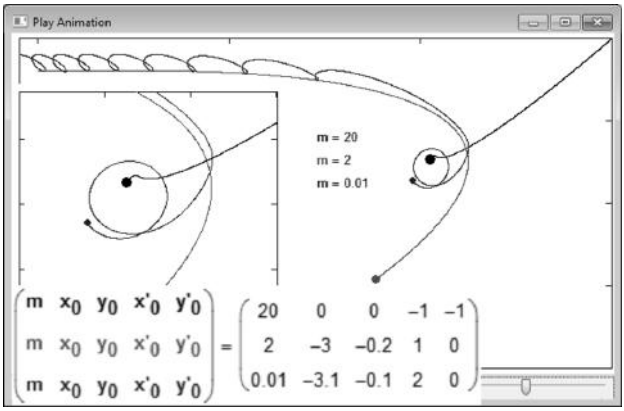


Рис. 9.7

*Кадр анимации перехват спутника (**https://www.ptcusercommunity.com/videos/5779**)*

Перехват спутника при создании анимации, показанной на рис. 9.7[11], был смоделирован с использованием метода (установки — см. рис. 9.2) `Radau` для функции

[9] Знать бы еще, что такое в количественном выражении «не слишком большие значения».

[10] Читаем у Даниила Хармса: «Прошло 35 колов времени, и хозяин принес Ивану Ивановичу антрекот на круглой деревянной дощечке».

[11] На рисунке слева показано увеличение области перехвата спутника

`Odesolve`. Если же метод решения сменить с `Radau` на `Adams/BDF`, то мы увидим качественно другую картину: перехват спутника не состоится, а будет наблюдаться только некоторая коррекция его орбиты вокруг красной планеты вследствие приближения черной планеты к системе «красная планета — синий спутник» — см. рис. 9.8.

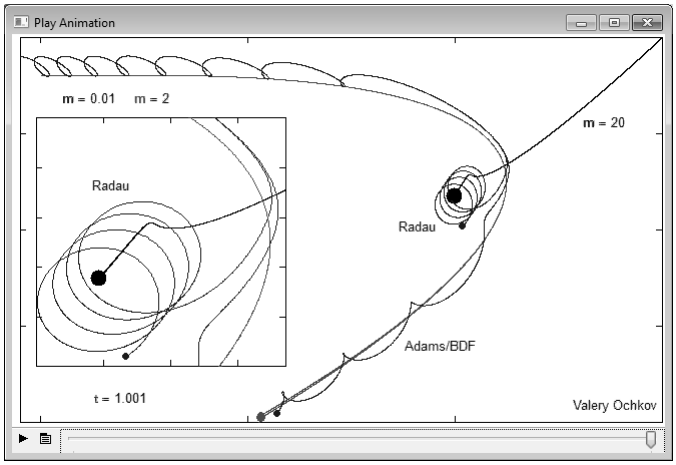


Рис. 9.8
*Результаты анимации при использовании двух разных численных методов (Radau и Adams/BDF) решения системы ОДУ (**https://www.ptcusercommunity.com/videos/5779**)*

Из рисунка 9.8 видно, что два разных метода решения системы ОДУ (`Radau` и `Adams/BDF`) сначала (до встречи с черной планетой) дают одинаковые результаты, а потом (после встречи с черной планетой) результаты расходятся: при методе `Radau`, повторяем, наблюдается перехват спутника черной планетой у красной планеты, а при методе `Adams/BDF` только коррекция его орбиты вблизи старого «патрона» — у красной планеты. Методы же `Fixed` и `Adaptive` (интегрирование с постоянным и переменным шагом) тут, как можно увидеть из соответствующей анимации, зайдя на вышеотмеченный сайт, тут совсем не годятся — срывы решения видны в самом начале процесса численного решения задачи (интегрирования).

Итак, мы имеем реальный физический процесс, его физическую модель, математическую модель физической модели (уравнения и начальные условия) и несколько вычислительных (численных) моделей того же процесса: расчетные модули и формулы для методов `Radau`, `Adams/BDF`, `Fixed` и `Adaptive` (см. рис. 9.3 выше). Численные модели дают некоторые приближения математической модели, которая осуществляет приближение физической модели, которая, в свою очередь, приближает реальный физический процесс. На каждом этапе моделирования мы по разным объективным причинам допускаем ошибки, а поэтому необходимо отделить несущественные ошибки от принципиальных. Ошибки указанных численных моделей являются принципиальными: четыре расчетные схемы дали четыре разных результата, которые сильно зависят от метода решения. И не количественно (см. рис. 9.4 и 9.5), а качественно (см. рис. 9.8). Это означает, что задача так и не решена в целом. Хотя первая фаза движения одинакова для двух методов (см. рис. 9.8), а поэтому, скорее всего, начальные результаты можно использовать при численном моделировании решаемой задачи. Но см. [7] и конец ниже.

Анимации, отображенные в одном кадре на рисунках 9.5–9.8, еще раз акцентируют наше внимание на «блеске и нищете» численных методов решения ОДУ: они позволяют худо-бедно решать задачи, которые «не по зубам» аналитическим методам, и создавать довольно занимательные и поучительные анимации. Но они

же могут нам «врать прямо в глаза». Но к этому, повторяем, мы еще вернемся ниже.

Тут затрагивается важный вопрос — можно ли использовать анимацию для оценки качества тех или иных методов решения систем ОДУ. Дерганье траекторий, «каляки-маляки» свидетельствуют о том, что использованный метод решения не годится, по крайней мере, для выбранного диапазона интегрирования. С другой стороны, правильность траекторий планет и спутников, их похожесть на реальные процессы, может служить неким признаком добротности использованных методов решения. Но не всегда — см. рис. 9.8, где обе траектории синего спутника «красивы» и похожи на реальные траектории.

Пример 4. Обмен спутниками. На рисунке 9.9 показан еще один нетривиальный и более сложный случай — обмен спутниками. Случаи, показанные ранее на рисунках 9.7 и 9.8, имеют житейские аналогии: существует некая супружеская пара (система «планета — спутник»), к которой приближается некий посторонний объект, вернее, субъект. Он может либо «отбить» спутник (рис. 9.7), либо внести в жизнь этой супружеской пары некие «пертурбации» (рис. 9.8). Когда на сайте **https://www.ptcusercommunity.com/videos/5783** публиковалась анимация, показанная на рис. 9.9, то она была нами прокомментирована так: «В жизни тоже такое встречается: две супружеские пары знакомятся, сближаются и в конце концов понимают, что когда-то они сделали не тот выбор; происходит законный (два развода и две новые свадьбы) или неформальный «обмен партнерами». На этот комментарий сразу отозвался модератор сайта и попросил его убрать, т. к. «на этот сайт заходят и несовершеннолетние», школьники, например, решающие свои домашние задания в среде Mathcad.

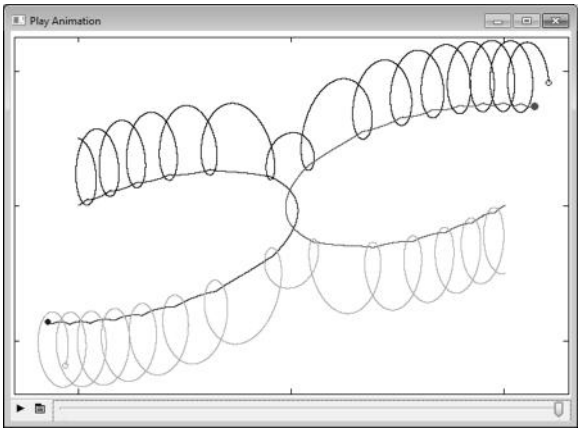


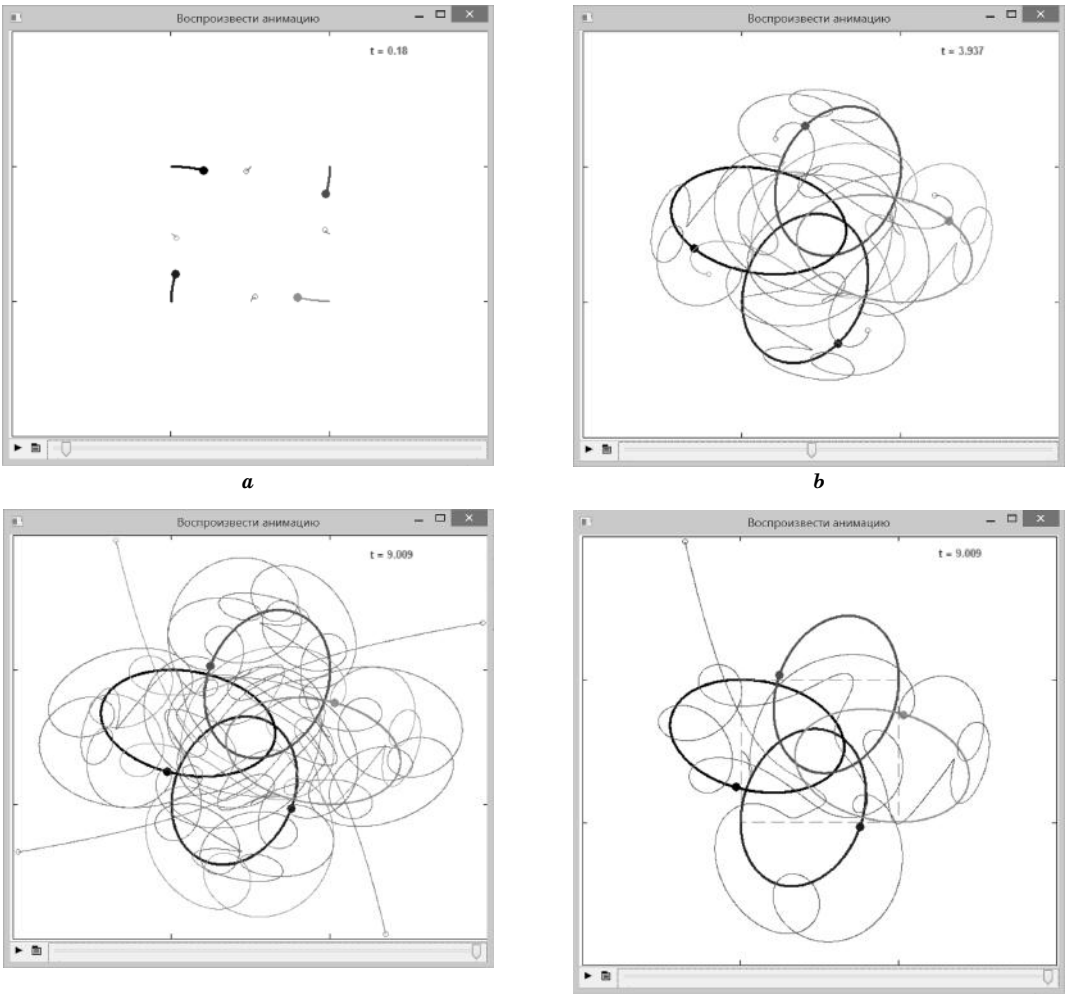
Рис. 9.9

*Кадр анимации обмена спутниками (**https://www.ptcusercommunity.com/thread/59968**)*

У рисунков 9.7–9.9 такое несколько фривольная интерпретация и потому, что нельзя серьезно относиться к этим «житейским» траекториям из-за, повторяем, сомнений в адекватности численных методов, использованных для решения этих задач. Но случаи интересные и базирующиеся на «адекватной» физики. И с этим нужно согласиться. Вернее так: можно согласиться, а можно не согласиться.

Пример 5 (гибридный: проверка старого и показ нового). Хоровод (па-де-катр) четырех планет с четырьмя спутниками. На рисунке 9.10*а* показан стартовый кадр анимации системы из восьми материальных точек: четырех планет (масса каждой 0,5) и четырех спутников (масса каждого 0.0001). Стартовые условия такие: планеты помещаются в углах квадрата со стороной, равной единице, и им придаются скорости, равные 0,5 и направленные вверх (синяя планета), вправо (черная), вниз (красная) и влево (коричневая). Такие уже пять, а не четыре плане-

ты без спутников кружатся в хороводе, показанном на рис. 9.5 (вернее, исполняют некий па-де-катр). Но в нашем случае в серединах сторон квадрата дополнительно помещаются спутники с нулевой начальной скоростью.



a *b* *c* *d*

Рис. 9.10

Кадры анимации движения 4 планет и 4 спутников:

a — старт; *b* — полет спутников вокруг планет; *c* — вылет спутников их «хоровода»; *d* — траектория одного спутника (остальные три траектории невидимы).

Анимацию движения этих восьми небесных тел описать словами почти невозможно. Нужно смотреть саму анимацию — **https://www.ptcusercommunity.com/thread/84910**. Каждый из спутников будет стараться поочередно «пристроиться» к каждой из четырех планет, выделывая всевозможные «па» (рис. 9.10*b*). В конце «танца» эти спутники вылетят из «хоровода» — см. рис. 9.10*c*. На вышеотмеченном сайте можно видеть анимацию движения одного из спутников, когда три остальные спутника не показаны (рис. 9.10*d*). Там четко видно, как спутник меняет партнеров в этом «танце». Такие же «телодвижения» в это время производят и другие спутники с единственной разницей: траектории их полетов повернуты относительно центральной точки

(центра масс) на 90° (рис. 9.10*c*). Остается открытым вопрос, вернутся ли назад в «хоровод» отлетевшие спутники. Численным экспериментом на этот вопрос ответить затруднительно — тут всю картину испортит погрешность вычислений (см., например, рис. 9.5*b*). Попытка ответа (вернее, один из путей решения этой отдельной задачи) будет дана в дивертисменте 1 к этюду. Симметрия же полёта планет и спутников, зафиксированная на рис. 9.10*c*, дает основание делать вывод о достаточной точности численного решения задачи.

Пример 6. Спутник Земли, или Часы Кеплера. Новая версия пакета Mathcad — Mathcad Prime пока не имеет анимации, но зато может решать ОДУ с единицами измерений физических величин. До этого (Mathcad 15) эти единицы игнорировались и использовались безразмерные единицы длины, скорости, массы и т. д., что чревато ошибками: где-то в системе уравнений по оплошности тройку в показателе степени заменили на двойку, система, тем не менее, решается, но выдает неверный ответ. В среде Mathcad Prime такая ошибка сразу будет замечена. При этом ведется пересчет единиц измерения, что также очень полезно и что отмечено на рисунках ниже.

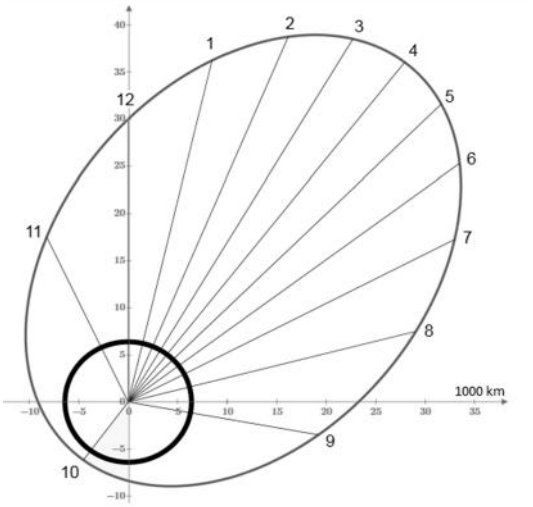
На рисунке 9.11 показана орбита вращения спутника Земли с реальными значениями расстояний, скоростей и, главное, гравитационной постоянной G. Подобраны такие исходные данные, чтобы период вращения спутника был равен 12 часам. Кроме того, на графике добавлены сектора эллиптической орбиты, которые иллюстрируют второй закон Кеплера: площади этих секторов в равные (часовые) промежутки времени равны (с точностью до сотых процента — см. рис. 9.11*b*) и при численном решении задачи, что является дополнительным подтверждением верности нашей модели[12]. Саму же анимацию можно рассматривать, как еще одни необычные математические часы [58], у которых стрелки движутся по эллиптическому циферблату с переменной скоростью[13]. Такие часы будут хорошо смотреться в музеях Кеплера, планетариях, обсерваториях...

Из рисунка 9.11*b* видно, что спутник в районе «восьми часов» максимально приближен к Земле (перигей) и в этой точке будет иметь максимальную скорость согласно второму закону Кеплера. В районе же двух часов (апогей) скорость спутника будет минимальной.

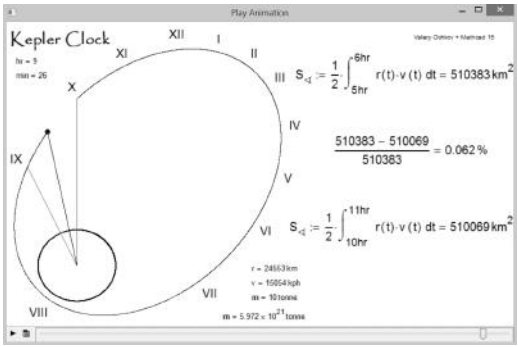
[12] Задача о двух телах имеет аналитическое решение, в частности, эллипс как траектория полета. Наш эллипс можно привести к каноническому виду и рассчитать значения его параметров — коэффициентов уравнения кривой второго порядка, осей, фокусов и прочее. Эту задачу мы попытаемся решить в дивертисменте 1.

[13] Попробуем составить такую формулу изобретения. Часы Кеплера представляют собой традиционные часы со стрелками и циферблатом. Отличие состоит в том, что циферблат часов Кеплера — это не окружность, а эллипс с числами 1, 2, 3... 12 (I, II, III... XII). Стрелки таких часов (часовая, минутная и секундная) движутся по кругу неравномерно. Длины стрелок также меняются по мере их движения. Концы стрелок описывают эллиптические траектории. Дополнительно на циферблате отображается сектор, охватывающий два соседних часа, между которыми в данный момент времени находится секундная стрелка. Этот сектор меняет свое расположение и форму каждые пять секунд, но площадь его остается постоянной согласно второму закону Кеплера. Вокруг места крепления стрелок, прорисована окружность, изображающая Землю, а на конце секундной стрелки помещен кружок, изображающий спутник Земли. Часы Кеплера могут быть только с одной стрелкой, которая показывает минуты часа, охваченного сектором эллипса (см. рис. 9.11*b*). Часы Кеплера часы будут очень уместны в школьных кабинетах математики и физики, в планетариях, обсерваториях, музеях, привлекая внимание людей, заставляя их задуматься о законах небесной механики, в частности, о втором законе Кеплера.

G	Earth	Satellite	L	v	t_{end}
$\left(\frac{N \cdot m^2}{kg^2}\right)$	(tonne)	(tonne)	(km)	(kph)	(hr)
$6.67384 \cdot 10^{-11}$	$5.972 \cdot 10^{21}$	10	30000	8670	12



a



b

Рис. 9.11

Часы Кеплера:

a — Mathcad-расчет; *b* — кадр анимации **(https://www.ptcusercommunity.com/thread/60078)**.

Вблизи Земли можно отходить от закона всемирного тяготения, переходить к потенциальному полю тяжести вблизи нашей планеты, характеризуемого ускорением свободного падения, и учитывать ее атмосферу. Такую задачу также несложно решить с привлечением Mathcad — см. рис. 9.12.

Пример 7. Прыжок парашютиста. Постановка задачи о парашютисте возникла под влиянием знаменитого прыжка с парашютом в 2012 г. с высоты почти 40 км (из стратосферы) австрийца Феликса Баумгартнера **(http://lenta.ru/news/2014/02/02/video)**. Подобный прыжок с еще большей высоты в октябре 2014 года повторил Алан Юстас — исполнительный директор фирмы Google **(http://www.profile.ru/mir/item/87941-ispolnitelnyj-direktor-google-pobil-rekord-vysoty-dlya-pryzhkov-iz-stratosfery).**

h_1	r_1	Масса	h_2	r_2	k	t_{end}
(м)	(см)	(кг)	(м)	(м)		(мин)
30000	30	110	1500	2.5	1.7	15

Изменение по высоте радиуса, площади сечения и объема парашютиста

$r(h) := \text{if}(h > h_2, r_1, r_2)$ Сечение$(h) := \pi \cdot r(h)^2$ Объем$(h) := \frac{4}{3}\pi \cdot r(h)^3$

Изменение плотности воздуха по высоте

$\rho_{возд}(h_1) = 0.01002 \frac{kg}{m^3}$

$\rho_{возд}(h_2) = 1.058 \frac{kg}{m^3}$

$\rho_{возд}(0\ m) = 1.225 \frac{kg}{m^3}$

$$\rho_{возд}(h) := \left| \begin{array}{l} L \leftarrow 0.0065 \frac{K}{m} \\ T_0 \leftarrow 288.15\ K \\ T \leftarrow T_0 - L \cdot h \\ M \leftarrow 28.9644 \frac{gm}{mole} \\ p \leftarrow 101325\ Pa \cdot \left(1 - \frac{L \cdot h}{T_0}\right)^{\frac{g \cdot M}{R \cdot L}} \\ \rho_{возд} \leftarrow \frac{p \cdot M}{R \cdot T} \end{array} \right.$$

Решить

Ограничения

$h(0\ c) = h_1$ $v(0\ c) = 0 \frac{м}{с}$ Начальные условия

$v(t) = \frac{d}{dt}h(t)$ Скорость - это производная пути по времени

Сила сопротивления воздуха

Масса $\cdot \frac{d}{dt}v(t) = k \cdot \rho_{возд}(h(t)) \cdot$ Сечение$(h(t)) \cdot v(t)^2 +$ Архимедова сила Вес

$+ g \cdot$ Объем$(h(t)) \cdot \rho_{возд}(h(t)) - g \cdot$ Масса

Решатель

$\begin{bmatrix} h \\ v \end{bmatrix} := \text{odesolve}\left(\begin{bmatrix} h(t) \\ v(t) \end{bmatrix}, t_{end}\right)$

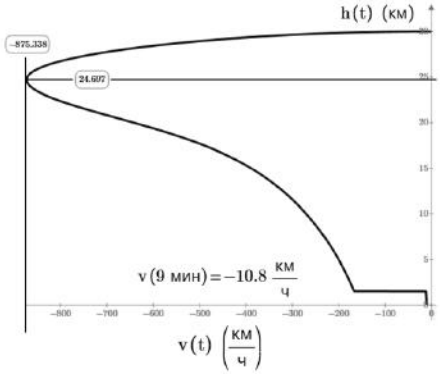


Рис. 9.12
Моделирование прыжка парашютиста

Рассмотрим такую *математическую модель*: парашютист в начале полета — это шар с радиусом r_1 и массой 110 кг. После раскрытия парашюта он превращается в шар с радиусом r_2. Масса парашютиста не меняется. Прыгает парашютист с высоты h_1, а раскрывает парашют на высоте h_2. Эти исходные данные вводятся в Mathcad-расчет в виде таблицы (рис. 9.12), у которой первая строка — это имена переменных, вторая — единицы измерения, а третья — числовые значения.

В задачу введены и дополнительные исходные данные: плотность воздуха $\rho_{возд}$ как функция высоты над уровнем моря (взято из Википедии), коэффициент трения парашютиста о воздух k (взято «из головы», но уточняется в процессе решения и сравнивания полученного ответа с реальной максимальной скоростью парашютиста) и предположительное время полета парашютиста t_{end} (это значение уточняется в процессе решения задачи). После ввода исходных данных записываются три вспомогательные функции: изменение в зависимости от высоты полета

радиуса «парашютиста» r, площади его поперечного сечения *Сечение* и его объема *Объем*. Эти функции имеют вид ступенек: до раскрытия парашюта ($h > h_2$) они возвращают одни значения, а после раскрытия — другие.

На рисунке 9.12 можно видеть дифференциальное уравнение, «уравнивающее» силы, действующие на парашютиста. Это уравнение не что иное, как математическая запись второго закона Ньютона, гласящего, что сумма сил, действующих на тело, равна произведению его массы на ускорение.

Какие силы действуют на парашютиста?

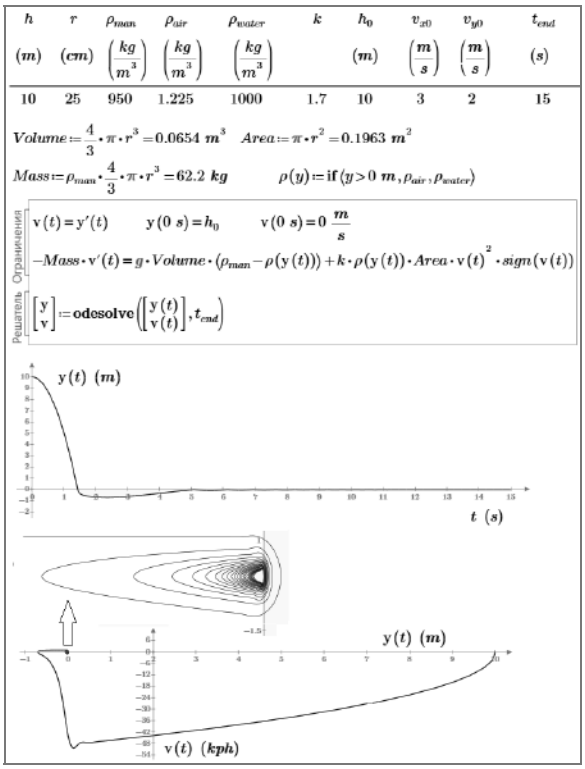
Первая сила — сила тяжести: произведение массы на ускорение свободного падения g. Эта сила направлена вниз, поэтому-то она в уравнении со знаком минус. Вторая сила — это сила сопротивления воздуха, которой обычно пренебрегают при рассмотрении классической задачи полета (падения) камня, но которой никак нельзя пренебречь в случае с парашютистом, иначе он разобьется в лепешку. Мы примем, что эта сила пропорциональна плотности воздуха, площади поперечного сечения падающего тела и квадрату его скорости. Коэффициент пропорциональности — это безразмерная переменная k. Третья сила — это сила Архимеда — вес вытесненного парашютистом воздуха. Если парашют еще не раскрыт, то этой силой можно пренебречь, но если наш парашютист раскрыл парашют, «раздувшись» (согласно нашей модели) до шара с пятиметровым диаметром и массой в сто десять килограмм, то эту силу учитывать уже нужно. Две эти силы направлены вверх, поэтому в их формулах мы видим знак плюс. Все эти три силы, повторяем, уравновешиваются силой инерции — произведением массы на ускорение.

Задачу можно усложнить — учитывать, например, изменение значения ускорения свободного падения по высоте, не шарообразную, а более сложную форму летящего парашютиста, изменение значения коэффициента k в зависимости от режима обтекания тела (ламинарный или турбулентный — все это является предметом изучения науки аэродинамики), горизонтальную составляющую полета парашютиста, связанную со скоростью самолета, из которого он выпрыгнул, и/или со скоростью ветра. Но и без этого наше решение получилось вполне правдоподобным: парашютист в свободном полете набирает максимальную скорость до 875 км/ч, его затем скорость падает вследствие роста плотности воздуха. После раскрытия парашюта скорость парашютиста резко уменьшается до 11 км/ч. С такой скоростью он и приземляется после 13 мин полета. Даже в школьном курсе физики известны задачи, рассматривающие движение тел с установившейся скоростью, которое возникает при действии на тело силы, направленной противоположно скорости тела, причем модуль силы увеличивается с увеличением скорости тела. Можно снять еще одно допущение. У нас при $h > h_2$ парашют раскрывается моментально. В реальности это происходит за одну-две секунды, что можно учесть, сделав функцию r(h) более сложной: не ступенькой, а неким пандусом.

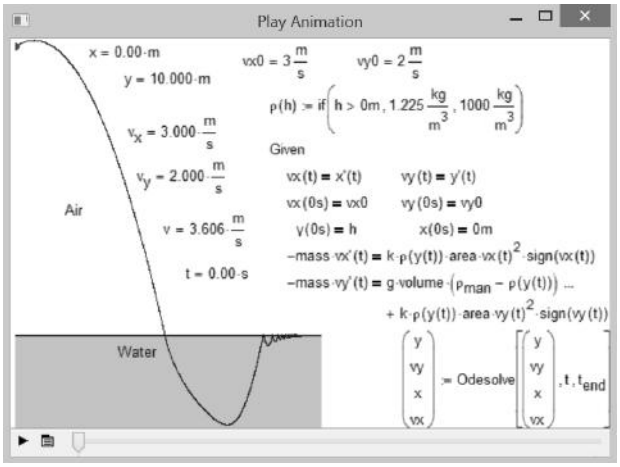
На сайте, поддерживающем этот этюд, есть расчет и анимация приземления спутника **(https://www.ptcusercommunity.com/thread/84707)** за счет кратковременного запуска двигателя, но без учета атмосферы Земли. Эту задачу несложно скрестить с задачей, показанной на рис. 9.12, и получить модель, близкую к реальности.

Пример 8. Прыжок с вышки в воду. На рисунке 9.12 была показана модель (и ее реализация в среде Mathcad) прыжка парашютиста на Землю, вернее на землю (не на планету, а на грунт). Но парашютист или космический аппарат может не только приземляться, но и приводняться. На заре космической эры советские аппараты приземлялись, а американские приводнялись. На рис. 9.13 показан расчет параметров прыжка человека в воду с вышки. Здесь также присутствует функция-ступенька, но связанная уже не с объемом «прыгуна», а с плотностью среды: воздух или вода. В расчете есть еще одна функция-ступенька — встроенная в Mathcad функция с именем sign, которая возвращает знак аргумента: минус единицу при отрицательном, нуль при нулевом и плюс единицу при положительном аргументе. Эта встроенная функция позволяет нам учитывать то положение, что сила сопро-

тивления среды всегда действует в направлении, противоположном скорости. Сам же знак значения скорости у нас пропадает из-за возведения этой величины в квадрат.



a



b

Рис. 9.13

Моделирование прыжка с вышки в воду:

a — Mathcad-решение; *b* — кадр анимации.

На рисунке 9.13*a* можно видеть графики параметров прыжка с вышки в воду. На одном из графиков отображен так называемый *аттрактор*: прыгун, вынырнув из воды, будет совершать затухающие колебательные движения на ее поверхности. Такой аттрактор можно увидеть на срезах деревянных досок с годовыми линиями. Анимация этой модели (рис. 9.13*b*) размещена по адресу **https://www.ptcusercommunity.com/videos/3597**.

Но в воду можно не только нырять — из воды можно также и «выпрыгивать». Если вернуться к главной теме этюда, то тут будет уместен еще один «ракетный» пример.

Пример 9. Старт ракеты из подводной лодки. Космические объекты могут не только приводняться, но и стартовать с поверхности воды, и даже из-под воды. На рисунке 9.14 можно видеть кадр анимации вертикального взлета и последующего падения в воду ракеты, запущенной из-под воды. Эта математическая модель отличается от предыдущих тем, что к силам, действующим на материальную точку, прибавилась сила тяги ракетного двигателя, а также тем, что масса материальной точки уменьшается вследствие расхода топлива и окислителя. Проблема применения второго закона Ньютона к объекту с переменной массой обсуждалась на сайте **https://www.ptcusercommunity.com/message/426431**.

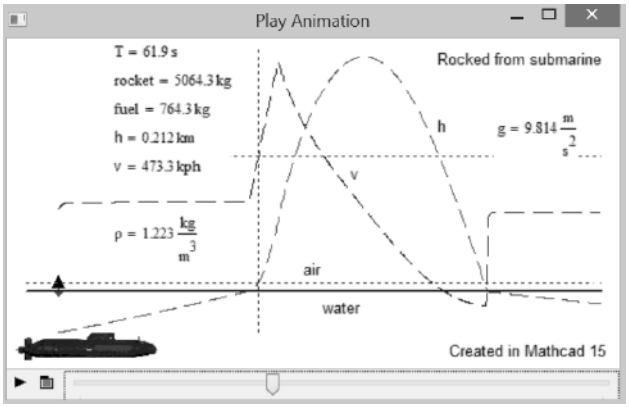


Рис. 9.14
Кадр анимации полета ракеты с подводной лодки
(https://www.ptcusercommunity.com/docs/DOC-3570)

На рисунке 9.14 показаны также графики изменения высоты подъема ракеты (h — красный пунктир) и ее скорости (v — синий пунктир) в зависимости от времени полета. На сайте **https://www.ptcusercommunity.com/thread/60347** можно видеть анимацию еще одной подобной задачи, когда спутник изменяет свою орбиту за счет кратковременного включения бортового двигателя.

Рассмотренные инструментальные средства Mathcad позволяют легко, быстро и с достаточной точностью отображать если не количественную, а только качественную сторону задачи, моделировать и анимировать различные случаи движения материальных точек и тел под действием сил гравитации, инерции, тяги двигателя и сопротивления среды. Предлагаем читателям рассмотреть такую модель: космический корабль на орбите Земли включает ракетный двигатель с известной тягой, стартует к Луне (**https://www.ptcusercommunity.com/thread/84708**) и после свободного полета при выключенном двигателе переходит на орбиту этого спутника Земли (**https://www.ptcusercommunity.com/thread/84709**). Если же тягу двигателя направить против скорости движения спутника Земли, то он «войдет в плотные слои атмосферы», сгорит или успешно приземлится, или приводнится с помощью парашюта. Элементы этих случаев движения тел мы рассмотрели выше, и их анимация есть на описываемом сайте.

При этом нужно только помнить одну важную вещь. Перефразируя известное выражение, можно сказать, что есть «ложь, наглая ложь и... численное решение дифференциальных уравнений». В этом можно еще раз убедиться, взглянув на рисунки 9.5–9.8. Мы ни в коей мере не утверждаем, что результаты численных методов не заслуживают доверия. К ним просто нужно относится очень осторожно. Как, например, к статистике, которая должна стоять в вышеприведенном крылатом выражении [59]. Численные методы решения дифференциальных уравнений уже давно успешно используется при осуществлении космических полетов. При этом данные телеметрии используют в том числе и для коррекции движения, которое было численно рассчитано заранее с определенной, часто недостаточной точностью. При этом и начальная скорость аппарата, заданная маршевыми двигателями, может отличаться от задуманной. Так и мы, принимая к исполнению, чью-то просьбу или приказание, понимаем, что в этой просьбе или приказании может быть вольная или невольная ложь, которую мы нивелируем в процессе выполнения просьбы, получая дополнительную информацию и принимая самостоятельные решения.

В настоящее время многие физические кабинеты школ и университетов оборудованы компьютерами с мультимедийными проекторами. В таком кабинете можно проводить физические опыты (изучать, например, колебание маятника), показывая на большом экране эксперимент так, чтобы всем было хорошо все видно. Но на этом же экране можно показывать решение дифференциального уравнения колебания маятника, сравнивать реальное физическое явление с его математической моделью, объясняя расхождения ограничениями и упрощениями модели.

Уравнения — и алгебраические, и дифференциальные — страшны для школьников и студентов не сами по себе, а методами их решения. Сейчас на компьютере решать такие уравнения можно довольно просто. Главное — составить уравнение или систему уравнений, понимая физику задачи, и правильно интерпретируя результаты расчетов... Поэтому такие задачи могут стать не «пытками», а удовольствием и для учеников, и для учителей...

На сайте: **http://communities.ptc.com/groups/dynamic-models-in-mathcad** авторы разместили большое количество различных динамических задач с их решениями в среде Mathcad [60] и анимацией средствами Mathcad (см. [44] и этюд 8). Вот некоторые из них:

- колебание одиночного маятника;
- колебание связанных маятников;
- вращение планет со спутниками (тема данного этюда; этим решениям посвящен отдельный подфорум **https://www.ptcusercommunity.com/groups/three-body-problem-with-animation**);
- старт ракеты из подводной лодки;
- ныряние человека в воду;
- скатывание саней с горки;
- движение автомобиля;
- движение по подземному прямому тоннелю гравитационного поезда [61, 62] и др.

ДИВЕРТИСМЕНТ 1. БЕЗЗАКОННАЯ КОМЕТА

«Как беззаконная комета в кругу расчисленных светил...»

А. С. Пушкин «Портрет»

На рисунке 9.11 отображено движение спутника по эллиптической орбите. Но есть небесные тела («беззаконные кометы», например), которые движутся по иным траекториям, и это отображено в эпиграфе к дивертисменту.

А траектории, повторяем, бывают такие: эллипс (частный случай — окружность), гипербола и парабола (редкий переходный случай от гиперболы к эллипсу).

Из наших расчетов, основанных на численном решении ОДУ с начальными условиями, часто можно видеть только какую-то дугу траектории, по которой трудно определить, что это — эллипс, гипербола или парабола. Но это можно понять, если вспомнить, что рассматриваемые кривые — это *плоские кривые второго порядка*. В уравнениях этих кривых (см. верхнюю часть рисунка 9.17) шесть коэффициентов. Если попытаться их вычислить, то по ним (графически или через вычисление инвариантов) можно определить тип кривой.

Тут на ум приходит еще одна цитата из А. С. Пушкина, но уже из «Каменного гостя»:

Дон Гуан:
Ее совсем не видно
Под этим вдовьим черным покрывалом,
Чуть узенькую пятку я заметил.
Лепорелло:
Довольно с вас. У вас воображенье
В минуту дорисует остальное;
Оно у нас проворней живописца,
Вам все равно, с чего бы ни начать,
С бровей ли, с ног ли.

Но у нас есть не только «воображенье», но и мощные вычислительные средства решения систем алгебраических уравнений (см. рис. 9.17). «Узенькая пятка» — это наши точки численного решение системы ОДУ (см. рис. 9.15 и 9.16), а «остальное» — это вся гипербола, вернее, коэффициенты ее уравнения. А то, что это гипербола, доказывает значение инварианта `D`, которое меньше нуля (у эллипса оно больше нуля, а у параболы равно нулю), и соответствующие графики на рисунке 9.18.

На рисунке 9.15 показан блок исходных данных и блок численного решения системы ОДУ, отображающий движение материальной точки (спутника) вблизи другой материальной точки (Земли) с массой, намного превышающей массу первой точки так, чтобы перемещением второй точки можно было пренебречь[14].

На рисунке 9.16 показано графическое отображение решения задачи о движении небесного тела малой массы (астероида, например) вблизи Земли за 12 ч.

Из рисунка 9.16 довольно трудно понять, чью дугу (часть) мы получили — эллипса, гиперболы или параболы. Можно увеличивать предел интегрирования[15] (увеличивать значение переменной t_{end}) и видеть, будет ли кривая загибаться (сворачиваться) в эллипс или уходить в бесконечность, помня при этом о точности решения (см. рисунки выше). Но можно поступить иначе — восстановить всю кривую второго порядка по коэффициентам ее уравнения. Как это можно сделать, показано на рисунке 9.17.

G	$Earth$	R	x_0	y_0	v_{x0}	v_{y0}	t_{end}
$\left(\frac{N \cdot m^2}{kg^2}\right)$	$(tonne)$	(km)			$\left(\frac{km}{s}\right)$	$\left(\frac{km}{s}\right)$	(hr)
$6.67384 \cdot 10^{-11}$	$5.972 \cdot 10^{21}$	6370	$2R$	$2R$	-0.5	-7	12

$$x(0\ s) = x_0 \qquad x'(0\ s) = v_{x0} \qquad x''(t) = -G \cdot \frac{Earth \cdot x(t)}{\left(\sqrt{x(t)^2 + y(t)^2}\right)^3}$$

$$y(0\ s) = y_0 \qquad y'(0\ s) = v_{y0} \qquad y''(t) = -G \cdot \frac{Earth \cdot y(t)}{\left(\sqrt{x(t)^2 + y(t)^2}\right)^3}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} := \mathbf{Odesolve}\left(\begin{bmatrix} x(t) \\ y(t) \end{bmatrix}, t_{end}\right)$$

Рис. 9.15
Блоки исходных данных и численного решения системы ОДУ

[14] В задаче, показанной на рисунке 9.11, мы этого не делали.

[15] Но, повторяем, нужно помнить о том, что это часто приводит к накоплению ошибки — см. рисунки 9.4*b* и 9.5*c* выше.

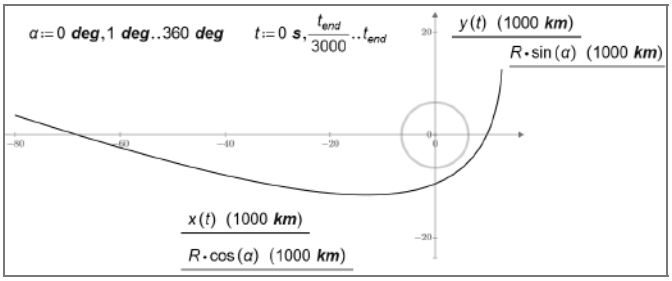


Рис. 9.16

Графическое отображение численного решения системы ОДУ

$$a_{11}\cdot x^2+2\,a_{12}\cdot x\cdot y+a_{22}\cdot y^2+2\,a_1\cdot x+2\,a_2\cdot y+a_0=0$$

Начальные приближения

$$[a_{11}\ a_{12}\ a_{22}\ a_1\ a_2\ a_0]:=\left[\frac{1}{m^2}\ \frac{1}{m^2}\ \frac{1}{m^2}\ \frac{1}{m}\ \frac{1}{m}\ 1\right]$$

Ограничения

$$a_{11}\cdot x(0\,hr)^2+2\,a_{12}\cdot x(0\,hr)\cdot y(0\,hr)+a_{22}\cdot y(0\,hr)^2+2\,a_1\cdot x(0\,hr)+2\,a_2\cdot y(0\,hr)+a_0=0$$
$$a_{11}\cdot x(1\,hr)^2+2\,a_{12}\cdot x(1\,hr)\cdot y(1\,hr)+a_{22}\cdot y(1\,hr)^2+2\,a_1\cdot x(1\,hr)+2\,a_2\cdot y(1\,hr)+a_0=0$$
$$a_{11}\cdot x(2\,hr)^2+2\,a_{12}\cdot x(2\,hr)\cdot y(2\,hr)+a_{22}\cdot y(2\,hr)^2+2\,a_1\cdot x(2\,hr)+2\,a_2\cdot y(2\,hr)+a_0=0$$
$$a_{11}\cdot x(3\,hr)^2+2\,a_{12}\cdot x(3\,hr)\cdot y(3\,hr)+a_{22}\cdot y(3\,hr)^2+2\,a_1\cdot x(3\,hr)+2\,a_2\cdot y(3\,hr)+a_0=0$$
$$a_{11}\cdot x(4\,hr)^2+2\,a_{12}\cdot x(4\,hr)\cdot y(4\,hr)+a_{22}\cdot y(4\,hr)^2+2\,a_1\cdot x(4\,hr)+2\,a_2\cdot y(4\,hr)+a_0=0$$
$$a_{11}\cdot x(5\,hr)^2+2\,a_{12}\cdot x(5\,hr)\cdot y(5\,hr)+a_{22}\cdot y(5\,hr)^2+2\,a_1\cdot x(5\,hr)+2\,a_2\cdot y(5\,hr)+a_0=0$$
$$a_{11}\cdot x(6\,hr)^2+2\,a_{12}\cdot x(6\,hr)\cdot y(6\,hr)+a_{22}\cdot y(6\,hr)^2+2\,a_1\cdot x(6\,hr)+2\,a_2\cdot y(6\,hr)+a_0=0$$
$$a_{11}\cdot x(7\,hr)^2+2\,a_{12}\cdot x(7\,hr)\cdot y(7\,hr)+a_{22}\cdot y(7\,hr)^2+2\,a_1\cdot x(7\,hr)+2\,a_2\cdot y(7\,hr)+a_0=0$$
$$a_{11}\cdot x(8\,hr)^2+2\,a_{12}\cdot x(8\,hr)\cdot y(8\,hr)+a_{22}\cdot y(8\,hr)^2+2\,a_1\cdot x(8\,hr)+2\,a_2\cdot y(8\,hr)+a_0=0$$
$$a_{11}\cdot x(9\,hr)^2+2\,a_{12}\cdot x(9\,hr)\cdot y(9\,hr)+a_{22}\cdot y(9\,hr)^2+2\,a_1\cdot x(9\,hr)+2\,a_2\cdot y(9\,hr)+a_0=0$$
$$a_{11}\cdot x(10\,hr)^2+2\,a_{12}\cdot x(10\,hr)\cdot y(10\,hr)+a_{22}\cdot y(10\,hr)^2+2\,a_1\cdot x(10\,hr)+2\,a_2\cdot y(10\,hr)+a_0=0$$
$$a_{11}\cdot x(11\,hr)^2+2\,a_{12}\cdot x(11\,hr)\cdot y(11\,hr)+a_{22}\cdot y(11\,hr)^2+2\,a_1\cdot x(11\,hr)+2\,a_2\cdot y(11\,hr)+a_0=0$$
$$a_{11}\cdot x(12\,hr)^2+2\,a_{12}\cdot x(12\,hr)\cdot y(12\,hr)+a_{22}\cdot y(12\,hr)^2+2\,a_1\cdot x(12\,hr)+2\,a_2\cdot y(12\,hr)+a_0=0$$

Решатель

$$\begin{bmatrix}a_{11}\\a_{12}\\a_{22}\\a_1\\a_2\\a_0\end{bmatrix}:=\mathbf{Find}\,(a_{11},a_{12},a_{22},a_1,a_2,a_0)=\begin{bmatrix}(101.180524192814\cdot 10^{-15})\,\frac{1}{m^2}\\(138.877437855728\cdot 10^{-15})\,\frac{1}{m^2}\\(78.4361030434229\cdot 10^{-15})\,\frac{1}{m^2}\\(2.9463522233284\cdot 10^{-6})\,\frac{1}{m}\\-3.1974809672896\cdot 10^{-6}\,\frac{1}{m}\\-67.8361105562363\end{bmatrix}$$

а

Решить $a_0 := -1$

Ограничения

$$[a_{11}\ a_{12}\ a_{22}\ a_1\ a_2] := \left[\frac{1}{m^2}\ \frac{1}{m^2}\ \frac{1}{m^2}\ \frac{1}{m}\ \frac{1}{m}\right]$$

$$a_{11}\cdot x(8\,hr)^2 + 2\,a_{12}\cdot x(8\,hr)\cdot y(8\,hr) + a_{22}\cdot y(8\,hr)^2 + 2\,a_1\cdot x(8\,hr) + 2\,a_2\cdot y(8\,hr) = -a_0$$

$$a_{11}\cdot x(9\,hr)^2 + 2\,a_{12}\cdot x(9\,hr)\cdot y(9\,hr) + a_{22}\cdot y(9\,hr)^2 + 2\,a_1\cdot x(9\,hr) + 2\,a_2\cdot y(9\,hr) = -a_0$$

$$a_{11}\cdot x(10\,hr)^2 + 2\,a_{12}\cdot x(10\,hr)\cdot y(10\,hr) + a_{22}\cdot y(10\,hr)^2 + 2\,a_1\cdot x(10\,hr) + 2\,a_2\cdot y(10\,hr) = -a_0$$

$$a_{11}\cdot x(11\,hr)^2 + 2\,a_{12}\cdot x(11\,hr)\cdot y(11\,hr) + a_{22}\cdot y(11\,hr)^2 + 2\,a_1\cdot x(11\,hr) + 2\,a_2\cdot y(11\,hr) = -a_0$$

$$a_{11}\cdot x(12\,hr)^2 + 2\,a_{12}\cdot x(12\,hr)\cdot y(12\,hr) + a_{22}\cdot y(12\,hr)^2 + 2\,a_1\cdot x(12\,hr) + 2\,a_2\cdot y(12\,hr) = -a_0$$

Решатель

$$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{22} \\ a_1 \\ a_2 \end{bmatrix} := \mathbf{Find}(a_{11}, a_{12}, a_{22}, a_1, a_2) = \begin{bmatrix} (1.4915434245398\cdot 10^{-15})\ \frac{1}{m^2} \\ (2.0472489722224\cdot 10^{-15})\ \frac{1}{m^2} \\ (1.1562583734727\cdot 10^{-15})\ \frac{1}{m^2} \\ (43.4333798629103\cdot 10^{-9})\ \frac{1}{m} \\ -47.1353805448646\cdot 10^{-9}\ \frac{1}{m} \end{bmatrix} \quad a_0 = -1$$

b

$$M := \begin{bmatrix} x(7\,hr)^2 & 2\,x(7\,hr)\cdot y(7\,hr) & y(7\,hr)^2 & 2\,x(7\,hr) & 2\,y(7\,hr) \\ x(8\,hr)^2 & 2\,x(8\,hr)\cdot y(8\,hr) & y(8\,hr)^2 & 2\,x(8\,hr) & 2\,y(8\,hr) \\ x(9\,hr)^2 & 2\,x(9\,hr)\cdot y(9\,hr) & y(9\,hr)^2 & 2\,x(9\,hr) & 2\,y(9\,hr) \\ x(10\,hr)^2 & 2\,x(10\,hr)\cdot y(10\,hr) & y(10\,hr)^2 & 2\,x(10\,hr) & 2\,y(10\,hr) \\ x(11\,hr)^2 & 2\,x(11\,hr)\cdot y(11\,hr) & y(11\,hr)^2 & 2\,x(11\,hr) & 2\,y(11\,hr) \end{bmatrix}$$

$$v := \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad M^{-1}\cdot v = \begin{bmatrix} (1.49154295494464\cdot 10^{-15})\ \frac{1}{m^2} \\ (2.04724825269486\cdot 10^{-15})\ \frac{1}{m^2} \\ (1.15625764036381\cdot 10^{-15})\ \frac{1}{m^2} \\ (43.4333638807231\cdot 10^{-9})\ \frac{1}{m} \\ -47.1353781403196\cdot 10^{-9}\ \frac{1}{m} \end{bmatrix}$$

c

Рис. 9.17

Численное решение системы алгебраических уравнений:

a — решение однородной системы из 13 уравнений с множеством пропорциональных решений; *b* — решение системы пяти уравнений с пятью неизвестными и единственным решением; *c* — решение системы линейных алгебраических уравнений с пятью неизвестными и единственным решением.

На рисунке 9.17 показано три решения. Первое (рис. 9.17*a*) можно назвать «лобовой атакой»: даны начальные приближения (предположения) к решению, записаны тринадцать уравнений, фиксирующих положение спутника в начальный момент времени (`t = 0`), после часа полета, после 2 ч полета и так далее до 12 ч полета и вызвана функция `Find`, выдавшая значения своих аргументов, обращающих уравнения в тождества. Если дать иные начальные приближения к решению, то будет выдан иной ответ, который тоже будет правильный. При этом значения шести неизвестных коэффициентов при первом наборе начальных приближений будут пропорциональны значениям этих же неизвестных при втором наборе начальных приближений. И это понятно, поскольку все коэффициенты в уравнении кривой, приведенной на рис. 9.17*a*, можно умножить на одно и то же число и получить новое уравнение относительно `x` и `y`, равносильное исходному. Это свойство позволяет заранее зафиксировать значение одного из коэффициентов и найти оставшиеся. Таким образом, задавая различные начальные приближения для всех

шести неизвестных коэффициентов, мы получаем различные, пропорциональные друг другу наборы коэффициентов, определяющих уравнение одной и той же кривой второго порядка.

На рисунках 9.17*b*, *c* в задаче оставлено только пять уравнений, т. к. известно, что кривую второго порядка можно восстановить по пяти точкам. При этом коэффициенту a_0 присвоено значение минус единицы. В этом случае решение будет единственным, и начальные приближения для оставшихся пяти коэффициентов уже не повлияют на ответ (см. рис. 9.17*b*), а саму задачу можно будет свести к решению системы *линейных* алгебраических уравнений, не требующему начальных приближений (см. рис. 9.17*c*). Линейную систему в среде Mathcad тоже можно решить с помощью функции `Find` (см. рис. 9.17*b*), но для этой задачи в среде Mathcad есть специализированные средства, например, умножение инвертированной матрицы коэффициентов при неизвестных на вектор свободных членов (см. рис. 9.17*c*).

На рисунке 9.18, во-первых, показано, что главный инвариант `D` уравнения плоской кривой второго порядка меньше нуля, и, во-вторых, построены две половинки гиперболы — верхняя (`top`) и нижняя (`up`). Аналитические выражения для них получены в результате решения уравнения кривой относительно переменной `y` и генерации двух функций с именем y_{top} и y_{up}. На верхнюю ветвь гиперболы нанизаны наши красные исходные точки.

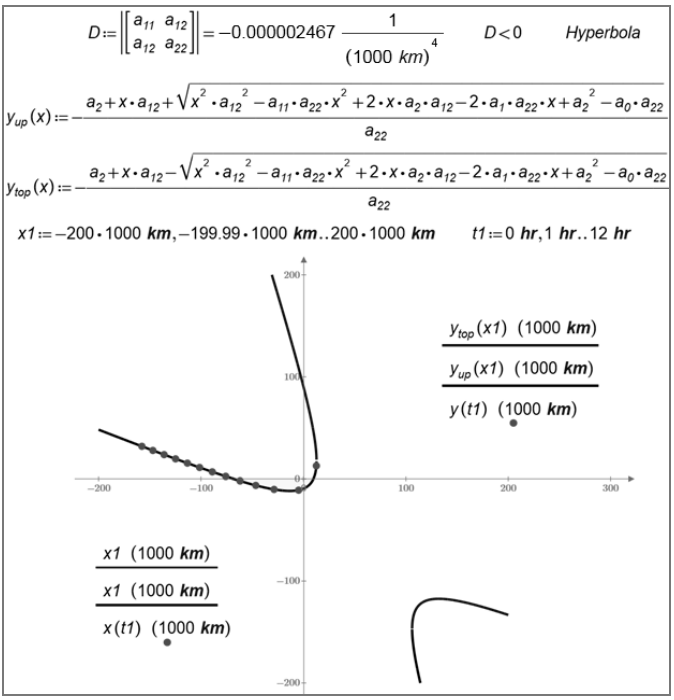


Рис. 9.18

Построение гиперболы по ее уравнению

Данный расчет можно продолжить, найдя, например, значения коэффициентов `a` и `b` канонического уравнения гиперболы:

$x^2 / a^2 - y^2 / b^2 = 1$

Первым шагом такого преобразования будет нахождение угла, на который нужно повернуть оси декартового графика, — см. рис. 9.19.

$$\phi := \mathbf{root}\left(\tan(\phi)^2 - \frac{a_{11}-a_{22}}{a_{12}}\tan(\phi) - 1, \phi, 0\ deg, 90\ deg\right) = 47.341\ deg$$

Рис. 9.19
Угол поворота осей

Предлагаем читателю самому закончить такое преобразование, имея под рукой все шесть коэффициентов a уравнения плоской кривой второго порядка. Угол поворота двух ветвей гиперболы мы уже нашли (см. рис. 9.19 — поиск одного из двух из корней квадратного уравнения в интервале от 0 до 90 угловых градусов). Осталось только определить значение сдвига ветвей гиперболы, чтобы она приняла канонический вид, симметричный относительно осей и начала координат.

Пять левых красных точек, показанные на рис. 9.18, можно рассматривать как данные телеметрии движения некоего астероида к Земле. Ученые, имея эти данные, могут рассчитать, как дальше будет лететь этот астероид: пролетит ли он мимо Земли (а именно это показано на рис. 9.16 и 9.18) или столкнется с Землей. Для предотвращения такой глобальной катастрофы на данный космический объект может быть, к примеру, послана команда для взрыва этого астероида — см. кинофильм «Армагеддон».

Когда этюд писался (май 2015), появилось сообщение, что космический грузовик «Прогресс», запущенный к Международной космической станции, не вышел на расчетную орбиту и должен через некоторое время упасть на Землю. Это побудило авторов попытаться смоделировать такое поведение неуправляемого искусственного спутника Земли — см. рис. 9.20–9.22.

На рисунке 9.20 в среде Mathcad Prime вводятся исходные данные: гравитационная постоянная G, параметры Земли (ее масса m_e и радиус R_e), параметры искусственного спутника (его масса m_s, плотность ρ_e и высота над поверхностью Земли в начальный момент h_0), коэффициент трения спутника о воздух k и время полета t_{end}. Далее в расчете, показанном на рис. 9.20, вводится функциональная зависимость плотности воздуха от высоты $\rho(h)$ (см. также рис. 9.12), рассчитывается скорость вращения спутника по круговой орбите v_0 на высоте h_0 над уровнем Земли (первая космическая скорость) и параметры искусственного спутника: его объем, диаметр и площадь поперечного сечения в предположении, что это шар с радиусом R_e и плотностью ρ_e.

G	m_e	R_e	m_s	ρ_s	h_0	k	t_{end}
$\left(\frac{N\cdot m^2}{kg^2}\right)$	$(tonne)$	(km)	$(tonne)$	$\left(\frac{kg}{m^3}\right)$	(km)		(hr)
$6.67384\cdot 10^{-11}$	$5.972\cdot 10^{21}$	6370	10	800	40	0.00003025	4.25897

$$\rho(h) := \left\| \begin{array}{l} L \leftarrow 0.0065\ \frac{K}{m} \\ T_0 \leftarrow 288.15\ K \\ T \leftarrow T_0 - L\cdot h \\ M \leftarrow 28.9644\ \frac{gm}{mole} \\ p \leftarrow 101325\ Pa\cdot\left(1-\frac{L\cdot h}{T_0}\right)^{\frac{g\cdot M}{R\cdot L}} \\ \rho_{возд} \leftarrow \frac{p\cdot M}{R\cdot T} \end{array} \right.$$

Air density

$$\rho(0\ m) = 1.225\ \frac{kg}{m^3} \qquad \rho(h_0) = 0.000061545\ \frac{kg}{m^3}$$

$$v_0 := \sqrt{\frac{G\cdot m_e}{R_e + h_0}} = 7.885\ \frac{km}{s}$$ Start satelitte velosity

Volume, diameter and section of the satelitte

$$V_s := \frac{m_s}{\rho_s} = 12.5\ m^3 \qquad d_s := \sqrt[3]{\frac{6\cdot V_s}{\pi}} = 2.879\ m \qquad S_s := \pi\cdot\left(\frac{d_s}{2}\right)^2 = 6.512\ m^2$$

Рис. 9.20
Моделирование торможения спутника в плотных слоях атмосферы: начало расчета

На рисунке 9.21 показаны операторы решения системы четырех обыкновенных дифференциальных уравнений, моделирующих движение и снижение искус-

ственного спутника на орбите Земли. К силе космического притяжения, определяемого законом всемирного тяготения, прибавилась сила торможения, связанная с трением о воздух. Эта сила пропорциональна (коэффициент k) площади поперечного сечения спутника S_s, плотности воздуха ρ и квадрату скорости (см. также рисунки 9.12, 9.13*a*, *b*). При этом в расчет вставлен не квадрат скорости, а произведение скорости на абсолютное значение скорости для того, чтобы правильно определить направление этой силы — она действует в направлении противоположном направлению вектора скорости.

Solve

$$\begin{cases} x(0\ s) = 0\ m \\ v_x(0\ s) = v_0 \\ y(0\ s) = R_e + h_0 \\ v_y(0\ s) = 0\ \dfrac{m}{s} \end{cases} \quad \begin{aligned} v_x(t) &= x'(t) & m_s \cdot v_x'(t) &= -G \cdot \frac{m_s \cdot m_e \cdot x(t)}{\left(\sqrt{x(t)^2 + y(t)^2}\right)^3} - k \cdot v_x(t) \cdot |v_x(t)| \cdot S_s \cdot \rho\left(\sqrt{x(t)^2 + y(t)^2} - R_e\right) \\ v_y(t) &= y'(t) & m_s \cdot v_y'(t) &= -G \cdot \frac{m_s \cdot m_e \cdot y(t)}{\left(\sqrt{x(t)^2 + y(t)^2}\right)^3} - k \cdot v_y(t) \cdot |v_y(t)| \cdot S_s \cdot \rho\left(\sqrt{x(t)^2 + y(t)^2} - R_e\right) \end{aligned}$$

$$\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} := \mathbf{Odesolve}\left(\begin{bmatrix} x(t) \\ y(t) \\ v_x(t) \\ v_y(t) \end{bmatrix}, t_{end}, 10^5\right)$$

Рис. 9.21

Моделирование торможения спутника в плотных слоях атмосферы: продолжение расчета

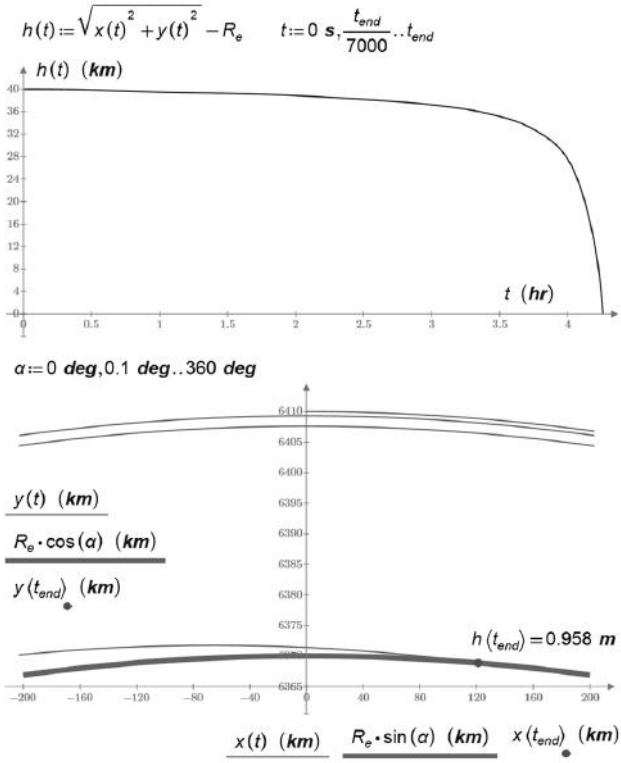


Рис. 9.22

Моделирование торможения спутника в плотных слоях атмосферы: окончание расчета

На рисунке 9.22 показано графическое отображение решения задачи о снижении орбиты спутника до момента касания поверхности Земли: зависимость высоты над уровнем моря h(t) от времени (верхний график) и фрагменты траектории орбиты (нижний график, где дуга y оси x — это поверхность Земли, куда падает спутник с высоты 6410 км, сделав три витка).

На сайте

https://www.ptcusercommunity.com/groups/three-body-problem-with-animation

есть модель и анимация случая, обратного, показанного на рис. 9.22, когда за счет включения двигателя спутник переходит на новую, более высокую орбиту.

Все файлы, приведенных в этюде расчетов (Mathcad 15 и Mathcad Prime 3), можно скачать с вышеотмеченного сайта. Там же размещены соответствующие анимации. Отдельные их адреса и кадры были приведены выше.

И ПРЕДПОСЛЕДНЕЕ ДЛЯ САМЫХ ЛЮБОПЫТНЫХ И ДОТОШНЫХ

Какие алгоритмы заложены в численные методы решения обыкновенных дифференциальных уравнений?

Самый простой метод решения таких задач предложил еще великий Эйлер[16]. Если говорить о задаче с начальными условиями[17] (задача Коши), то метод Эйлера реализуется так. Если известно начальное (первое) значение искомой функции, то отрезок интегрирования разбивается на n коротких отрезков длиной Δ и рассчитывается значение искомой функции во второй точке по значению производной в первой точке — а это правая часть дифференциального уравнения: $y_2 = y_1 + y'_1 \cdot \Delta$. Эта операция проводится в цикле ($y_i = y_{i-1} + y'_{i-1} \cdot \Delta$)[18], в котором формируются два вектора, хранящие дискретные значения аргумента и искомой функции. Если число точек n достаточно большое и искомая функция не слишком «крутая»[19], то метод Эйлера дает вполне приемлемые решения, хотя он и не встроен в Mathcad (см. список встроенных методов на рис. 9.2). Мы сейчас описали приложение метода Эйлера к одному дифференциальному уравнению первого порядка. В наших же задачах небесной механики рассматривалась система уравнений второго порядка: в математических моделях фигурировало ускорение — вторая производная пути по времени. Но метод Эйлера можно применить не только к одному уравнению, но и к системе, а одно уравнение второго порядка можно преобразовать в два уравнения первого порядка. Такое решение показано на рис. 9.23 применительно к еще одной классической задаче, связанной законом всемирного тяготения и силой тяжести, — к задаче о колебании маятника[20].

---

[16] Сейчас у нас все чаще говорят не Эйлер, а как во всем мире — Ойлер. Такая метаморфоза случилась и с другим великим математиком — с Ньютоном. Сначала (по времена Ломоносова) его у нас называли Невтоном (как написано (Newton), так и произносилось), затем Ньютоном с ударением на последнем слоге, а сейчас как в Англии — с ударением на первом слоге.

[17] А еще есть так называемая краевая задача, когда условия задаются на концах отрезка интегрирования (см. рис. 7.5 на стр. 144). Ниже будет кратко описан один из способов решения этой задачи.

[18] В программах на рисунках 9.23 и 9.25 нижний индекс переменной (элемент вектора) заменен на «верхний индекс» (столбец матрицы с одной строкой). Это было сделано для того, чтобы выводилась итоговая матрица решений. Эти нюансы было описаны на форуме

https://www. ptcusercommunity.com/thread/85205.

[19] Под словом «крутой» можно понимать очень многое. Это как в молодежном сленге. В наших задачах спутники и планеты вели себя «круто» и ставили в тупик довольно сложные методы решения дифференциальных уравнений.

[20] На сайте PTC Community можно найти авторскую анимацию одиночного маятника в вязкой среде, на эластичной нити, а также анимации двух, трех и четырех связанных маятников на жесткой несгибаемой нити. Маятники при этом описывают замысловатые кривые, подобные тем, какие показаны на рис. 9.10. И там также можно четко видеть срыв численного решения.

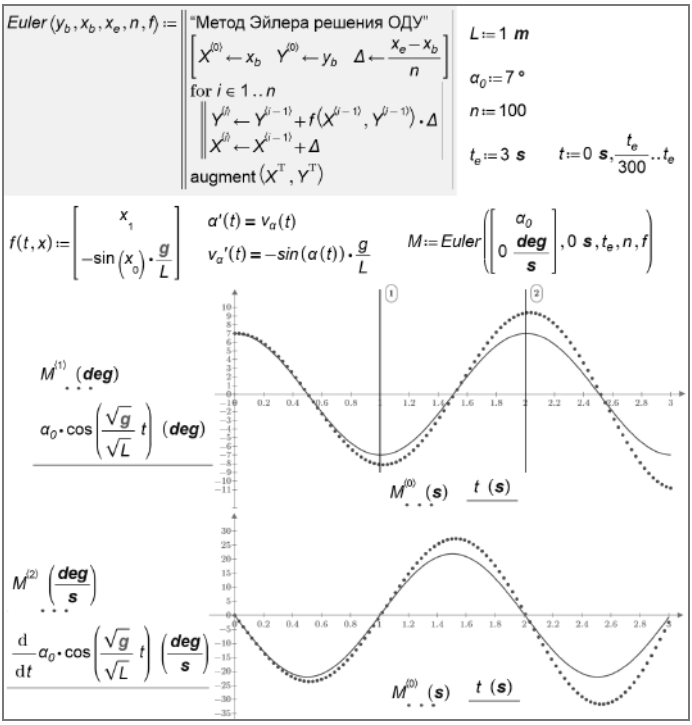
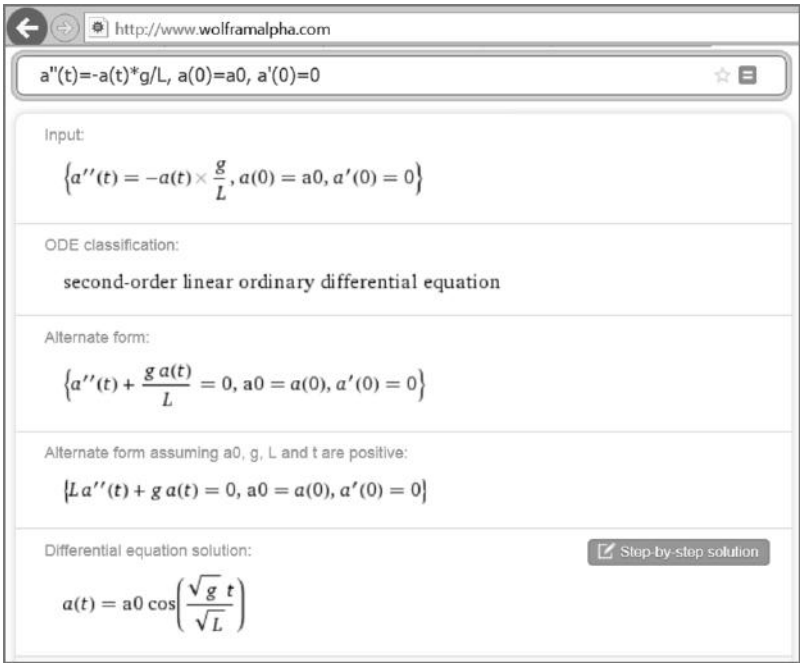


Рис. 9.23
Метод Эйлера, примененный для решения системы двух обыкновенных дифференциальных уравнений

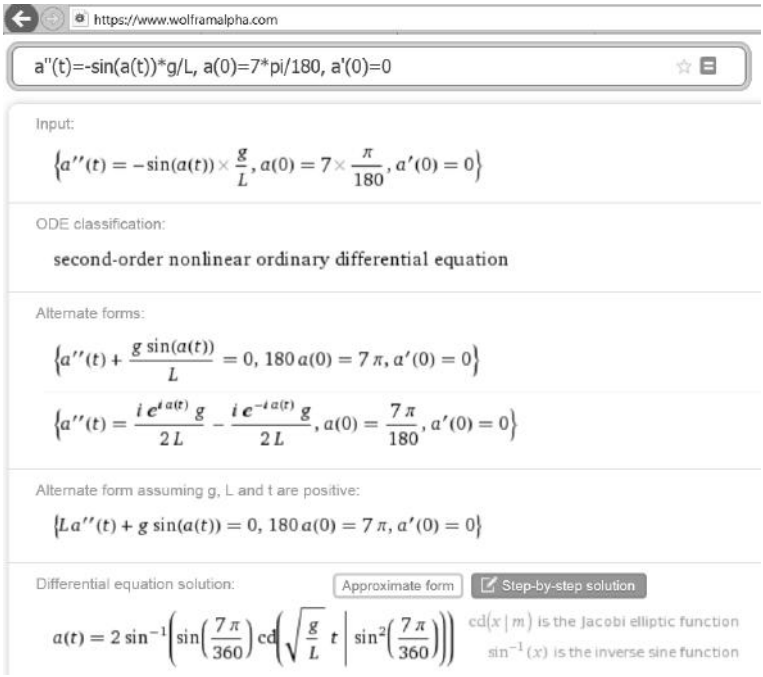
Математическая модель колебания маятника учитывает тот факт, что его угловое ускорение пропорционально произведению ускорения свободного падения на синус угла отклонения нити маятника от вертикали и обратно пропорционально длине нити маятника. Если синус угла заменить на сам угол (а это можно сделать при значениях угла по модулю меньших, чем приблизительно 7°), то такое уравнение (уравнение *математического* маятника) можно решить аналитически — см. рис. 9.24*a*, где для этого использован сайт программы Mathematica.

Если же в уравнение движения маятника вернуть синус, то решение на вышеотмеченном сайте также будет получено, но аналитическим его можно назвать с натяжкой, т. к. в нем будет присутствовать *специальная* функция — эллиптическая функция Якоби (`cd` — см. рис. 9.24*b*). Забегая вперед к похожей задаче — задаче о гравитационном поезде, можно отметить, что там синуса в уравнении нет изначально (см. рис. 9.30). Но в уравнение движения такого полуфантастического поезда можно ввести силу сопротивления воздуха, пропорциональную его скорости в квадрате. В этом случае в решении на рис. 9.30 появится еще одна специальная функция — функция Бесселя, которая, кстати встроена в Mathcad[21]. Под аналитическим решением обычно понимают решение только с элементарными функциями. Но термин «аналитический» имеет слишком много различных толкований. Многие полагают, что решение с помощью любых известных спецфункций и даже любых рядов (и даже интегралов) считается аналитическим. Компьютер только добавил разночтения в этом вопросе.

[21] Был проект: переиздать репринтно «старые добрые советские справочники», в том числе и по функциям Бесселя. Но тут же нашлись люди, которые то ли в шутку, то ли всерьез предложили переиздать и знаменитые таблица Брадиса с логарифмами, тригонометрическими и прочими элементарными функциями, на которых «выросло несколько поколений советских людей».



a



b

Рис. 9.24

Аналитическое решение дифференциального уравнения маятника:
a — математического; *b* — физического.

Но вернемся к численным методам решения.

Из рисунка 9.23 видно, что при `n = 100` (число разбиений интервала интегрирования) решение системы двух ОДУ методом Эйлера получилось довольно грубое, значительно отклоняющееся от истинного. Мы специально на графике отметили линией «истинное» (аналитическое) решение, а численное (приближенное) отметили точками, чтобы подчеркнуть еще раз, что методом Эйлера получена не искомая функция, а набор точек, которые еще как-то нужно преобразовать в функцию, используя, например, методы интерполяции [59], которые мы рассмотрим в этюде 14.

Наша функция `Euler` (рис. 9.23) в случае одного дифференциального уравнения первого порядка возвращает искомые точки в виде матрицы с двумя столбцами — дискретные значения аргумента и соответствующие дискретные значения функции. Если же мы имеем не одно уравнение, а систему уравнений, то в возвещаемой матрице появятся дополнительные столбцы. На рисунке 9.23 специально показаны два графика (график отклонения по времени нити маятника от вертикали и график угловой скорости маятника) с «извлечением» нужных столбцов матрицы `M`: $M^{<0>}$, $M^{<1>}$ и $M^{<0>}$.

Если в решении, показанном на рис. 9.23, значение `n` увеличить со 100 до 1000 или до еще большего значения, то точки на графике сольются в линию (иллюзия аналитического решения с функцией в ответе) и лягут строго на кривую — решение будет более-менее точным. Но это соответственно увеличит время счета[22]. Из-за этого и по другими причинам (из-за «крутизны», т. е. очень быстрого изменения искомых функций на некоторых участках) были предложены более «хитрые» методы численного решения ОДУ. Самый известный из них разработан в 1900 г. немецкими математиками К. Рунге и М. Куттой. Метод Рунге — Кутты (рис. 9.25) ведет расчет очередной точки значения искомой функции с некой «пристрелкой» на половинном значении Δ. Он встроен практически во все компьютерные вычислительные программы, включая и пакет Mathcad — см. выше рис. 9.2. Потом (в XX в.) были разработаны еще более сложные алгоритмы численного решения ОДУ — метод Рунге — Кутты с переменным шагом Δ, метод Adams, метод BDF, метод Radau и т. д. Мы с этими методами «поиграли», решая задачу о перехвате спутника — см. рис. 9.8. Выбор метода решения ОДУ для конкретной задачи — это тонкая наука и высокое искусство. В функцию `Odesolve`, которую мы использовали в расчетах, дополнительно заложен алгоритм интерполяции, чтобы выдавалась не матрица дискретных значений, а именно функция (набор функций). Из-за этого иногда возникает недопонимание — некоторые считают, что если эта встроенная функция возвращает не таблицу, а искомую функцию, то получено аналитическое решение задачи.

Если же нужно решить не задачу с начальными условиями (задачу Коши), а краевую задачу, то применяют метод стрельбы. Краевая задача применительно к задаче о колебании маятника может быть сформулирована так: известны значения угла отклонения нити маятника от вертикали в два момента времени — определить положение нити в другие моменты времени. Метод стрельбы заключается в следующем. Задается недостающее начальное условие (угловая скорость нити, если иметь в виду задачу о маятнике). Задача при этом сводится[23] к известной задаче с начальными условиями, решая которую находят нужное значение на другом конце отрезка интегрирования. Если оно не совпадает с изначально заданным

[22] Сейчас на мощных компьютерах такое изменение во времени счета почти не чувствуется, но в те времена, когда создавались описываемые методы и когда вообще не было компьютеров, эта разница во времени счета была существенной и даже определяющей при выборе того или иного метода решения.

[23] Как вскипятить воду в чайнике? Ответ: нужно налить в него воды и поставить на огонь. Новая задача: в чайнике уже есть вода. Ответ математика: воду нужно вылить. После этого задача сводится к предыдущей. Шутки шутками, но многие утром выливают воду из чайника, считая, что она за ночь испортилась, и наливают в чайник новой из крана или из фильтра.

(«промах»), то ведут «корректировку огня» — меняют в нужную сторону значение нужного параметра в начальной точке: скорость нити маятника или угол наклона пушки. Эта операция повторяется до тех пор, пока задача не будет решена с заданной точностью. Можно замерить положение спутника в два момента времени и по вышеописанной методике рассчитать его траекторию (орбиту — краевая задача). Можно это сделать также, замерив не только положение спутника, но и его скорость в какой-то момент времени (задача с начальными условиями). Если же нет влияния третьих космических тел, то можно замерить положение спутника в пяти точках движения (в разных моментах времени) и решить систему уже не дифференциальных, а линейных алгебраических уравнений (см. рис. 9.17*c*), имея в виду, что спутник движется строго по эллиптической (круговой), гиперболической или параболической (редкий случай) траектории. Если же космических тел больше трех, то, повторяем, аналитических решений тут нет[24].

На авторском сайте **http://twtmas.mpei.ac.ru/mas/Worksheets/DE_shot.mcd** помещен интерактивный расчет методом стрельбы задачи о развитии эпидемии, сводящейся в простейшем варианте к решению системы двух обыкновенных дифференциальных уравнений первого порядка (**http://twt.mpei.ac.ru/ochkov/mc8Pro.book/5_text.htm**).

Часто можно слышать споры о том, какие пакеты численно или аналитически лучше решают системы ОДУ. Но все эти споры разбиваются о такой факт.

Математическое моделирование более-менее сложных (реальных) физических объектов часто сводится к решению... нескольких тысяч дифференциальных уравнений. И часто оказывается, что никакие разработанные и оттестированные средства, встроенные в математические пакеты Mathcad, Maple, Mathematica, ANSYS, COMSOL и др., не годятся для их решения. Тут приходится составлять индивидуальную расчетную схему со множеством допущений, упрощений, линеаризаций и других «хитростей». А это может сделать только специалист в данной области моделирования (космические полеты, энергетика, радиотехника и т. д. и т. п.), знающий, что тут важно, а что второстепенно. В паре с хорошим математиком, конечно.

В пакете Mathcad (в отличие от Maple и Mathematica) нет средств аналитического (символьного) решения дифференциальных уравнений. Из-за этого мы воспользовались сторонними средствами — см., например, рис. 9.24.

В невозможности непосредственного аналитического решения ОДУ часто упрекают пакет Mathcad, вернее, его разработчиков. Но тут нужно понимать, что более-менее сложное дифференциальное уравнение символьно не решается, а если и решается, то решение получается очень громоздим и неудобным для дальнейшего использования. В этом можно убедиться, если в исходное уравнение, записанное в окошке на рис. 9.24*a*, вернуть синус (см. рис. 2.24*b*), то есть математический маятник превратить в физический.

На сайте **https://www.ptcusercommunity.com/videos/1471** можно видеть анимацию (см. один ее кадр на рис. 9.26) численного решении ОДУ методами Эйлера (рис. 9.23) и Рунге — Кутты (рис. 9.25). На анимации видно, что «Эйлера слегка заносит на поворотах», а Рунге в паре с Куттой «четко чеканят шаг по прочерченной линии аналитического решения». Аналогичное поведение зафиксировано и на рисунках 9.22 и 9.26.

[24] Их математики и астрономы искали столетиями, пока не убедились, что их нет. Хотя — как считать. Некоторые задачи теплопроводности, сводящиеся к решению дифференциальных уравнений, не имели аналитических решений. Но потом математики придумали новые функции (функции Бесселя) и стали считать, что аналитические решения есть, но в них присутствуют эти самые функции Бесселя. По ним даже составили специальные таблицы, подобные таблицам синуса или логарифма. В принципе для нашей задачи о движении спутника и двух планет (рис. 9.7 на стр. 204) можно тоже придумать и затабулировать специальные функции, формирующее аналитическое решение этой задачи.

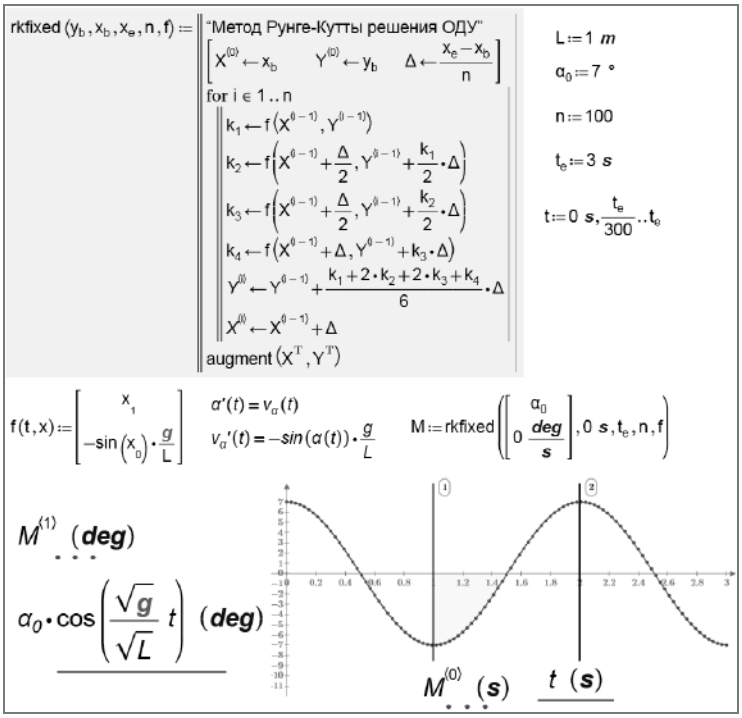


Рис. 9.25

Метод Рунге — Кутты, примененный для решения системы двух обыкновенных дифференциальных уравнений

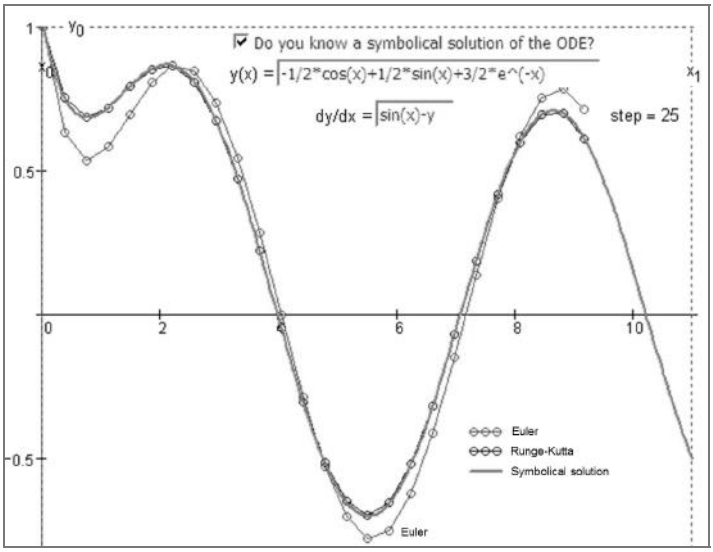


Рис. 9.26

Кадр анимации сравнения метода Эйлера и метода Рунге — Кутты

На сайте **https://www.ptcusercommunity.com/thread/85205** обсуждался процесс создания программ, показанных на рис. 9.23 и 9.25. Там же можно найти программу и пример анимации численного решения системы ОДУ методом Рун-

ге — Кутты с переменным шагом (Adaptive — см. рис. 9.2), когда шаг меняется в зависимости от величины производных — правых частей ОДУ.

Заканчивая этюд, можно привести еще одну цитату из А. С. Пушкина: «О сколько нам открытий чудных готовит просвещенья дух...», заменив в нем слово «просвещенья» на слово IT.

ДИВЕРТИСМЕНТ 2. ПУТЕШЕСТВИЕ ИЗ ПЕТЕРБУРГА В МОСКВУ, ИЛИ СВЕТ В КОНЦЕ ТУННЕЛЯ

Рассказывают, что Николай I перед строительством железной дороги из Санкт-Петербурга в Москву положил на карту линейку и провел карандашом прямую линию между этими двумя столицами Российской империи. В районе Валдайской возвышенности карандаш наскочил на палец императора, и в этом месте дорога сделала небольшой крюк, о котором мы упомянем в конце.

А вот еще один анекдот, но уже не исторический, а наших дней. Один человек сдавал в бухгалтерию отчет о командировке, в которой цена железнодорожного билета из Москвы в Петербург была несколько выше цены обратного билета. На вопрос бухгалтера, откуда взялась такая разница, подотчетное лицо посоветовало посмотреть на... глобус: из Москвы в Санкт-Петербург поезд поднимается вверх, а на обратном пути катится под горку...

Николай I для еще большего сокращения пути должен был не просто прочертить карандашом прямую линию на карте, а... просверлить в глобусе прямое отверстие, соединяющее Москву с Петербургом!

Шутки шутками, но уже давно обсуждается полуфантастический проект так называемого *гравитационного поезда*, катящегося без трения на магнитной подвеске в прямолинейном подземном туннеле, из которого выкачан воздух (рис. 9.27). Первую половину пути такой поезд будет катиться под горку без какой-либо тяги локомотива, а вторую половину пути будет по инерции подниматься вверх, замедляясь без тормозов до самого пункта назначения, где он и остановится.

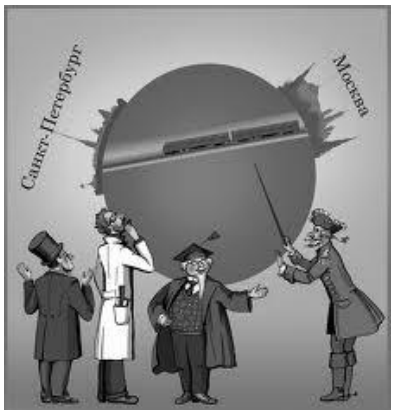


Рис. 9.27
Гравитационный поезд

В Интернете (а рис. 9.27 взят именно оттуда) есть множество готовых формул, по которым можно оценить, сколько времени такой поезд будет в пути и какой максимальной скорости он достигнет в середине туннеля. Но мы сейчас не будем считать по этим формулам, а проанализируем баланс сил, действующих на гравитационный поезд, и решим это уравнение — получим функцию положения поезда в туннеле в зависимости от времени. Отказ от готовых формул позволит нам в дальнейшем усложнить модель, приблизить ее к реальности, учтя силы трения.

На рисунке 9.28 изображена простейшая расчетная модель гравитационного поезда: на планете Земля (идеальный шар с радиусом R) сделан прямолинейный туннель длиной L, по которому идет поезд. Начало декартовых координат, от которого будет вестись отсчет, находится в центре Земли.

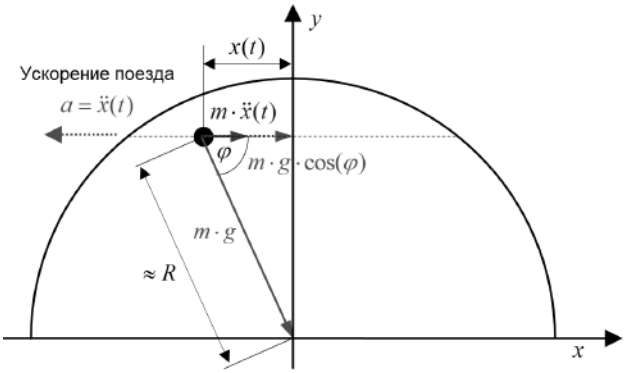


Рис. 9.28
Схема задачи о гравитационном поезде

Несложно показать, что на наш поезд (на физическую материальную точку) вдоль координаты x будет действовать ускоряющая сила (первая половина пути) или тормозящая сила (вторая половина пути), равная весу тела (m · g), умноженному на отношение x/R значения координаты x к радиусу Земли R. Это положение точки будет зависеть от времени — будет функцией x(t). Если от этой функции взять первую производную x′(t), то мы получим скорость поезда, а если вторую производную x″(t) — то его ускорение. Дифференциальное уравнение движения нашего гравитационного поезда будет иметь вид:

$$m \cdot x''(t) = -m \cdot g \cdot x(t) / R,$$

обусловленный вторым законом Ньютона: сумма сил, действующих на тело, равна произведению его массы на его ускорение.

На рисунке 9.29 показана оценка расстояния между Москвой и Питером «по хорде и по прямой» с использованием авторского Mathcad-сервера, куда введены географические координаты (широта[25] и долгота) центров двух российских столиц (угловые градусы, минуты и секунды), взятых опять же из Википедии. Принято, что Земля — это идеальный шар.

На рисунке 9.30 показано решение дифференциального уравнения движения гравитационного поезда с помощью сайта Интернета. В диалоговое окно не только введено само уравнение — в нем также отмечены начальное положение поезда x(0) = -L/2 и его нулевая начальная скорость x′(0) = 0.

Решение, показанное на рисунке 9.30, отображено графически в среде Mathcad на рисунке 9.31. Расстояние от Москвы до Питера мы округлили до 600 км, а радиус Земли до 6400 км.

[25] «А нам не страшен ни вал девятый, ни холод вечной мерзлоты — ведь мы ребята, ведь мы ребята семидесятой широты» — поется в одной песне. Жители города, где выпущена в свет эта книга, заменяют в этой песне слово «семидесятой» на слово «шестидесятой». Кстати, до русской революции 1917 г. долгота отмечалась не по гринвичскому, а по пулковскому меридиану, расположенному вблизи Петербурга. В расчет, показанный на рис. 9.29, можно ввести эту опцию — выбор нулевого меридиана.

http://twt.mpei.ac.ru/MCS/Worksheets/Earth-D-L.xmcd

Расчет расстояния по хорде (ортодромия) между двумя точками на шаре (глобусе)

R/km := 6371

	deg	min	sec	
Географическая широта первого города φ_1	55	45	21	◉ сш ○ юш
Географическая долгота первого города λ_1	37	37	04	◉ вд ○ зд
Географическая широта второго города φ_2	59	57	00	◉ сш ○ юш
Географическая долгота второго города λ_2	30	19	00	◉ вд ○ зд

Recalculate

$\varphi_1 = 0.9731229$ rad $\quad \lambda_1 = 0.6565541$ rad $\quad \varphi_2 = 1.046325$ rad $\quad \lambda_2 = 0.5291257$ rad

Угловая длина ортодромии $\delta := \arccos(\sin\varphi_1 \sin\varphi_2 + \cos\varphi_1 \cos\varphi_2 \cos(\lambda_2 - \lambda_1)) = 0.09967$ rad

$\delta = 5.7105846$ deg $\quad \delta = 6.3450939$ grad

Длина ортодромии: $D := R\,\delta = 634.988$ km

Длина по прямой между точками $L := 2\,R \sin\frac{\delta}{2} = 634.725$ km

Максимальная "глубина" прямой $h := R - R\cos\frac{\delta}{2} = 7.909$ km

Рис. 9.29

Оценка расстояния от Москва до Петербурга

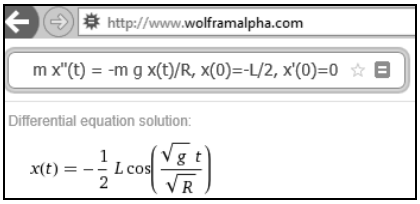


Рис. 9.30

Символьное решение задачи о гравитационном поезде

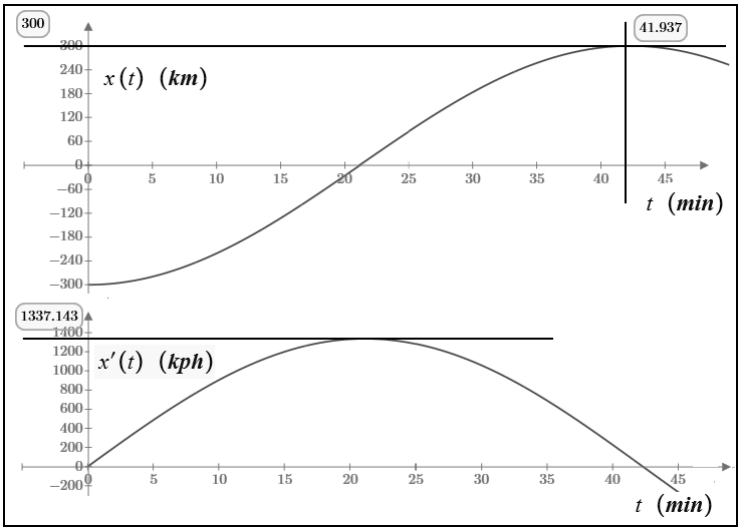


Рис. 9.31

График движения гравитационного поезда

Наш фантастический поезд будет в пути чуть больше 42 мин и достигнет в середине туннеля скорости 1337 км/ч. Тут можно было бы сказать, что скорость поезда здесь превысила скорость звука, но в нашем туннеле воздуха нет (мы ведь в такой модели пренебрегаем сопротивлением воздуха) и, значит, нет и звука.

Из рисунков 9.30 и 9.31 видно, что наш поезд будет подобно маятнику совершать в туннеле колебательные движения от одного города к другому (в нашем случае — от Петербурга к Москве и обратно) с периодом, не зависящим от расстояния между городами. Это расстояние будет влиять только на среднюю скорость и максимальную скорость поезда в середине пути. Эти расчеты в среде Mathcad показаны на рисунке 9.32.

$$R := 6400\ \mathbf{km} \qquad L := 600\ \mathbf{km}$$

$$x_0 := -\frac{L}{2} \qquad y_0 := \sqrt{R^2 - x_0^{\,2}} = 6392.965\ \mathbf{km} \qquad h := R - y_0 = 7.035\ \mathbf{km}$$

$$x(t) := -\frac{1}{2} L \cdot \cos\left(\sqrt{\frac{g}{R}} \cdot t\right) \qquad t := \frac{\pi}{\sqrt{\frac{g}{R}}} = 42.299\ \mathbf{min}$$

$$v(t) := \frac{\mathrm{d}}{\mathrm{d}t} x(t) \rightarrow \frac{L \cdot \sin\left(t \cdot \sqrt{\frac{g}{R}}\right) \cdot \sqrt{\frac{g}{R}}}{2} \qquad v_{max} := \frac{L}{2} \cdot \sqrt{\frac{g}{R}} = 1336.9\ \mathbf{kph}$$

Рис. 9.32

Расчет параметров туннеля Москва — Петербург

На рисунке 9.32 после ввода исходных данных (R и L) рассчитываются начальные координаты поезда x_0 и y_0 и значение максимальной глубины туннеля h, которое для трассы Москва — Петербург составит чуть больше семи километров. Это очень важный параметр. Дело в том, что в нашей математической модели ускорение свободного падения g принимается за константу 9.807 м/с^2, встроенную в Mathcad. В более глубоких туннелях нужно будет учитывать изменение значения g в зависимости от глубины туннеля. А как меняется эта величина? Это отдельный вопрос. Если, например, туннель прокопать через центр Земли (а такой фантастический проект тоже обсуждается), то в середине шахты такого *гравитационного лифта* значение ускорения свободного падения должно быть нулевым.

Уравнение, показанное на рисунке 9.30, можно дополнить силой сопротивления воздуха (если считать, что воздух в туннеле все же есть) и силой трения колес о рельсы, т. е. приблизить наш гравитационный поезд к реальным условиям. Силу сопротивления воздуха обычно принимают пропорциональной плотности воздуха ρ, умноженной на площадь поперечного сечения поезда S и квадрат его скорости x□(t). Сила трения колес о рельсы пропорциональна той составляющей веса поезда, которая параллельна оси y. Такое усложненное дифференциальное уравнение уже нельзя будет решить аналитически, т. е. нельзя будет получить формулу для функции x(t). Это уравнение нужно будет решать численно, приближенно, т. е. получать таблицу значений функции x(t) при разных значениях t. На рисунке 9.33 показано это решение в среде Mathcad.

Две силы трения нужно умножить на встроенную в Mathcad функцию-ступеньку sign, которая возвращает нуль, если ее аргумент меньше или равен нулю, и единицу — в противном случае. Это сделано для того, чтобы сила трения поезда о воздух всегда действовала против движения поезда, а сила трения колес о рельсы равнялась нулю при нулевой скорости поезда. При высоких скоростях поезд будет тормозиться в основном за счет силы встречного ветра, а при низких скоростях — за счет силы трения качения колес. Такое можно наблюдать у приземляющегося самолета. Сначала он тормозит за счет закрылков и тормозного па-

рашюта, а затем за счет тормозов шасси. Из уравнения на рисунке 9.33 можно убрать квадратный корень, так как для туннеля Москва — Петербург отношение `x(t)/R` очень мало. Но для других туннелей это отношение будет достаточно большим, и его нельзя будет игнорировать. В более глубоких туннелях нужно будет также учитывать изменение плотности воздуха.

R	L	m	F	S	ρ	k	f	t_e
(km)	(km)	$(tonne)$	(kgf)	(m^2)	$\left(\frac{kg}{m^3}\right)$			(hr)
6400	600	15	121.2	3	1.25	0.02	0.00003	24

$$\alpha := 2 \cdot \operatorname{asin}\left(\frac{L}{2 \cdot R}\right) = 5.373\,°\qquad x_0 := -R \cdot \sin\left(\frac{\alpha}{2}\right) = -300\ km$$

Решатель Ограничения

Решить

$$x(0\ s) = x_0 \qquad x'(0\ s) = 0\ kph$$

$$m \cdot x''(t) = F - m \cdot g \cdot \frac{x(t)}{R} - \left(k \cdot \rho \cdot S \cdot \frac{x'(t)^2}{2} + f \cdot m \cdot g \cdot \sqrt{1 - \left(\frac{x(t)}{R}\right)^2}\right) \cdot sign\left(x'(t)\right)$$

$$x := \mathbf{Odesolve}\left(x(t), t_e\right)$$

Рис. 9.33

Численное решение задачи о гравитационном поезде

При учете сил трения и правильно выбранной силе тяги локомотива `F` (у нас это 121.2 килограмм силы) поезд в туннеле благополучно доедет до конечной точки и покатится назад (если его не задержать, подложив, например, под его колеса тормозной башмак), повторяя движение затухающего маятника (рис. 9.34), с окончательной остановкой не в середине туннеля, а где-то под Валдайской возвышенностью — там, где поезд под небольшим уклоном будет удерживаться силой тяги локомотива `F`.

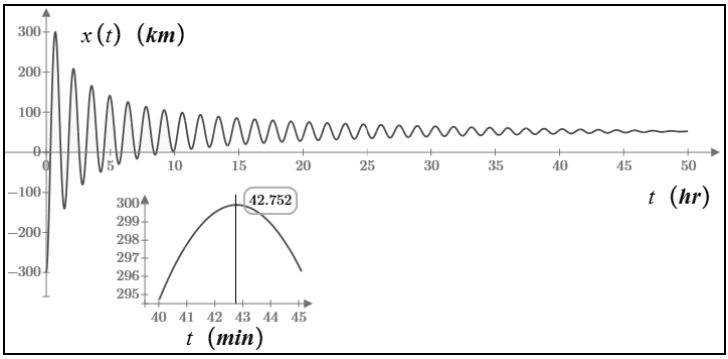


Рис. 9.34

График движения гравитационного поезда с трением

На рисунке 9.33 показана встроенная функция Mathcad `Odesolve`, численно решающая (solve) наше обыкновенное (o) дифференциальное (d) уравнение (e — equation). До уравнения зафиксировано начальное положение поезда `x(0 s) = x₀` и его скорость `x′(0 s) = 0` км/ч (`kph`). От этой точки функция `Odesolve` будет поточечно рассчитывать значения создаваемой функции `x(t)`, отображенной на графике на рисунке 9.34.

Мы доказали и теоретически, и нашим численным экспериментом, что время движения гравитационного поезда не зависит от длины туннеля и равно этим самым 42 мин, которые можно считать некоей константой, связанной с гравитационной постоянной и плотностью Земли и характеризующей нашу планету. При иных длинах туннеля будет меняться только скорость поезда — средняя и максимальная. В туннеле Москва — Петербург среднюю скорость легко рассчитать:

$$600 \text{ км} / 42 \text{ мин} = 857 \text{ км/ч}.$$

Эта скорость меньше скорости пассажирского самолета (900–1000 км/ч), но больше скорости сверхскоростных поездов (400–600 км/ч), которые пока катятся не между Москвой и Петербургом, а по другим маршрутам. Будем надеяться, что в будущем скорость движения поездов между Москвой и Петербургом повысится с теперешних 150–200 км/ч до некоего психологического барьера 857 км/ч и выше, и реальный поезд на данном маршруте сможет обогнать гравитационный.

Палец Николая I на линии «Москва — Петербург» — это не просто исторический анекдот. Это объезд реальным поездом реального большого и глубокого оврага, через который построили мост только в 2000 г. при реконструкции дороги. Подробнее обо всех вымыслах и реалиях этой истории см.: **http://af1461.livejournal.com/212024.html**.

На сайте: **https://www.ptcusercommunity.com/videos/1982** можно увидеть анимацию движения гравитационного поезда.

Кстати о реальных, а не о фантастических туннелях. Знаете ли вы, по какой траектории они строятся, если нет каких-то особых ограничений?! Тут обычно отвечают: по прямой линии или по дуге окружности. Но это не так. В прямом туннеле будет скапливаться вода, и ее нужно будет непрерывно откачивать. Дугообразный же туннель копать довольно сложно. Туннели обычно строят так. Из двух точек, расположенных у противоположных склонов горы, два проходческих щита начинают прокладывать туннель строго по прямой линии, несколько поднимающейся над горизонтальной линией. Щиты должны встретиться где-то в центре туннеля[26] несколько выше уровня стартовых точек. В таком крышеобразном туннеле не только не будет скапливаться вода, но и при необходимости из него за счет своего веса сможет выкатиться заглохший транспорт. Свет в конце такого туннеля можно увидеть, только дойдя до его середины... Кстати говоря, несложно подсчитать, что если железную дорогу между Москвой и Петербургом проложить не на поверхности Земли (по дуге идеального шара с радиусом 6400 км), а в туннеле строго по прямой (см. рис. 9.27 и 9.28), то расстояние между этими двумя столицами России сократится всего лишь на 220 м.

С этим «странным» свойством окружности связана одна любопытная математическая задача. Представим себе, что Земля — это идеальный шар, по экватору туго обтянутый верёвкой. Мы разрезаем эту верёвку и надставляем её куском всего лишь в 1 м длины, а потом верёвку равномерно распределяем над линией экватора в воздухе. Вопрос: может ли какое-нибудь животное подлезть под этой верёвкой, не деформируя ее форму окружности? Второй вопрос: какое животное может пролезть под верёвкой, если первоначально вместо Земли мы по «экватору» обтянули футбольный мяч? Ответ в обоих случаях одинаков: пролезет любое животное, чей рост в сантиметрах не превышает величины $50/\pi$, т. е. почти 16 см. Всё дело в формуле длины окружности. Сначала имеем $C = 2\pi R$, потом $C + 100 = 2\pi(R + h)$, поэтому $100 = 2\pi h$ и $h = 50/\pi$.

А по какому маршруту, вернее, по какой траектории поезд может делать объезд? Как тут правильно спрофилировать рельсы? Представьте себе, что вы инже-

[26] Еще один анекдот. В комиссию по сооружению туннеля под Ла-Маншем пришли два брата-шотландца с лопатами и сказали, что они могут выкопать туннель так: один будет копать со стороны Англии, а другой — со стороны континентальной Европы. В комиссии усмехнулись и спросили: «А что будет, если вы промахнетесь?» «Ничего страшного, — ответили братья, — Вы получите два туннеля по цене одного!» Управление проходческими щитами — это высокая наука и искусство.

нер-путеец и вам нужно спроектировать поворот на 90° трамвая на перекрестке — с улицы в переулок. Первое решение, какое приходит в голову, — это проложить рельсы дугой четверти окружности. Но в этом случае кривизна рельсов на стыке будет скачком меняться с нулевого значения (прямые рельсы) до значения `1/R` (`R` — радиус поворота[27]), а потом опять же скачком возвращаются к нулевому значению. А это нехорошо. Техника, как и природа, не терпит острых углов. Поэтому подобные повороты делаются так. Сначала вставляют рельсы, кривизна которых плавно меняется от `0` до `1/R`, затем рельсы с радиусом `R`, а потом рельсы, кривизна которых плавно меняется от `1/R` до `0`. Представим себе, что Николай I, обводя карандашом свой палец, изобразил на карте кривую, показанную на рис. 9.35, — кривую нормального распределения[28]. Ее кривизна плавно меняется от очень малого значения, близкого к нулю (а в технических задачах допустимо ее считать равной нулю), до какого-то достаточно большого ненулевого значения и потом опять возвращается к очень малой. На рис. 9.35 показаны три кадра анимации движения точки (поезда Москва — Петербург) вдоль кривой нормального распределения с касательной и нормалью, начинающейся в точке и заканчивающейся в центре окружности, характеризующей кривизну кривой. При движении по такому маршруту пассажиры не будут чувствовать резких боковых толчков (см. авторскую анимацию **https://www.ptcusercommunity.com/videos/5727)**.

Вернемся к туннелям. Вопрос: а по какой траектории нужно строить гравитационный туннель, чтобы уменьшить время поезда в пути? Обычно эту задачу *вариационного исчисления* проецируют на... катание с горок — как нужно спроектировать горку, чтобы скатиться с нее за минимальное время? На рисунке 9.36 показан кадр анимации математической модели горок, которые у нас в России называют американскими, а в Америке русскими. Задан вид кривой (искаженная синусоида, переходящая в горизонтальную прямую[29]), и далее решено дифференциальное уравнение, подобное тому, что было составлено для моделирования движения поезда в гравитационном туннеле (см. рис. 9.33), учитывающее наклон и движущую силу вследствие этого наклона, сопротивление санок и сопротивление воздуха. Можно менять (варьировать!) вид выражения, задающего профиль горки, и замерять время спуска, а можно обратиться к сайтам Интернета, где эта задача описана. Если задать в качестве такой функции, например, параболу, проходящую через начальную и конечную точки горки, то можно попытаться рассчитать параметры такой параболы, при которых время спуска (с учетом или без учета сил трения — см. рис. 9.36) будет минимальным. Но парабола тут будет не самой оптимальной кривой. Доказано, что для такой скоростной горки больше подходит перевернутая циклоида (см. анимацию ее создания на авторском сайте **https://www.ptcusercommunity.com/videos/2077)**. Это, если не учитывать силы трения. А если их учитывать? Какая тут должна быть кривая?

Кстати о параболе. На школьных уроках математики параболу вводят не с того, так сказать, конца: рассказывают о формуле, описывающей параболу (о квадратном уравнении), а не о геометрическом месте точек на Евклидовой плоскости, равноудаленных от фокуса параболы и ее директрисы. На авторском сайте по адресу **https://www.ptcusercommunity.com/thread/136959** помещена анимация, иллюстрирующее это свойство параболы. На другом сайте по адресу **https://www.ptcusercommunity.com/thread/136960** анимация показывает свойство другой плоской кривой второго порядка — гиперболы, которую мы рассматривали в первом дивертисменте к этому этюду (см. рис. 9.18 на стр. 218).

---

[27] Тут, конечно, имеется в виду пара рельсов с радиусами, отличающимися на величину ширины колеи. Раньше трамваи в Москве визжали на поворотах из-за того, что у них не было межколесных дифференциалов и колеса вращались с одной угловой скоростью, а линейная скорость контакта колес с рельсами должна быть разной на поворотах. Проскальзывание колес было причиной этого трамвайного визга, наполнявшего старую Москву и другие «трамвайные» города.

[28] О ней мы подробнее поговорим в этюдах 12 и 14.

[29] Тут ездоки будут ощущать легкий удар, вследствие резкого изменения кривизны кривой — см. выше.

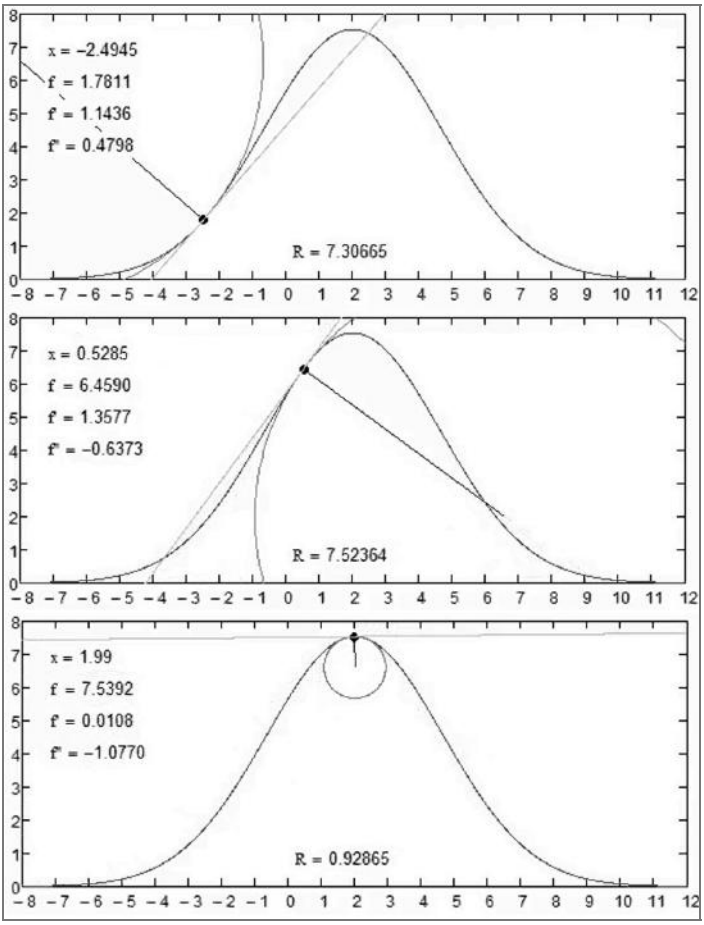


Рис. 9.35
Палец императора на трассе «Москва — Петербург»
*(см. **https://www.ptcusercommunity.com/videos/5727**)*

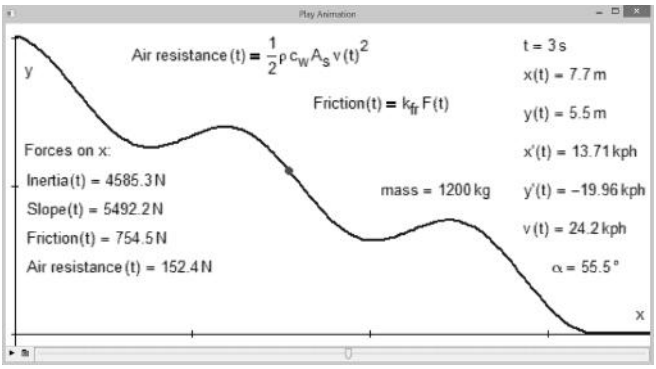


Рис. 9.36
Кадр анимации математической модели русских/американских горок
*(см. **https://www.ptcusercommunity.com/docs/DOC-2141**)*

ДИВЕРТИСМЕНТ 3. СИЛА, МАССА, ИМПУЛЬС, УСКОРЕНИЕ...

Диалог в учительской: Преподаватель Закона Божьего, обращаясь к учителю физики:
«Сегодня Ваш любимчик двойку получил. Спросил я его, что такое Божья Сила.
Так он мне ответил, что это произведение Божьей Массы на Божье Ускорение».
Учитель физики: «Я ему и по физике двойку поставлю.
Ведь произведение Божьей Массы на Божье Ускорение
дает силу с божественностью в квадрате, а не в первой степени».

У этого старого «физико-теологического» анекдота есть вполне серьезное продолжение.

Второй закон Ньютона в его общей постановке гласит, что сила это первая производная *импульса* (произведения массы на скорость) по времени. Но если масса постоянна (не зависит от времени), то ее можно вынести из-под знака производной, и тогда сила будет равна произведению массы на ускорение — на первую производную скорости по времени. В такой редакции почти все люди, включая и вышеописанного гимназиста, помнят второй закон Ньютона. Но есть случаи, когда масса величина непостоянная. Решение одной такой задачи показано на рис. 9.37: ракета имеет стартовую массу 270 т, из которых 250 т — это топливо и окислитель, которые расходуются со скоростью 5 т в секунду. Тяга ракетного двигателя — 300 т силы[30]. Спрашивается, какая у ракеты будет конечная скорость [83].

В этой задаче масса рассматриваемого объекта (материальной точки) величина переменная. В какой редакции нужно применить в этой задаче второй закон Ньютона? Тут нужно вспомнить об инерциальных и неинерциальных системах отсчета, не только о ньютоновской, но и о лагранжевой и гамильтоновой механике, а также о других тонкостях раздела физики под названием динамика. Впрочем, наш вопрос о том, какая кривая на рис. 9.37 более правильная, сам по себе не совсем корректен. В настоящее время, почти все физические журналы не принимают к рассмотрению не только статьи с описанием вечных двигателей, но и статьи по небесной механике с опорой на ньютоновскую механику. Но наша книга не о механике, а о том, как старые не очень сложные задачи решать с помощью современных компьютерных средств. Современная же физика настолько сложна, что ее положения и теоремы многие не понимают, а просто в них верят или не верят. Так что термин «физико-теологический», приведенный в начале этого дивертисмента, имеет вполне законное основание[31]. Как вам понравится ученая степень «кандидат физико-теологических наук»!?

Кстати, о вечных двигателях, вернее, о вечно двигающихся механизмах. Если реальные такие устройства невозможны (их «смерть» — это силы трения), то виртуальные вполне. За работу, читатель. Инструкции в этюде 8.

На рисунке 9.37 применены обе редакции и два ответа отличаются в разы. Какой из них верен? Этот вопрос обсуждался на сайте

https://www. ptcusercommunity.com/message/426478.

На ракету кроме силы тяги действует и вторая сила — сила тяжести. Все это уравновешивается либо производной от импульса по времени (первое решение), либо произведением массы на ускорение (второе решение).

Решение этого вопроса мы оставляем читателю, который может учесть тут и третью силу — силу сопротивления воздуха. При старте ракеты плотность воздуха наибольшая, но скорость ракеты мала. С разгоном ракеты плотность воздуха

[30] Обычно в таких задачах задают не тягу двигателя, а скорость истечения реактивной струи. Если эту величину перемножить на массовый расход топлива и окислителя, то и получится эта самая сила тяги. Но задание скорости истечения струи более «физично» в том плане, что в такой постановке задачи можно применить закон движения центра масс: ракета и продукты сгорания топлива разлетаются в разные стороны, но центр их масс остается на одном месте.

[31] Авторское исследование соотношения науки, религии и культуры можно посмотреть здесь **http://twt.mpei.ac.ru/ochkov/WSCR/WaterSCR.pdf.**

уменьшается, но растет скорость ракеты. В какой момент времени сопротивление воздуха будет максимально?

$$\frac{d}{dt}(m(t)\cdot v(t)) \rightarrow m(t)\cdot\frac{d}{dt}v(t)+v(t)\cdot\frac{d}{dt}m(t)$$

m_0	M	F	φ	q
$(tonne)$	$(tonne)$	$(tonnef)$	(deg)	$\left(\frac{tonne}{s}\right)$
20	250	300	30	5

$m(t) := m_0 + M - q\cdot t \qquad m(0\ s) = 270\ tonne \qquad t_{end} := \frac{M}{q} = 50\ s$

$h(0\ s) = 0\ m \qquad v(0\ s) = 0\ kph \qquad v(t) = h'(t)$

$m(t)\cdot v'(t) + m(t)\cdot g(h(t),\varphi) = F$

$\begin{bmatrix} h_1 \\ v_1 \end{bmatrix} := \mathbf{odesolve}\left(\begin{bmatrix} h(t) \\ v(t) \end{bmatrix}, t_{end}\right) \qquad t := 0\ s, \frac{t_{end}}{300} .. t_{end}$

$h(0\ s) = 0\ m \qquad v(0\ s) = 0\ kph \qquad v(t) = h'(t)$

$(m(t)\cdot v'(t) + m'(t)\cdot v(t)) + m(t)\cdot g(h(t),\varphi) = F$

$\begin{bmatrix} h \\ v \end{bmatrix} := \mathbf{odesolve}\left(\begin{bmatrix} h(t) \\ v(t) \end{bmatrix}, t_{end}\right) \qquad t := 0\ s, \frac{t_{end}}{300} .. t_{end}$

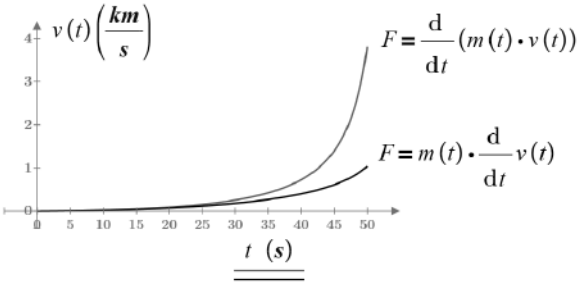


Рис. 9.37
Второй закон Ньютона и старт ракеты

Задача о ракете открывает очень интересную тему — тему многоступенчатых ракет, позволяющих при одинаковой стартовой массе ракеты выводить на баллистическую или околоземную орбиту более весомую полезную нагрузку за счет сброса отработавших ступеней. Эти задачи с попытками их решения помещены в этюде 24.

И еще один важный момент.

На рисунке 9.37 в свернутой области[32], расположенной в начале расчета, помещена функция, возвращающая ускорение свободного падения g в зависимости

[32] Сразу за этой областью помещено преобразование, показывающее, чему равна производная от произведения двух функций: пакет Mathcad — это, помимо прочего, и хороший математический справочник, который всегда под рукой, когда решаешь задачу на компьютере.

от высоты над уровнем моря h и широты местности φ. Эта функция показана на рис. 9.38 в раскрытой области.

$$g(h,\varphi) := \left(9.780327 \cdot \left(1 + 0.0053024 \cdot \sin(\varphi)^2 - 0.0000058 \cdot \sin(2 \cdot \varphi)^2\right) - 3.086 \cdot 10^{-6} \cdot \frac{h}{m}\right) \cdot \frac{m}{s^2}$$

Рис. 9.38
Ускорение свободного падения: функция пользователя

На рисунке 9.39 показан авторский сайт, где ускорение свободного падения рассчитывается в интерактивном режиме.

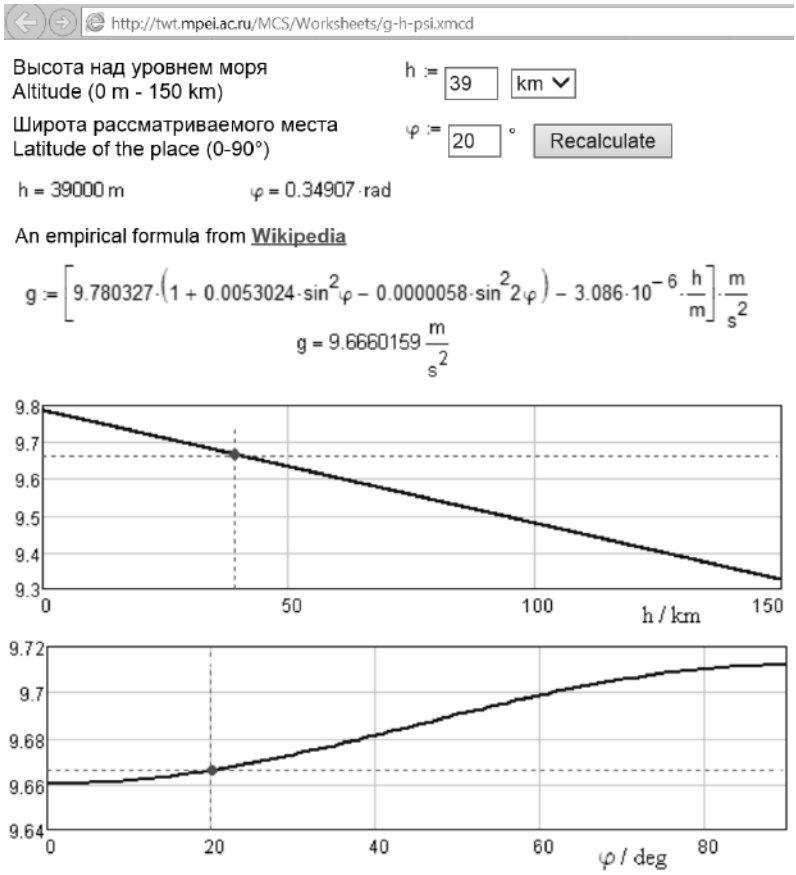


Рис. 9.39
Ускорение свободного падения: онлайн-расчет

Читатель, сравни, как отображается степень синуса на рис. 9.38 и 9.39! На первом рисунке эта степень показана в стиле пакета Mathcad — со скобками и степенью за скобками. На втором же рисунке степень синуса отображена в общематематической манере — без скобок и со степенью, стоящей сразу за словом sin. Дело в том, что создавая сетевой расчет, показанный на рис. 9.39, авторы слегка схитрили — поместили для обозрения посетителями сайта не саму Mathcad-формулу, а ее подправленный рисунок. Но сейчас даже чистые математики стали прописывать степень тригонометрических и прочих элементарных функций в ма-

нере математических пакетов. Эти пакеты и математики как бы идут навстречу друг другу, вырабатывая единые правила записи формул.

Онлайн-расчет, показанный на рис. 9.39, помещен по адресу

http://twt. mpei.ac.ru/ochkov/VPU_Book_New/mas/others.html

в разделе **Прочее**. Там же помимо ссылки на живой расчет, помещены ссылки на соответствующие функции для популярных расчетных программ, а также ссылка на «облачную» функцию — см., например, рис. 1.27 на стр. 37. Там же эта формула записана в виде текста, который несложно перенести в языки программирования.

И самое последнее.

Детство и отрочество авторов книги прошло в те времена, когда запускался первый спутник Земли, отправлялись зонды на Луну, был осуществлен первый космический полет человека, первый выход человека в открытый космос в скафандре... Других особых успехов, успехов мирового уровня тогда у страны не было... В то время почти все мальчишки мечтали стать космонавтами. Один из авторов не исключение. Но пойти учится на летчика, из которых набирались космонавты, ему не удалось — зрение подвело. Потом этот автор попал на срочную службу в советскую армию в ракетную часть, но опять же из-за зрения не в ракетчики, а в «обслугу» — был начальником котельной в воинской части, занимавшейся телеметрией ракетных пусков и поиском упавших на землю боеголовок **(http://twt.mpei.ac.ru/ochkov/Army.htm)**. Но теперь этот автор, имея под рукой Mathcad с его анимацией, может совершать виртуальные полеты в космос...

Этот же автор когда-то давно узнал в школе, как просто можно нарисовать эллипс, по которому вращаются спутники. Основное характеристическое свойство эллипса: сумма расстояний от точки эллипса до каждого из его фокусов постоянна. Этот автор пришел домой, вбил в стену два гвоздя и привязал к ним веревочку. Получилась цепная линия, которая была описана в этюде 7. Затем автор взял карандаш, натянул им веревочку и нарисовал часть верхней половинки эллипса — см. рис. 9.40. Но весь эллипс нарисовать не удалось — мать увидела, как сын портит стену, и сделала ему нагоняй. Но сейчас этому автору удалось все-таки доделать эту работу в среде Mathcad 15 — создать анимацию рисования эллипса, размещенную на сайте **https://www.ptcusercommunity. com/videos/2089**.

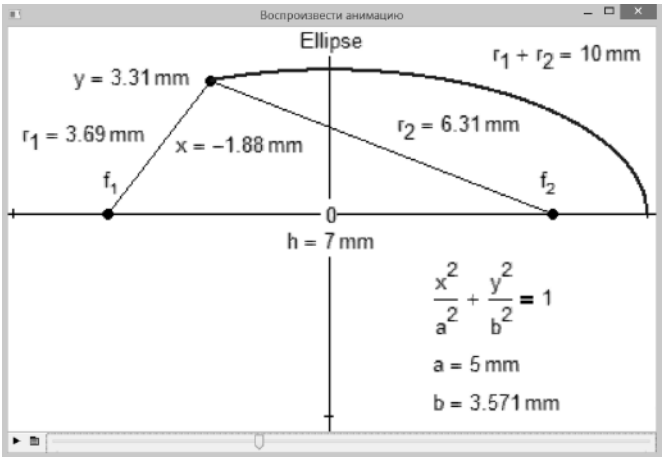


Рис. 9.40

Кадр анимации рисование эллипса с помощью двух гвоздиков и веревочки

На форуме
https://www.ptcusercommunity.com/groups/fine-math-curves-in-mathvad[33]
авторы собрали целую коллекцию созданных с помощью Mathcad анимаций замечательных кривых — циклоида, суперциклоида, гипоциклоида, гипотрохоида, кардиоида, синусоида и т. д.

Задания читателям

1. Придумать и реализовать на компьютере новые интересные «хороводы» небесных тел.

2. Создать анимацию, включающую в себя следующие этапы полета на Луну: старт с поверхности Земли, переход на орбиту Земли, старт с орбиты Земли к Луне, движение к Луне с выключенными двигателями, переход на орбиту Луны, прилунение, а далее возвращение на Землю.

3. Задав геометрию крышеподобного туннеля, рассчитать точку, в которой человек, сидящий в автомобиле или идущий пешком, «увидит свет в конце туннеля».

4. Создать анимацию движения поезда на поворотах, подобную той, какая показана на рис. 9.35, но опирающуюся не на кривую нормального распределения, а на отрезки кривых, кривизна которых плавно меняется от `0` до `1/R` (первая кривая), затем с постоянной кривизной `1/R` (вторая кривая), а потом рельсы, кривизна которых плавно меняется от `1/R` до `0` (третья кривая).

5. Обычно в расчетах, подобным тому, какой показан на рис. 9.37, не учитывают географическую широту мест запуска ракеты. Но известно, что космодромы нужно размещать ближе к экватору. Показать это расчетом!

6. Создать расчет расстояния между двумя географическими точками Земли, подобный тому, какой показан на рис. 9.29, но в котором будут видны все использованные формулы.

7. Создать анимации рисования не только эллипса (см. рис. 9.40), но и параболы, гиперболы и других интересных кривых, опираясь на их определение как на геометрическое место точек на плоскости. К примеру, если сумму, использованную в анимации на рис. 9.40, заменить на произведение, то можно получить так называемые овалы Кассини. У этих овалов, как и у эллипса может быть более двух фокусов. Постройте такие замкнутые кривые!

[33] Да, да! Увы, именно Mathvad, а не Mathcad. Эту опечатку уже не исправить — слишком много сделано ссылок на этот сайт.

Этюд 10

РЕКУРСИЯ, ИЛИ ХОД КОНЕМ

В этюде будет рассказано, что такое рекурсия и как можно и нужно ее применять для решения различных математических задач на компьютере. Будет также рассказано о фракталах — простых и оптимизированных.

Математика: факториал, объем параллелепипеда, локальный экстремум функции одной переменной, дифференциальное уравнение, линии уровня, седловая точка, ряд, локальный экстремум функции двух переменных, фрактал, числа Фибоначчи, числа Аккермана.

Физика: шахматы.

Информатика: рекурсивный алгоритм, компьютерный анализ функциональных зависимостей.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127820**

Германн сошёл с ума. Он сидит в Обуховской больнице
в 17-м нумере, не отвечает ни на какие вопросы
и бормочет необыкновенно скоро:
«Тройка, семёрка, туз! Тройка, семёрка, дама!..»

А. С. Пушкин «Пиковая дама»

Почти все знают, что такое факториал числа. Это произведение всех натуральных чисел от 1 до n включительно. Для факториала введено обозначение `n!`, которое используется и в среде Mathcad. Так, например, $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$. Такое классическое определение факториала не охватывает нуль. Для удобства общей записи формул, в которых 0! обычно входит в виде сомножителя, математики договорились считать $0! = 1$.

Замечательно то, что факториал нуля легко подсчитать (не определить, а, именно подсчитать) с помощью рекурсии (от лат. «recursio» — возвращение). На рисунке 10.1 помещена программа-функция для Mathcad Prime подсчета факториала с использованием рекурсии.

```
Factorial(n) := | if n = 5
                |   | return 120
                | if n > 5
                |   | return (Factorial(n − 1) · n)
                | if n < 5
                |   | return Factorial(n + 1) / (n + 1)

Factorial(4) = 24        Factorial(6) = 720

Factorial(5) = 120       Factorial(0) = 1
```

Рис. 10.1
Рекурсивная функция «Факториал»

Рекурсивная функция — это функция, которая при вычислении вызывает сама себя. Она обычно зависит от параметра, при изменении которого происходит рекурсивный спуск или подъем.

Вот как определяет рекурсию один юмористический справочник по информатике: «Рекурсия — см. рекурсия». Проще всего проиллюстрировать рекурсию геометрически: многократно (даже бесконечное число раз) повторить какие-то геометрические элементы (возможно, с изменением масштаба), т. е. получить са-

моподобные геометрические объекты. Классическим примером бесконечной рекурсии являются изображения в двух поставленных друг напротив друга зеркалах: в них образуются два коридора из уменьшающихся и затухающих отражений зеркал. Такие зеркала можно видеть в купе поезда. При рекурсии очень легко математически посчитать размерность геометрического объекта. Математики заметили, что некоторые рекурсивные объекты имеют дробную размерность. Такие объекты назвали фракталами. Вспомним, что точка имеет размерность 0, линии (кривые и прямые) — 1 («длину»); плоскость и поверхности — 2 («площадь»); трехмерное пространство и тела — 3 («объем»). Чтобы представить себе геометрический объект дробной размерности, найдите в Интернете любой сайт с изображениями фракталов. Или взгляните на рис. 10.5 ниже.

Если вычислительный алгоритм основан на принципе разбиения основной задачи на подзадачи, каждая из которых повторяет основную, то мы получаем рекурсивный алгоритм. Такой алгоритм решает задачу путем ее сведения к решению одной или нескольких таких же задач, но в более простом варианте. Пример такого алгоритма представлен на рисунке 10.1. Вычисление факториала начинается со значения `n = 5`, а затем идет или рекурсивный подъем, или рекурсивный спуск. Заметим, что можно было бы начать с любого натурального числа n, чей факториал нам известен. Например, взять `10! = 3628800` и `n = 10` или `n = 1` и `n! = 1`.

Словесное описание используемой функции `Factorial` такое. Если n равно 5, то `n!` равен 120 (см. первые две строки программы на рис. 10.1). Если n больше 5, то факториал такого целого числа равен `(n - 1)!·n`. Пример 6! = 5!·6 = 720. Если `n` меньше 5, то `n!` равен `(n + 1)! / (n + 1)`. Примеры: 4! = 5! / 5 = 24, 3! = 4! / 4 = 6, 2! = 3! / 3 = 2, 1! = 2! / 2 = 1 и, наконец, 0! = 1! / 1 = 1, что и требовалось доказать.

В Интернете есть очень интересный сайт **http://oeis.org** «Открытая энциклопедия целочисленных последовательностей». Если в окошко этого сайта ввести последовательность чисел 1, 1, 2, 6, 24, 120, 720 (см. выше) и нажать кнопку `Search` (`Искать`), то сайт выдаст ответ, что тут «запрятан» факториал, и будет много интересного рассказано об этом математическом операторе.

Очень забавно вводить на этом сайте различные последовательности целых чисел и узнавать, какой закономерности они подчиняются. Так, например, если ввести числа 1, 2, 3, 5, 8, 13, 21, то будет получен ответ, что это числа Фибоначчи — числовой ряд, где каждое очередное число — это сумма двух предыдущих чисел (см. рис. 10.2).

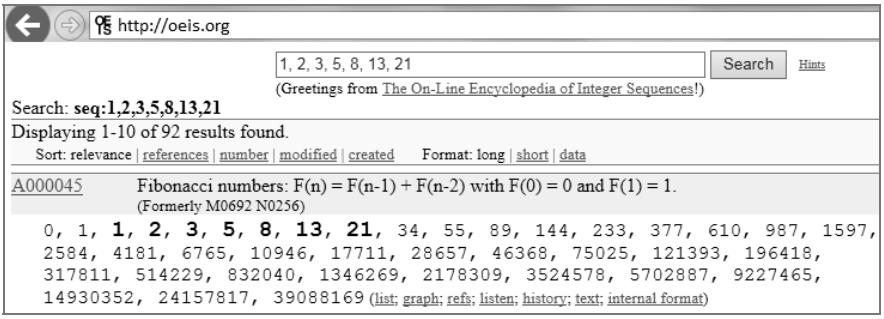


Рис. 10.2

Сайт определения закономерностей в целочисленных последовательностях

Интересно найти закономерность той или иной числовой последовательности самому или с помощью сайта, показанного на рис. 10.2. Но еще интересней «утереть нос» этому сайту — ввести числовую последовательность с известной закономерностью и увидеть, что сайт дал слабину и не нашел ответа. После этого можно будет зарегистрироваться на этом сайте и ввести в него новую информацию. Попробуем это сделать. Что такое числа Фибоначчи (см. выше), знают многие. А вот что такое *краси-*

вые числа Фибоначчи, мало кто знает. «Красивые» числа — это 1, 3 и 7. Если их сложить, то получится еще одно «симпатичное» число — 11. Вспомним эпиграф к данному этюду: «Тройка, семерка, туз (11)». На рисунке 10.3 показана рекурсивная функция, возвращающая красивые числа Фибоначчи не только при положительном, но и при отрицательном аргументе[1]. В этой более полной последовательности чисел появилась и пятерка, которой также нельзя отказать в «изящности», если вспомнить школьные и институтские оценки знаний.

```
Fibonacci(n) := | if n = 1
                |   | return 3
                | if n = 2
                |   | return 7
                | if n = 3
                |   | return 11
                | if n > 3
                |   | return (Fibonacci(n−3) + Fibonacci(n−2) + Fibonacci(n−1))
                | if n < 1
                |   | return (Fibonacci(n+3) − Fibonacci(n+2) − Fibonacci(n+1))
```

$$n := -7..7 \qquad n = \begin{bmatrix} -7 \\ -6 \\ -5 \\ -4 \\ -3 \\ -2 \\ -1 \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{bmatrix} \qquad Fibonacci(n) = \begin{bmatrix} 11 \\ -1 \\ -5 \\ 5 \\ -1 \\ -1 \\ 3 \\ 1 \\ 3 \\ 7 \\ 11 \\ 21 \\ 39 \\ 71 \\ 131 \end{bmatrix}$$

Рис. 10.3
Прекрасные числа Фибоначчи

Так вот, если в окошко упомянутого сайта (рис. 10.2) ввести числа 1, 3, 7, 11, 21, 39, то он «раскусит» эту последовательность, отметив, что каждое очередное число — это сумма трех (а не двух — см. рис. 10.2) предыдущих. Но если этому сайту оказать «медвежью услугу» — расширить ряд до −1, 3, 1, 3, 7, 11, 21, 39 (приписать в начале последовательности чисел минус 1 и тройку), то ответа не последует — «каша будет испорчена маслом».

Предлагаем читателям «поиграть» с сайтом **oeis.org**, вводя в него числовые последовательности, созданные программированием (см. рис. 10.1 и 10.3), или опираясь на свою собственную смекалку. Сайт **oeis.org**, к примеру, знает, что 1, 2, 5, 10, 20, 50 и 100 — это номиналы банкнот США, а 5, 10, 20, 50, 100, 200, 500 — это банкноты евро, и прогнозирует будущие банкноты в случае гиперинфляции. Но этот сайт не знает, что 1, 2, 3, 5, 10, 25, 50, 100 — это номиналы банкнот бывшего СССР[2]. Неизвестно сайту также и то, что 7, 9, 11, 13, 15, 19, 22, 28, 48 — это стоимость (в копейках) мороженого в том же бывшем Советском Союзе. В программе, показанной на рисунке 10.3, можно поменять опорные числа («тройку, семёрку,

[1] Когда-то давно один из авторов этой книги увлекался формализмом в программировании и написал «красивую» программу на языке Паскаль вычисления чисел Фибоначчи, где число знаков на программной строке было равно… очередному числу Фибоначчи.

[2] Здесь везде по 7 банкнот. Считается, что устойчивая банковская система должна базироваться на «красивом» числе банкнот. В СССР, кстати, до денежной реформы 1961 г. и монет было семь — 1, 2, 3, 5, 10, 15 и 20 копеек. Оптимальное число банкнот и монет, а также гирек для весов (разновесов) — это интересная математическая задача.

туза» на «тройку, семёрку, даму» (еще одну тройку — см. эпиграф), получить новую последовательность целых чисел и пропустить ее через сито сайта **oeis.org**.

Но вернемся к нашей основной теме — к рекурсии.

Расчет факториала (рис. 10.1) и чисел Фибоначчи (обычных или прекрасных) можно вести и без рекурсии: знаешь предыдущие числа — высчитываешь очередное, очередное становится одним из предыдущих, а сама операция продолжается до тех пор, пока не доберемся до искомого числа. Такой счет (рекуррентность вместо рекурсии — примеры ниже на рис. 10.16 и на стр. 17 на рисунке 3.1) ведется быстрее и не требует значительных ресурсов памяти компьютера. Но рекурсия очень упрощает написание программы. Классический пример — программа, решающая головоломку «Ханойской башни».

Суть головоломки. Имеется три стержня, на первый из которых (у нас в программе на рис. 10.4 он маркирован как A) нанизаны диски на манер детской пирамидки: самый большой диск внизу — самый маленький наверху. Предлагается переложить эти диски на стержень C, беря их по одному. Для временного складирования разрешается использовать третий стержень B. При этом нельзя класть большой диск на меньший. Программе на рис. 10.4 достаточно сообщить только число дисков в пирамиде `n`. После запуска программа будет возвращать порядок перекладки дисков:

`n = 2: AB, AC и BC (три хода)`

`n = 3: AC, AB, CB, AC, BA, BC` и, наконец, AC (семь ходов)

`n = 4: ... (15 ходов)` и т. д.

$n := 11 \qquad p := 2^n - 1 = 2047 \qquad V_0 := \text{"begin"} \qquad V_{p+1} := \text{"end"}$

```
HT(n, V, x, y, z) := | p ← 2^n − 1
                     | i ← 1
                     | if n = 1
                     |   | while V_i ≠ 0
                     |   |   | i ← i + 1
                     |   | V_i ← concat(x, z)
                     | else
                     |   | V ← HT(n − 1, V, x, z, y)
                     |   | while V_i ≠ 0
                     |   |   | i ← i + 1
                     |   | V_i ← concat(x, z)
                     |   | V ← HT(n − 1, V, y, x, z)
                     | V
```

$HT(n, V, \text{"A"}, \text{"B"}, \text{"C"})^T = [\overset{0}{\text{"begin"}}\ \overset{1}{\text{"AC"}}\ \overset{2}{\text{"AB"}}\ \overset{3}{\text{"CB"}}\ \overset{4}{\text{"AC"}}\ \overset{5}{\text{"BA"}}\ \overset{\dots}{\dots}\ \overset{2048}{\dots}]$

Рис. 10.4
Ханойская башня с рекурсией

Число перестановок `p` в общем случае равно $2^n - 1$. Задача с `n` дисками легко сводится к задаче с `n - 1` дисками, а задача с `n - 1` дисками сводится к задаче с `n - 2` дисками и т. д. до задачи с двумя дисками, которая решается просто — диск со стержня A перекладываем на диск C (см. на рис. 10.4 фрагмент программы `If n = 1...`). Отсюда и рекурсия в программе на рис. 10.4.

По древней легенде, тибетские монахи уже несколько тысячелетий перекладывают 64 золотых диска, нанизывая их на алмазные стержни. Когда головоломка будет решена и на стержне C окажутся все диски, наступит конец света. Спасает нас лишь то, что при 64 дисках для решения головоломки потребуются свыше...

триллиона лет, если на каждый ход тратить по секунде. Это без учета ложных ходов и времени, необходимого на ручной или компьютерный расчет порядка перестановки дисков. В Интернете, кстати, есть виртуальные игры «Ханойская башня». Откройте ее и поиграйте с 5–10 дисками.

Теперь о ранее упоминавшемся фрактале. Фрактал — это некое зримое воплощение рекурсии. На сайте **https://www.ptcusercommunity.com/videos/2426** собран 101 фрактал, созданный[3] в среде Mathcad 15. Развитие фрактала от простой формы к сложной хорошо иллюстрировать средствами анимации Mathcad. Так на рис. 10.5 показаны пять кадров развития фрактала, который называется фракталом Леви по имени исследователя, его придумавшего: берется отрезок прямой линии (`n = 0` — нулевой кадр анимации; он не показан на рис. 10.5), который ломается пополам и из которого строится прямой угол (`n = 1` — первый кадр анимации). Далее эта операция повторяется снова и снова с уменьшающимися отрезками прямых, пока не появится некая «кудрявая» фигура (`n = 21`), напоминающая накаченного, как раньше говорили, культуриста — современного бодибилдера (второе название фрактала).

Рекурсия в среде Mathcad помогает решить старинную задачу о перемещении коня по шахматной доске так, чтобы не ходить на одну клетку два раза. Задача похожа на задачу коммивояжера, которую мы рассмотрим в этюде 15, но в шахматной задаче не требуется, чтобы конь вернулся на исходную позицию. Анимация этого решения расположена на сайте **https://www.ptcusercommunity.com/videos/1484**, а ее три кадра показаны на рис. 10.6. На авторском сайте нужно указать начальную клетку шахматной доски (например, `b - 3` как на рис. 10.6) и Mathcad рассчитает путь коня — решение задачи:

http://twtmas.mpei.ac.ru/mas/Worksheets/DG/knight.mcd.

Шахматные задачи сыграли важную роль в судьбе первого автора этой книги. Будучи на научной стажировке в Германии в 1983 г., он приобрел на одну месячную стипендию (1200 DM) компьютер Sharp MZ80. К нему прилагалась кассета с программой для игры в шахматы. Запустив ее, можно было не только играть с компьютером, но и решать шахматные задачи. Одна из них была опубликована в журнале «Наука и жизнь», который выписывала университетская библиотека. Ее условие было стандартным — мат в три хода. Но когда фигуры были расставлены на виртуальной шахматной доске купленного компьютера, то он сделал мат в... два хода. Автор не удержался и написал в редакцию журнала о таком неожиданном решении. Оказалось, что это была старинная шахматная задача с дополнительным усложняющим условием — поставить мат, но не бить определенное поле. Компьютер этого условия, естественно, не знал и сделал мат в два, а не в три хода. Завязалась переписка автора с редакцией журнала, которая вылилась в рубрики «Человек и компьютер», «Советы начинающему программисту» и др., материалы некоторых выпусков можно видеть в этой книге. Например, в журнале был объявлен конкурс на самую короткую программу на языке Бейсик (а он был очень популярен в то время) по переводу римских чисел в арабские и наоборот — см. этюд 3. Потом появились книги в соавторстве с Ю. В. Пухначевым — сотрудником журнала «Наука и жизнь» — «24 этюда на Бейсике», «128 советов начинающему программисту» и др. И пошло-поехало. Настоящая книга подводит некий итог этой работы по популяризации математических методов решения задач на компьютере.

[3] С сайта можно скачать соответствующий файл, созданный В. И. Коробовым.

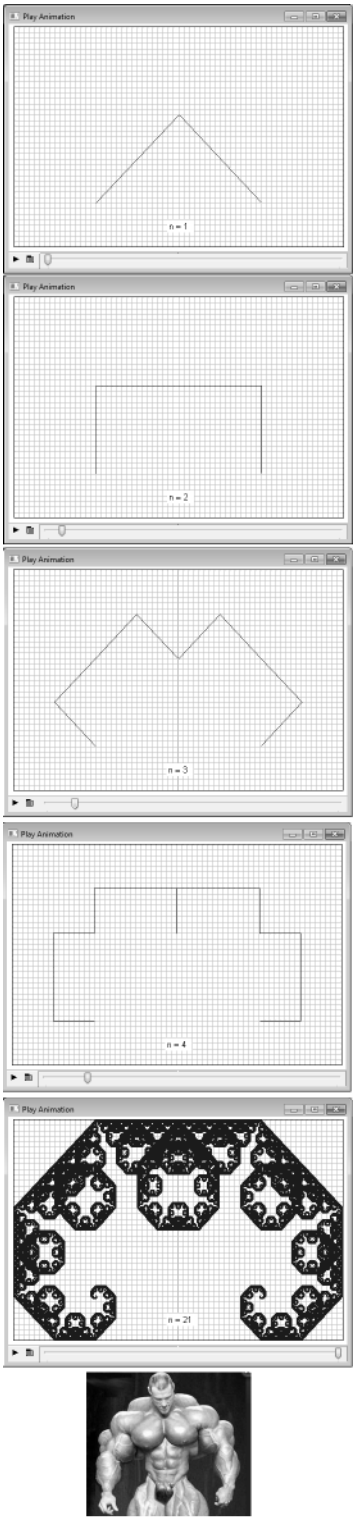


Рис. 10.5
Фрактал Бодибилдинг

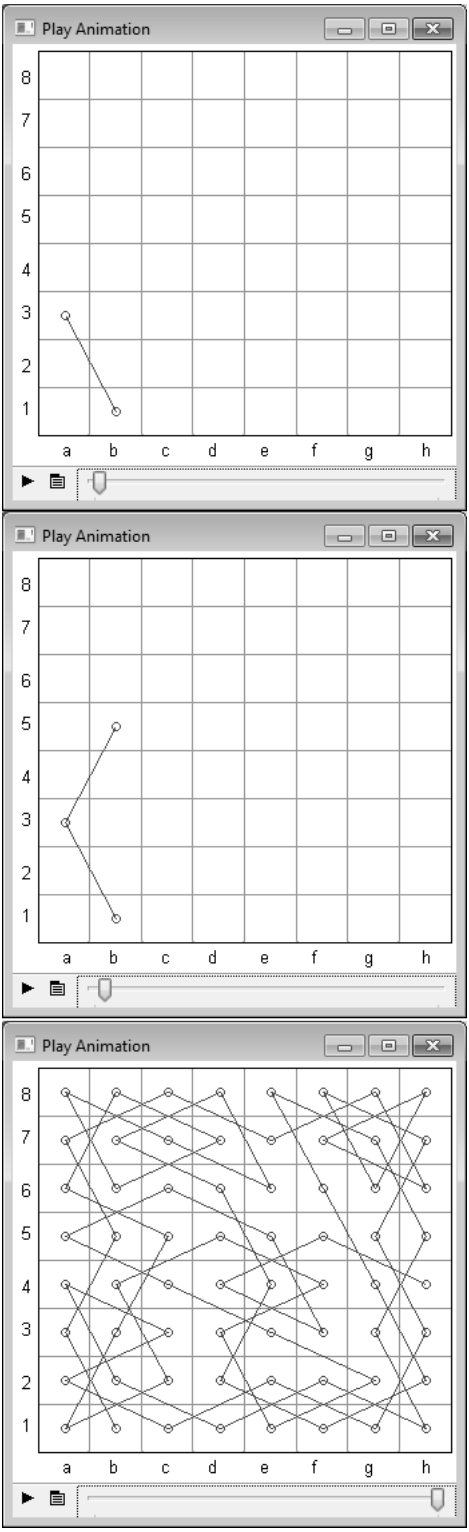


Рис. 10.6
Конь на шахматной доске

ДИВЕРТИСМЕНТ. ОПТИМИЗИРОВАННЫЙ ФРАКТАЛ

Есть такая народная[4] задача оптимизации. Берется квадратный лист бумаги (картона, жести и т. д.), в углах которого вырезаются четыре одинаковых квадрата меньшего, естественно, размера. Далее из такой крестообразной заготовки складывается коробка. На рисунке 10.7 показаны кадры анимации, сделанной в среде Mathcad 15 (см. подробнее об этом в этюде 8), превращения квадратной заготовки в коробку. Саму анимацию можно скачать с сайта книги.

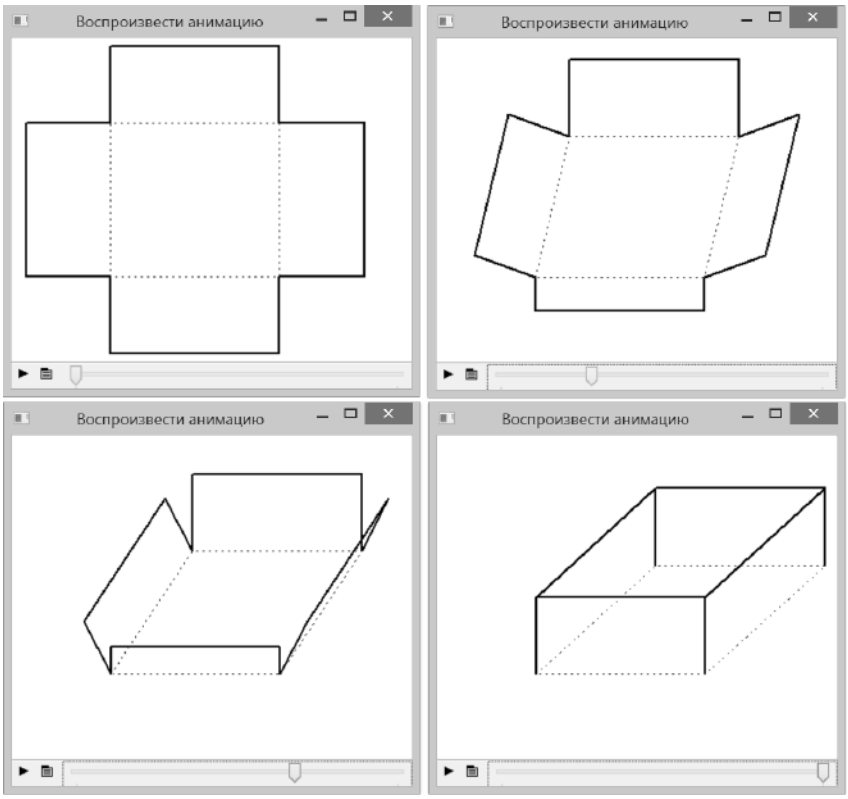


Рис. 10.7
Кадры анимации изготовления коробки из квадратной заготовки

Анимация делалась не просто просто, а предельно просто: в расчет вводились векторы, хранящие координаты углов крестообразной заготовки и свернутой коробки. Затем значения координат углов заготовки плавно (с изменением значения системной переменной `FRAME` от 0 до 999) менялись до значений координат сложенной коробки.

Но вернемся к самой задаче. Спрашивается, какой должен быть размер вырезаемых квадратов, чтобы объем полученной коробки был максимальным?

Сходу, как правило, люди тут выдают два ответа:

1) коробка должна быть в виде куба, а длина стороны вырезаемого квадрата должна равняться трети длины стороны исходного квадрата;

2) длина стороны вырезаемого квадрата должна равняться одной четвертой длины стороны исходного квадрата.

[4] «Народная» в том смысле, что авторам не удалось найти ее создателя. В Интернете по соответствующему ключу поиска натыкаешься на множество текстов, рисунков и видео по этой задаче, но без указания авторства.

Но оба ответа неверны.

Заблуждаются многие и в ответе на усложненную данную задачу, которая будет описана ниже (рис. 10.13, [81]).

На рисунке 10.8 показано решение задачи о коробке с помощью символьной математики Mathcad: создается функция пользователя с именем `V` и с двумя аргументами: `A` — длина стороны исходной квадратной заготовки и `x` — длина стороны вырезаемых четырех квадратов. Функция `V` возвращает объем свернутой коробки. От этой функции (кубический полином) берется первая производная по `x` (получается квадратный полином), у которой ищутся нули. Их два: `A/2` — локальный минимум (нулевой объем коробки) функции `V` и `A/6` — искомый локальный максимум функции `V`, т. е. решение нашей «народной» задачи оптимизации раскроя коробки: длина стороны вырезаемых квадратов должна равняться одной шестой длины стороны исходного квадрата. На рисунке 10.8 это решение проверено графически для коробки, сделанной из квадратной заготовки со стороной, равной одному метру.

$$V_1(A,x) := (A-2\,x)^2 \cdot x \qquad \frac{d}{dx}V_1(A,x)=0 \xrightarrow{solve,x} \begin{bmatrix} \frac{A}{2} \\ \frac{A}{6} \end{bmatrix} \qquad V_1\left(1\ m, \frac{1}{6}\ m\right) = 74.074\ L$$

$$x := 0\ m, 1\ cm .. 0.5\ m$$

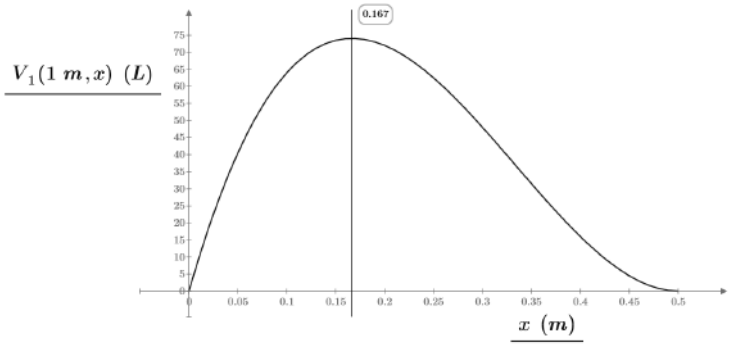


Рис. 10.8

Решение задачи о коробке максимального объема

Это решение можно также получить, обратившись к сайту пакета Mathematica, или (в русле импортозамещения ;-)) — к авторскому Mathcad-северу — см. рис. 10.9–10.11.

Сайт на рисунке 10.9 строит график функции, сайт на рисунке 10.10 выдает ее производную[5], а сайт на рисунке 10.11 — ищет нули производной.

Показанное выше решение, повторяем, можно найти во многих бумажных и интернетовских источниках. Но задача об оптимальной коробке имеет более свежее и довольно интересное «фрактальное» продолжение. А фракталы наряду с рекурсией — это тема данного этюда. Так вот, можно четыре квадрата, отрезанные от исходной заготовки (см. левый верхний кадр анимации на рис. 10.7), не выбрасывать, а пустить в дело — сделать из них четыре маленькие коробочки по той же схеме раскроя. Из шестнадцати (4·4) новых отрезанных квадратиков опять же можно сделать новые коробочки. Из шестидесяти четырех (16·4) новых отрезанных квадратиков опять же можно сделать еще меньшие новые коробочки и т. д. до бесконечности. Вот вам и некий фрактал, требующий *оптимизации* — определения размеров сторон квадратных вырезок, при которых суммарный объем всех полученных коробок будет максимальным. Тут опять же сходу приходит

[5] Этот сайт примечателен тем, что производная выводится через предел, определяющий производную — ренессанс в вычислениях — тема этюда 1.

на ум тривиальное решение — все коробки должны иметь ранее найденную пропорцию 1/6. Но так ли это?

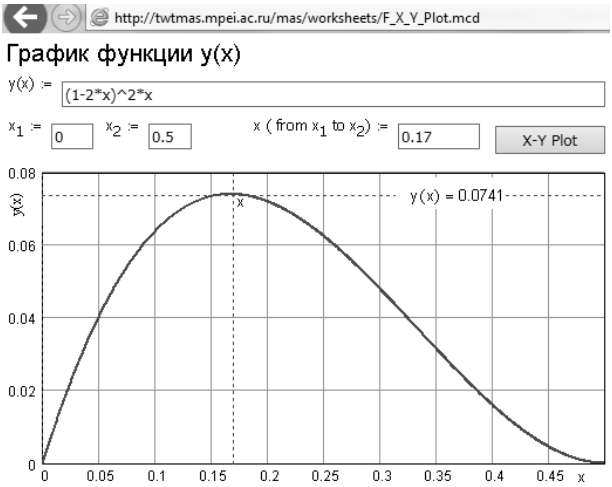


Рис. 10.9

Построение графика функции y(x) с помощью Mathcad-севера

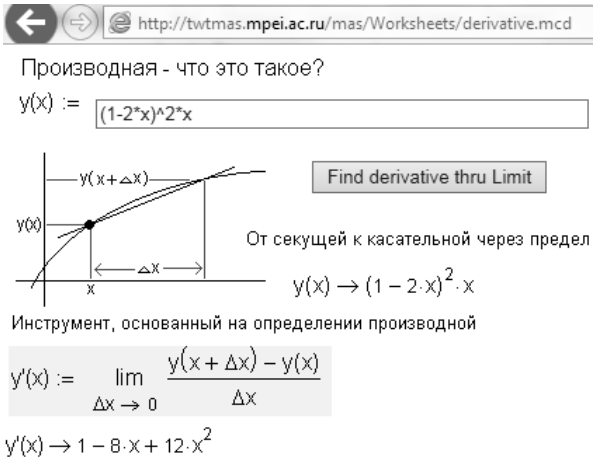


Рис. 10.10

Поиск производной функции y(x) с помощью Mathcad-севера

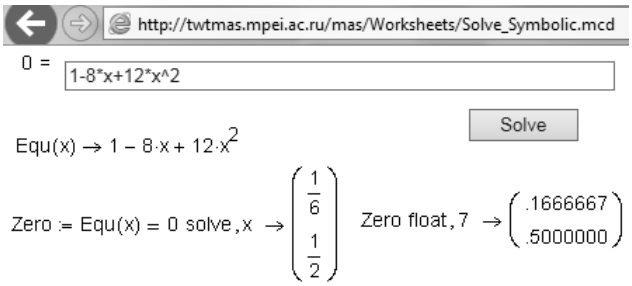


Рис. 10.11

Поиск нулей функции y(x) с помощью Mathcad-севера

На рисунке 10.12 показано нахождение оптимального размера сторон четырех квадратов, вырезаемых из исходного квадрата со стороной A и из которых вырезаются 16 квадратов меньшего размера. Ясно, что при втором (последнем) шаге раскроя нужно сохранить пропорцию 1/6, чтобы суммарный объем четырех коробок малого размера был максимален. При этом пропорция раскроя исходной заготовки будет несколько отличаться от 1/6.

$$V_5(A,x) := V_1(A,x) + 4\cdot\left(x - 2\,\frac{x}{6}\right)^2\cdot\frac{x}{6}$$

$$\frac{d}{dx}V_5(A,x) = 0 \xrightarrow{solve,x} \begin{bmatrix} \frac{9\cdot A}{29} + \frac{3\cdot\sqrt{7}\cdot A}{58} \\ \frac{9\cdot A}{29} - \frac{3\cdot\sqrt{7}\cdot A}{58} \end{bmatrix}$$

$$A := 1\ m \qquad \begin{bmatrix} \frac{9\cdot A}{29} + \frac{3\cdot\sqrt{7}\cdot A}{58} \\ \frac{9\cdot A}{29} - \frac{3\cdot\sqrt{7}\cdot A}{58} \end{bmatrix} = \begin{bmatrix} 44.719 \\ 17.35 \end{bmatrix} cm$$

$$V_5\left(A, \frac{9\cdot A}{29} - \frac{3\cdot\sqrt{7}\cdot A}{58}\right) = 75.529\ L$$

Рис. 10.12
Задача о пяти коробках

Задачу о пяти коробках — одной большой и четырех маленьких, можно решить и по-иному, забыв на время про 1/6 и проанализировав функцию V_5 не одного, а двух аргументов — пропорция для первого шага раскроя x и пропорция для второго шага раскроя y (см. рис. 10.13).

В решении на рисунке 10.13 вводится функция пользователя V_5 с двумя аргументами[6]: x и y. Далее решается система двух алгебраических уравнений: равенство нулю частных производных функции V_5 по x и по y. Это необходимое условие экстремума для функции двух аргументов. Координаты точек экстремума функции V_5 обязаны удовлетворять этой системе. Далее можно в каждой найденной точке проверить выполнение достаточного условия экстремума. Но можно результат получить не аналитически, а графически — построить линии уровня функции двух переменных (см. рис. 10.13). Полученная система уравнений имеет четыре корня, один из которых совпадет с решением, показанным на рис. 10.9, и является искомой точкой максимума. Три другие корня — это точка минимума (0,5 0,5) и две так называемых «седловых точки», что ясно видно из графиков линий уровня, показанных на рис. 10.13. Минимумы, максимумы и «седла» можно увидеть и на поверхности, построенной с помощью авторского Mathcad-сайта — см. рис. 10.14.

[6] От единиц измерения мы тут отказались и, кроме того, допустили, что A = 1.

Суммарный объем пяти коробок $V_5(x,y) := (1-2\,x)^2 \cdot x + 4 \cdot (x-2\,y \cdot x)^2 \cdot y \cdot x$

$$a := \begin{bmatrix} \frac{d}{dx} V_5(x,y) = 0 \\ \frac{d}{dy} V_5(x,y) = 0 \end{bmatrix} \xrightarrow{solve,\begin{bmatrix} x \\ y \end{bmatrix}} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{6} & \frac{1}{2} \\ \frac{3 \cdot \sqrt{7}}{58} + \frac{9}{29} & \frac{1}{6} \\ \frac{9}{29} - \frac{3 \cdot \sqrt{7}}{58} & \frac{1}{6} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.1666667 & 0.5 \\ 0.447194 & 0.1666667 \\ 0.1734956 & 0.1666667 \end{bmatrix}$$

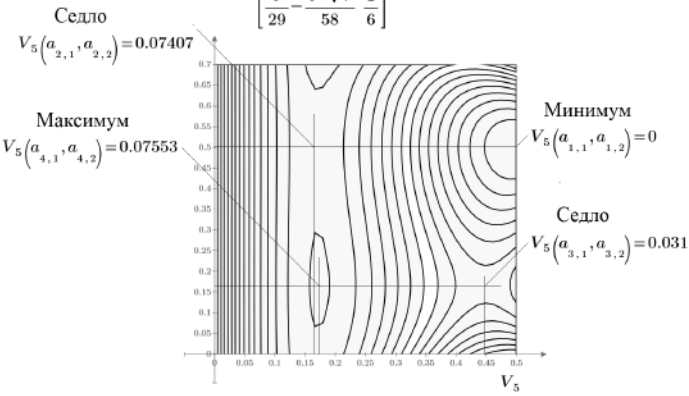


Рис. 10.13

Задача о пяти коробках — решение 2

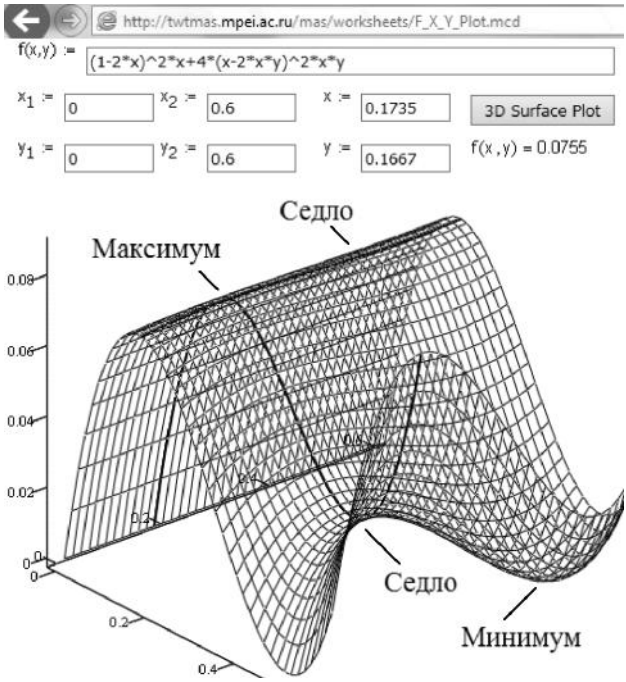


Рис. 10.14

Построение поверхности функции двух аргументов с помощью Mathcad-севера

На рисунке 10.14 максимум попадает на почти горизонтальную прямую, и создается впечатление, что точки максимума заполняют всю прямую. Нужно развернуть график или изменить масштаб по вертикальной оси, чтобы максимум вы-

делялся на общем фоне. У автора есть мечта установить в аудитории лазерный 3D-проектор, напустить в помещение тумана и показывать поверхности и другие объемные объекты, подобные тому, какой показан на рис. 10.14, под потолком, поворачивая их по трем осям, отмечая особые точки. Сейчас так проводятся различные шоу[7].

Пойдем дальше! Из вырезанных 16 квадратов также можно изготовить новые коробки, получив 64 новых квадратных обрезков, из которых... Вот вам и обещанный фрактал. Его можно отобразить графически, если, например, на плоскости выставить все коробки, не двигая их, а только загибая вверх их стенки.

На рисунке 10.15 показана формула, по которой можно рассчитать суммарный объем коробок при любом числе шагов раскроя квадратной заготовки и при любой пропорции раскроя. Аргументом функции V_n должен быть вектор, число элементов которого определяет количество шагов раскроя, а значения элементов — пропорции раскроя.

$$V_n(x) := \sum_{i=0}^{\mathrm{last}(x)} \left(4^i \cdot \prod_{j=0}^{i} x_j \cdot \left(\mathrm{if}\left(i=0,1,\prod_{j=0}^{i-1} x_j\right) - 2\cdot\prod_{j=0}^{i} x_j\right)^2\right)$$

$$V_n\left(\begin{bmatrix}\frac{1}{6}\end{bmatrix}\right)=0.07407 \qquad V_n\left(\begin{bmatrix}\frac{9}{29}-\frac{3\cdot\sqrt{7}}{58}\\ \frac{1}{6}\end{bmatrix}\right)=0.07553 \qquad V_{21}(x):=V_n\left(\begin{bmatrix}x\\ \frac{9}{29}-\frac{3\cdot\sqrt{7}}{58}\\ \frac{1}{6}\end{bmatrix}\right)$$

$$x_3 := 0.2$$

$$x_3 := \mathbf{Maximize}\left(V_{21}, x_3\right) = 0.173645$$

$$V_{21}\left(x_3\right) = 0.0755599$$

$$x := 0, 0.001 .. 0.5$$

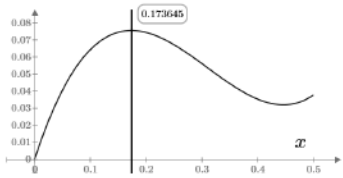


Рис. 10.15
Задача о произвольном числе шагов квадратной заготовки

На рисунке 10.15 формула с операторами суммы и произведения протестирована на решениях, найденных для одной коробки (рис. 10.8–10.11) и для пяти коробок (рис. 10.12 и 10.13). Все совпало. Задача о 21 коробке (одна большая, четыре средние и 16 маленьких) решена средствами не символьной, а численной математики — вызовом функции `Maximaze` (см. этюд 4). Символьная математика не справится с этой задачей из-за того, в частности, что в анализируемой функции V_n присутствует функция `if` («гроза» символьной математики). Можно, конечно, попытаться избавиться от функции `if`, заменив ее функцией-ступенькой (функцией Хэвисайда), но можно предположить, что ответ (если он все-таки будет получен) окажется слишком громоздким. Кроме того, есть более простой способ решения данной задачи при числе шагов раскроя квадратной заготовки, стремящемся к... бесконечности. А фрактал и бесконечность — это, как говорится, «два сапога — пара».

История этого решения такова. В [16] была сделана попытка анализа раскроя 5, 21 и 85 (1 + 4 + 16 + 64) коробок. После выхода книги в свет было получено письмо от анонимного читателя, который предложил решение для любого числа шагов раскроя, последний из которых имеет пропорцию 1/6. На рисунке 10.16 показана эта рекуррентная формула: задается значение $x_0 = 1/6$ (пропорция раскроя самых маленьких коробочек), далее находится пропорция для предыдущего шага x_1 и т. д.

[7] См. сноску 2 на стр. 137.

$$N := 7 \qquad x_0 := \frac{1}{6} \qquad i := 0..N$$

$$x_{i+1} := \frac{1}{4 + 2 \cdot \sqrt{1 - 2\,x_i \cdot (1 - 2\,x_i)}}$$

$$V_n(\mathrm{reverse}(x)) = 0.075560532301892$$

$$n := \sum_{i=0}^{N} 4^i = 21845$$

$$x = \begin{bmatrix} 0.166666666666667 \\ 0.173495621841487 \\ 0.173644979419639 \\ 0.1736481106798 \\ 0.173648176263916 \\ 0.173648177637545 \\ 0.173648177666315 \\ 0.173648177666917 \\ 0.17364817766693 \end{bmatrix}$$

Рис. 10.16

Задача о бесконечном числе коробок с максимальным суммарным объемом

Решение, показанное на рис. 10.16, было опубликовано в [10]. Кто автор этого решения и верно ли оно — остается загадкой.

Но до бесконечности нам еще очень далеко. В расчете на рисунке 10.16 было сделано только восемь шагов в раскрое коробки, для которых были найдены оптимальные пропорции вырезок. Если число шагов раскроя увеличивать, то будет достигнут предел точности численной математики Mathcad, и изменения в ответе (в таблице на рис. 10.16) не будут фиксироваться.

Оптимизацию бесконечного числа раскроя квадратной заготовки можно провести, если перейти к символьной математике Mathcad и сделать допущение — пропорцию раскроя квадратов сделать постоянной для всех шагов раскроя.

На рисунках 10.17–10.19 показано решение такой задачи.

$$\sum_{i=1}^{\infty} 4^{i-1} \cdot x^i \cdot \left(x^{i-1} - 2\,x^i\right)^2 \rightarrow \left\| \begin{array}{l} \text{if } x = -\left(\frac{1}{4} \cdot 2^{\frac{1}{3}}\right) - \frac{2^{\frac{1}{3}} \cdot \sqrt{3}}{4} \cdot 1\mathrm{i} \vee x = -\left(\frac{1}{4} \cdot 2^{\frac{1}{3}}\right) + \frac{2^{\frac{1}{3}} \cdot \sqrt{3}}{4} \cdot 1\mathrm{i} \vee x = \frac{1}{2} \cdot 2^{\frac{1}{3}} \\ \left\| \frac{x - x \cdot \infty + \infty}{x^2} \right. \\ \text{else if } x \neq -\left(\frac{1}{4} \cdot 2^{\frac{1}{3}}\right) - \frac{2^{\frac{1}{3}} \cdot \sqrt{3}}{4} \cdot 1\mathrm{i} \wedge x \neq -\left(\frac{1}{4} \cdot 2^{\frac{1}{3}}\right) + \frac{2^{\frac{1}{3}} \cdot \sqrt{3}}{4} \cdot 1\mathrm{i} \wedge x \neq \frac{1}{2} \cdot 2^{\frac{1}{3}} \\ \left\| \frac{(2 \cdot x - 1)^2 \cdot \left(\lim\limits_{i \to \infty} 2^{2 \cdot i} \cdot x^{3 \cdot i} - 4 \cdot x^3\right)}{4 \cdot x^2 \cdot (4 \cdot x^3 - 1)} \right. \end{array} \right.$$

Рис. 10.17

Задача о бесконечном числе коробок с максимальным суммарным объемом и с одинаковыми пропорциями раскроя

На рисунке 10.17 с помощью оператора суммы формируется формула, задающая суммарный объем бесконечного числа коробок с пропорцией x в раскрое, и сделана попытка найти формулу для этого ряда. Пакет Mathcad 15 выдал «абсолютно точный и абсолютно бесполезный ответ», в то время как Mathcad 12 выдавал вполне приемлемый результат, и мы его покажем ниже. Это связано с тем, что в эти версии Mathcad встроена разная символьная математика, и мы об этом упоминали во Введении[8].

Была сделана попытка избавиться от суммы сходящегося ряда на сайте пакета Mathematica — см. рис. 10.18. Этот сайт не стал выдавать «страшную» формулу (см. рис. 10.13), а сообщил, что ряд сходится при определенных значениях параметра x.

[8] Вот что написано во Введении: «Mathcad изначально создавался как пакет численной математики, к которому впоследствии (в 5-й версии) было прикреплено ядро символьной математики из пакета Maple, которое затем (в 13-й версии) было заменено на ядро символьной математики из пакета MuPAD».

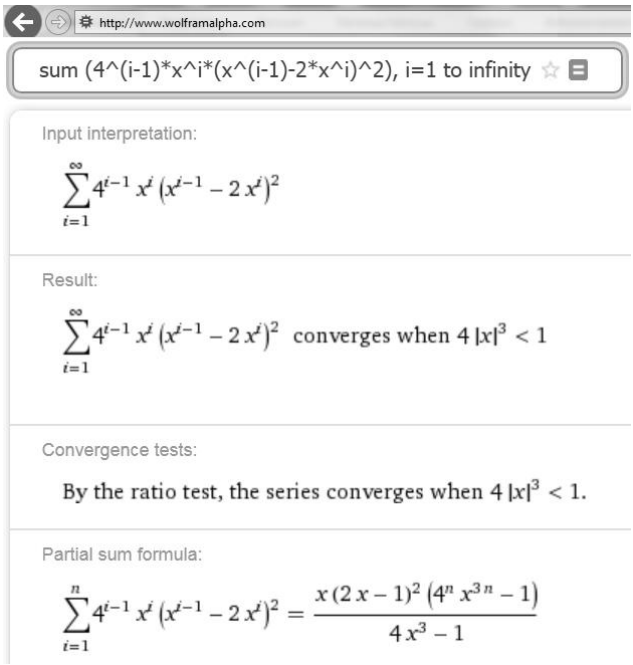


Рис. 10.18
Попытка упрощения суммы

$$\sum_{i=1}^{\infty} 4^{i-1} \cdot x^i \cdot \left(x^{i-1} - 2\,x^i\right)^2 \xrightarrow{assume,\,|x| < \sqrt[3]{\frac{1}{4}}} -\frac{x \cdot (2 \cdot x - 1)^2}{4 \cdot x^3 - 1}$$

$$\frac{d}{dx}\frac{x \cdot (2 \cdot x - 1)^2}{1 - 4 \cdot x^3} = 0 \xrightarrow[float,\,25]{solve} \begin{bmatrix} 0.5 \\ 0.76604444311189780352023927 \\ -0.93969262078590838405410 93 \\ 0.17364817766693034885171 66 \end{bmatrix}$$

Рис. 10.19
Решение задачи о бесконечном числе коробок с максимальным суммарным объемом и с одинаковыми пропорциями раскроя

Дополнительное условие по параметру x было через ключевое слово assume (предполагать) сообщено пакету Mathcad Prime, и он справился с задачей: нашел сумму сходящегося ряда и нашел корни соответствующего уравнения — см. рис. 10.19. Их оказалось четыре, последний из них — решение задачи. Это число (0.1736481...) можно считать некой новой математической константой, о которых мы рассуждали в конце этюда 4 (см. рис. 4.51).

Максимизация объема коробок — это одна из тем школьного факультатива, который ведет один из авторов в московском лицее № 1502 при Московском энергетическом институте, — см. рис. 8.11 на стр. 190.

Задания читателям

1. Создать программы расчета факториала и чисел Фибоначчи без рекурсии (рекуррентно) и сравнить время их выполнения со временем выполнения аналогичных программ с рекурсией.

2. Создать программу сортировки массива чисел с помощью рекурсии.

3. Найти в Интернете описание чисел Аккермана и создать программу с двойной рекурсией расчета этих чисел.

4. Создать анимацию свертывания крестообразной заготовки в коробку, но не на плоскости (см. рис. 10.7), а в объеме, добавив в рисунок перспективы. На рисунке 10.7 ее явно не хватает.

5. Создать анимацию изготовления пяти коробок — одной большой в центре и четырех маленьких по углам.

6. Получите формулу для функций V_{21}, V_{85} и т. д., по аналогии с формулой для функции $V_5(A, x)$, показанной на рис. 10.9. Попытайтесь решить данные задачи аналитически — символьным решением алгебраического уравнения.

7. Попытаться решить систему уравнений для случая с 21 коробками: составить функцию трех аргументов и использовать необходимое условие экстремума — приравнять все три частные производные нулю.

Этюд 11

ДА — НЕТ В СРЕДЕ MATHCAD, ИЛИ ОРИГАМИ БУЛЯ

В этюде будет рассказано о четкой и нечеткой логике, об особенностях ее применения в программировании.

Математика: булева функция, булева алгебра, экстремум булевой функции, нечеткая логика.

Физика: переключатели электрических схем, машина для голосования.

Информатика: булевы операторы, древовидный оператор, векторизация компьютерных вычислений.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127818**

В среде Mathcad 15 есть операторы, реализованные в виде префиксного и инфиксного операторов, — см. правую часть рис. 11.1, которые возвращают только два значения: 1 (не ноль) или 0 (ноль): Да — Нет, True (Истина) — False (Ложь). Эти операторы собраны на панели инструментов Boolean (английская версия Mathcad 15 — см. левую часть рис. 11.1) и Сравнение (русская версия Mathcad Prime — см. рис. 11.2).

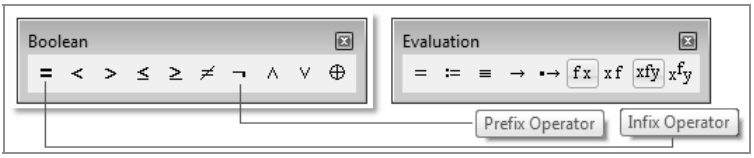


Рис. 11.1
Панель булевых операторов в среде Mathcad 15 (слева)

Рис. 11.2
Панель булевых операторов в среде Mathcad Prime

Иногда булевы[1] операторы (или, иначе, булевы функции) называют *двоичными* или *логическими*. Операндами этих операторов (аргументами этих функций) в среде Mathcad могут быть и вещественные числа, но при этом число, отличное от нуля, воспринимается как истина, а ноль как ложь.

В некоторых языках программирования (Pascal, например) начать работу с булевыми и другими типами переменных можно только после того, как они будут перечислены с указанием типа переменной. Это сделано для экономии памяти компьютера: под хранение булевой переменной выделяется намного меньше места, чем под хранение целочисленной, а тем более вещественной. Сейчас память компьютера, как правило, не является лимитирующим фактором при решении

[1] Слово «булевый» происходит от имени Джорджа Буля — английского математика и логика, профессора математики Королевского колледжа Корка (ныне Университетский колледж Корк). Д. Буль — создатель символической логики.

задач, и от такого предварительного объявления типов переменных отказались. Так, по крайней мере, делается в среде Mathcad.

Поговорим о булевых функциях [63], отталкиваясь от проблем, возникающих при работе с ними в среде Mathcad или в иных математических программах, а также в языках программирования, да и вообще, при использовании цифровой вычислительной техники, в основе которой лежит двоичный «атом» — элементарный элемент памяти, находящийся в одном из двух состояниях (заряжено-разряжено, намагничено-размагничено и т. д.). Из «атомов» (биты) составляются «молекулы» (байты), которые, в свою очередь, формируют новые «соединения» — переменные, массивы переменных (см. этюд 2) и т. д. — все то, чем оперируют программисты. Арифметические и прочие действия над числами в переменных и массивах переменных — это по своей сути не что иное, как разнообразные булевы (побитовые) операции.

1. АЗЫ БУЛЕВОЙ МАТЕМАТИКИ

1.1. Булевы функции одного аргумента

Базовых булевых функций одного аргумента *четыре*: см. таблицу 11.1, но на практике работают только с одной — с f_1, которую называют *отрицанием* (*инверсией*). Остальные три функции возвращают либо свой аргумент (функция f_2), либо константы 1 (f_3) и 0 (f_4).

Таблица 11.1

Булевы функции одного аргумента

a	f_1	f_2	f_3	f_4
0	1	0	1	0
1	0	1	1	0
Обозначение	¬a	a	1	0
	Not(a) !a ā			

В первой строке графы «Обозначение» таблицы 11.1 (и таблицы 11.2 ниже) показаны операторы Mathcad, которыми булевы функции реализуется. Символ ¬ (префиксный[2] оператор отрицания или инверсии) расположен на панелях, показанных на рис. 11.1 и 11.2.

Из этих функций можно составлять и более сложные функции одного аргумента, но они, как правило, не востребованы. Посмотрим, как, например, действует функция $f_2(f_4(f_1(a)))$. Если $a = 1$, то $f_1(a) = 0$, $f_4(f_1(a)) = 0$, $f_2(f_4(f_1(a))) = 0$. Если же $a = 0$, то $f_1(a) = 1$, $f_4(f_1(a)) = 0$, $f_2(f_4(f_1(a))) = 0$. Таблица значений говорит нам, что $f_2(f_4(f_1(a))) = f_4(a)$, поэтому сложную функцию есть смысл заменить простой базовой булевой функцией.

1.2. Булевы функции двух аргументов

Базовых булевых функций двух аргументов шестнадцать: см. таблицу 11.2.

Таблица 11.2 условно делится на две половины — на «именную» (f_1 – f_8) и безымянную (f_9 – f_{16}). Вот имена первых восьми функций. Вернее, семи (еще одна

[2] Определение «префиксный» означает, что символ оператора расположен до операнда. В среде Mathcad есть и постфиксные встроенные операторы — оператор факториала, например, (5!), где символ оператора (!) стоит после операнда (5). У инфиксного оператора два аргумента, а его символ стоит между ними (пример — оператор сложения с двумя слагаемыми-операндами). В среде Mathcad 15 есть также и древовидный пользовательский оператор — см. рис. 11.7 и 11.12 ниже.

«великолепная семерка» — см. этюд 4): две функции (f_5 и f_6) называются одинаково — импликация или логическое следование.

Таблица 11.2

Булевы функции двух аргументов

a b	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
0 0	0	0	1	0	1	1	1	1	0	0	1	1	0	0	1	0
0 1	0	1	0	1	0	1	0	1	0	1	1	0	0	1	1	0
1 0	0	1	0	1	1	0	0	1	1	0	0	1	1	0	1	0
1 1	1	1	1	0	1	1	0	0	0	0	0	0	1	1	1	0
Обозначение	$\wedge$	$\vee$	=	$\oplus$ $\neq$	$\geq$	$\leq$			>	<	$\neg$a	$\neg$b	a	b	1	0
	* × • И And & && min	+ ИЛИ Or || max	$\leftrightarrow$ $\equiv$ $\Leftrightarrow$ Eqv ==	<> >< Xor !=	$\rightarrow$ $\supseteq$ $\Rightarrow$ Imp	$\rightarrow$ $\supseteq$ $\Rightarrow$ Imp	$\downarrow$ $\neg$And	| $\neg$Or								

f_1 — конъюнкция (логическое умножение);

f_2 — дизъюнкция (логическое сложение);

f_3 — равнозначность (эквивалентность, тождественность);

f_4 — неравнозначность (неэквивалентность, разделительная дизъюнкция, сумма по модулю 2);

f_5 и f_6 — импликация (логическое следование): f_5 — импликация от a к b, f_6 — импликация от b к a;

f_7 — функция (стрелка) Пирса (функция Вебба, функция Даггера, антидизъюнкция — о ней мы поговорим особо: см. рис. 11.7 и 11.8);

f_8 — функция (штрих) Шеффера (антиконъюнкция).

Остальные восемь функций таблицы 11.2 (f_9 - f_{16}, как и три последние функции таблицы 11.1), повторяем, не имеют ни имен, ни практического применения[3]. Это либо константы (f_{15} и f_{16}), либо функции только одного аргумента (f_{11} - f_{14}). Имя, да и то условно, можно дать только функциям f_9 и f_{10}: инверсия (отрицание) импликации. Но эти две функции, как правило, применяются только к вещественным аргументам: пример π > e = 1 (да, численное значение отношения длины окружности к ее диаметру больше, чем численное значение основания натурального логарифма).

На рисунке 11.3 показаны операторы вызова в среде Mathcad 15 некоторых базовых булевых функций двух аргументов из таблицы 11.2. Эти операторы реализованы в виде инфиксных операторов — операторов, символ которого, как уже отмечено, находится между двух операндов-аргументов. Стрелка над операндами-векторами на рис. 11.3 — это специальный оператор векторизации, т. е. поэлементного выполнения действия над векторами.

У некоторых операторов, показанных на рис. 11.3, могут быть только булевы операнды (0 или 1), а у других — еще и вещественные: 0.1, 0,9, 2,5 и т. д. На рисунке 11.4 показана работа операторов Исключающее или[4] (Xor) и Не равно с вещест-

[3] В среде Mathcad Prime появился еще один оператор, возвращающий нуль или единицу: ∈ — *оператор принадлежности*. Этот инфиксный оператор определяет, принадлежит ли его первый операнд к одному из возможных числовых множеств (второй операнд): **C** — комплексные числа, **Q** — рациональные числа (этот оператор относится к символьной математике), **R** — вещественные числа и **Z** — целые числа (см. рис. 12.1 на стр. 273).

[4] Кстати говоря, наше разговорное «или» — это зачастую не простое «или», а именно «исключающее или». Когда мы спрашиваем, к примеру: «Ты идешь в кино или останешься дома?», мы вкладываем в этот вопрос именно «исключающее или», т. к. человек не может быть одновременно и в кино, и дома. И еще. В письменных текстах (и в данной книге тоже) появился гибрид «и» и «или», который только недавно редакторы перестали вымарывать из рукописей: «и/или».

венными операндами: не с 0 и 1, а с 0.1 и 0,9. Эти операторы при булевых операндах (рис. 11.3) дали одинаковые ответы, а при вещественных операндах (рис. 11.4) разные.

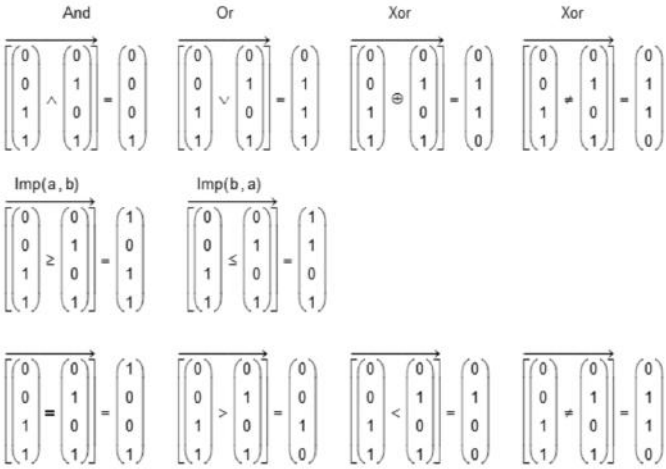


Рис. 11.3
Операторы вызова булевых функций в Mathcad 15

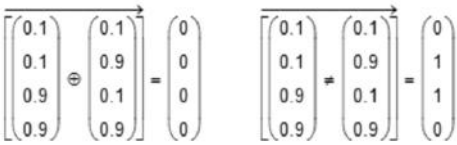


Рис. 11.4
Оператор «Исключающее или» и «Не равно» с вещественными аргументами

Ответ первого оператора на рис. 11.4 (одни нули) объясняется тем, что числа, отличные от нуля, булевыми функциями пакета Mathcad воспринимаются как единицы (`Истина`). Это связано с тем, что в ранних версиях Mathcad не было операторов `And` и `Or`, и данные булевы операции реализовывались через арифметические операторы умножения и сложения — см. рис. 11.5. При сложении же двух единиц (оператор `Or`) получается двойка (`Истина`) — число, отличное от нуля.

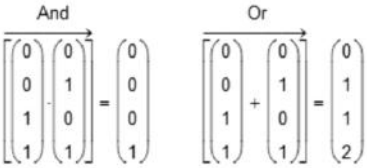


Рис. 11.5
Реализация логического умножения и логического сложения в пакете Mathcad ранних версий

Булевы функции `And` и `Or` часто иллюстрируют электрической цепью: последовательное соединение выключателей (устройств, находящихся в двух возможных положениях 0 и 1) — это конъюнкция (логическое умножение), а параллельное — дизъюнкция (логическое сложение). На рис. 11.6 показан также менее тривиальный пример — электрический аналог эквиваленции (`Eqv`): схема соединения двух выклю-

чателей, так чтобы свет независимо зажигался и тушился из двух разных мест. Такие выключатели будут полезны в длинном коридоре или на лестнице многоэтажного дома, на дачном участке и в других местах. На авторском сайте **http://twtmas.mpei.ac.ru/mas/Worksheets/Book_MC_12/4_25_Logic.mcd** можно поработать с таким виртуальным выключателем.

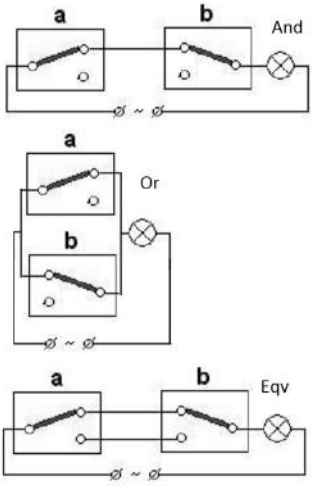


Рис. 11.6

Электрические аналоги трех булевых функций And, Or и Eqv

В таблицах 11.1 и 11.2 сделана попытка собрать все имена базовых функций и символы базовых операторов, использующихся для реализации булевой алгебры. Список, конечно, неполный. Можно расширить его примерами из других языков программирования (Pascal, Fortran и др.) и математических программ (Maple, Matlab, Mathematica и др.).

Можно отметить некую *избыточность* функции, собранных в таблицах 11.1 и 11.2. В конкретных языках программирования есть некий ограниченный набор встроенных булевых функций и операторов, с помощью которых можно составить любые другие булевы функции. Доказать достаточность такого набора для реализации любых булевых операторов — непростая математическая проблема. Если выбрано конечное число функций, описаны подчиняющиеся особым правилам действия с ними и доказана возможность реализации всех других функций посредством эти функций и действий, то говорят, что построена алгебра[5].

Вот перечень таких функций и операторов, встроенных в популярные программные среды:

- язык программирования BASIC: `Not`, `And`, `Or`, `Xor` и `Imp`;
- язык программирования C: `!`, `&` (логическое умножение), `&&` (побитовая конъюнкция), `! =`, ‖ и `= =`;
- математическая программа Mathcad: ¬, ∧, ∨ и ⊕ (см. рис. 11.1 и 11.2).

[5] Алгеброй называется непустое множество элементов любой природы с введенными для них алгебраическими операциями, удовлетворяющими заданным аксиомам. Объектом изучения алгебры являются не элементы, а алгебраические операции. Известны алгебры нескольких типов. Один из наиболее важных и наиболее изученных типов — *группы*, т. е. алгебры с одной ассоциативной бинарной операцией, содержащие единицу и для каждого элемента имеющие обратный ему элемент. Важнейшие типы алгебр с двумя бинарными операциями — это *кольца* и *поля*. Операции в них обычно называются сложением и умножением.

Недостающие двоичные функции (операторы) программист может ввести в программу сам через механизм пользовательских функций, и мы это покажем ниже.

Разделение двоичных функций и операторов на основные (базисные) и вспомогательные появилось задолго до компьютеров и узаконилось в виде двоичных алгебр (в скобках отмечен их базис):

- алгебра логики (¬, &, ∨, → и ↔);
- булева алгебра (¬, & и ∨);
- алгебра Жегалкина (&, ∨ и ⊕);
- алгебра Пирса (↓);
- алгебра Шеффера (|).

Две последние двоичные алгебры примечательны тем, что в их базисе всего лишь одна двоичная функция ↓ или |, опираясь на которую можно построить все остальные.

На рисунке 11.8 показан документ Mathcad 15, где с опорой на функцию Пирса (↓ — см. рис 11.7) построены другие булевы функции: одна функция одного аргумента (отрицание, инверсия — `Not`) и пять функций двух аргументов: `And`, `Or`, `Imp`, штрих Шеффера (|) и `Eqv`. Последние три функции (`Imp`, штрих Шеффера и `Eqv`) определены с использованием ранее определенных функций. Это сделано для большей компактности рисунка, но от механизма вложения пользовательских функций (`Imp(a, b): = Or(Not(a), b)`), например, можно отказаться и оперировать «для чистоты эксперимента» только функцией (штрихом) Пирса.

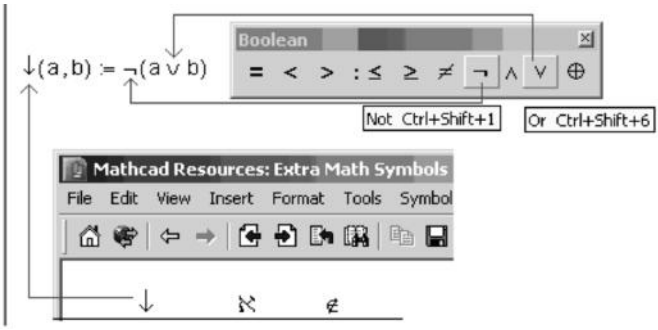


Рис. 11.7

Создание в среде Mathcad булевой функции Пирса с опорой на отрицание и логическое сложение

На алгебру Пирса и алгебру Шеффера в свое время возлагали большие надежды в смысле построения цифрового компьютера из однотипных элементов. Потом от этой идеи отказались по ряду причин, главная из которых в том, что любой компьютер и так состоит только из однотипных элементов — из транзисторов, объединенных в интегральные микросхемы (чипы).

Можно отметить и некую *недостаточность* набора математических инструментов, отображенных в таблицах 11.1 и 11.2. Возьмем, например, наверное, самую популярную функцию булевой алгебры — конъюнкцию (следствие ее популярности и в том, что у нее больше всего имен и символов для обозначения — см. столбец f_1 в таблице 11.2). Столбец истинности конъюнкции в таблице истинности (а так называют таблицы 11.1 и 11.2) по идее должен быть такой, как показано в таблице 11.3.

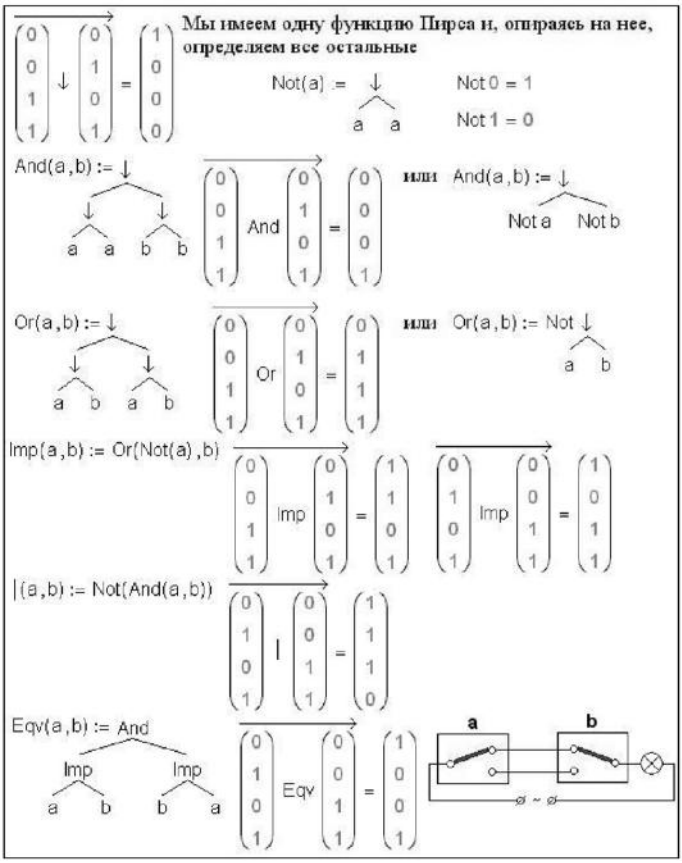


Рис. 11.8
Построение булевых функций с опорой на функцию Пирса

Таблица 11.3

Уточненная конъюнкция

a b	a And b (f_1)
0 –	0
– 0	0
1 0	0
1 1	1

Прочерк рядом с нулем в первом столбце таблицы 11.3 означает то, что если первый (a) аргумент равен нулю, то незачем проверять, чему равен второй аргумент (b), и наоборот. Такое предусматривают, создавая некоторые («неленивые») языки программирования — язык C, например. При программировании же в среде языка BASIC условный переход по конъюнкции можно записать так:

`If a And b Then...` (1-й способ)

но лучше так:

`If a Then If b Then...` или `If b Then If a Then...` (2-й способ)

Второй способ записи позволяет не только ускорять расчеты, но и избегать некоторых ошибок — логическое выражение b может иметь смысл только в том слу-

чае, если на альтернативный вопрос a дан положительный ответ. Вот типичный пример такой «программистской» ситуации:

```
If I> 0 Then If V(i) >V(i-1) Then...
```

Можно сказать, что в языке BASIC есть две конъюнкции: `And` и `Then If`.

В таблицах 11.1 и 11.2 мы, повторяем, собрали булевы функции одного (таблица 11.1) и двух (таблица 11.2) аргументов. Но, возвращаясь к конъюнкции, можно сказать, что эта функция имеет не два, а... полтора аргумента — см. таблицу 11.3.

Такую же нецелочисленность количества аргументов можно наблюдать и для других булевых функций:

Таблица 11.4

Булевы функции полутора, одного и нуля аргументов

a b	a Or b (f_2)
0 0	0
0 1	1
1 –	1
1 –	1

a b	a (f_{11})	¬a (f_{13})
0 –	0	1
0 –	0	1
1 –	1	0
1 –	1	0

a b	b (f_{12})	¬b (f_{14})
– 0	1	0
– 1	1	1
– 0	0	0
– 1	0	1

a b	1 (f_{15})	0 (f_{16})
– –	1	0
– –	1	0
– –	1	0
– –	1	0

Можно отметить, что в таблицу 11.2 попали операторы, изначально предназначенные для работы не с двоичными, а с вещественными операндами: «>», «<», «≥», «≤», «=» и «≠». Но если принять во внимание тот факт, что нуль и единица входят во множество вещественных чисел, то включение этих операторов в таблицу 11.2 вполне закономерно. В этом ряду («>», «<», «≥», «≤», «=» и «≠») также можно отметить и избыточность, и недостаточность. С избыточностью все более-менее ясно («больше», например, — это инверсия от оператора «меньше или равно» и т. д.). Недостаточность же можно наблюдать в том, например, что при работе с вещественными переменными вместо оператора «равно» более уместно использовать оператор «примерно равно», которого нет в списках встроенных. Можно также вспомнить о существовании понятий «намного больше» или «намного меньше». Эти операторы соотношения также возвращают двоичные значения, но имеют фактически уже не два, а три аргумента (операнда): сравниваемую пару вещественных чисел и некое наше контекстное представление о том, что такое «примерно» или «намного».

На рисунке 11.9 показано создание в среде Mathcad 15 булевого оператора `Примерно равно`. Мы использовали его в программах на рис. 4.25 (стр. 86) и 4.26 (стр. 87).

В среде Mathcad Prime появилась опция, позволяющая превращать оператор «равно» в оператор «примерно равно» — см. рис. 11.10.

Булевы функции в Mathcad можно использовать не только для логических вычислений. Ими можно пользоваться и при решении уравнений.

Пример 1. Решить уравнение

$$\sqrt[4]{\sin \pi x} - \sqrt{\cos 1.7x} = \sqrt[4]{16 - x^2} - \sqrt{x^6 + x^5 - 4x + 9}.$$

Для решения можно было бы воспользоваться одним из операторов-решателей, описанных в этюде 4. А предварительно хорошо было бы построить графики обеих частей уравнения (заметим, что это достаточно сделать лишь на отрезке [−4, 4], поскольку должно выполняться неравенство $16 \geq x^2$) и локализовать имеющиеся корни. Но можно поступить иначе.

Введем следующие логические переменные: $a := \sin \pi x \geq 0$; $b := \cos 1.7x \geq 0$; $c := x^6 + x^5 - 4x + 9 \geq 0$; $p := \sqrt[4]{\sin \pi x} - \sqrt{\cos 1.7x}$; ; $q := \sqrt[4]{16 - x^2} - \sqrt{x^6 + x^5 - 4x + 9}$.

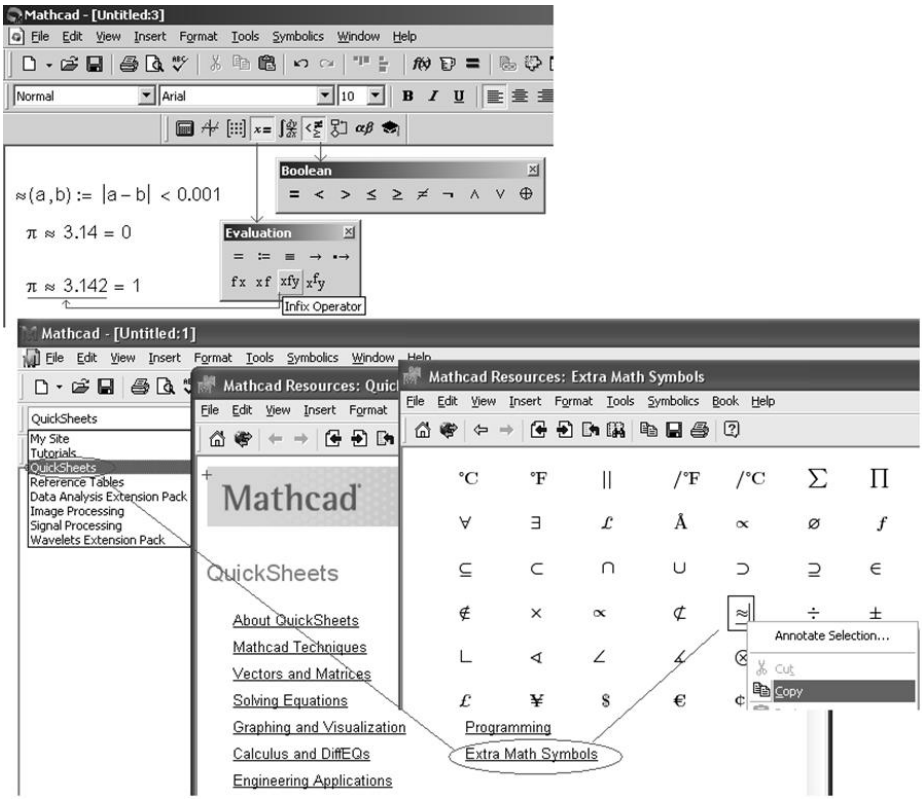


Рис. 11.9
Создание в среде Mathcad 15 булевого оператора ***Примерно равно***

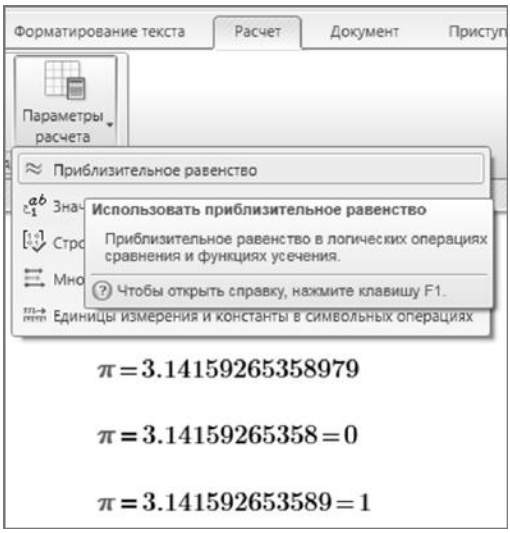


Рис. 11.10
Примерно равно *в среде Mathcad Prime*

Очевидно, что корни уравнения будут найдены приближенно (за исключением очевидного целого корня x = 0), поэтому следует выбрать нужный шаг, и для каждой точки указанного выше отрезка, полученной с помощью этого шага нужно проверить истинность высказывания

$f_{13}(a) \wedge f_{13}(b) \wedge f_{13}(c) \wedge (f_{13}(p) \wedge f_{13}(q) \vee f_{11}(p) \wedge f_{11}(q)) \wedge (p \approx q)$.

Проверяемое высказывание можно упростить, если преобразовать исходное уравнение к виду $\sqrt[4]{\sin \pi x} + \sqrt[4]{16 - x^2} = \sqrt{x^6 + x^5 - 4x + 9} + \sqrt{\cos 1.7x}$. Такой вид позволит не проверять истинность утверждения о том, что обе части уравнения в каждой допустимой точке имеют одинаковые знаки. И для $r := \sqrt[4]{\sin \pi x} + \sqrt[4]{16 - x^2}$ и $t := \sqrt{x^6 + x^5 - 4x + 9} + \sqrt{\cos 1.7x}$ потребуется в каждой полученной точке проверить истинность высказывания

$f_{13}(a) \wedge f_{13}(b) \wedge f_{13}(c) \wedge (r \approx t)$.

Пример 2. Решить уравнение $\sqrt[4]{\sin \pi x} + \sqrt{\cos 1.7y} = \sqrt[4]{16 - x^4} + \sqrt{9 - y^2}$.

Решением одного уравнения с двумя переменными может быть любое (в том числе и бесконечное) число точек на плоскости. Например, уравнение $\sqrt{-x^2 - y^2} = 0$ имеет единственное решение (0, 0), а уравнение $x^2 + y^2 = 1$ имеет бесконечно много решений — точек, образующих окружность единичного радиуса и центром в нуле. Решения уравнений с двумя неизвестными обсуждаются в этюде 4. Здесь же мы снова применим логические переменные: $a := \sin \pi x \geq 0$; $b := \cos 1.7y \geq 0$; $p := \sqrt[4]{\sin \pi x} + \sqrt{\cos 1.7y}$; $q := \sqrt[4]{16 - x^4} + \sqrt{9 - y^2}$, и будем проверять истинность высказывания

$f_{13}(a) \wedge f_{13}(b) \wedge (p \approx q)$

в точках прямоугольника $-2 \leq x \leq 2$, $-3 \leq y \leq 3$, заданных с определенным шагом по каждой из переменных. Все найденные таким способом точки дадут нам приближенное решение исходного уравнения.

Классическая булева алгебра оперирует аргументами-высказываниями, которые могут быть истинными или ложными. Например, высказывание a: «Студент сдал экзамен по математике», высказывание b: «Студент сдал экзамен по информатике», высказывание $a \wedge b$: «Студент сдал экзамены по математике и по информатике». Высказывания могут быть естественным образом связаны с множествами[6]. Например, A — множество студентов, сдавших экзамен по математике; B — множество студентов, сдавших экзамен по информатике. Вопрос может быть поставлен так: «Истинно или ложно утверждение о том, что данный студент принадлежит множеству $A \cdot B$?

Такое естественное соответствие булевой алгебры и алгебры множеств позволяет пользоваться преимуществами той или иной интерпретацией высказываний. Так, геометрическое наглядное представление отношений между высказываниями (множествами) в булевой алгебре изображают с помощью кругов[7] или иных фигур.

[6] Традиционно высказывания, или логические переменные, обозначают строчными буквами, а множества — прописными. Также традиционно отличаются обозначения логических операций в булевой алгебре и в алгебре множеств.

[7] При решении целого ряда задач Леонард Эйлер использовал идею изображения множеств с помощью кругов. Однако этим методом для геометрической интерпретации логических связей между понятиями ещё до Эйлера пользовался выдающийся немецкий философ и математик Готфрид Вильгельм Лейбниц. Методом кругов Эйлера пользовался и немецкий математик Эрнст Шрёдер в книге «Алгебра логики».

Такое изображение называют диаграммой Венна[8]. Элемент 1 булевой алгебры представляется как универсальное множество, или истинное высказывание, его можно изображать в виде квадрата и обозначать буквой Ω. Элементу 0 соответствует пустое множество, или ложное высказывание, его можно обозначать $\varnothing$. Некоторое непустое множество A, или некоторое высказывание a, представляется в виде круга или иной фигуры, расположенной в квадрате. То, что лежит за пределами множества A, является дополнением A или отрицанием высказывания a, т. е. $\neg A$. Сумма (дизъюнкция) двух множеств A и B представляется в виде объединения изображающих их фигур и может быть обозначена значком «+». Произведение (конъюнкция) двух множеств A и B представляется в виде общей части изображающих их фигур и может быть обозначена значком «·». Диаграмма Венна используются для наглядной иллюстрации справедливости аксиом и теорем булевой алгебры, а также для представления отношений между логическими высказываниями. Являясь частным случаем кругов Эйлера, диаграмма Венна изображает конечную булеву алгебру, наглядно описывая все ее свойства и комбинации этих свойств. При $n = 3$ эта диаграмма обычно изображается в виде трёх кругов с центрами в вершинах равностороннего треугольника и одинаковым радиусом, немного меньшим длины стороны треугольника. Вся эта конструкция помещена в квадрат.

С помощью диаграмм Венна можно упрощать или составлять из табличных функций различные булевы функции. Она позволяет понять справедливость логических тождеств. Очевидно, что равным (левой и правой) частям логических выражений соответствуют одинаковые фигуры на диаграмме Венна. Так, объединением круга A и сегмента $A \cdot B$ является круг A, поэтому, получаем, что $A + A \cdot B = A$. Заметим, что для любого высказывания (множества) A булево выражение $a + (\neg a)$ является истинным, т. е. $A + (\neg A) = \Omega$. Это же равносильно свойству $\neg(\neg a) = a$, которое еще называют «Законом исключенного третьего». Действительно, если A — это круг внутри квадрата, то $(\neg A)$ — это часть квадрата без круга, но тогда $\neg(\neg A)$ — это часть квадрата, из которого исключена часть квадрата без круга, т. е. просто круг.

Перечислим основные свойства булевой алгебры. Их легко понять, придавая высказываниям a, b, c значения «истина» и «ложь», или заштриховывая определенные области на диаграмме Венна для множеств A, B и C.

1. $A + A = A$.
2. $A \cdot A = A$.
3. $A + B = B + A$.
4. $A \cdot B = B \cdot A$.
5. $A + A \cdot B = A$.
6. $A + (B + C) = (A + B) + C$.
7. $A \cdot (B \cdot C) = (A \cdot B) \cdot C$.
8. $A \cdot (B + C) = A \cdot B + A \cdot C$.
9. $(A + B) \cdot (A + C) = A + B \cdot \mathrm{C}$.
10. $A + \Omega = \Omega$.
11. $\neg\Omega = \varnothing$.
12. $\neg\varnothing = \Omega$.
13. $A + \varnothing = A$.
14. $A \cdot \Omega = A$.
15. $A \cdot \varnothing = \varnothing$.
16. $A + (\neg A) = \Omega$.
17. $A \cdot (\neg A) = \varnothing$.
18. $\neg(A + B) = (\neg A) \cdot (\neg B)$.
19. $\neg(A \cdot B) = (\neg A) + (\neg B)$.
20. $\neg(\neg A) = A$.

[8] В математическую логику такие изображения были введены английским математиком Джоном Венном, подробно описавшим их в своей книге «Символическая логика», изданной в Лондоне в 1881 г.

Указанные свойства наиболее часто используют для доказательства равенств, задающих сложные высказывания или множества.

Например, требуется упростить логическое высказывание

```
f(a, b, c) = a ∨ a ∧ b ∨ b ∧ c ∨ (¬a) ∧ c.
```

Конечно, можно, используя базовые логические функции, составить таблицу значений этой функции, а потом поискать другую, более простую функцию с точно такими же значениями (если таковая существует). Но мы поступим следующим образом. Нарисуем диаграмму Венна в случае пересечения всех трех множеств (для удобства на рис. 11.11 круги мы заменили прямоугольниками). При указанном расположении прямоугольников мы видим, что

$$A + A \cdot B + B \cdot C + (\neg A) \cdot C = A + C.$$

Теперь строго докажем это равенство, используя свойства булевой алгебры:

$$\begin{gathered} A + A \cdot B + B \cdot C + (\neg A) \cdot C = \\ = (A + (\neg A) \cdot C) + (A \cdot B + B \cdot C) = \\ = (A + (\neg A) \cdot C) + B \cdot (A + C) = \\ = (A + (\neg A)) \cdot (A + C) + B \cdot (A + C) = \\ = \Omega \cdot (A + C) + B \cdot (A + C) = \\ = (\Omega + B) \cdot (A + C) = \Omega \cdot (A + C) = A + C. \end{gathered}$$

Не всегда удается получить строгое доказательство предполагаемого равенства. А если равенство не верное, то найти решение еще труднее, так как, проводя преобразования, надо привести равенство к противоречию. Графический метод с использованием диаграмм Венна не является строгим или полным решением в том случае, если равенство справедливо, поскольку в случае графического решения надо будет рассмотреть все возможные расположения всех множеств. Но диаграммы дают однозначный ответ в том случае, когда равенство не справедливо. Действительно, если удалось найти хотя бы один пример, когда равенство не верное, значит, оно точно не выполняется.

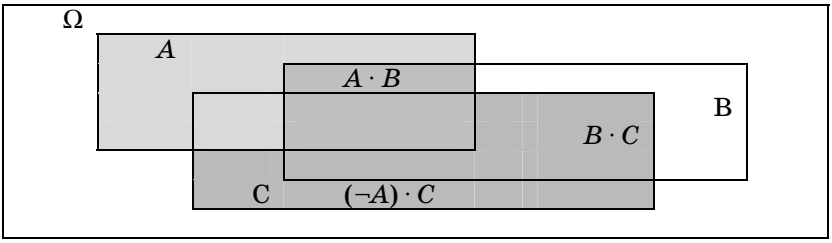


Рис. 11.11
Диаграмма Венна для $A + A \cdot B + B \cdot C + (\neg A) \cdot C$

Очевидно, что использование диаграммы особенно удобно, если высказываний достаточно много.

Если говорить не о классической булевой алгебре, а о реальной практике программирования, то следует признать, что переменные, фигурирующие в таблицах 11.1 и 11.2, могут принимать не два (0 или 1), а три значения: 0, 1 и неопределенно. Эту особенность мы уже зафиксировали в таблицах 11.3 и 11.4, где вместо конкретных значений аргументов (0 или 1) стоит прочерк. В языках программирования есть инструменты, обработки таких «прочерков» в таблицах истинности (см. пример на рис. 4.28 и 4.29 на стр. 88. Если аргумент булевой функции не определен, то расчет может либо прерываться сообщением об ошибке, либо идти по третьему сценарию.

Аргументы булевых функций могут принимать не два и не три, а... бесконечное множество вещественных значений. Это множество делится на две предельно неравные части: на нуль и на ненуль (¬0, если говорить языком таблицы 11.1 —

на числа, отличные от нуля, которые, повторяем, булевыми функциями пакета Mathcad воспринимаются как единицы). Бывает и так, что булева функция возвращает не только нули и единицы. Вот, например, как работает функция `Or` в одной из реализаций языка BASIC (вернее, вариант языка BASIC, которым комплектовалась популярная в свое время ПЭВМ «Искра-226» — **http://www.ic.kz/~ksxi/musei/int6.htm** калька машины «Wang-2000»): см. таблицу 11.5.

Таблица 11.5

«Дизъюнктивная конъюнкция»

a	b	a Or b
0	0	0
0	¬0	1
¬0	0	1
¬0	¬0	2

Можно допустить и такую работу расширенного оператора `Or`: см. таблицу 11.6.

Таблица 11.6

Расширенная конъюнкция (дизъюнкция)

a	b	a Or b
0	0	0
0	¬0	1
¬0	0	2
¬0	¬0	3

Одно дело, когда первый аргумент (операнд) не равно нулю, другое — когда второй, и третье — когда оба одновременно.

Подытоживая наш разбор таблиц 11.1 и 11.2, можно сказать, что описываемые булевы функции в реальных компьютерных реализациях могут иметь недвоичные аргументы и возвращать опять же недвоичные результаты. Но особого недвоичного смысла в этом нет. Просто, повторяем, по технологическим причинам вещественные переменные в описываемых реализациях языков программирования (BASIC, Mathcad) «по совместительству» исполняют роль двоичных. При этом булевы функции воспринимают свои вещественные аргументы «двоично»: нуль есть нуль («`Нет`», «`False`»), а все остальное единица («`Да`», «`True`»).

Эта, можно сказать, «категоричность» описываемых встроенных функций вступает в противоречие с положениями *теории нечетких множеств* (fuzzy sets — см. этюд 12) и теории нечеткой логики (fuzzy logic [64, 65]). Необходимо, например, статистически обрабатывать на компьютере не только «черно-белые» (двоичные) ответы анкетируемых типа «`Да (1)`» — «`Нет (0)`», но и «цветные» (вещественные) ответы: «`Да (1)`», «`Скорее да, чем нет (0.75, например)`», «`Ни да, ни нет (0,5)`», «`Скорее нет, чем да (0.25, например)`» и «`Нет (0)`». Если говорить не о статистике, а об электротехнике и вернуться к электрическим цепям, которыми часто иллюстрируют работу булевых функций (см. рис. 11.6), то можно упомянуть тот факт, что сейчас в быту получают распространение выключатели, плавно меняющие накал ламп от 100% до нуля. Еще раньше такие устройства стали применять в театрах и кинозалах. Медики уверяют, что плавный переход от света к темноте через полумрак не портит зрение. (В кинотеатрах свет тушат плавно, конечно, не по медицинским соображениям, а по другим причинам — если резко погасить свет, то может возникнуть паника.)

Можно привести множество других примеров, толкающих к тому, что аргументы функций, перечисленных в таблицах 11.1 и 11.2, могут и должны быть не только двоичными, но и вещественными числами от нуля до единицы. И функции, перечисленные в таблицах 11.1 и 11.2, должны возвращать вещественные значения, плавно меняющиеся от нуля до единицы. Вот как, например, можно задать «плавную» функцию отрицания — см. рис. 11.12.

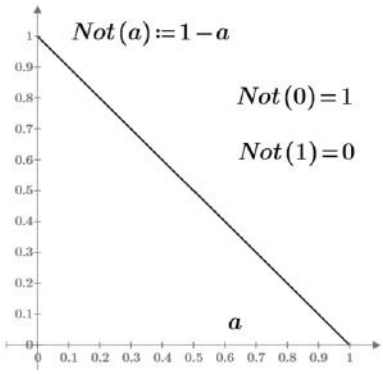


Рис. 11.12
Нечеткое отрицание

«Плавная» конъюнкции (`And`) и «плавная» дизъюнкция (`Or`)[9] получаются сами собой, если вспомнить о том, что одно из обозначений конъюнкции — это `min` (минимум — см. столбец f_1 в таблице 11.2), а одно из обозначений дизъюнкции — это `max` (максимум — см. столбец f_2):

```
And(a, b) : = min(a, b) Or(a, b) : = min(a, b).
```

На рисунках 11.13–11.16 показано трехмерное графическое отображение нечетких булевых функций `And`, `Or` и `Eqv`. Последняя функция представлена в двух вариантах `Eqv` (рис. 11.15) и `Eqv1` (рис. 11.16).

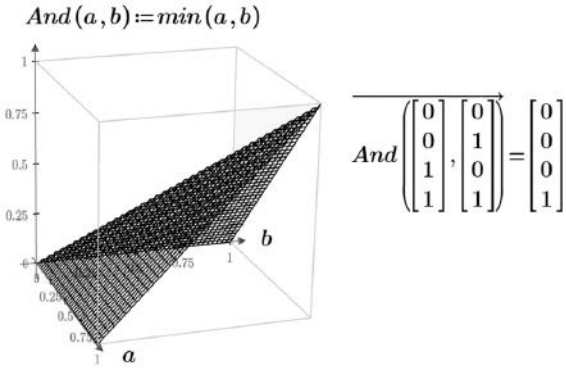


Рис. 11.13
График нечеткого ***And***

[9] Некоторым пользователям Mathcad трудно запомнить, что ^ — это логическое умножение, а ∨ — логическое сложение. Поэтому не будет лишним ввести в расчет пользовательские функции `And` и `Or` даже в том случае, если мы работаем только с четкой логикой.

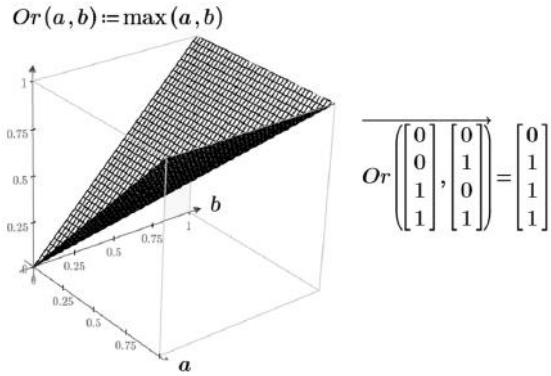


Рис. 11.14
График нечеткого ***Or***

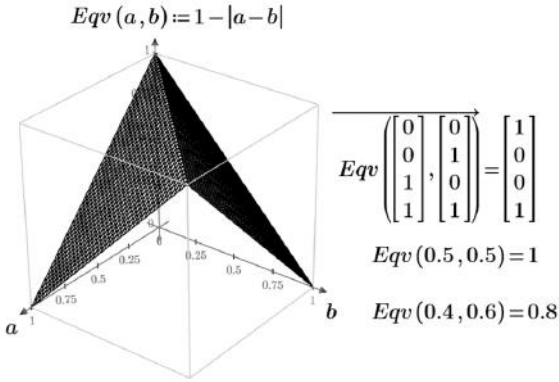


Рис. 11.15
График нечеткого ***Eqv***

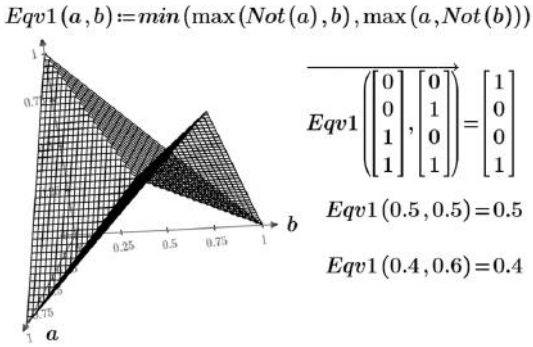


Рис. 11.16
График нечеткого ***Eqv*** *(второй вариант)*

Ломаные поверхности, показанные на рисунках 11.13–11.16 и 11.19 ниже, похожи на японские оригами — фигурки, сложенные из листа бумаги.

На рисунке 11.17 показано создание нечетких булевых функции и их отображение на графиках линий одного уровня (на контурных графиках).

$$Not(a) := 1 - a \qquad And(a,b) := min(a,b) \qquad Or(a,b) := \max(a,b)$$

$$\downarrow(a,b) := Not(And(a,b)) \qquad |(a,b) := Not(Or(a,b)) \qquad Imp(a,b) := Or(a, Not(b))$$

$$Eqv(a,b) := And(Imp(a,b), Imp(b,a)) \qquad Xor(a,b) := Not(Eqv(a,b))$$

$$a := \begin{bmatrix}0\\0\\1\\1\end{bmatrix} \quad b := \begin{bmatrix}0\\1\\0\\1\end{bmatrix} \quad \overrightarrow{Imp(a,b)} = \begin{bmatrix}1\\0\\1\\1\end{bmatrix} \quad \overrightarrow{Imp(b,a)} = \begin{bmatrix}1\\1\\0\\1\end{bmatrix} \quad \overrightarrow{Eqv(a,b)} = \begin{bmatrix}1\\0\\0\\1\end{bmatrix} \quad \overrightarrow{\downarrow(a,b)} = \begin{bmatrix}1\\1\\1\\0\end{bmatrix} \quad \overrightarrow{|(a,b)} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \quad \overrightarrow{Xor(a,b)} = \begin{bmatrix}0\\1\\1\\0\end{bmatrix}$$

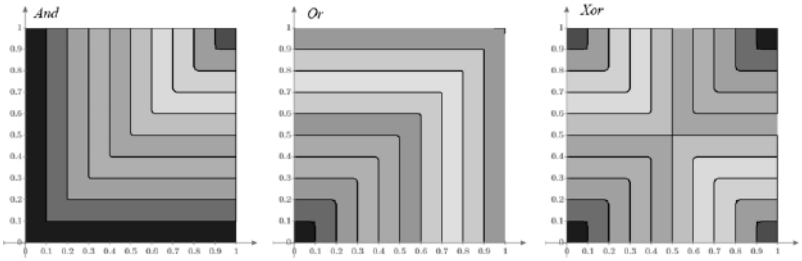


Рис. 11.17
Нечеткие булевы функции

1.3. Функции многих аргументов

Булевых функций трех аргументов 256, а четырех... Читатель, определи это сам!

Функции `And` и `Or`, если их отожествлять с функциями `min` и `max` (см. выше), могут иметь переменное число аргументов — с четкими (`0` или `1`) или нечеткими (от `0` до `1`) значениями. Этим свойством обладают и некоторые другие булевы функции. Какие? Читатель, определи опять же это сам!

На рисунках 11.18 и 11.19 показано формирование в среде Mathcad нечеткой функции трех аргументов, возвращающей... решение жюри присяжных, которые могут выдавать уже не «черно-белые» ответы (виновен — невиновен), а... «цветные»: виновен на 30%, невиновен на 70% и т. д. В электрическом аналоге машинки для голосования выключатели (0 или 1) заменены на реостаты (от 0 до 1).

Печальное примечание. Говорят, что в США электрический стул приводят в смертельное действие несколько человек. При этом настоящий рубильник приводится в действие только одним человеком. Остальные участники этой экзекуции включают фальшивые рубильники. И никто не знает, где фальшивый, а где настоящий рубильник. Такая несколько ханжеская процедура дает возможность каждому такому палачу думать, что не он, а кто-то другой был причиной смерти человека. Если же все настоящие рубильники заменить на реостаты (см. ниже рис. 11.18), плавно меняющими напряжение, то смерть можно заменить на нелетальное наказание: приговоренный преступник получит удар током (сильный или слабый), но останется жив.

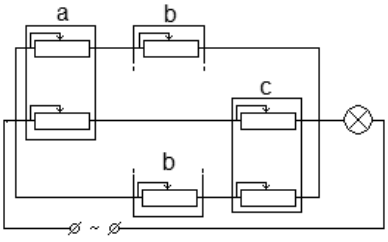


Рис. 11.18
Машина для голосования: параллельное (Or) соединение последовательно (And) соединенных выключателей (реостатов)

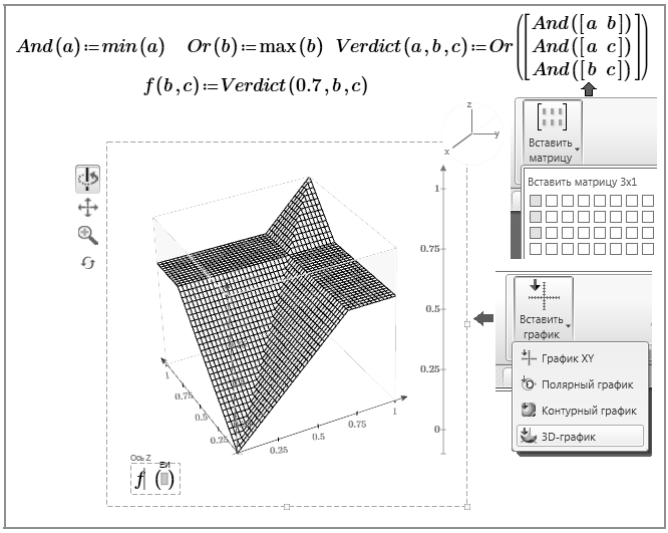


Рис. 11.19
Функция нечеткого голосование трех членов жюри

Функция `Verdict`, показанная на рис. 11.19, при двоичных аргументах возвращает двоичный ответ, а при вещественных — вещественный, естественно. Показан соответствующий «булев кубик» (оригами) при `a = 0.7` — мы видим гибрид конъюнкции с дизъюнкцией: нечеткое (от 0 до 1) мнение одного члена жюри плавно переводит вердикт из области `Or` (см. рис. 11.14) в область `And` (см. рис. 11.13).

Задания читателям

1. Создать функцию, подобную той, какая показана на рис. 11.19, но не для трех, а для любого числа членов жюри присяжных, выносящих свой вердикт нечетко — вещественным числом от 0 до 1.
2. Создать булеву функцию включения лампочки с любого места.
3. Найти в Интернете логические головоломки и попытаться решить их с помощью средств, описанных в данном этюде.

Этюд 12

НЕЧЕТКОЕ МНОЖЕСТВО, ИЛИ ОПТИМАЛЬНОЕ ПОЖАРНОЕ ВЕДРО

В этюде будет рассказано о решении задач оптимизации и о том, что такое теория нечетких множеств и как ее можно приложить к «четким» цифровым компьютерам.

Математика: множество, подмножество, объем цилиндра, объем конуса, объем шара, предел последовательности, площадь поверхности, минимум функции, производная, функция нормального распределения, Гауссова кривая, линии уровня функции двух переменных, функция принадлежности элемента нечеткому множеству, сложение нечетких множеств, перемножение нечетких множеств, целевая функция.

Физика: объем тела, фильтрование.

Информатика: оператор принадлежности к множеству, компьютерные аналитические преобразования, полностью размерная функция, функции для решения задач оптимизации с ограничениями.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127817**

В математике есть очень важные понятия *множество* и *подмножество*.

Есть, например, множество геометрических тел. В этом множестве есть подмножество конусов. В этом подмножестве есть (под)подмножество прямых круговых конусов.

В среде Mathcad Prime есть оператор, определяющий, принадлежит ли значение определенному числовому множеству, и мы уже касались его в предыдущем этюде. Так, из рисунка 12.1 можно видеть, что число π принадлежит (1 — да) числовым множествам «Комплексные числа `C`» и «Действительные числа `R`», но не принадлежит (0 — нет) множествам «Рациональные числа `Q`» и «Целые числа `Z`». В программе, показанной на рисунке 17.2 на стр. 368, при решении задачи о рыбаках и рыбке мы будем использовать оператор принадлежности числа к множеству `Z` целых чисел.

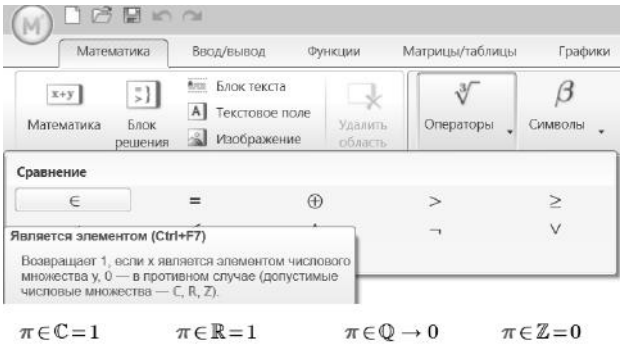


Рис. 12.1

Оператор принадлежности к множеству чисел

В среде Mathcad есть и другие функции для работы с множеством, возвращающие 0 или 1. Вот некоторые из них:

`IsString` — принадлежит ли аргумент функции к строке:

```
IsString(«123») = 1, но IsString(123) = 0;
```

`IsArray` — является ли аргумент массивом (an array — вектор или матрица — см. этюд 2);

`IsFunction` — является ли аргумент функцией;

`IsScalar` — является ли аргумент скаляром;

`USUnitsOf` — определяет, к какому множеству физических величин (длина, объем, площадь и т. д.) принадлежит аргумент.

С понятием «множество» мы часто сталкиваемся и в обыденной жизни.

Пример. Есть множество ведер. В этом множестве есть подмножество пожарных ведер.

Давайте поговорим о них.

Пожарное ведро

Пожарное ведро отличается от обычного ведра тем, что оно выполнено в виде *прямого кругового конуса*. Обычное же ведро — это обычно усеченный прямой круговой конус или прямой круговой цилиндр. Пожарное ведро делается в виде конуса для того, чтобы у него не было дна, на которое его можно было поставить. Воду в такое ведро набрать можно, но ее сразу нужно будет вылить на очаг возгорания, не выпуская ведра из рук. Использовать в быту такое ведро для других целей (для уборки помещений, например) практически невозможно. Правда, один из авторов этой книги, проходя в свое время срочную службу в армии (см. **http://twt.mpei.ac.ru/ochkov/Army.htm**) и моя там, как водится, полы в казарме, ставил пожарное ведро (а другого не было) в перевернутую табуретку. Отсюда наверно и его «любовь» к пожарным ведрам, вылившаяся на страницы этой и других его книг по Mathcad. Заезжая на свой участок в дачном кооперативе, этот автор первым делом видит пожарный щит с пожарным ведром. Еще один атрибут «бездонного» пожарного ведра — это его красный цвет. Но эта деталь хоть и важная, но лишняя для нашего дальнейшего повествования.

Конус наряду с шаром-мячом и кубиками — это одно из первых геометрических тел, с которыми сталкивается ребёнок. Вспомним игрушечную пирамидку, состоящую из отдельных дисков, нанизанных на центральный стержень, или головоломку «Ханойские башни», о которой рассказано в этюде 10 (см. рис. 10.4 на стр. 243). Эти игрушки подсказывают, как можно доказать, что объем конуса равен ровно одной трети объема цилиндра, в который вписан конус. На рисунке 12.2 показан Mathcad-документ с расчетом объема детской пирамидки с радиусом основания `r` и высотой `h`, состоящей из `n` дисков — прямых круговой цилиндров без отверстия в середине.

Более-менее сложная часть этого расчета — это определение, как будет меняться радиус `R` очередного *i*-го диска. На рисунке 12.3 оператор цикла с параметром `i` для расчета суммы заменен на сам оператор суммы. Затем с помощью оператора определения предела (если число дисков стремится к бесконечности, то ступенчатая пирамидка превращается в гладкий конус) сделано то, что и требовалось доказать — объем конуса равен одной трети объема цилиндра, в который вписан конус. Когда ребёнок собирает пирамидку, то можно считать, что он готовится к изучению азов интегрального исчисления. А чему равен объем шара? Можно обратиться к справочникам на книжной полке или в Интернете и найти ответ, а

можно вывести соответствующую формулу, вообразив себе ту же детскую пирамидку, выполненную не в виде конуса, а в виде полушария.

$$V(r,h,n) := \left\| \begin{array}{l} \Delta h \leftarrow \dfrac{h}{n} \\ V \leftarrow 0 \\ \text{for } i \in 0..n-1 \\ \quad \left\| \begin{array}{l} R \leftarrow r \cdot \left(1 - \dfrac{i \cdot \Delta h}{h}\right) \\ v \leftarrow \pi \cdot R^2 \cdot \Delta h \\ V \leftarrow V + v \end{array} \right. \\ V \end{array} \right.$$

$r := 10\ cm \qquad h := 20\ cm$

$n := 7 \qquad V(r,h,n) = 2.565\ L$

$n := 1000000 \qquad V(r,h,n) = 2.094\ L$

$\dfrac{1}{3} \cdot \pi \cdot r^2 \cdot h = 2.094\ L$

Рис. 12.2

Расчет объема детской пирамидки-конуса, состоящего из n дисков-цилиндров

$$V(r,h,n) := \sum_{i=0}^{n-1} \pi \cdot \left(r \cdot \left(1 - \frac{i \cdot \frac{h}{n}}{h}\right)\right)^2 \cdot \frac{h}{n}$$

$$\lim_{n \to \infty} V(r,h,n) \to \frac{\pi \cdot h \cdot r^2}{3}$$

Рис. 12.3

Сумма и предел при расчете объема детской пирамидки

Прямой круговой конус (далее просто конус) имеет два параметра — радиус основания r и высоту h, по которым можно вычислить не только его объем V (см. рис. 12.3), но и площадь полной поверхности S, состоящей из боковой поверхности и поверхности круга-основания.

С конусом связано много интересных задач оптимизации. Давайте решим некоторые из них.

Задача 1. Определить отношение радиуса основания конуса r к его высоте h, при котором площадь его полной поверхности S будет минимальна. Дети любят пускать мыльные пузыри или смотреть, как это делают взрослые. Представим себе такой фокус. Пускаются мыльные пузыри, но не в виде сфер (замкнутых поверхностей с минимальным значением площади поверхности при заданном значении объема), а в виде конусов с таким же свойством.

На рисунке 12.4 показан, скажем так, *инженерный подход* к решению этой задачи для конуса с объемом 100 m^3. Инженеру дано задание спроектировать емкость для хранения какого-нибудь сыпучего материала (песка или цемента, например) в виде перевернутого конуса с крышкой и потратить на это минимум металла.

$$V(r,h) := \frac{1}{3}\pi \cdot r^2 \cdot h \qquad S(r,h) := \pi \cdot r \cdot \sqrt{r^2 + h^2} + \pi \cdot r^2$$

Решить

Начальные приближения

$r := 1\ m$

$h := 1\ m$

Ограничения

$V(r,h) = 100\ m^3$

Решатель

$$\begin{bmatrix} r \\ h \end{bmatrix} := \text{Minimize}(S, r, h) = \begin{bmatrix} 3.232 \\ 9.142 \end{bmatrix} m$$

$$\frac{r}{h} = 0.354 \qquad V(r,h) = 100\ m^3 \qquad S(r,h) = 131.269\ m^2$$

Рис. 12.4
Конус с минимальной площадью поверхности: решение 1

Коническую емкость заданного объема можно построить с разными соотношениями r и h. В Mathcad-расчете, показанном на рисунке 12.4, определяются такие значения r и h, при которых объем конуса остается равным 100 m^3 (ограничение), а площадь его поверхности при этом будет минимальна. В расчете использован блок Решить с тремя областями: область *начальных приближений* к решению, область *ограничений* и область *решения* со встроенной функцией Minimize. Эта функция по особому численному алгоритму (см. этюд 13) меняет значения переменных r и h, начиная от начальных приближений, так, чтобы функция S приняла минимальное значение, а ограничения бы не нарушались. В нашей задаче ограничение одно — объем конуса должен оставаться равным заданным 100 m^3. Но в эту область можно добавить и дополнительные ограничения, например, такое. Если в емкости хранится сыпучий материал (вспомним песочные часы), то угол наклона боковой поверхности нашего перевернутого конуса должен быть не менее определенного значения. Иначе сыпучий материал застрянет в конусе.

Можно немного подумать и сообразить[1], что наша задача о минимальной площади поверхности конуса с заданным объемом имеет не два (r и h), а один (r или h) параметр. Если задан объем конуса V и, например, его радиус r, то несложно определить его высоту h. Этот подход к решению нашей задачи оптимизации размеров конуса отображен на рисунке 12.5.

В решении, показанном на рисунке 12.5, пользовательская функция с именем S после несложных ручных аналитических преобразований имеет уже не два (r и h, как в решении на рис. 12.4), а один (r) аргумент. Это позволило построить декартов график изменения значения функции S от аргумента r при заданном объеме конуса V, на котором четко видна точка минимума, значение которого уточнено с помощью той же функции Minimize.

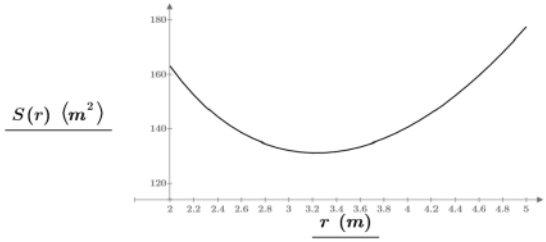
Строго говоря, в этом случае функция S является функцией двух переменных r и V (см. рис. 12.4). Фиксирование значения V : = 100 m^3 позволяет построить график сечения поверхности S = S(r, V) плоскостью V = 100 m^3 (см. рис. 12.5). Для понимания, где расположены точки минимума указанной функции достаточно построить ее трехмерный график.

[1] «Можно немного подумать и сообразить», что наша парадоксальная транспортная задача, которую мы будем решать в этюде 15 (см. рис. 15.16 и 15.17 на стр. 351), тоже имеет меньшее число неизвестных — не четыре, а только одно.

$V := 100\ m^3 \qquad V = \frac{1}{3}\pi \cdot r^2 \cdot h \qquad h = \frac{3 \cdot V}{\pi \cdot r^2}$

$S = \pi \cdot r \cdot \sqrt{r^2 + h^2} + \pi \cdot r^2 \qquad S(r) := \pi \cdot r \cdot \sqrt{r^2 + \left(\frac{3 \cdot V}{\pi \cdot r^2}\right)^2} + \pi \cdot r^2$

$r := 2\ m, 2.1\ m .. 5\ m$



$r := 3.2\ m \qquad \mathrm{r} := \mathrm{Minimize}(S, r) = 3.232\ m$

$h := \frac{3 \cdot V}{\pi \cdot \mathrm{r}^2} = 9.142\ m \qquad \frac{1}{3}\pi \cdot \mathrm{r}^2 \cdot h = 100\ m^3 \qquad S(\mathrm{r}) = 131.269\ m^2$

Рис. 12.5
Конус с минимальной площадью поверхности: решение 2

Но решение, реализованное на рисунке 12.4, тоже имеет право на жизнь. В нем реализован KISS-принцип. KISS — это аббревиатура английской фразы «Keep it simple, stupid» — «Делай это проще, дурачок!» Она призывает решать поставленные задачи простыми методами и прибегать к изощренным алгоритмам только тогда, когда простые методы дают сбой или сложны в реализации. «Изощрения» же нам понадобятся в еще одном методе решении этой задачи оптимизации — см. рисунок 12.6.

На рисунке 12.6 показан уже не инженерный, а *математический* подход к решению задачи. Вернемся к рисунку 12.4. Там рассчитываются не только конкретные значения `r` и `h`, но и их отношение, которое получилось равным 0.354. Можно менять значение заданного объема конуса `V` и видеть, что значение этой дроби не меняется. Но это можно показать и, главное, доказать по-другому. Математический подход к решению задачи базируется на том факте, что в точке минимума производная гладкой непрерывной функции равна нулю. Это в частности выражается в том, что на графике рисунка 12.5 касательная к кривой `S(r)` в точке минимума (`r = 3.232 m`) параллельна оси ординат.

В решении на рисунке 12.6 задействованы четыре оператора символьной математики Mathcad:

1. Оператор `solve` поиска корня уравнения или нуля функции. Этот оператор в нашем решении используется дважды: для решения уравнения (для вывода формулы расчета высоты конуса по его объему и радиусу основания — см. третий оператор на рисунке 12.6) и для нахождения нулей производной функции `S(V, r)` по `r` при `V = constant`. В первом случае корень уравнения можно найти и в уме (это, кстати, мы проделали во втором варианте решения задачи на рисунке 12.5). Но в третьем варианте решения (рисунок 12.6 — поиск корней производной) это сделать без компьютера вряд ли удастся. Вернее, удастся хорошему математику, но не простому инженеру.

2. Оператор `substitute` замены переменной на другую переменную или выражение. В решении на рисунке 12.5 мы вручную провели замену переменной `h` на выражение `(3 · V / π) / r²`. На рисунке 12.6 показано, как это автоматически делается в среде Mathcad.

$$V(r,h)=\frac{1}{3}\pi\cdot r^2\cdot h \qquad S(r,h)=\pi\cdot r\cdot\sqrt{r^2+h^2}+\pi\cdot r^2 \qquad V=\frac{1}{3}\pi\cdot r^2\cdot h \xrightarrow{solve,h} \frac{3\cdot V}{\pi\cdot r^2}$$

copy

$$\pi\cdot r\cdot\sqrt{r^2+h^2}+\pi\cdot r^2 \xrightarrow{substitute,\,h=\frac{3\cdot V}{\pi\cdot r^2}} r\cdot\left(\pi\cdot r+\sqrt{\frac{\pi^2\cdot r^6+9\cdot V^2}{r^4}}\right)$$

copy

$$\frac{d}{dr}r\cdot\left(\pi\cdot r+\sqrt{\frac{\pi^2\cdot r^6+9\cdot V^2}{r^4}}\right) \xrightarrow{simplify} \frac{\sqrt{\pi^2\cdot r^2+\frac{9\cdot V^2}{r^4}}\cdot\left(2\cdot\pi^2\cdot r^6-9\cdot V^2+2\cdot\pi\cdot r^5\cdot\sqrt{\pi^2\cdot r^2+\frac{9\cdot V^2}{r^4}}\right)}{\pi^2\cdot r^6+9\cdot V^2}$$

copy

$$\frac{\sqrt{\pi^2\cdot r^2+\frac{9\cdot V^2}{r^4}}\cdot\left(2\cdot\pi^2\cdot r^6-9\cdot V^2+2\cdot\pi\cdot r^5\cdot\sqrt{\pi^2\cdot r^2+\frac{9\cdot V^2}{r^4}}\right)}{\pi^2\cdot r^6+9\cdot V^2} \xrightarrow{solve,r}$$

$$\frac{8^{\frac{5}{6}}\cdot 9^{\frac{1}{6}}\cdot V^{\frac{1}{3}}}{8\cdot\pi^{\frac{1}{3}}}$$

$$-\frac{8^{\frac{5}{6}}\cdot 9^{\frac{1}{6}}\cdot V^{\frac{1}{3}}}{8\cdot\pi^{\frac{1}{3}}}$$

И еще 8 нулей

copy

$$\frac{r_{opt}}{h_{opt}}=\frac{\frac{8^6\cdot 9^6\cdot V^3}{8\cdot\pi^{\frac{1}{3}}}}{\pi\cdot\left(\frac{8^{\frac{5}{6}}\cdot 9^{\frac{1}{6}}\cdot V^{\frac{1}{3}}}{8\cdot\pi^{\frac{1}{3}}}\right)^2} \xrightarrow{simplify} \frac{r_{opt}}{h_{opt}}=\frac{\sqrt{2}}{4}$$

copy

Рис. 12.6

Конус с минимальной площадью поверхности: решение 3

3. Оператор d■ / d■ взятия производной.

4. Оператор `simplify` упрощения выражения. Мы упростили производную для того, чтобы она уместилась на рисунке 12.6.

Производная по радиусу основания от функции площади поверхности конуса имеет 10 нулей — два действительных противоположных по знаку (они показаны на рисунке 12.6), и восемь комплексных с мнимой частью. Первый нуль взят (скопирован и вставлен) для того, чтобы доказать, что отношение радиуса основания конуса с минимальной площадью поверхности к его высоте при любом объеме конуса равно корню из двух, деленному на четыре, т. е. равно приблизительно 0.354 (см. рисунок 12.4).

Читатель в качестве упражнения может попытаться найти подобные оптимальные соотношения радиуса `r` к высоте `h` и для других геометрических тел: для цилиндра с круглым плоским дном, но без круглой плоской крышки; для цилиндра с круглым плоским дном и с круглой плоской крышкой; для конуса, но не с круглой плоской крышкой (рис. 12.4–12.6), а с крышкой в виде полушария и т. д. В подобных, но уже полых телах (оболочках) можно ставить горизонтальные или вертикальные перегородки, увеличивающие суммарную площадь поверхности такого тела и меняющие оптимальное отношение `r` к `h`. Все эти задачи несложно решить «инженерно» (численно) для частных случаев с заданным объемом, но только некоторые из них можно будет решить «математически» (символьно) для общего случая без фиксации объема. Успехов! Если же особых удач не будет, то можно обратиться к сайту **https://www. ptcusercommunity.com/community/ mathcad**, где эти

задачи решены с помощью не только пакета Mathcad, но и с помощью пакетов Maple и Mathematica.

Задача 2. Имеется коническая воронка, в которую вкладывают фильтр, свернутый из круглой фильтровальной бумаги. Многие из нас работали с такими фильтрами на уроках химии в школе и вузе. Спрашивается, каким должен быть угол «раструба» конуса воронки (β — см. рисунок 12.7), при котором скорость фильтрования была бы максимальна[2]?

Решение этой задачи оптимизации показано на рисунке 12.7.

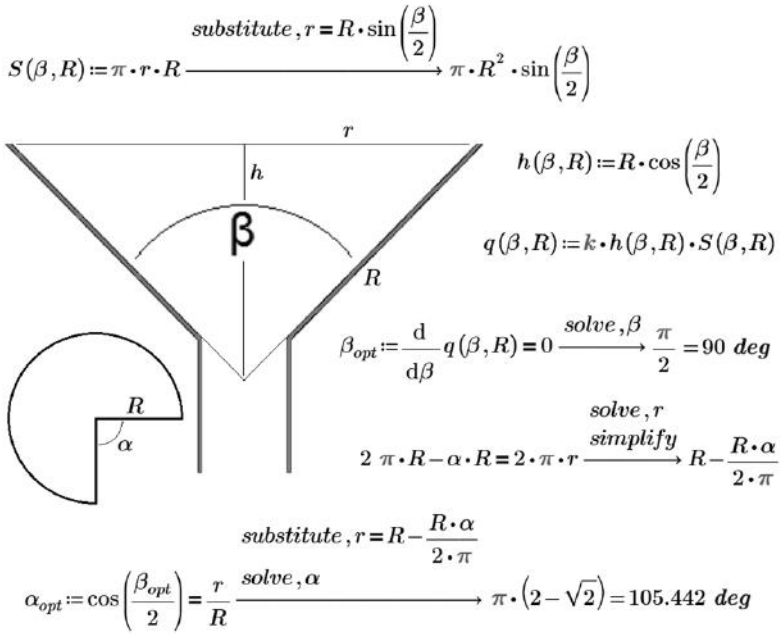


Рис. 12.7
Решение задачи об оптимальном бумажном коническом фильтре

В решении принято, что скорость фильтрования (расход жидкости q) пропорциональна произведению высоты столба жидкости h на площадь фильтрования S. «Компромисс» между высотой столба жидкости и площадью фильтрования равен 90° (прямой угол). С таким углом раструба, кстати, сделаны почти все конические воронки для перелива жидкостей. Читатель может при желании проверить это.

В нашей задаче о фильтре в конической воронке параметр R — это радиус круглой бумажной заготовки, из которой загибом сектора изготавливают конический колпачок-фильтр с радиусом r в основании и с высотой h.

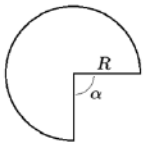
Несложно подсчитать (а это сделано двумя последними операторами, показанными на рисунке 12.7) угол сектора α, который нужно вырезать из круглой заготовки, чтобы из оставшегося сектора свернуть конус с раструбом в 90 градусов. Этот угол (α_{opt}) равен примерно 105 с половиной угловых градусов (deg).

А какой сектор нужно вырезать из круглой заготовки с радиусом R, чтобы получить коническую емкость (наше пожарное ведро, например) максимального объема?

Решение этой **третьей** задачи оптимизации конуса показано на рис. 12.8. Мы сразу отказались от «инженерного» подхода и воспользовались математическим.

[2] Батунер Л. М., Позин М. Е. Математические методы в химической технике. — Л.: Химия, 1960.

$$2\cdot\pi\cdot R-\alpha\cdot R=2\cdot\pi\cdot r \xrightarrow[\text{simplify}]{\text{solve},r} R-\frac{R\cdot\alpha}{2\cdot\pi}$$



$$r(\alpha,R):=R-\frac{R\cdot\alpha}{2\cdot\pi} \qquad h(\alpha,R):=\sqrt{R^2-r(\alpha,R)^2}$$

$$V(\alpha,R):=\frac{1}{3}\cdot\pi\cdot r(\alpha,R)^2\cdot h(\alpha,R)$$

$$\frac{d}{d\alpha}V(\alpha,R)\xrightarrow{\text{simplify}}-\frac{R^2\cdot\sqrt{-\left(R^2\cdot\alpha\cdot(\alpha-4\cdot\pi)\right)}\cdot\left(18\cdot\pi\cdot\alpha^2-3\cdot\alpha^3-28\cdot\pi^2\cdot\alpha+8\cdot\pi^3\right)}{24\cdot\pi^2\cdot\alpha\cdot(\alpha-4\cdot\pi)}$$

$$\frac{R^2\cdot\sqrt{-\left(R^2\cdot\alpha\cdot(\alpha-4\cdot\pi)\right)}\cdot\left(18\cdot\pi\cdot\alpha^2-3\cdot\alpha^3-28\cdot\pi^2\cdot\alpha+8\cdot\pi^3\right)}{24\cdot\pi^2\cdot\alpha\cdot(\alpha-4\cdot\pi)}\xrightarrow{\text{solve},\alpha}\begin{bmatrix}2\cdot\pi\\ \frac{2\cdot\pi\cdot\left(\sqrt{6}+3\right)}{3}\\ -\frac{2\cdot\pi\cdot\left(\sqrt{6}-3\right)}{3}\end{bmatrix}$$

$$a_{opt}:=\frac{2\cdot\pi\cdot\left(3-\sqrt{6}\right)}{3}=66.061\ deg$$

$$\beta_{opt}:=2\cdot\operatorname{asin}\left(\frac{r\left(a_{opt},R\right)}{R}\right)\xrightarrow{\text{simplify}}2\cdot\operatorname{asin}\left(\frac{\sqrt{6}}{3}\right)=109.471\ deg$$

Рис. 12.8

Максимальный объем конуса, свернутого из круглой заготовки

При решении задачи о максимальном объеме конуса, свернутого из круглой заготовки, создаются три функции пользователя `r(α, R)`, `h(α, R)` и `V(α, R)` (целевая функция оптимизации). У последней функции берется производная по α. Один из нулей производной будет решением нашей задачи: из круглой заготовки нужно вырезать сектор с примерно 66 градусами в вершине, остаток (больший сектор) свернуть в конус[3] с 109,5 угловыми градусами в «раструбе». Из вырезанного же сектора также можно свернуть еще один конус. Спрашивается (новая задача для читателя), каков будет оптимальный раскрой круглой заготовки на два сектора, из которых можно свернуть два конуса с максимальным суммарным объемом. А если круглую заготовку разрезать не на два, а на три и большее количество секторов — заготовок для конусов? Превысит ли суммарный объем этих конусов суммарный объем двух конусов? Предлагаем читателям провести самим эти расчеты численными и символьными методами.

Вот мы, сворачивая конусы, и подошли к нашему пожарному ведру.

Пожарное ведро, повторяем, делается в виде конуса для того, чтобы его нельзя было поставить на пол и использовать не по прямому назначению (для стирки, например) — такое ведро свалится на бок. Такую же примерно форму (рюмка с отбитой ножкой) имеет бокал «Пей до дна!»

Решая задачу о коническом пожарном ведре (см. рис. 12.8), мы под оптимизацией понимали *максимизацию* объема ведра при выбранной технологии его изготовления — вырезании сектора из круглой заготовки с последующим сворачиванием этого сектора в конус. Но при этом ведро теряло не только свою *оптимальность*, но и свое название «ведро»: оно превращалось во что-то похожее на... тазик (пожарный тазик?) или перевернутую вьетнамскую шляпу. Мы гнались за *числом* (объем ведра) и потеряли *суть* (функцию, ипостась ведра) — емкость для удобной переноски жидкости.

Тут вспоминается мультфильм по сказке Альфа Прейсена о козленке (см. **http://www.skazochki.narod.ru/skazki/kozel.html**), который научился *считать* и всех встречных-поперечных пересчитывал: «Я — это раз, теленок — это два, ко-

[3] Когда-то давно в наших магазинах не было готовых бумажных или пластиковых пакетов, и продавцы очень ловко сворачивали из листа бумаги конус-кулек, в который насыпали сахар или крупу, клали макароны и другие продукты.

рова — это три, бык — это четыре...» У козленка из-за этого со всей пересчитанной живностью возникали крупные неприятности: «Ах! Ты меня сосчитал?! Ну, держись!» Но все кончилось хорошо. На то она и сказка.

В этой истории, как и в любой другой запоминающейся сказке, есть глубокий смысл. Стоит нам что-то пересчитать[4], как мы вступаем с этой пересчитанной субстанцией в некий конфликт. Природа не любит не только острых углов, но и *счета*, который в ряде случаев просто убивает ее. Это можно наблюдать не только в биологии и физике, где инструменты познания часто неузнаваемо портят или даже убивают сам объект исследования, но и в *computer science*. И не только в области приложения компьютеров (*счёта*, грубо говоря) к решению естественнонаучных задач, но и в области применения компьютеров к самим компьютерам.

Врачу для установления диагноза незачем знать точное *числовое* значение температуры тела больного — 36.6, 38,9 и т. д. Достаточно выразить показания термометра диапазонами (отдельными *множествами* температур), о которых медики договорились заранее — «температура пониженная», «температура нормальная», «температура повышенная», «температура высокая», «температура очень высокая (жар)». Границы этих оценок (множеств температур) хоть и интуитивно понятны, но, тем не менее, размыты — «пушисты». Это определяется не только современными представлениями (теории нечетких («пушистых») множеств, *ТНМ*), но и погрешностью самого термометра, методикой измерения температуры и др. Выпускник медицинского вуза, не задумываясь, скажет вам, где лежит граница между высокой и очень высокой температурой. Опытный же врач может этого и не знать, хотя диагнозы он выставляет не хуже начинающего доктора. Опытный врач знает, что границы не только размыты, но и контекстнозависимы. Даже параметры больного, выраженные не в вещественном, а в булевом виде (реакция Вассермана, наличие палочки Коха, анализ на СПИД и т. д.), имеют также «пушистые» границы. Об этом хорошо знают лаборанты, проводящие анализы. Если заглянуть в любой справочник терапевта, где описаны симптомы болезней, то, как правило, конкретных чисел (температура тела, артериальное давление, содержание гемоглобина в крови и т. д.) там не увидишь. Одни слова — «повышено», «понижено» и т. д. Программы выставления диагноза по введенным в компьютер параметрам больного не получили широкого практического применения. Одна из трудностей в этом деле — перевод параметра (числа) в симптом (в категорию).

Принято выделять три революции, переводившие программирование на новые уровни: *структурное*, *объектно-ориентированное* и *визуальное*. Но эта революционность была больше обращена на *программы* (искусство ради искусства) и почти не касалась *объекта* программирования — тех моделей реального мира, свойства и события которого программно имитируются. Более того, ретроспективный взгляд может уловить и некую контрреволюционность в отказе, например, от аналоговых вычислительных машин и переход к цифровой технике. Хотя в последнее время здесь наблюдается какое-то подобие ренессанса — возрождение принципов аналогового моделирования на современных цифровых компьютерах. Это можно упомянуть и в технологии визуального программирования, где воссоздаются прежние элементы управления (Controls), а также некие аналоги сумматоров, интеграторов и др. Но виртуальность этих неоаналоговых машин подразумевает и их строгую детерминированность (четкость), что влечет за собой не только положительные, но и отрицательные последствия. Сами же принципы ТНМ давно уже у программистов под рукой. В буквальном смысле. Мышь компьютера реагирует на два события — щелчок (click) и двойной щелчок (doubleclick). А чем, собственно, один двойной щелчок отличается от двух одинарных? Длительностью паузы между щелчками. Если ее выразить на языке человеческого общения (очень короткая, короткая, длинная и т. д.), то это будет типичным примером множеств с «пуши-

[4] А в век цифровых компьютеров мы это делаем все чаще и чаще.

стыми» границами. На сколько нужно увеличить короткую паузу, чтобы она превратилась в длинную и чтобы двойной щелчок воспринимался компьютером как два одинарных? А сколько зернышек нужно добавить в горсть, чтобы она превратилась в кучу? И на сколько сантиметров самая низкая гора выше самого высокого холма?

В разгар структурной революции, когда во всех программистских «храмах» предавали анафеме ключевое слово `GOTO`, часто можно было услышать такую фразу: «Практически невозможно научить хорошо программировать студентов, ориентированных первоначально на BASIC: как потенциальные программисты они умственно оболванены без надежды на исцеление». Встречались и более категоричные предупреждения типа: «Осторожно! Занятие программированием может лишить вас будущего. Не думайте, что, научившись программировать, вы чего-то добьетесь в жизни». Традиционное программирование как бы заставляет программиста смотреть на многоцветный мир сквозь черно-белые очки: булева переменная может принимать только два значения (да/нет), а вещественная — строго определенное в оговоренном диапазоне с фиксированной длиной мантиссы и т. д. Истина, как всегда, лежит посередине. Но и крайние точки зрения не бесполезны — они как бы подпирают истину с двух сторон, не давая ей скатиться к крайностям. Да и сами истины в чем-то «пушисты». А одна из них может звучать так: «Если хочешь познать мир (нечеткий, «пушистый», бесчисловой) и управлять им, то опасайся традиционных языков программирования и математических программ с их строгой детерминированностью».

Но вернемся к нашей задаче о пожарном ведре и попытаемся решить ее уже с привлечением аппарата ТНМ.

Проведем своеобразный опрос общественного мнения и узнаем как можно больше о параметрах оптимального пожарного ведра: о его удобной геометрии (о радиусе основания конуса и высоте) и о его оптимальном объеме (о весе ведра с водой). Вот здесь-то и проявятся во всей своей красе положения ТНМ. Сколько воды нужно добавить в ведро, чтобы оно из легкого превратилось в тяжелое? На сколько нужно увеличить или уменьшить радиус или высоту ведра, чтобы при переноске в нем воды оно перестало быть удобным? Вот эти «*сколько*» и являются типичными представителями нечетных множеств. В среде Mathcad, как и в других популярных пакетах, нет типов переменных для хранения таких величин, но мы, тем не менее, постараемся решить поставленную задачу.

Представления людей об оптимальном (удобном) радиусе основания конуса пожарного ведра можно получить так: изготовить много ведер различной геометрии, дать людям поносить в них воду и оценить по такой шкале:

- удобное (1);
- скорее удобное, чем неудобное (0.67);
- затрудняюсь ответить (0,5);
- скорее неудобное, чем удобное (0.34);
- неудобное (0).

Можно принимать во внимание и другие оценки в диапазоне от нуля до единицы.

На рисунке 12.9 показано начало Mathcad-документа с решением задачи об оптимальном пожарном ведре с привлечением элементов теории нечетких множеств. Если в теории четких множеств некий элемент может (значение 1) или не может (значение 0) принадлежать некоему четкому множеству (см. этюд 11), то в теории нечетких множеств этот элемент может принадлежать некому нечеткому множеству с вероятностью от нуля до единицы. Этот параметр задается *функцией принадлежности* — центральным понятием теории нечетких множеств.

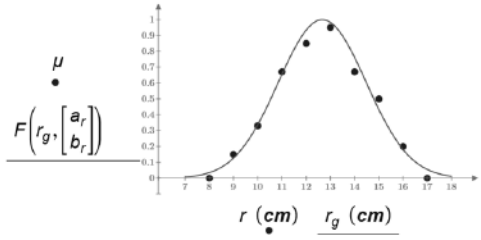
$F(x,a) := \left\| \begin{array}{l} \text{"Функция нормального распределения"} \\ e^{a_0 \cdot (x - a_1)^2} \end{array} \right.$

1. Представления об оптимальном радиусе ведра

r (cm)	μ
8	0
9	0.15
10	0.33
11	0.67
12	0.85
13	0.95
14	0.67
15	0.5
16	0.2
17	0

$\begin{bmatrix} a_r \\ b_r \end{bmatrix} := \text{genfit}\left(r, \mu, \begin{bmatrix} -0.173\ cm^{-2} \\ 13\ cm \end{bmatrix}, F\right) = \begin{bmatrix} -1534.0016\ \frac{1}{m^2} \\ 0.1266\ m \end{bmatrix}$

$r_g := 7\ cm, 7.1\ cm .. 18\ cm$



Функция принадлежности по радиусу ведра $\mu_r(r) := F\left(r, \begin{bmatrix} a_r \\ b_r \end{bmatrix}\right)$

Рис. 12.9

Задача об оптимальном пожарном ведре: начало расчета

Первым оператором в расчете на рисунке 12.9 вводится функция `F` с аргументом `x` и параметром `a`. Аргумент `x` — это скаляр, а параметр `a` — вектор с двумя элементами a_0 и a_1. Такое не совсем обычное определение функции связано с тем, что далее в расчете она будет вызываться в качестве аргумента встроенной функции `genfit`, особенности которой будут раскрыты ниже.

Функция `F` — это упрощенный вариант функции нормального распределения, стандартный вид которой показан на рисунке 12.10. Мы подобную функцию уже использовали в этюде 9 для прокладки рельсов, по которым поезд или трамвай будет поворачивать без толчков — см. рис. 9.35 на стр. 234.

В формуле на рисунке 12.10 переменная μ — это математическое ожидание (среднее значение), а переменная σ — среднеквадратическое отклонение.

В решении на рисунке 12.9 мы, повторяем, несколько упростили формулу нормального распределения — сделали так, чтобы значение μ всегда оставалось в пределах от нуля до единицы.

В расчете на рисунке 12.9 представления людей об оптимальном радиусе пожарного ведра заносятся в таблицу с двумя столбцами — в два вектора с именами `r` и μ. Мы ограничились десятью точками (оценками людей радиуса ведра), но их может быть намного больше: сколько людей — столько и мнений. Читатель при желании может опросить своих друзей и дополнить векторы `r` и μ новыми парами чисел. Данные опроса обрабатываются методом наименьших квадратов, когда в качестве аппроксимирующей кривой взята кривая нормального распределения. Это делается с помощью встроенной в Mathcad функции `genfit`, предназначенной для сглаживания («подгонки» — **fit**ting) точек пользовательской функцией общего (**gen**eral) вида. В нашем случае эта функция `F` — упрощенная функция нормального распределения с двумя аргументами. Но у сглаживающей функции может быть любое число аргументов. Поэтому-то эти аргументы хранятся в виде вектора с переменным числом элементов.

$$f(x,\sigma,\mu) := \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$x := -5, -4.99 .. 5$

$f(x, \sigma \leftarrow 1, \mu \leftarrow 2)$

$f(x, \sigma \leftarrow 2, \mu \leftarrow 0)$

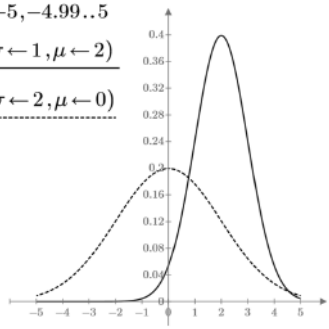


Рис. 12.10
Функция нормального распределения

Функция же genfit имеет четыре аргумента: два вектора с координатами исходных отрабатываемых точек (r и μ), вектор первых приближений и имя сглаживающей функции (F). Тут очень важно задать правильные значения первого приближения с правильными единицами измерения. Иначе функция genfit будет возвращать не ответ (вектор значений коэффициентов сглаживающей функции), а сообщение об ошибке. Правильные единицы измерения определить несложно — достаточно проанализировать поведение сглаживающей функции. Задать значение первого приближения для коэффициента a_1 также не представляет особого труда — это значение элемента вектора r при максимальном значении вектора μ. Труднее задать значение коэффициента a_0. Тут можно либо вводить случайные числа и смотреть, каким получается график, либо, не полагаясь на случай, поступить так, как показано на рисунке 12.11: определить значение коэффициента a_0 для случая, когда сглаживающая кривая проходила бы через одну из исходных точек — через точку r = 15 cm и μ = 0,5, например. На рисунке 12.11 создается вспомогательная функция с именем f, с аргументом x и параметром a. У этой функции с помощью встроенной функции root вблизи точки начального предположения ($1/cm^2$) ищется нуль, который будет начальным предположением для функции genfit. Без такого несколько хитрого приема задать правильное значение начального предположения для коэффициента a_0 будет довольно сложно.

$$f(x,a) := e^{a \cdot (x - 13\,cm)^2}$$

$$a := \frac{1}{cm^2} \qquad \mathrm{root}(f(15\,cm, a) - 0.5, a) = -0.173\,\frac{1}{cm^2}$$

Рис. 12.11
Определение начального приближения для коэффициента a

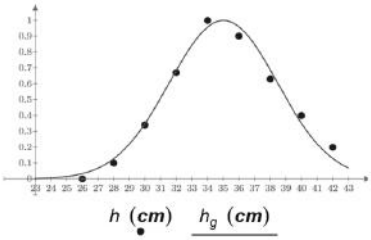
По функции F и найденным оптимальным коэффициентам a_r и b_r создается функция $\mu_r(r)$ — функция принадлежности конкретного значения радиуса основания пожарного ведра нечеткому множеству «оптимальный радиус основания пожарного ведра» — см. рисунок 12.9.

2. Представления об оптимальной высоте ведра

h (cm)	μ
26	0
28	0.1
30	0.34
32	0.67
34	1
36	0.9
38	0.63
40	0.4
42	0.2
44	0

$$\begin{bmatrix} a_h \\ b_h \end{bmatrix} := \text{genfit}\left(h, \mu, \begin{bmatrix} -0.1\ cm^{-2} \\ 35\ cm \end{bmatrix}, F\right) = \begin{bmatrix} -419.8702\ \frac{1}{m^2} \\ 0.3501\ m \end{bmatrix}$$

$$h_g := 23\ cm, 23.1\ cm .. 43\ cm$$



Функция принадлежности по высоте ведра $\mu_h(h) := F\left(h, \begin{bmatrix} a_h \\ b_h \end{bmatrix}\right)$

Рис. 12.12

Задача об оптимальном пожарном ведре: продолжение расчета

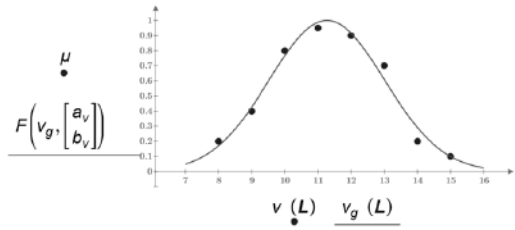
Такие же «опросы общественного мнения» с их последующей статистической обработкой проведены и в отношении высоты пожарного ведра (рисунок 12.12) и его объема (рисунок 12.13).

3. Представления об оптимальном объеме воды в ведре

v (L)	μ
8	0.2
9	0.4
10	0.8
11	0.95
12	0.9
13	0.7
14	0.2
15	0.1

$$\begin{bmatrix} a_v \\ b_v \end{bmatrix} := \text{genfit}\left(v, \mu, \begin{bmatrix} -0.01\ L^{-2} \\ 10\ L \end{bmatrix}, F\right) = \begin{bmatrix} -165.7121 \cdot 10^3\ \frac{1}{m^6} \\ (11.2689 \cdot 10^{-3})\ m^3 \end{bmatrix}$$

$$v_g := 7\ L, 7.1\ L .. 16\ L$$



Функция принадлежности по объему воды в ведра $\mu_v(v) := F\left(v, \begin{bmatrix} a_v \\ b_v \end{bmatrix}\right)$

Рис. 12.13

Задача об оптимальном пожарном ведре: продолжение расчета

Наше пожарное ведро должно быть оптимальным **И** по радиусу основания, **И** по высоте, **И** по объему. В математике нечетких множеств операцию логического умножения (And, И) выполняет команда определения *минимума*, и мы уже писали об этом в этюде 11 (см. рис. 11.13 на стр. 234). На рисунке 12.14 (пункт 4) формируется функция принадлежности $\mu_{rh}(r, h)$ конкретного ведра с параметрами r и h нечеткому множеству «оптимальное пожарное ведро». Для этого в расчете задействована встроенная в Mathcad функция min, имеющая в качестве аргумента вектор с тремя элементами: заданными ранее функциями μ_r, μ_h и μ_v (см. рисунки 12.9,

12.12 и 12.13). Сама же функция μ_{rh}(r, h) будет иметь только два аргумента, т. к. объем ведра зависит от его размеров.

4. Генерация функции принадлежности по радиусу и высоте ведра
Оптимальность по радиусу И высоте И объему воды

$$\mu_{rh}(r,h) := min\left(\begin{bmatrix} \mu_r(r) \\ \mu_h(h) \\ \mu_v\left(\frac{1}{3}\cdot\pi\cdot r^2\ h\right) \end{bmatrix}\right)$$

5. Поиск параметров самого оптимального ведра

$$r := 15\ cm \qquad h := 40\ cm \qquad \begin{bmatrix} r \\ h \end{bmatrix} := \mathbf{Maximize}\,(\mu_{rh}, r, h) = \begin{bmatrix} 14.89 \\ 39.27 \end{bmatrix} cm$$

$$v := \frac{1}{3}\cdot\pi\cdot r^2\cdot h = 9.12\ L$$

$$\mu_r(r) = 0.46622857$$
$$\mu_h(h) = 0.46628344$$
$$\mu_v(v) = 0.4662453 \qquad \mu_{rh}(r,h) = 0.4662286$$

Рис. 12.14
Определение параметров оптимального пожарного ведра

Нам нужно как-то «сложить», вернее, подытожить мнения людей об оптимальных параметрах пожарного ведра, зафиксированных на рисунках 12.9, 12.12 и 12.13. В ТНМ нет понятий сложения, вычитания, умножения и т. д., лежащих в основе традиционной «четкой» математики и реализованных в среде Mathcad инфиксными операторами +, –, и т. д. В ТНМ *умножение* (*пересечение* множеств, And), повторяем, заменено на операцию поиска *минимума* (min), а *сложение* (*слияние* множеств, Or) — на поиск *максимума* (max). Математика четких множеств является частным случаем математики нечетких множеств — в программах вместо функции (оператора) And можно использовать функцию поиска минимума, а вместо функции Or — функцию поиска максимума, и мы это, повторяем, уже делали в этюде 11. В нашей задаче функция принадлежности μ_{rh} получается путем *нечеткого умножения* (min) функций μ_r, μ_h и μ_v — нечеткое множество «удобное ведро» лежит на *пересечении* трех других нечетких множеств: «удобный радиус ведра» (рисунок 12.9), «удобная высота ведра» (рисунок 12.12) и «удобный объем ведра» (рисунок 12.13). Функцию же μ_{rh} удалось сделать двухаргументной с аргументами r и h, а не трехаргументной (аргументы r, h и v) за счет, повторяем, ввода в расчет формулы объема конуса, связывающего между собой эти три аргумента.

В пункте 5 на рисунке 12.14 с помощью встроенной в Mathcad функции Maximize ведется поиск параметров оптимального пожарного ведра.

На рисунках 12.15 и 12.16 показаны линии уровня и поверхность двухаргументной (с аргументами r и h) функции принадлежности конкретного пожарного ведра нечеткому множеству «оптимальное пожарное ведро».

На рисунке 12.16 можно увидеть... пожарное ведро, помятое вследствие использования его не по прямому назначению — не для тушения пожара, а для... мытья полов со вставкой ведра в перевернутую табуретку (см. выше).

В начале текста был упомянут пожарный щит на дачном участке одного из авторов. Так вот, этот автор после выкладок, изложенных выше, решил измерить параметры реального пожарного ведра. Но когда он подошел с линейкой к этому щиту, то оказалось, что никакого пожарного инвентаря на щите нет, а все это (ломик, лопата, топор и само пожарное ведро) просто нарисовано красной краской на зеленом пожарном щите. Автору осталось только ухмыльнуться, вспомнить нарисованный очаг в каморке папы Карло и ни с чем вернуться назад. Но запрос в поисковой системе Интернета привел автора к одному магазину пожарного инвентаря, где продаются конические пожарные ведра с габаритами 280 на 380 мм, объе-

мом 8 литров и массой 1.1 кг по цене 110 руб. за штуку. Это (кроме цены) оказалось довольно близко к тому, что мы получили нашими «нечеткими» вычислениями.

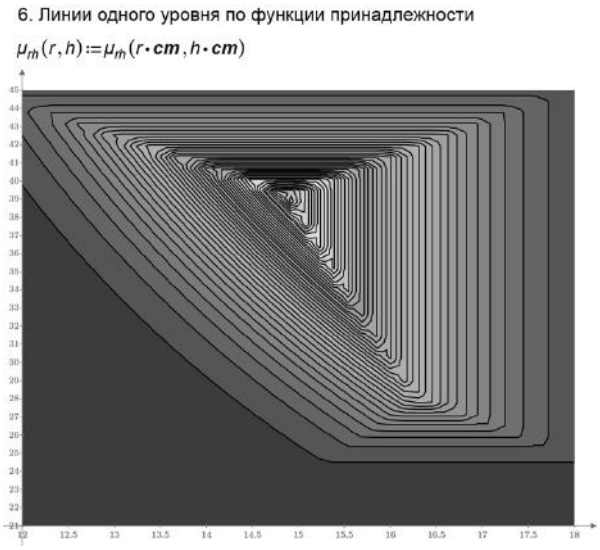


Рис. 12.15
Линии уровня функции принадлежности «оптимальное пожарное ведро»

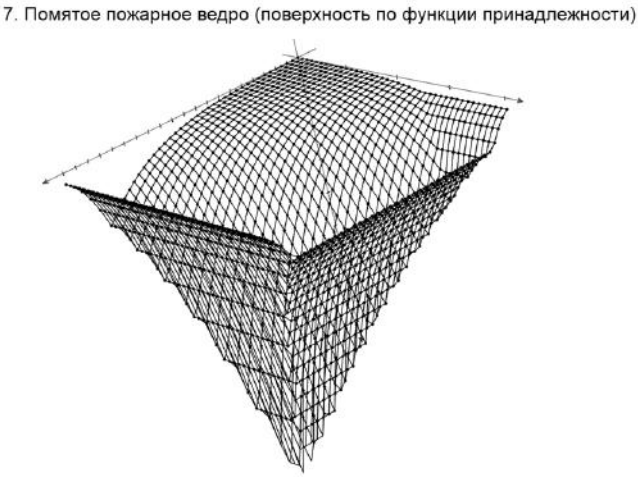


Рис. 12.16
Поверхность функции принадлежности «оптимальное пожарное ведро»

На сайте этого магазина приведены наиболее распространенные ответы на вопрос, почему пожарное ведро имеет форму конуса.

1. С производственной точки зрения, такая форма позволяет не припаивать днище, следовательно, производственный процесс заметно упрощается, а ведро становится дешевле.

2. С бытовой точки зрения (самое распространенное мнение) такое ведро не украдут с пожарного щита, для бытового использования оно непригодно[5].

3. Самое основное мнение. С точки зрения пожарной науки такая форма ведра позволяет быстро справиться с возгоранием: во-первых, конусовидная форма ведра позволяет в зимнее время пробивать лунки в пожарных водоемах; во-вторых, особая форма пожарного ведра позволяет избежать расплескивания воды при тушении. Из обычного ведра вода выливается не равномерно, а из пожарного ведра вода выливается целенаправленной струей.

Но вернемся к основной теме.

Даже не очень внимательный читатель заметит неточности, допущенные при решении задачи об удобном пожарном ведре. Вот три из них:

- ведро никогда не наполняется до краев;
- авторы уж очень вольно обращаются с такими понятиями, как объем, вес и масса ведра, путая их[6];
- не учтен вес пустого ведра, а также материал, из которого оно сделано.

Однако стоит еще раз мельком взглянуть на графики, иллюстрирующие нечеткие множества на рисунках 12.9, 12.12 и 12.3, чтобы понять важнейшую особенность решения задач с привлечением аппарата ТНМ. Наше решение вычленяет, если так можно выразиться, *суть* задачи, оставляя без внимания различные *мелочи*: плотность воды, вес пустого ведра, степень его наполнения и др.

Эта особенность в настоящее время реализована, например, в системах автоматического регулирования, где регуляторы, настроенные с учетом положений ТНМ, более «внимательны» к основному сигналу и менее восприимчивы к шуму. Ведь, для управления работой, например, электростанции нам не нужно знать конкретное значение давление пара в котле, например. Нам нужно знать, не повышенное ли или не пониженное ли оно, и исходя из этого принимать решения. А понятия «пониженное давление» или «повышенное давление» (как, впрочем, и понятия «пониженная температура» или «повышенная температура» — см. выше) — это типичные представители нечетких множеств. Оказалось, хотя это и кажется парадоксальным, что традиционные «четкие» алгоритмы управления качественно проигрывают «нечетким» либо являются их частными случаями. В теории автоматического регулирования наблюдался некий застой, т. к. никакие новые алгоритмы не могли сравниться со старым добрым пропорционально-интегрально-дифференциальным (ПИД) алгоритмом (законом) управления. Принципы ПИД-регулирования можно узреть, например, в процедуре принятия решения о выдаче кредита клиенту банка, когда принимающий решение банкир учитывает, во-первых, количество денег на текущем счете просящего (*пропорциональная* составляющая — чем богаче клиент, тем больше денег ему можно дать в долг), во-вторых, динамику изменения текущего счета клиента (*дифференциальная* составляющая — дела клиента на подъеме или в упадке) и, в-третьих, среднее количество денег у клиента за последние, к примеру, пять лет (*интегральная* составляющая — не занял ли клиент вчера денег на стороне, чтобы создать видимость своего благополучия). Можно учитывать и другие составляющие, но... три — красивое число. Кроме того, в конце концов, решения о выдаче того же кредита принимается чаще всего «по наитию», с учетом недостатка даже размытой, «пушистой» информации. О банкирах, хороших и плохих, можно прочесть также в этюде 15.

ПИД-алгоритм регулирования как-то незаметно был фетишизирован. Идеи нечеткого управления — это свежая струя в теории автоматического регулирова-

[5] Можно подумать, что это связанно с особенностями российского менталитета, но в Интернете можно найти картинки иностранных пожарных ведер с надписью «fire», у которых дно тоже не плоское, а, например, сферическое с припаянной к нему дополнительной ручкой для удобства выливания воды.

[6] Да и с такими понятиями, как «нечеткое множество», «функция принадлежности» авторы этой книги также обращаются слишком «нечетко», и это отметили некоторые читатели статьи [64].

ния, основные положения которой в настоящее время подвергаются ревизии. Правда, есть и другое мнение. Некоторые ученые полагают, что использование аппарата ТНМ в теории автоматического регулирования и в кибернетике вообще — это попытки замены одной *неопределенности* на другую (шило на мыло, грубо говоря). Наблюдающиеся эффекты повышения качества управления скептики объясняют тем, что на регуляторы лишний раз обратили внимание (принцип доброго слова, которое и кошке приятно). Кроме того, некоторые исследователи полагают, что ТНМ (ей всего лишь 40 лет, а открыл ее миру Л. А. Заде — американец иранского происхождения) — это хорошо забытое старое. По традиции, четкие множества принято иллюстрировать кругами с резко оконтуренными границами. Нечеткие же множества — это круги, образованные отдельными точками: в центре круга точек много, а ближе к периферии их густота уменьшается до нуля; круг как бы растушевывается (становится «пушистым») на краях. Такие «нечеткие множества» можно увидеть... в тире — на стене, куда вывешиваются мишени. Следы от пуль образуют *случайные* множества, математика которых известна. Оказалось, что для оперирования нечеткими множествами годится уже давно разработанный аппарат случайных множеств...

Мы говорим *нечеткое множество*. А множество *чего*? Если быть последовательным, то приходится констатировать, что элементом нечеткого множества оказывается... новое нечеткое множество новых нечетких множеств и т. д. Вернемся к классическому примеру — к *куче зерна*. Элементом этого нечеткого множества будет *миллион зерен*, например. Но миллион зерен — это никакой не четкий *элемент*, а новое *нечеткое множество*. Ведь считая зерна (вручную или автоматически), не мудрено и ошибиться — принять за миллион 999 997 зерен, например. Тут можно сказать, что элемент 999 997 имеет значение функции принадлежности к множеству «миллион», равное 0,999997. Кроме того, само зерно — это опять же не элемент, а новое нечеткое множество: есть полноценное зерно, а есть два сросшихся зерна, недоразвитое зерно или просто шелуха. Считая зерна, человек должен какие-то отбраковывать, принимать два зерна за одно, а в другом случае одно зерно за два. Нечеткое множество не так-то просто запихнуть в цифровой компьютер с классическими языками: элементами массива (вектора) должны быть новые массивы массивов (вложенные векторы и матрицы, если говорить о Mathcad). Классическая математика четких множеств (теория чисел, арифметика и т. д.) — это крюк, с помощью которого *человек разумный* фиксирует (детерминирует) себя в скользком и нечетком окружающем мире. А крюк, как известно, — инструмент довольно грубый, нередко портящий то, за что им цепляются. Термины, отображающие нечеткие множества (а их достаточно и в этой книге — «много», «слегка», «чуть-чуть» и т. д. и т. п.), трудно «запихнуть» в компьютер еще и потому, что они *контекстно зависимы*. Одно дело сказать «Дай мне немного семечек (зерна)» человеку, у которого семечки в кармане, а другое дело — человеку, сидящему за рулем грузовика с семечками.

Можно ли усмотреть некий *кризис* в теории и практике программирования, связанный с противоречием между четкой структурой программ (данных) и нечетким миром? Следует ли разрабатывать «нечеткие» языки программирования для реализации «нечетких» алгоритмов и для размещения «нечетких» данных? Мнения здесь разные. Программисты (а за ними последнее слово) худо-бедно научились «запихивать» нечеткий мир в строго детерминированный компьютер.

Наш подход к выбору оптимального пожарного ведра можно применить и к выбору оптимальной провисающей цепи, о которой было рассказано в этюдах 4 и 7. В конце этюда 4 были определены параметры цепи с минимальными силами разрыва у точек крепления, расположенных на одном уровне. Было показано, что у такой цепи отношение ее длины к расстоянию между точками крепления должно быть равно 1.258 (см. такую цепь на рис. 4.49 и 4.51 на стр. 101 и 103). Но будет ли такая цепь красивая? Ведь цепи часто используются чисто в декоративных целях — ими, к примеру, огораживают различные памятники. Можно на улице по-

весить цепи различной формы (с различным значением отношения длины цепи к расстоянию между точками крепления) и попросить прохожих (экспертов) дать свои оценки «красивости» по шкале, какое мы применяли для пожарного ведра: цепь некрасивая; скорее цепь некрасивая, чем красивая; не могу определить; скорее цепь красивая и цепь красивая. Можно ожидать, что красивой окажется цепь с отношением ее длины к расстоянию между точками крепления, равным 1.258. Ведь часто оказывается так, что крепко построенный мост оказывается и красивым. И наоборот!

ДИВЕРТИСМЕНТ. ПОЛНОСТЬЮ РАЗМЕРНАЯ ФУНКЦИЯ

При решении задачи о пожарном ведре мы использовали единицы длины и объема (вместимости). Поговорим об этом более детально.

На рисунке 12.17 изображена операция создания и вызова в среде Mathcad функции с именем V_c объема не конуса, а цилиндра.

$$V_c(d,h) := \frac{\pi \cdot d^2}{4} \cdot h$$

$$V_c(20mm, 15mm) = 4.712\,cm^3$$

Рис. 12.17

Традиционное создание и «размерный» вызов функции пользователя

Из рисунка 12.17 видно, что если переменные d и h имеют размерность длины, то функция безаварийно возвращает значение объема цилиндра с размерностью длины в кубе. Но эта же функция безаварийно принимает значение своих аргументов с единицами любых других размерностей и безразмерными единицами (рис. 12.18).

$$V_c(d,h) := \frac{\pi \cdot d^2}{4} \cdot h$$

$$d := 20 \qquad h := 15kg$$

$$V_c(d,h) = 4712.4\,kg$$

Рис. 12.18

Вызов функции объема цилиндра с различными размерностями аргументов

И неважно, что функция в ситуации, показанной на рис. 12.18, выдает значение с размерностью, не равной длине в кубе (следствие!). Важна причина такой ненормальной работы. Чтобы разобраться, как выйти из этого положения, вспомним замечательный принцип, сформулированный Фурье в его классической работе «Аналитическая теория теплоты», которая вышла в свет в 1822 г. Этот принцип называется теперь «принципом размерной однородности» и утверждает, что любое физическое уравнение (которое, как правило, представляет собой алгебраическую сумму двух или более членов) должно содержать только члены, размерность которых одинакова. Чтобы пояснить, в чем разница между обычными алгебраическими и «физическими» уравнениями, обратимся к примеру, приведенному в основополагающей работе Бриджмена[7]. Анализируя задачу о падении тела под действием силы тяжести, Бриджмен замечает, что справедливы, по меньшей мере, два равенства, связывающие пройденное расстояние (s), скорость (v), время (t) и ускорение свободного падения (g):

$$v = g \cdot t, \quad s = g \cdot t^2 / 2.$$

[7] Бриджмен П. Анализ размерностей. — Ижевск : НИЦ «Регулярная и хаотическая динамика», 2001.

Если теперь чисто алгебраически сложить оба равенства, получится уравнение

$$v - s = g \cdot t - g \cdot t^2 / 2,$$

которое выглядит правдоподобным, но неверно с точки зрения размерностей — скорость нельзя складывать с расстоянием! Имея в распоряжении инструмент контроля размерностей, внедренный в Mathcad, мы можем проконтролировать размерность вводимой величины путем добавления «размерного нуля», а именно: прибавлением и вычитанием некоторой величины заданной размерности. На рисунке 12.19 показано, как можно создать функцию пользователя, принимающую в качестве аргументов величины с заданной пользователем же размерностью.

Из рисунка 12.19 видно, что вновь созданная функция, возвращающая объем цилиндра, принимает значение аргументов только в единицах длины — с размерностью длины. И выдает эта функция значения только с единицами (размерностью) длины в кубе (объема).

Рис. 12.19
Создание и вызов «размерной» функции пользователя

Рис. 12.20
Ошибка при вызове функции пользователя

Следующий пример (рис. 12.20) показывает, что в среде Mathcad ведется контроль только соответствия размерностей физических величин, а не самих физических величин.

На рисунке 12.20 показано, что ошибка при вызове функции, возвращающей объем цилиндра, может возникнуть, если поменять местами аргументы функции: на место диаметра основания цилиндра поставить его высоту, а на место высоты — диаметр. Этой ошибки можно избежать, усилив инструмент контроля, нацелив его не на размерности, а на *физические величины*. Дело в том, что диаметр основания цилиндра и его высота — это две разные физические величины, имеющие одинаковые размерности длины. Впервые, в наиболее законченной форме, эта идея была эффективно развита в книге Хантли[8], который, в частности, предложил использовать «векторные размерности», т. е. приписывать разнонаправленным величинам различные символы размерности.

Таким образом, единицы измерения в среде Mathcad нужно привязывать не к размерностям (длина — метры, футы, мили и т. д., масса — килограммы, фунты и т. д.), а к физическим величинам. При этом нужно будет принять, что метры диаметра основания нашего цилиндра и метры его высоты — это разные единицы измерения, и это их «различие», в частности, должно выражаться в том, что эти

[8] Хантли Г. Анализ размерностей. — М. : Мир, 1970.

метры нельзя, например, складывать: возникнет ошибка, как если бы мы складывали длину и массу. Конечно, с точки зрения математики, это выглядит абсурдным. Но таковы уж законы программирования: для достижения главной цели (контроль размерностей) иногда можно и пожертвовать здравым смыслом!

В пакет Mathcad вшиты восемь размерностей: длина, время, масса, сила тока, температура, сила света, количество вещества («великолепная семерка» SI) и стоимость (только в 14-й и 15-й версиях и в Mathcad Prime). Если же в расчет необходимо ввести размерность, не перечисленную выше, например, количество информации с единицами бит, байт и т. д. или стоимость (в версиях Mathcad до 14-й), то приходится под эту размерность резервировать одну из свободных размерностей, размерность, которая не задействована в данном расчете. На рис. 12.21 показано, как можно две одинаковые размерности (диаметр основания цилиндра и его высоту) разделить на две разные физические величины и прикрепить к ним две различные единицы длины.

Рис. 12.21

Расчет объема цилиндра с диаметром основания (первая физическая величина) и высотой (вторая физическая величина)

Теперь вольная или невольная перестановка местами аргументов у функции, возвращающей объем цилиндра, будет прерываться сообщением об ошибке, которую можно расшифровать так: «Тут стоит не та физическая величина — высота цилиндра вместо диаметра его основания!» Правда, в сообщении об ошибке, показанном на рис. 12.21, стоит иное: «Тут должна быть длина, а стоит сила света!» Но если вспомнить, что «безработная» в данном расчете кандела (cd) была нами связана с недостающей физической величиной, то все встанет на свои места.

И последнее.

Встроенные в Mathcad инструменты решения задач оптимизации иногда оказываются бессильными при большом числе переменных оптимизации и сложной целевой функции. Тут приходится разрабатывать индивидуальную схему решения или искать решение на стороне. Так, фирма «Сигма Технология», например, предлагает пользователям Mathcad и других популярных расчетных программ пользовательские функции для решения сложных задач оптимизации. Подробнее — см. **www.iosotech.com/ru**.

Задание читателям

Попытаться решить оптимизационные и прочие задачи данной книги с помощью инструментов нечетких множеств.

Этюд 13

ПУТЕШЕСТВИЕ ОКРУЖНОСТИ В ТРЕУГОЛЬНИКЕ, А ТРЕУГОЛЬНИКА В ЛОЖБИНЕ, ИЛИ САМ СЕБЕ КОМПЬЮТЕРНЫЙ РЕЖИССЕР

В этюде будет рассказано, как можно анимировать численные методы решению задач на компьютере.

Математика: декартовы координаты, треугольник, вписанная окружность, свойства биссектрис треугольника, теорема Пифагора, теорема Герона, площадь треугольника, система уравнений, уравнение прямой на плоскости, линии уровня функции двух переменных, лемниската Бернулли.

Физика: сила Кориолиса, закручивание воды в воронке.

Информатика: трассировка промежуточных данных численных методов решения уравнений, символьная математика, анимация, симплекс-метод, метод «Два шага», цикл с предпроверкой, цикл с параметром, условный оператор.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127816**

Задача. Треугольник задан декартовыми координатами своих вершин (x_1, y_1), (x_2, y_2) и (x_3, y_3). Найти параметры вписанной в этот треугольник окружности: ее радиус r и декартовы координаты центра (x_r, y_r) (см. рис. 13.1).

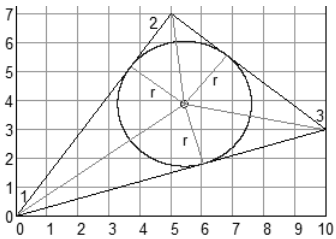


Рис. 13.1
Задача об окружности, вписанной в треугольник

Можно попытаться поискать в справочниках или Интернете готовые формулы, по которым рассчитываются значения искомых величин r, x_r и y_r по значениям заданных переменных x_1, y_1, x_2, y_2, x_3 и y_3. Но готовых формул для всех трех величин r, x_r и y_r ни в справочниках, ни в Интернете найти не удастся (их и быть не может, вернее, пока не может: читаем этюд дальше). Можно найти только одну очень изящную формулу, по которой рассчитывается радиус вписанной в треугольник окружности в зависимости от его площади (S) и полупериметра (p[1]) или в зависимости от длин сторон треугольника (a, b и c) и полупериметра (площадь тре-

[1] Нормальный человек под переменной p будет понимать периметр (**p** — perimeter) треугольника — сумму длин его сторон. Но в этой формуле радиуса вписанной окружности для ее простоты математики под переменную p зарезервировали полупериметр. А сам периметр обозначается заглавной буквой P, так же как площадь обозначается заглавной буквой S. И это очень путает многих. Истинно сказано: «Иная простота хуже воровства». Автора, например, до сих пор смущает и путает тот факт, что в формуле площади круга стоит его радиус в квадрате, а в формуле длины окружности этот радиус нужно помножить еще и на двойку, т. е. работать уже с диаметром, а не радиусом. Отсюда вывод. Вводя в расчет даже простейшую формулу (формулу для площади круга, например), чтобы не ошибиться, стоит себя подстраховать: искать формулу не в своей голове, а в справочниках — бумажных или интернетовских. Вернее, так: сначала попытаться найти в голове (это очень хорошо для тренировки памяти), а потом сравнить найденное с формулой из справочника.

угольника выражена формулой Герона $S_\Delta = \sqrt{p(p-a)(p-b)(p-c)}$, которую изучают в школе):

$$r = \frac{S}{p} = \sqrt{\frac{(p-a)(p-b)(p-c)}{p}}.$$

Величины S, p, a, b, и c нужно будет рассчитывать по другим формулам, которые опять же нужно где-то искать или выводить самому. Длины сторон треугольника можно, например, вычислить по значениям декартовых координат вершин с привлечением теоремы еще одного древнего грека Пифагора — см. рис. 13.2.

$x_1 := 0m \quad y_1 := 0m \quad x_2 := 5m \quad y_2 := 7m$

$a := \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = 8.602\,m$

$x_3 := 10m \quad y_3 := 3m$

$b := \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} = 6.403\,m$

$c := \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2} = 10.44\,m$

$p := \frac{a+b+c}{2} = 12.723\,m$

$S := \sqrt{p\cdot(p-a)\cdot(p-b)\cdot(p-c)} = 27.5\,m^2$

$r := \frac{S}{p} = 2.161\,m$

Рис. 13.2

Расчет в среде Mathcad радиуса окружности, вписанной в треугольник

Рисунок 13.2 — это протокол расчета радиуса вписанной в треугольник окружности, выполненный в среде Mathcad. Координаты вершин треугольника на рис. 13.2 даны в метрах. Это покоробит многих «чистых» математиков. Они сразу скажут, причем тут единицы длины[2]? Почему здесь именно метры, а не сантиметры, километры или футы с дюймами?! А вот почему! Ввод в расчет единиц измерения позволяет вести контроль размерностей и выявлять возможные ошибки при вводе формул. Забыл человек, например, возвести переменную в квадрат или по ошибке возвел ее в куб. Расчет без метров эту оплошность «проглотит», а если ввести размерные величины, то такая ошибка компьютером сразу будет замечена и исправлена [4].

При решении задачи о треугольнике, вписанном в окружность, можно не искать готовые решения, а поступить так, как было описано в [32] — проанализировать задачу, вспомнить, например, что радиус, проведенный в точку касания, перпендикулярен касательной (см. ниже). В результате такого анализа нетрудно составить систему трех алгебраических уравнений, описывающих положение окружности в треугольнике, и решить эту систему на компьютере аналитически (символьно — см. рис. 13.3) или численно (приближенно — см. рис. 13.4). Раньше по понятным причинам нельзя было и помыслить о такой «лобовой атаке» на задачу — не было доступных компьютерных средств решения систем нелинейных

[2] В старых версиях Mathcad была возможность оперировать не конкретными единицами измерения (метрами, секундами, килограммами и т. д.), а физическими величинами (длиной, временем, массой и т. д.), что более логично для нашей задачи. Но это уже будет «коробить» инженеров, любящих конкретику в расчетах. Так что всем не угодишь. Кого-то, например, может даже «коробить» такое «чисто математическое» выражение $x + x^2$: нельзя складывать переменную и ее квадрат!

алгебраических уравнений[3]. Теперь же — другое дело! Почти у каждого студента или школьника под рукой есть компьютер, в который несложно загрузить соответствующие программные средства решения такого рода задач!

Три алгебраические уравнения получаются, если исходный треугольник разбить на шесть прямоугольных треугольников (вернемся к рис. 13.1). В каждом из них одна из вершин упирается в центр окружности, а один из катетов является радиусом окружности. Гипотенузами этих прямоугольных треугольников будут отрезки прямых, соединяющих искомый центр окружности с вершинами треугольника. Наши уравнения, показанные на рисунках 13.2 и 13.3, фиксируют тот факт, что длины сторон исходного треугольника — это суммы длин пар катетов смежных прямоугольных треугольников.

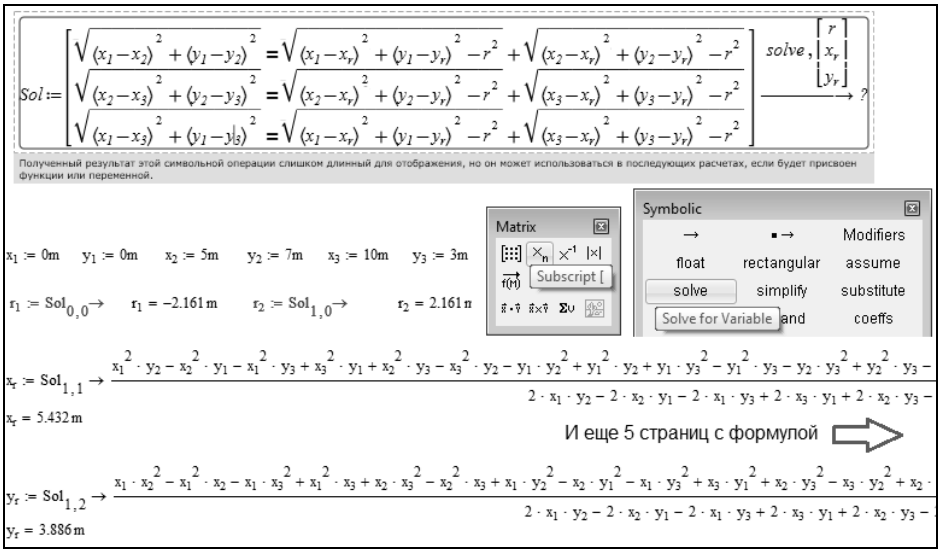


Рис. 13.3

Аналитическое решение задачи об окружности, вписанной в треугольник

На рис. 13.3 показано использование в среде Mathcad аналитического (символьного) оператора `solve` при поиске *формул* для переменных r, x_r и y_r. Mathcad решил поставленную задачу, но не смог выдать «на печать» очень объемный ответ[4] — матрицу с двумя столбцами (два решения системы) и тремя строками (три формулы для трех неизвестных системы). Сообщение об этой ошибке «вывалилось» из «покрасневшего от натуги» оператора `solve`. Руководствуясь этим сообщением, мы вывели «на печать» отдельные элементы матрицы-ответа `Sol`, да и то только малые части формул для x_r и y_r. Формулу для r пакет Mathcad «на печать» выдать так и не смог. Эти формулы можно попытаться упростить, но они все равно останутся слишком объемными. Поэтому-то их и нет в справочниках или Интернете. Но в принципе, несмотря на их очень большой размер, в настоящее время их можно поместить в интернет-справочники в виде, например, очень длинных текстов, но не для изучения в образовательных целях, а для копирования и вставки в компьютерные программы, где необходимо рассчитать положение вписанной в треугольник окружности по координатам вершин треугольника.

[3] Эти уравнения не алгебраические (состоящие из полиномов), а иррациональные [3]. Термин «алгебраический» тут означает, что это не дифференциальное уравнение.

[4] Мы об этом уже писали в этюде 4 — см. рис. 4.3 на стр. 65.

По трем формулам (двум видимым и одной невидимой — см. рис. 13.3) были подсчитаны значения искомых величин r, x_r и y_r при заданных ранее значениях координат вершин треугольника.

Если же нужно решить нашу задачу уже не на плоскости, а в пространстве (найти параметры сферы, вписанной в тетраэдр, например), то аналитические подходы тут уже не сработают. В этом случае придется сразу прибегать к численным методам решения задач. Проиллюстрируем их на уже нами решенной «плоской» задаче об окружности и треугольнике (см. рис. 13.4).

$x_1 := 0m \quad y_1 := 0m \quad x_2 := 5m \quad y_2 := 7m \quad x_3 := 10m \quad y_3 := 3m$

Given

Первое предположение $\quad (r \;\; x_r \;\; y_r) := (1 \;\; 10 \;\; 10)\,m$

$$\sqrt{(x_1-x_2)^2+(y_1-y_2)^2} = \sqrt{(x_1-x_r)^2+(y_1-y_r)^2-r^2}+\sqrt{(x_2-x_r)^2+(y_2-y_r)^2-r^2}$$

$$\sqrt{(x_2-x_3)^2+(y_2-y_3)^2} = \sqrt{(x_2-x_r)^2+(y_2-y_r)^2-r^2}+\sqrt{(x_3-x_r)^2+(y_3-y_r)^2-r^2}$$

$$\sqrt{(x_1-x_3)^2+(y_1-y_3)^2} = \sqrt{(x_1-x_r)^2+(y_1-y_r)^2-r^2}+\sqrt{(x_3-x_r)^2+(y_3-y_r)^2-r^2}$$

Решение $\quad \begin{pmatrix} r \\ x_r \\ y_r \end{pmatrix} := \mathrm{Find}(r, x_r, y_r) = \begin{pmatrix} 2.161 \\ 5.432 \\ 3.886 \end{pmatrix} m$

Рис. 13.4

Численное решение задачи об окружности, вписанной в треугольник

На рисунке 13.4 показано *численное* решение нашей системы трех нелинейных уравнений. Для этого записываются исходные данные (координаты вершин треугольника), вводится ключевое слово `Given` (`Дано`), задаются значения первого предположения решения и записывается сама система уравнений. Встроенная в Mathcad функция `Find` (`Найти`) начинает по особому численному алгоритму менять значения своих аргументов (неизвестных r, x_r и y_r) так, чтобы прийти в конце концов к решению — найти значения этих переменных, подстановка которых в исходные уравнения превращает их в тождества. Вернее, *почти* (примерно, *приближенно*) в тождества: правые и левые части уравнений должны по модулю отличаться друг от друга на наперед заданное число `CTOL`. По умолчанию `CTOL` = 10^{-3}, а единицы измерения при этом равны базовыми единицам СИ — метрам в нашем случае. То есть наша геометрическая задача решена с точностью до миллиметра (10^{-3} m). Если бы мы наш треугольник и окружность измеряли не метрами, а сантиметрами (что более естественно — вспомним, как мы вычерчивали геометрические фигуры в линованной тетрадке по математике), то ответ был бы слишком грубым. Тут нужно было бы менять значение системной переменной `CTOL` на значение 10^{-7}, например.

А по какому алгоритму функция `Find` «толкает» значения неизвестных от начального предположения, заданного пользователем, к искомому решению?! Можно узнать название этого алгоритма (метода) через нажатие правой кнопки мыши на слове `Find` и поискать информацию по конкретному методу в том же Интернете (см. рис. 13.5). А можно поступить несколько иначе — не так скучно.

Пакет Mathcad 15, в среде которого численно решалась задача (рис. 13.4), имеет очень полезную возможность записи промежуточных значений искомых величин при реализации тех или иных встроенных инструментов численного решения задач. В нашем случае можно вспомнить крылатую фразу и сказать: «Цель ничто — движение всё!». Нас особо не интересует цель — **место** «вписывания» окружности в треугольник (мы его уже нашли аналитически — см. рис. 13.3 и численно — см. рис. 13.4), но то, **как** окружность туда «закатывается численно», то есть ее **движение**.

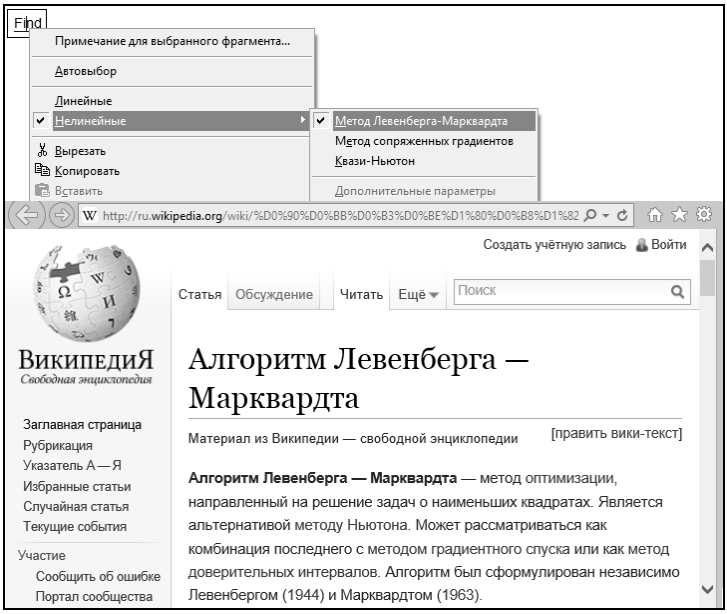


Рис. 13.5

Выбор метода решения системы уравнений в среде Mathcad 15 и справка в Интернете

На рисунке 13.6 показана траектория движения центра окружности от точки начального предположения (`x = 10 m`, `y = 10 m`) до «цели» — до решения (`x = 5.432 m`, `y = 3.886 m`). Мы специально задали пакету Mathcad довольно трудное задание — поместили точку начального предположения вне треугольника, чтобы посмотреть, как пакет «будет идти к цели»: «Много ты, компьютер, о себе воображаешь — посмотрим, проглотишь ли ты вот это!» А нужно было бы, конечно, задать значения первого предположения вблизи решения — где-то в центре исходного треугольника. Но, повторяем, «Цель ничто — движение всё!»

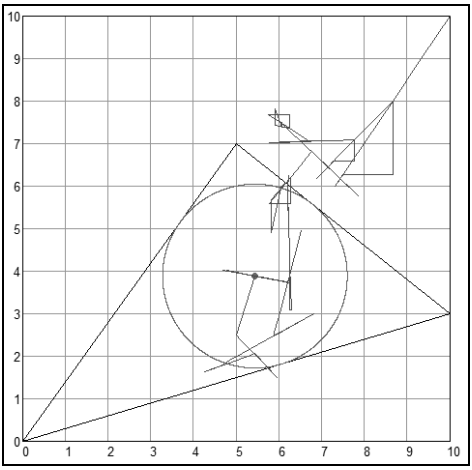


Рис. 13.6

Траектория движения центра окружности в треугольник

Но интересно просмотреть это движение окружности в **анимации**, инструменты создания которой есть в Mathcad 15 [44], и мы их довольно подробно описали в

этюде 8. На рис. 13.7 показан начальный кадр этой анимации (`n = 0`, а всего кадров будет 373): окружность (видна только ее дуга-четверть) с радиусом `1 m` находится в точке `x = 10 m`, `y = 10 m`. Затем (первые 94 кадра анимации) окружность срывается с места и ударяется о ближайшую сторону треугольника (рис. 13.8), подскакивает и ударяется о треугольник снова и снова, как бы пытаясь пробить в нем дыру.

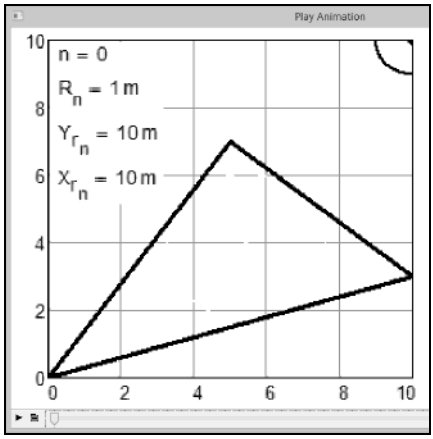


Рис. 13.7
Начальный кадр анимации движения окружности в треугольник

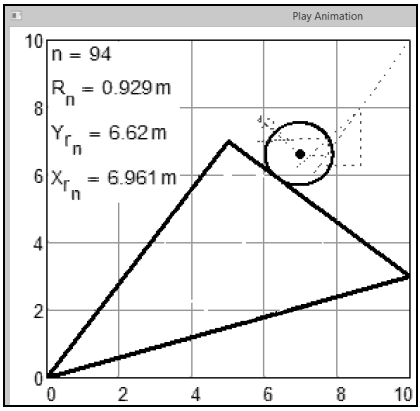


Рис. 13.8
94-й кадр анимации движения окружности в треугольник

Но такой «лобовой таран» не удается и окружность начинает кататься по стороне треугольника, пытаясь как-то проникнуть в него (рис. 13.8). Эти попытки оказываются также безуспешными. Тогда наша «живая и настырная» окружность начинает уменьшаться в размерах почти до точки (рис. 13.9), находит некую гипотетическую «дырочку» в стороне треугольника и проникает внутрь его (рис. 13.10).

Затем точка в треугольнике увеличивается до окружности и начинает «биться» внутри треугольника (рис. 13.11), меняя свое положение и радиус, пока не «успокоиться» — не найдет в нем свое место: решение задачи (рис. 13.12 и 13.6).

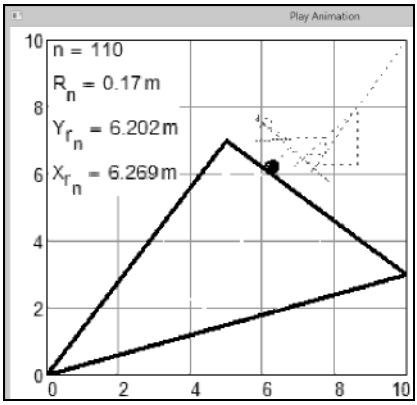


Рис. 13.9
110-й кадр анимации движения окружности в треугольник

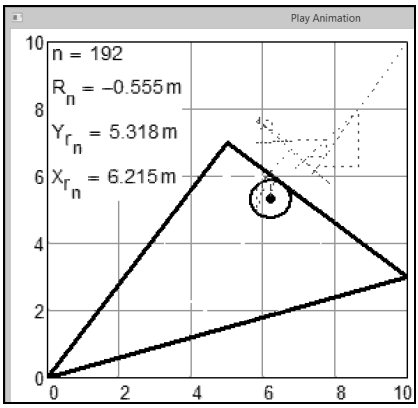


Рис. 13.10
192-й кадр анимации движения окружности в треугольник

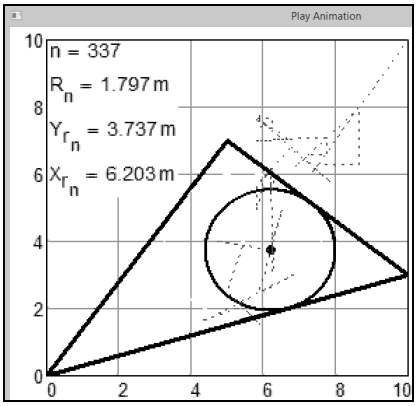


Рис. 13.11
337-й кадр анимации движения окружности в треугольник

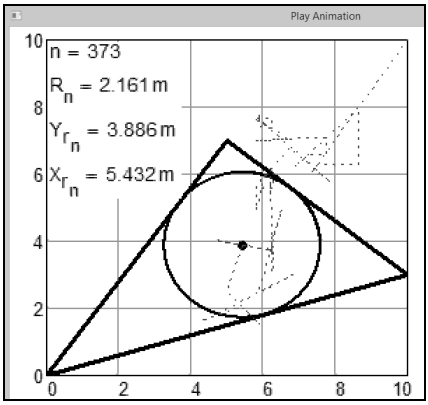


Рис. 13.12
Финальный кадр анимации движения окружности в треугольник

Такое «живое» поведение окружности при «численном» ее вписывании в треугольник, отображенном на рис. 13.7–13.12, видно только при хорошем пространственном воображении. Но можно рисунки с кадрами анимации вырезать из книги, сшить их в блокнот, быстро перелистывать его, перегнув и зажав пальцами страницы и отпуская их[5]. Такие книжки-игрушки (флит-бук) с незамысловатыми сюжетами были очень популярны до изобретения кино и телевидения. Так примерно и поступает пакет Mathcad 15 при создании анимации, «сшивая в блокнот» отдельные кадры, а затем показывая их со скоростью, достаточной для анимирования. Анимация движения окружности в треугольник (рис. 13.7–13.12) опубликована на сайте PTC Community под номером 3972[6].

Как можно выводить промежуточные данные при запуске встроенных в Mathcad 15 инструментов численного решения задач?

Для этого в среде Mathcad 15 есть три инструмента: встроенная функция `trace`, режим работы `Toggle Debugging` (`Режим отладки`) и окно Trace Window, куда заносятся промежуточные данные. Все они показаны на рис. 13.13.

Рис. 13.13
Инструменты вывода промежуточных результатов численного поиска решения

[5] Эврика! Можно попросить издательство отмакетировать данную книгу так, чтобы рисунки 13.7–13.12 располагались в одном месте разных страниц и их можно было «оживлять», быстро перелистывая страницы книги. Но проще, конечно, войти в Интернет по адресу, отмеченному в ссылке ниже, и просмотреть анимацию на компьютере.

[6] Полный адрес этой анимации **http://communities.ptc.com/videos/3972**. PTC Community, повторяем, это форум, где пользователи Mathcad и других программ фирмы PTC задают вопросы и обмениваются информацией.

Функция `trace` должна быть вставлена в одно из алгебраических уравнений (см. верхний левый угол на рис. 13.13). Аргументами функции `trace` будут поочередно: радиус окружности (как на рис. 13.13) и координаты центра, начальные, промежуточные и конечные значения которых будут автоматически заноситься в окно `Trace Window` (левый нижний угол рис. 13.13), если включен режим отладки (правая часть рис. 13.13). Эти массивы данных нужно будет скопировать, перенести в расчет в виде векторов, отобразить на графике и создать по ним анимацию движения окружности от точки первого предположения до конечной точки. При этом можно менять начальные размер и положение окружности, а также метод численного решения задачи и точность поиска корня алгебраического уравнения. Так можно «снимать довольно увлекательное кино», т. е. заниматься своего рода *компьютерным изобразительным искусством* [42], базирующемся на прикладной математике.

В этюде 4 на рисунках 4.24 (стр. 85) и 4.27 (стр. 87) показано использование функции `trace` при поиске нуля функции методом секущих и методом половинного деления.

На сайте PTC Community можно увидеть анимацию (номер 5015) численного решения «обратной» задачи — задачи об окружности, описанной вокруг треугольника. Окружность срывается с места в центр треугольника почти к решению, а затем «в сомнениях» (опять живой организм!) начинает рыскать в нем, отыскивая окончательное решение.

Ну а если особо не увлекаться «съемками кино с окружностью и треугольником в главных ролях», а решать задачу о вписывании окружности в треугольник традиционными методами, то нам понадобятся три теоремы — «Нет ничего практичней хорошей теории!»

Теорема 1. Все три биссектрисы треугольника пересекаются в одной точке.

Теорема 2. Центр вписанной в треугольник окружности лежит в точке пересечения биссектрис.

Теорема 3. Биссектриса угла треугольника делит противоположную сторону на отрезки, пропорциональные прилежащим сторонам.

Кстати, в предыдущих расчетах мы опирались еще на одну важнейшую теорему: радиус окружности, проведенный в точку касания, перпендикулярен касательной. Именно это и дало нам возможность нарисовать шесть прямоугольных треугольников и составить для них систему нелинейных алгебраических уравнений, которые мы решили аналитически и численно.

Итак, опираясь на три вышеперечисленные теоремы, наметим план решения поставленной задачи. Нам достаточно найти точку пересечения лишь двух каких-либо биссектрис (теорема 1). Для этого нам понадобятся уравнения этих биссектрис. Уравнение прямой (биссектрисы) получим, зная две точки, лежащие на этой прямой.

Мы исходим из того, что длины сторон исходного треугольника уже найдены. Обозначим a_{12} длину стороны A_1A_2, a_{13} длину стороны A_1A_3 и a_{23} длину стороны A_2A_3 (см. рис. 13.1). Пусть A_3B_3 — биссектриса угла A_3, а точка B_3 лежит на стороне A_1A_2. Аналогично A_2B_2 — биссектриса угла A_2, а точка B_2 лежит на стороне A_1A_3. Тогда (теорема 3) имеем соотношения $\frac{A_2B_3}{A_1B_3} = \frac{a_{23}}{a_{13}}$ и $\frac{A_3B_2}{A_1B_2} = \frac{a_{23}}{a_{12}}$.

Если известны две точки плоскости $A(x_A;\ y_A)$, $B(x_B;\ y_B)$, то координаты точки $M(x_M;\ y_M)$, которая делит отрезок AB в отношении $\frac{AM}{BM} = \lambda$, выражаются формулами:

$$x_M = \frac{x_A + \lambda x_B}{1+\lambda},\quad y_M = \frac{y_A + \lambda y_B}{1+\lambda}.$$

Используя указанные формулы, легко вычисляем координаты точек $B_3(x_{B_3}, y_{B_3})$ и $B_2(x_{B_2}, y_{B_2})$.

Уравнение прямой в общем случае имеет вид $Kx+Ly+P=0$. Но поскольку в начале координат находится вершина A_1, то искомые биссектрисы A_2B_2 и A_3B_3 через начало координат не проходят, а поэтому уравнение каждой биссектрисы будем искать в упрощенном виде $kx+ly+1=0$. Искомые коэффициенты получатся из системы, полученной при подстановке в уравнение прямой координат двух точек, лежащих на этой прямой. Так, уравнение для биссектрисы A_2B_2 будет иметь вид $k_2x+l_2y+1=0$, где $\begin{cases} k_2x_2+l_2y_2+1=0 \\ k_2x_{B2}+l_2y_{B2}+1=0 \end{cases}$, а для биссектрисы A_3B_3 будет иметь вид $k_2x + l_2y + 1=0$, где $\begin{cases} k_3x_3+l_3y_3+1=0 \\ k_3x_{B3}+l_3y_{B3}+1=0 \end{cases}$. Осталось найти точку (x,y) пересечения этих биссектрис, поскольку (теорема 2) она и является центром окружности. Координаты (x,y), таким образом, находятся из системы шести линейных уравнений $\begin{cases} k_2x_2+l_2y_2+1=0 \\ k_2x_{B2}+l_2y_{B2}+1=0 \\ k_3x_3+l_3y_3+1=0 \\ k_3x_{B3}+l_3y_{B3}+1=0 \\ k_2x+l_2y+1=0 \\ k_3x+l_3y+1=0 \end{cases}$, в которой числа x_2, x_3, y_2, y_3 известны по условию, а числа $x_{B2}, x_{B3}, y_{B2}, y_{B3}$ уже найдены.

Ух! Вернемся к «съемкам фильмов» с нелинейными уравнениями.

Система алгебраических уравнений, показанная на рис. 13.3 и 13.4, как мы уже отметили, имеет два решения: одно с положительным значением r и второе с таким же по модулю, но отрицательным значением r. На рис. 13.14 показан кадр еще одного «кинофильма» — решения системы двух алгебраических уравнений: уравнения эллипса и уравнения лемнискаты Бернулли, которая имеет уже не два, а шесть действительных корней. Вокруг этих двух замкнутых кривых и точек их пересечения (корни системы двух уравнений) в анимации вращается по окружности точка первого предположения. Другая точка (текущее решение системы уравнений — точка пересечения эллипса с лемнискатой) перескакивает с одного места на другое. Эта авторская анимация на сайте PTC Community имеет номер 2418. Она показывает, как точка первого предположения влияет на решение системы — на выбор того или иного корня. Эта анимация акцентирует внимание на важности выбора начальной точки поиска решения.

Иногда вместо термина «первое **предположение**», очень важного при использовании численных методов решения задач, используют термин «начальное **приближение**», что не совсем верно (это мы отмечали и в этюде 4), и анимация, показанная на рис. 13.14, доказывает это: очень часто решение — это не точка, ближайшая к исходной, а совсем другая (это зафиксировано на рис. 13.14), а в ряде случаев — даже одна из самых дальних. Все зависит от системы уравнений и метода, примененного для ее численного решения.

В анимации, показанной на рис. 13.14, точка первого предположения движется по окружности вокруг эллипса и лемнискаты Бернулли. Другая точка, отмечающая корень системы уравнений, будет перескакивать с мест пересечения двух замкнутых кривых. При этом, как, в частности, зафиксировано на рис. 13.14, очередной корень далеко не всегда будет ближайшем к точке на окружности.

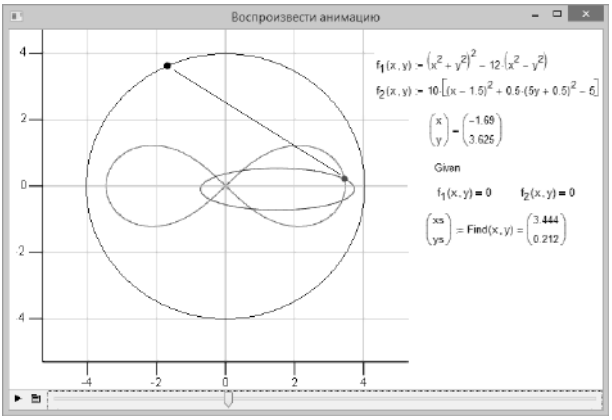


Рис. 13.14
Кадр анимации связи первого предположения с корнем системы уравнений

Наблюдая в анимации движение окружности (рис. 13.1–13.13), мы видим, что она ведет себя как некий **живой организм**, забивающийся в свою «треугольную норку». Тут как раз впору вспомнить, что некоторые численные методы решения задач имеют «животные» названия. Так, один из методов поиска минимума функции называется методом амебы. Понять это название можно, посмотрев, как треугольник, а не окружность может вести себя как **живое существо** и переползать куда-то, решая математическую задачу.

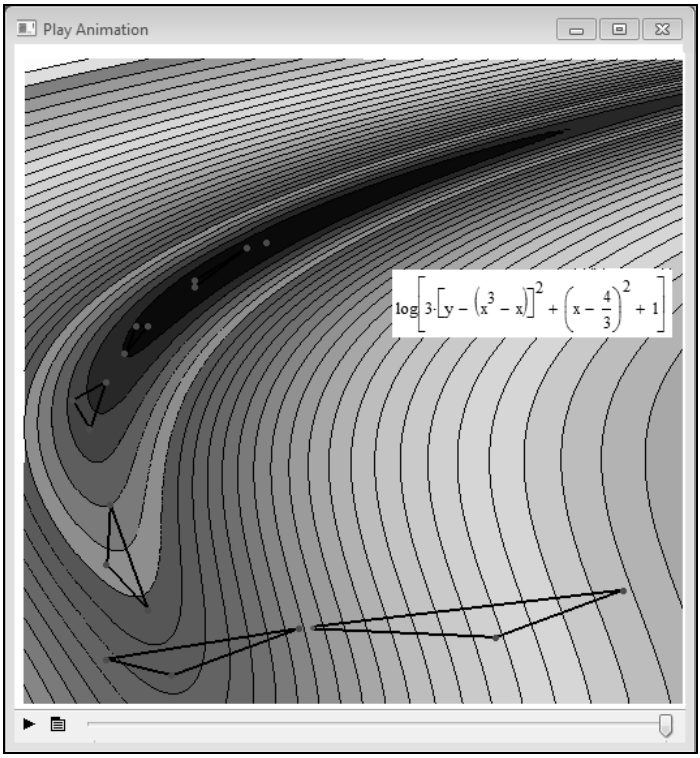


Рис. 13.15
Переползание треугольника в ложбину

Кадры анимации переползания треугольника показаны на рис. 13.15 (анимация под номером 4984 — автор программы и анимации В. Р. Сабанин): футуристическая треугольная амеба переваливаясь с боку на бок ползет из правого нижнего угла в некую ложбину (минимум функции двух аргументов), уменьшаясь при этом в размерах (сжимаясь), пока не найдет искомую точку, обрамленную на рис. 13.15 линиями одного уровня.

Можно найти в Интернете описание этого метода оптимизации (другие его названия — метод Нелдера — Мида, метод деформируемого многогранника, симплекс-метод) и понять его суть. Метод довольно сложен и в описании, и в реализации, но эффективен: точка минимума находится быстро (за малое число ходов) и довольно точно. Если у анализируемой функции не два, а три аргумента, то треугольник заменяется тетраэдром. Если переменных больше трех, то выбирается соответствующий многогранник.

На рис. 13.16 показан менее быстрый, но довольно простой в описании и реализации авторский метод оптимизации, который условно можно назвать «Два шага» (номер анимации 1460).

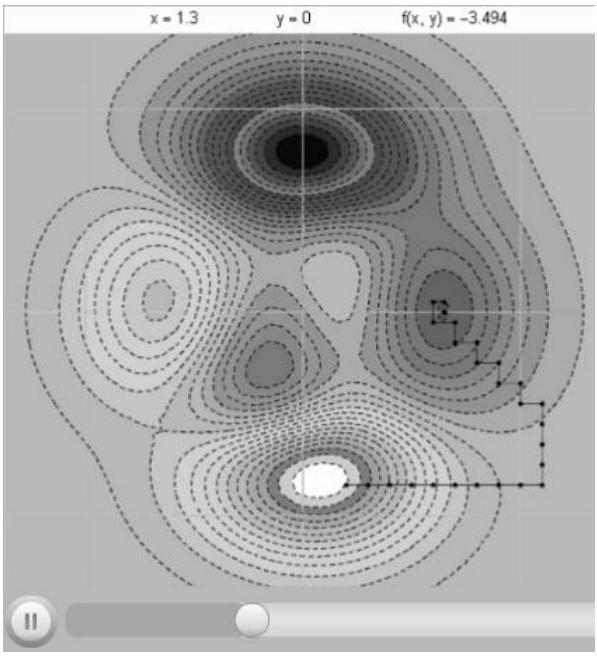


Рис. 13.16
Анимация метода «Два шага»

Программа, еще раз подчеркиваем, довольно проста — больше времени ушло на поиск ее названия, чем на ее создание. Можно было бы, конечно, тут применить стандартный метод градиентного спуска, при котором шаги делаются по направлению градиента (при поиске максимума) или минус градиента (при поиске минимума), но была поставлена цель написать самую простую в реализации программу, особо не думая о ее скорости. Программа была названа «**Два шага — Two step**». Есть такой бальный танец (один персонаж кинофильма «Свадьба в Малиновке» называл его «В ту степь»). Программа при поиске минимума делает два шага от очередной точки приближения (от «печки», если опять же вспомнить танцы) и переносит опорную точку туда, где значение оптимизируемой функции минимально. Эти два шага длиной `X` делаются по всем координатам (их число `L`) в отрицательном (`-D`) и положительном (`D`) в направлениях. По мере приближения к

минимуму шаги укорачиваются. «Танец» повторяется до тех пор, пока шаг не станет короче заданного значения погрешности (`D < CTOL` — см. рис. 13.17).

```
MinTwoStep(f, x, D) := | (L ← last(x)  f_min ← f(x)  j ← p ← n ← 0  M^⟨0⟩ ← x)
                       | while  D > CTOL
                       |   | p ← 1
                       |   | while  p
                       |   |   | p ← 0
                       |   |   | for  i ∈ ORIGIN .. L
                       |   |   |   for  X ∈ −D, D
                       |   |   |     | (x_i ← x_i + X  F ← f(x))
                       |   |   |     | (p ← X  j ← i  f_min ← F)  if  F < f_min
                       |   |   |     | x_i ← x_i − X
                       |   |   | (x_j ← x_j + p  n ← n + 1  M^⟨n⟩ ← x)
                       |   | D ← D/2
                       | M
```

Рис. 13.17
Mathcad-программа «Два шага»

Программа «выводит» на печать не только финальную точку, а весь путь к этой точке (матрица `M`), который несложно отобразить на графике (см. рис. 13.18 и 13.19) и анимировать (см. рис. 13.16).

В программе интересен не сам алгоритм, а след, оставляемый программой на плоскости `X-Y` («в степи») при оптимизации некоторой функции двух переменных — см. рис. 13.18 и 13.19.

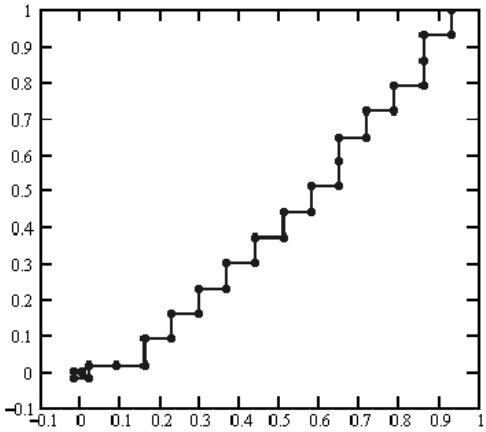


Рис. 13.18
Начальные шаги приближения к минимуму

Форма следа программы «Тустеп» позволила дать ей и другие названия. Считается, что чем больше у человека имен, тем больше у него небесных покровителей. Это, наверное, можно отнести и к программам (методам) решения задач.

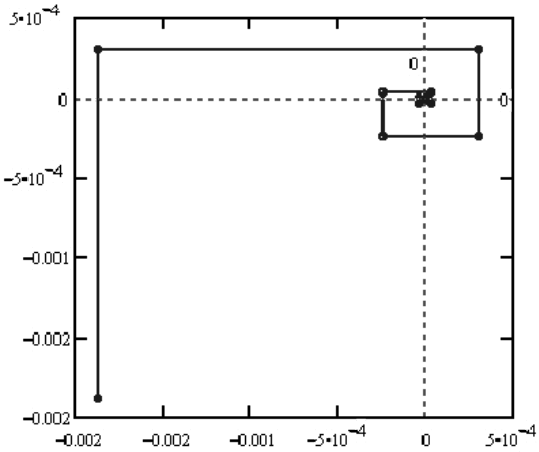


Рис. 13.19
Конечные шаги приближения к минимуму

Если наполнить ванну водой, бросить в нее, к примеру, перышко и выдернуть пробку, то перышко сначала будет более-менее спокойно двигаться к сливному отверстию, а потом закрутится в водовороте. Примерно так, как показано на рис. 13.18 и 13.19: при начальных шагах приближения (рис. 13.18) промежуточные точки спокойно дрейфуют к минимуму, а затем «закручиваются в вихре танца» (рис. 13.19), который позволяет называть данный метод оптимизации не только «Тустеп», но и «**Вальс**». Но если быть совсем точным, то его следует именовать также и методом ***наискорейшего спуска воды***. И вот почему.

Вода в ванне или в каком-либо ином сосуде с дыркой на дне всегда закручивается в одну сторону. Даже если воду раскрутить рукой в другом направлении, то, преодолев насилие, она вернет свою природную закрутку. Это связано с силой Кориолиса, которая в свою очередь зависит от вращения Земли вокруг своей оси. В северном полушарии вода закручивается по часовой стрелке, в южном — против, на экваторе — сливается без вихрей (а на полюсах — совсем не течет: замерзает: шутка!). Это явление основательно изучалось в Alma Mater авторов книги (в Московском энергетическом институте) применительно к барабанным паровым котлам. Дело в том, что вода из барабана котла, поступая в опускные трубы, а затем в подъемные трубы, расположенные в топке котла, может закручиваться и захватывать вследствие этого большие порции водяного пара. Это ухудшает теплообмен и приводит к авариям — разрывам паровых труб от перегрева. Подробнее об этом здесь: **http://twt.mpei.ru/ ochkov/KotelVikhr.html**.

Для проверки этой гипотезы авторы переслали по Интернету файл с программой оптимизации коллеге в Австралию (чем черт не шутит) и к полному своему удивлению, переходящему в удовлетворение, узнали, что след поиска минимума в южном полушарии закручивается против часовой стрелки. «Австралийский» след оптимизации из-за экономии места здесь не приводится. Но его можно увидеть, посмотрев на рис. 13.18 и 13.19 через зеркало.

Причина иной закрутки следа поиска минимума южнее экватора, конечно, не в воде, хотя сила Кориолиса здесь присутствует: при пересылке файла в Австралию в код программы (рис. 13.17) вкралось маленькое изменение. Вместо строки `for X ∈ -D, D` появилась другая — `for X ∈ D, -D`. На поиск минимума это не повлияло (перебор координат оптимизируемой функции ведется всего лишь с другого конца), но, тем не менее, след стал закручиваться по-иному. А виной тому все та же сила Кориолиса, заставляющая жесткие диски серверов и маршрутизаторов в южном полушарии Земли вращаться несколько иначе, чем в северном полуша-

рии. Отсюда и незначительные сбои при пересечении файлами экватора. Их выловить довольно трудно, т. к. при контрольной обратной пересылке файла ошибка исправляется по принципу минус на минус дает плюс. (Продолжение темы закрутки следа поиска минимума — **http:// twt.mpei.ac.ru/ochkov/SI.)**

Современные программные средства позволяют легко и быстро анимировать численные методы решения математических задач. Такая анимация позволяет лучше понимать эти методы, особенности и ограничения при их применении для решения прикладных задач на компьютере. Кроме того, такая анимация позволит оживить, сделать более занимательными занятия по изучению данных методов на семинарах и уроках информатики и прикладной математики.

Авторы не устают повторять своим студентам, что от учебы нужно стараться в первую очередь получать не знания и навыки, а... удовольствие. Учеба — это работа, очень трудная работа, а без удовольствия, без любви к труду даже необременительная, престижная и высокооплачиваемая работа может превратиться в каторгу. Удовольствие же немыслимо без шутки, без легкого и ироничного отношения к самым серьезным вещам. А что может быть «серьезнее» математики? Наша средняя школа со своей зубрежкой ответов для ЕГЭ напрочь отбивает у многих школьников тягу к получению знаний, к «получению удовольствия от процесса получения знаний». Эту книгу можно рассматривать как попытку вернуть учащимся интерес к этой самой главной человеческой способности — к получению знаний. Многие студенты тянутся к компьютеру, но будущая их инженерная специальность — это не всегда компьютерные технологии (IT), а, например, теплотехника [2]. Авторы надеются, что данная книга поможет этим студентам хотя бы через компьютер и Интернет возродить в себе способность к получению удовольствия от учебы в техническом вузе, основа которой является математика. Недаром Иммануил Кант говаривал, что «в каждой естественной науке заключено столько истины, сколько в ней есть математики».

Задание читателям

Создать объемную анимацию движения к точке минимума методом «Два шага». «Плоский» случай показан на рис. 3.18 и 3.19.

Этюд 14

ИНТЕРПОЛЯЦИЯ, ЭКСТРАПОЛЯЦИЯ И СГЛАЖИВАНИЕ, ИЛИ ЛОЖЬ, НАГЛАЯ ЛОЖЬ И СТАТИСТИКА[1]

В этюде будет рассказано о некоторых методах статистической обработки массивов данных.

Математика: функция, корреляция, интерполяции (линейная и сплайнами), экстраполяция, регрессионный анализ, статистическая выборка, таблица значений функции, точность вычислений, диаграмма рассеяния, гистограмма, система двух линейных уравнений, метод наименьших квадратов, производная, сплайн, функция нормального распределения, Гауссова кривая, экспоненциальная зависимость, график функции двух переменных, линии уровня, минимум функции двух переменных.

Физика: статистическая обработка экспериментальных данных, погрешность измерений, теплопроводность, вес и рост человека.

Информатика: база данных с полями и записями, интерактивные расчеты в Интернете, числа с плавающей точкой, расчеты по эмпирическим и псевдоэмпирическим формулам, компьютерные средства аналитических преобразований (символьная математика), облачная функция, генетический алгоритм оптимизации.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127815**

Как справедливо было отмечено в [66, с. 241], «преувеличение роли информатики столь же, а может быть, и более опасно, чем преуменьшение этой роли». Подводя итоги последнего десятилетия, мы можем с уверенностью сказать, что современные студенты доверяют Википедии больше, чем своему лектору, а их вера в возможности вычислительных пакетов вообще не имеет границ. Не будучи способными разобраться в математических методах и алгоритмах, «спрятанных» в недрах математических вычислительных программ, молодые исследователи в своих работах допускают принципиальные и неустранимые ошибки. Выходом и такого положения является как можно более раннее (еще в школе) знакомство учащихся с базовыми основами численных методов с помощью примеров, доступных пониманию школьников. Решение обыденных математических задач, переработанных для образования школьников с помощью информатики, и будет реальным шагом на пути фундаментализации обучения информатике [67].

Продемонстрируем реальную попытку объединения информатики и математики в рамках единой образовательной области, определим точки соприкосновения математики и информатики [68]. Приведем примеры результативного взаимодействия этих наук, используя понятные всем объекты и задачи.

Сначала дадим расшифровку некоторых терминов, которые помогут нам понять суть проблемы, поднимаемой в этом этюде [69].

Функция (от лат. *functio* — исполнение, осуществление) — это закон `f`, по которому каждому элементу `x` из одного числового множества ставится в соответствие некоторый определенный элемент `y` из другого числового множества, так что `y = f(x)`.

Корреляция (от лат. *correlatio* — соотношение) — вероятностная или статистическая зависимость, не имеющая строго функционального характера (см. выше) из-за невозможности точно учесть влияние множества одновременно меняющихся факторов.

[1] Эту фразу приписывают многим известным людям — зарядите ее в Google и посмотрите, кто так высказывался, и главное, по какому поводу.

Интерполяция (от лат. *interpolation* — изменение, переделка) — приближенное или точное нахождение какой-либо величины по известным отдельным значениям этой величины, т. е. восстановление (точное или приближенное) функции (см. выше) по ее нескольким известным значениям.

Регрессия (от лат. *regressio* — обратное движение) — такая зависимость среднего значения какой-либо величины от некоторой другой величины или от нескольких величин, что одному и тому же значению независимой переменной x могут соответствовать в зависимости от случая различные значения величины y.

Примеры функции и корреляции были даны в конце этюда 1, когда рассматривалась связь атмосферного давления, температуры кипения воды и высоты над уровнем моря.

А теперь к делу.

При обработке экспериментальных данных мы обычно сталкиваемся с двумя принципиально разными случаями: или полученные данные связаны некоторой функциональной зависимостью, формула которой нам неизвестна, или таковой функциональной зависимости не существует, хотя и наблюдается корреляция.

Но есть еще и третий вариант: две величины действительно связаны функциональной зависимостью, и теоретически каждому значению аргумента x соответствует ровно одно значение y. Но при проведении эксперимента не удается получить значения с достаточной степенью точности, и поэтому экспериментальные результаты представляются в виде таблиц значений (x_i, y_i), $y_i = f(x_i) \pm \varepsilon_i$, где ε_i — погрешности измерений. Если в этом случае просто соединить экспериментальные точки отрезками прямых, то получится ломаная, не имеющая ничего общего с той функциональной зависимостью, которая реально существует. Причем форма этой ломаной из-за ошибок измерений при повторном эксперименте вряд ли не воспроизведется.

В первом из рассмотренных случаев для получения приближенной формулы зависимости двух величин применяют интерполяцию. А вот во втором и третьем используют регрессионный анализ. Приведем примеры.

Студентам, решающим задачи по физике, химии и другим учебным дисциплинам, часто приходится заглядывать в справочники, где некая функция представлена в виде таблицы: теплопроводность какого-либо металла в зависимости от температуры, плотность водного раствора какой-либо соли при фиксированной концентрации зависит от той же температуры и т. д. и т. п.[2] В этих таблицах пары чисел «температура — свойство» показаны лишь для некоторых значений температуры, например, 0, 10, 20, 30 и т. д. градусов по шкале Цельсия. А значение свойства (теплопроводности металла или плотности раствора) нужно, допустим, взять при 17°С. Подразумевается, что величины, указанные в таблицах, являются точными (правильнее сказать, получены с достаточной степенью точности). Поскольку физические законы позволяют считать рассматриваемые величины функционально зависимыми, то применяют интерполяцию. Простейшую — линейную или более сложную — интерполяцию многочленами либо сплайнами. При линейной интерполяции мы мысленно, на бумаге или на дисплее компьютера фиксируем две точки, проводим через них прямую линию и на ней находим нужное промежуточное значение. При нелинейной интерполяции отыскивается обладающая определенной гладкостью функция, график которой проходит через указанные точки (интерполяционные узлы).

Но всегда следует помнить, что в случае отсутствия каких-либо определенных функциональных связей между двумя величинами интерполяция может привести к ложному результату.

[2] Раньше приходилось определять по таблицам значение тригонометрических, логарифмических, показательных и других элементарных функций. Сейчас активно используются таблицы значений специальных функций, например, дифференциальной и интегральной функций Лапласа.

Когда-то перед лекцией на тему «Регрессионный анализ» по курсу «Информационные технологии» первый автор книги подбирал пример статистической выборки для такого анализа. Но когда он «взошел на кафедру» и взглянул на аудиторию (рис. 14.1), то он понял, что эта выборка находится прямо перед его глазами. Была проведена перекличка студентов. Юноши вставали и сообщали свой *вес*[3] и *рост* (студентки по понятным причинам, были освобождены от этой процедуры). Числа заносились в два вектора с именами `Вес` и `Рост` в среде математической программы Mathcad. В векторах получилось по 50 элементов. Эти массивы чисел послужили хорошей затравкой для лекции.

Рис. 14.1
Студенты — «подопытные статистические кролики»
(правая половина аудитории: часть выборки)

Как уже отмечено, существует два основных способа первичной обработки полученных данных: аппроксимация (проведение линии вблизи точек) и интерполяция (проведение линии через выбранные точки). Каждый из этих способов даст некоторое представление о зависимости (функциональной или статистической) двух величин. У каждого из способов есть сильные и слабые стороны, каждый имеет свои особенности компьютерной реализации.

Начнем с интерполяции. Она (теоретически) может быть осуществлена с помощью любой известной функции, но на практике предпочитают использовать многочлены. И в первую очередь — многочлены первой степени, чьими графиками являются прямые линии на плоскости. Такую интерполяцию называют линейной.

Линейная интерполяция, формально примененная к паре величин «Рост — Вес», показана на рис. 14.2. Были взяты данные двух студентов («маленький студент» и «большой студент») и через соответствующие точки (узлы интерполяции) была проведена прямая линия на плоскости. Были также определены параметры этой прямой — коэффициенты `a` и `b` уравнения `y = a + b x`.

Как рассчитывались коэффициенты `a` и `b`, показано на рис. 14.3. Можно в справочниках поискать соответствующие формулы, а можно в среде Mathcad решить пару линейных уравнений (рис. 14.3) и получить ответ.

На рисунке 14.3 в таблицу вводятся вес и рост «маленького и большого студентов», затем записывается система двух линейных алгебраических уравнений прохождения прямой через две точки, формируется квадратная матрица `M` коэффициентов при неизвестных `a` и `b` и вектор свободных членов `v`. Решение системы найдено с помощью встроенной в Mathcad функции `lsolve`. Нашу систему двух линейных уравнений можно решить и аналитически (рис. 14.4), получив формулу для

[3] Имелась в виду, конечно, масса студента, а не его вес. В середине XX в., когда наша страна переходила на международную систему измерений (СИ), предписывающую вес измерять не в килограммах силы, а в ньютонах, школьники и студенты шутили примерно так — заходили в магазин и просили продавщицу отвесить 500 ньютонов колбасы.

определения коэффициентов a и b, входящих в уравнение прямой линии, проходящей через две заданные точки с координатами на плоскости (x_1, y_1) и (x_2, y_2).

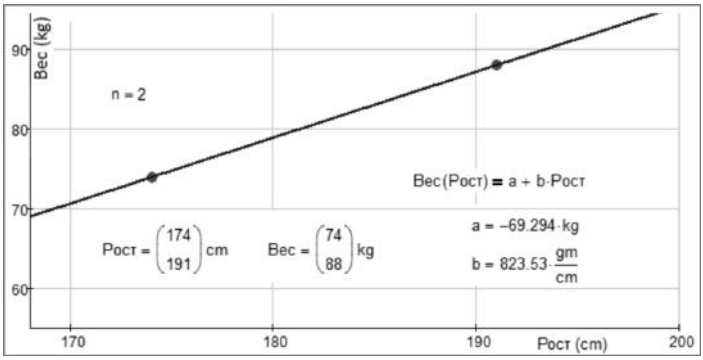


Рис. 14.2
Линейная интерполяция по двум студентам

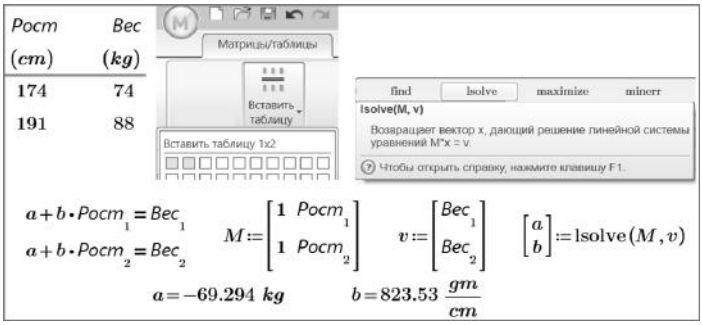


Рис. 14.3
Численное решение системы линейных уравнений для прямой, проходящей через две точки на плоскости

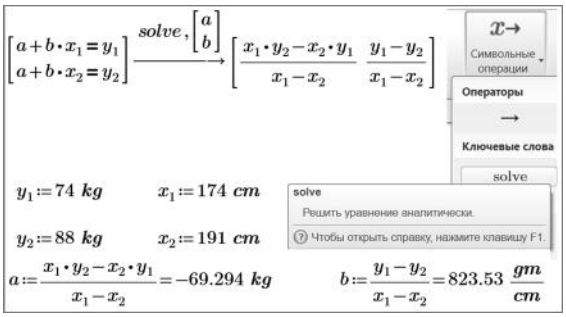


Рис. 14.4
Символьное решение системы линейных уравнений прямой, проходящей через две тоски на плоскости

Если выбирать пару студентов случайным образом, то наша прямая линия может быть различными способами ориентирована на плоскости «Рост — Вес», поэтому доверия к такому способу описания зависимости величин в этом случае нет. Линейная интерполяция тут не годится: она еще раз «льет воду на мельницу» тех, кто в шутку или всерьез утверждает, что есть «ложь, наглая ложь и… статистика».

Теперь для обработки результатов попробуем использовать линейную аппроксимацию, т. е. провести некоторую прямую линию вблизи точек. В статистике такая аппроксимация, построенная в результате применения определенного способа оптимизации, называется линейной регрессией. Для поиска линейной регрессии нужно построить диаграмму рассеяния, т. е. на декартову плоскость «высыпать» все 50 точек со статистическими данными студентов, а затем постараться как-то провести прямую линию, отображающую взаимозависимость (корреляцию) роста и веса человека, не противоречащую основным тенденциям поведения измеренных величин — см. рис. 14.5.

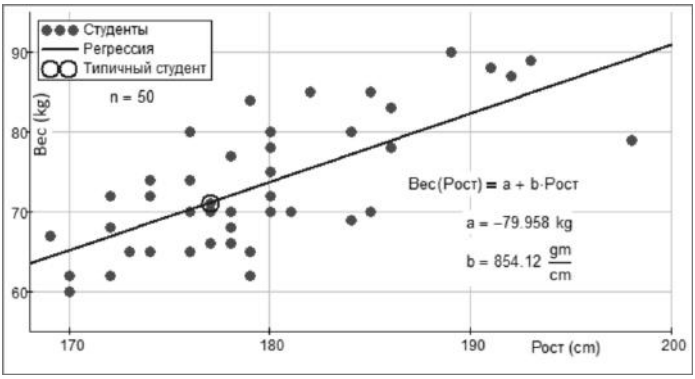


Рис.14.5
Линейное сглаживание (аппроксимация)

Когда-то давно в докалькуляторную и докомпьютерную эру такая работа делалась примерно так: на миллиметровой бумаге проставлялись точки исходных статистических данных, а затем с помощью прозрачной линейки выставлялась некая прямая линия, наилучшим (оптимальным) образом описывающая связь роста человека и его веса. При этом каждый проводящий такую линию мог руководствоваться какими-то своими субъективными понятиями об оптимальности положения этой прямой на плоскости, то есть о значениях коэффициентов `a` и `b`, задающих уравнение прямой линии `y = a + b · x` (константы `a` и `b`, кстати, рассчитаны и прописаны на рис. 14.2 и 14.5). На компьютере эту работу можно автоматизировать, если выбрать какой-то объективный критерий оптимизации. Одним из таких критериев является критерий минимума суммы квадратов «вертикальных» отклонений точек от кривой (в нашем случае от прямой линии). Сам же метод такого расчета называется *методом наименьших квадратов* (МНК). Если возникнет спор о том, чья прямая линия, проведенная в массиве точек, более правильная, то можно циркулем и линейкой замерить отклонения по вертикали точек от прямой, возвести каждое такое отклонение в квадрат, просуммировать эти квадраты и сравнить полученные суммы. У кого эта сумма окажется меньше, у того и прямая линия правильнее иллюстрирует исследуемую статистическую зависимость. Если у нас есть только две точки или точек больше двух, но все они лежат на одной прямой, то, очевидно, сумма квадратов отклонений будет равна нулю. Но если же не все точки лежат на одной прямой (рис. 14.5), то можно предположить, что есть такое положение сглаживающей прямой (такие значения коэффициентов `a` и `b`), при котором сумма квадратов отклонений будет *минимальной*. На рис. 14.6 показан Mathcad-документ, решающий эту задачу оптимизации.

$$CKO(a,b) := \sum_{i=1}^{n} \left[Вес_i - \left(a + b \cdot Рост_i\right)\right]^2$$

$$a := -100kg \qquad b := 1\,\frac{kg}{cm} \qquad CKO(a,b) = 3396\ kg^2$$

$$\begin{pmatrix} a \\ b \end{pmatrix} := Minimize(CKO, a, b)$$

$$a = -79.958\ kg \qquad b = 854.12\,\frac{gm}{cm} \qquad CKO(a,b) = 1505\ kg^2$$

Рис. 14.6

Метод наименьших квадратов как задача оптимизации

В Mathcad-документе, показанном на рис. 14.6, формируется функция пользователя с именем `CKO` (сумма квадратов отклонений — целевая функция оптимизации), имеющая два аргумента — `a` и `b`. Далее задаются начальные значения этих переменных оптимизации с опорой на известную эмпирическую формулу, гласящую, что рост взрослого человека в сантиметрах — это его вес в килограммах плюс сто (см. рис. 1.2 на стр. 16). Наша задача — проверить эту гипотезу на данной *статистической выборке* — на 50 студентах. Затем в расчете на рис. 14.6 вызывается встроенная в Mathcad функция `Minimize`, которая начинает менять заданные нами значения переменных `a` и `b` так, чтобы функция `CKO` приняла минимальное значение. Как это делается — это отдельный разговор. Мы же видим, что при начальных значениях коэффициентов `a` и `b` функция `CKO` возвращала значение 3396 kg^2, а при оптимальных значениях — 1505 kg^2, и верим, что меньшего значения в пределах заданной точности найти не удаться. На рис. 14.5 можно видеть эти оптимальные значения коэффициентов `a` и `b` и «оптимальную» прямую линию, «пронизывающую» массив точек. К оптимальности этого решения мы еще вернемся ниже.

Задачу, сформулированную на рис. 14.6, можно решить и аналитически, если вспомнить, что у дифференцируемой функции двух аргументов в точке минимума частные производные по обоим аргументам равны нулю (условие необходимое, но не достаточное). На рисунке 14.7 показано символьное решение системы двух уравнений равенства нулю частных производных функции `CKO (a, b)` по аргументам `a` и `b`.

$$CKO(a,b) := \sum_{i=1}^{n}\left(Y_i - \left(a + b \cdot X_i\right)\right)^2$$

$$\begin{bmatrix} \frac{d}{da}CKO(a,b) \\ \frac{d}{db}CKO(a,b) \end{bmatrix} \xrightarrow{solve,\begin{bmatrix} a \\ b \end{bmatrix}} \begin{bmatrix} \dfrac{\left(\sum_{i=1}^{n} Y_i\right)\cdot\sum_{i=1}^{n} X_i^2 - \left(\sum_{i=1}^{n} X_i\right)\cdot\sum_{i=1}^{n} X_i \cdot Y_i}{n\cdot\sum_{i=1}^{n} X_i^2 - \left(\sum_{i=1}^{n} X_i\right)^2} & -\dfrac{\left(\sum_{i=1}^{n} X_i\right)\cdot\sum_{i=1}^{n} Y_i - n\cdot\sum_{i=1}^{n} X_i \cdot Y_i}{n\cdot\sum_{i=1}^{n} X_i^2 - \left(\sum_{i=1}^{n} X_i\right)^2} \end{bmatrix}$$

$$a := \frac{\left(\sum_{i=1}^{n} Y_i\right)\cdot\sum_{i=1}^{n} X_i^2 - \left(\sum_{i=1}^{n} X_i\right)\cdot\sum_{i=1}^{n} X_i \cdot Y_i}{n\cdot\sum_{i=1}^{n} X_i^2 - \left(\sum_{i=1}^{n} X_i\right)^2} \qquad b := \frac{n\cdot\sum_{i=1}^{n} X_i \cdot Y_i - \left(\sum_{i=1}^{n} X_i\right)\cdot\sum_{i=1}^{n} Y_i}{n\cdot\sum_{i=1}^{n} X_i^2 - \left(\sum_{i=1}^{n} X_i\right)^2}$$

Рис. 14.7

Символьное решение задачи оптимизации — поиска минимума функции двух аргументов

Формулы для констант a и b, выведенные на рис. 14.17, пригодятся для работы в среде Mathcad Express, где функции intercept и slope заглушены.

У функции СКО суммируются *квадраты* «вертикальных»[4] отклонений точки от прямой линии. И это понятно: если убрать квадрат, то суммирование приведет к некорректной постановке задачи: будут суммироваться отклонения с разными знаками. А можно ли возведение в квадрат заменить взятием абсолютного значения? Для этого достаточно внести соответствующее изменение в первый оператор расчета, показанного на рис. 14.6. Но в этом случае функция Minimize будет иметь плохую сходимость и выдавать совершенно разные ответы при разных начальных значениях переменных a и b. Дело в том, что функция абсолютного значения угловатая, если так можно выразиться. А природа не любит острых углов. Кроме того, стоит взглянуть на рис. 14.8, где сделана неудачная попытка аналитического решения системы двух уравнений — равенство нулю частных производных не от функции СКО (a, b), а от функции СМО(a, b): **С**умма **М**одулей **О**тклонений. Дело в том, что функция $y = |x - x_0|$ не имеет производной в точке x_0, тем же свойством обладают и любые другие функции, содержащие модуль. Помимо этого, сама операция численного дифференцирования является некорректной, т. е. может выдавать не соответствующие действительности результаты.

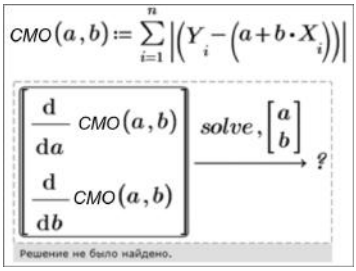


Рис. 14.8
Неудачная попытка решения системы двух уравнений

На сайте **http://www.math.wpi.edu/Course_Materials/SAS/lablets/7.3/7.3c/index.html** можно видеть описание и сравнение методов наименьших квадратов и наименьших модулей.

Одна точка на рис. 14.5 обведена кружочком — это, можно сказать, некий типичный студент, отклонение веса которого от средней линии (стандарта) минимально.

В связи с этим возникает интересное предложение по подведению итогов различных конкурсов красоты. Сейчас в них слишком много субъективности, а значит — обид, слез и даже судебных тяжб. В финалы таких конкурсов обычно попадают «красавицы-раскрасавицы» (см., например, рис. 14.1), из которых довольно трудно выбрать самую оптимальную, пардон, самую красивую мисс или миссис. Так вот, можно у этих финалисток замерить вес и рост или другие размеры (пресловутые 90–60–90, например), провести через точки линию и выбрать победительницу так, как это показано кружочком на рис. 14.5.

Но вернемся к нашим более серьезным делам. Встроенная в Mathcad функция Minimize, показанная на рис. 14.6, позволяет нам реализовать метод наименьших квадратов в его общей постановке: пользовательская (целевая) функция с именем СКО в принципе может иметь любое число аргументов, а ее правая часть — любую функциональную зависимость для сглаживания точек. Но для частных случаев в

[4] Можно попробовать просуммировать квадраты «горизонтальных» отклонений — значений не масс студентов, а их роста. С «перпендикулярными» отклонениями поработать не удастся, так как наши два вектора с данными студентов имеют разные размерности — масса и длина. Но «вертикальные» отклонения тут более уместны — рост человека (аргумент) влияет на его вес (функция), а не наоборот.

среде Mathcad есть более простые инструменты решения задачи статистической обработки данных методом наименьших квадратов. Так, на рисунке 14.9 показан вызов двух встроенных в Mathcad функций `intercept` и `slope`, возвращающих значение коэффициентов `a` и `b` сглаживающей прямой `y = a + b · x`. В эти две функции заложены формулы, выведенные нами на рис. 14.7. Они также использованы в расчете на рис. 14.9. Кроме того, на рисунке 14.9 показана работа еще одной функции, возвращающей коэффициенты линейной регрессии, — функции `line`. Английские слова *an intercept* и *a slope* можно перевести как «пересечение» и «наклон»: значение коэффициента `a` — это значение абсциссы при пересечении нашей прямой с осью `Y`, а значение константы `b` — это тангенс угла наклона прямой к оси `X`. Вернее так. Про тангенс и угол можно было бы упомянуть, если бы наши исходные векторы были безразмерными. При размерных векторах (масса и длина, как в нашей задаче) это просто отношение приращения по вертикали (в граммах) к приращению по горизонтали (в сантиметрах).

$$a := \mathrm{intercept}(Pocm, Bec) = -79.958\ kg$$

$$b := \mathrm{slope}(Pocm, Bec) = 854.12\ \frac{gm}{cm}$$

или $\begin{bmatrix} a \\ b \end{bmatrix} := \mathrm{line}(Pocm, Bec)$

$$a = -79.958\ kg \qquad b = 854.12\ \frac{gm}{cm}$$

или $n = 50$

$$a := \frac{\left(\sum_{i=1}^{n}\left(Bec_i \cdot Pocm_i\right)\right) \cdot \sum_{i=1}^{n} Pocm_i - \left(\sum_{i=1}^{n}\left(Pocm_i\right)^2\right) \cdot \sum_{i=1}^{n} Bec_i}{\left(\sum_{i=1}^{n} Pocm_i\right)^2 - n \cdot \sum_{i=1}^{n}\left(Pocm_i\right)^2} = -79.958\ kg$$

$$b := \frac{\left(\sum_{i=1}^{n} Bec_i\right) \cdot \sum_{i=1}^{n} Pocm_i - n \cdot \sum_{i=1}^{n}\left(Bec_i \cdot Pocm_i\right)}{\left(\sum_{i=1}^{n} Pocm_i\right)^2 - n \cdot \sum_{i=1}^{n}\left(Pocm_i\right)^2} = 854.12\ \frac{gm}{cm}$$

Рис. 14.9
Инструменты пакета Mathcad для решения линейной регрессионной задачи методом наименьших квадратов

Формулы, выведенные для функций `intercept` и `slope` (рис. 14.7 и 14.9), можно, повторяем, использовать в бесплатном пакете Mathcad Express, где сами эти две встроенные функции заглушены.

Через точки, показанные на рис. 14.5, можно провести только прямую линию. Но если исследователь видит в исходных разбросанных точках не прямую, а, например, параболу или полином более высокой степени с коэффициентами `a`, `b`, `c`, `d` и т. д., то он может вызвать Mathcad-функцию `regress`, у которой уже не два (см. рис. 14.9), а три аргумента: вектор ординат точек, вектор абсцисс точек и степень полинома: 1 (наш случай, показанный на рис. 14.5, 14.7, 14.8 и 14.9), 2 (парабола), 3 — полином третьей степени, 4 — четвертой (см. рис. 14.10 ниже) и т. д. до того момента, когда степень полинома станет равна числу точек минус единица.

Если степень полинома равна числу точек минус единица, то аппроксимация (проведение кривой вблизи точек) переходит в интерполяцию (проведение кривой через точки). Но тут не так все просто. Авторская анимация на сайте **http://communities.ptc.com/videos/1473** (рис. 14.10) показывает, как меняется форма аппроксимирующей кривой вблизи 15 точек и как добавляются новые коэффициенты k_i при изменении степени полинома от 1 до 14.

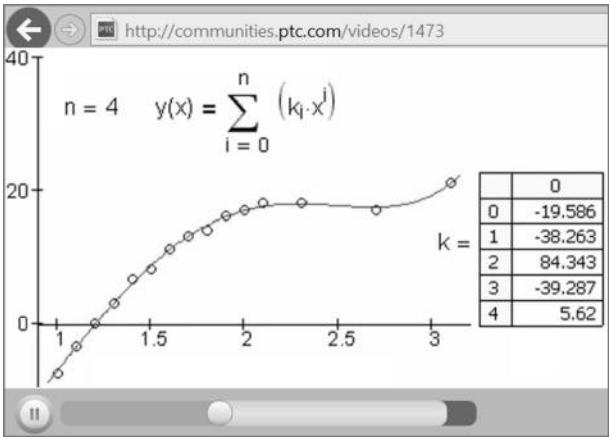


Рис. 14.10

Кадр анимация работы функции ***regress****: через 15 точек проводится линия полинома 4-й степени*

При приближении значения степени регрессионного полинома к значению числа заданных точек (рис. 14.11) может наблюдаться, так называемая, *осцилляция* (от лат. ōscillātio — качание, раскачивание). Видно, как кривая регрессии приближается к точкам (при `n = 14`, повторяем, аппроксимация переходит в интерполяцию), но при этом в промежутках между точками кривая может «раскачиваться» — далеко уходить от точек, сводя на нет всю интерполяцию и аппроксимацию.

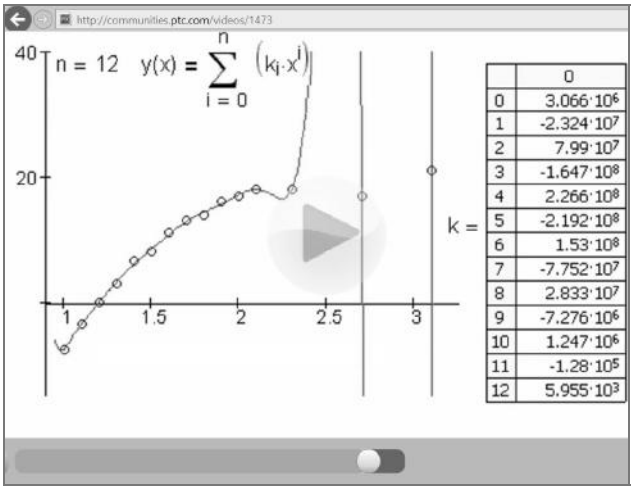


Рис. 14.11

Кадр анимация работы функции ***regress****: через 15 точек проводится линия (часть ее обрезана) полинома 12-й степени*

В интервалах между тремя правыми точками на рис. 14.11 из-за осцилляции мы видим не «статистику» и не просто «ложь», а «наглую ложь» (см. второе название этюда).

Функцию `regress` можно применять не только к линии, но и к поверхности, обрабатывая два вектора (вернее, матрицу с двумя столбцами) и одну квадратную матрицу. Подробнее об этом рассказано в [2].

Но вернемся к терминам, описанными нами в самом начале этюда. Взаимозависимость роста и веса человека — это типичный пример *корреляции*, а наш поиск значений коэффициентов a и b линейной *функции* — классический пример *регрессионного анализа*.

В традиционной математике, как правило, мы имеем дело не с корреляцией, а с четкими функциональными зависимостями. Каждому значению плоского угла, например, соответствует четкое значение синуса, а каждому значению радиуса окружности — значение ее длины и т. д. Но в реальной жизни все намного сложнее. Как зависит теплопроводность металла, о которой мы упоминали выше, от температуры? С одной стороны — это четкая функциональная зависимость, протабулированная во многих справочниках[5] по свойствам веществ. Но с другой стороны, на это физическое свойство металла могут влиять и некие неучтенные факторы — незначительные примеси, форма образца при замере теплопроводности, возраст («усталость») металла и т. д. и т. п. На сайте **http://twt.mpei.ac.ru/MCS/Worksheets/Thermal/T-T-2-Tab-3-04.xmcd** можно провести аппроксимацию табличных данных по теплопроводности разных металлов с помощью полинома переменной степени и вычислить искомое значение. Плотность раствора соли также может неоднозначно зависеть от температуры и концентрации: соль может иметь примеси, вода, в которой ее растворили, может быть тоже не совсем чистой и т. д. На сайте **http://twt.mpei.ac.ru/TTHB/1/Water-chem/C-NaCl.html** расположены онлайн-расчеты плотности водного раствора этой соли в зависимости от температуры и концентрации. Тут уже статистическая обработка ведется не на плоскости, а в пространстве: у соответствующей функции не один, а два аргумента. Технологии получения подобных зависимостей по табличным и прочим данным посвящена первая глава «Свойства рабочих тел и теплоносителей для теплотехнических расчетов» книги [2].

Видя в книге или Интернете таблицу с парами значений, повторяем, не так просто определить, что зафиксировано в таблице — функция или корреляция, и что применить к этим данным — интерполяцию (проведение линии строго через точки — рис. 14.1) или аппроксимацию (сглаживание, проведение линии вблизи точек — рис. 14.5 и 14.10).

Мы уже упоминали линейную интерполяцию — см. рис. 14.2. В среде Mathcad есть встроенная функция `linterp (X, Y, x)`, которая формирует функцию пользователя с аргументом `x` для интерполяции (`interp`) отрезками прямой линии (l — эль) точек с координатами, хранящимися в векторах `X` и `Y`. При такой *кусочно-линейной интерполяции* необходимо, чтобы исходные векторы данных были одного размера, хранили вещественные числа, а в векторе `X` элементы располагались в порядке возрастания. На авторском сайте **http://communities. ptc.com/videos/1512** показана анимация работы функции `linterp` — поведение прямой линии через пару очередных точек при линейной интерполяции данных.

Линейная интерполяция очень проста в реализации. Но «иная простота хуже воровства»: при такой упрощенной интерполяции функция получается угловатой, а природа, повторяем, не терпит острых углов и старается их сгладить. Вспомним, как морские волны превращают острые обломки камней в гладкую гальку. Но главное то, что «ломаная» функция, состоящая из отрезков прямой линии, имеет в качестве первой производной набор констант. Вторая же производная такой функции вообще равна нулю (украдена!) во всех точках, кроме «угловых», в которых вторая производная просто не существует. Это часто затрудняет применение к такой функции некоторых численных методов, например, поиск нуля или минимума — см. пример на рис. 14.8, где была задействована «угловатая» функция.

[5] В справочниках по традиции, идущей еще от докомпьютерных времен, помещаются не формулы для компьютерного расчета, а именно таблицы для ручных расчетов. В современных интернетовских справочниках, которые создают в том числе и авторы этой книги, применяется компромиссный вариант: дается таблица, по которой интерполяцией находится нужное промежуточное значение.

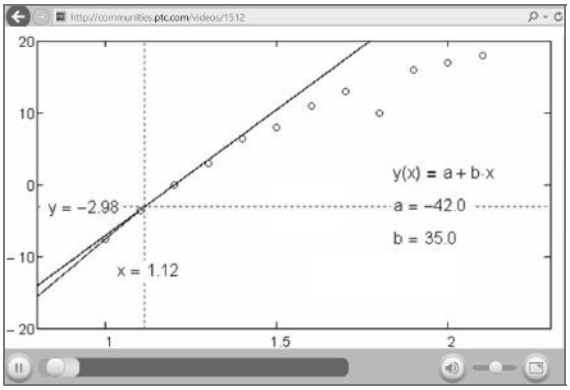


Рис. 14.12
Кадр анимации линейной интерполяции

Если через две точки можно провести прямую линию (см. рис. 14.2 и 14.12), то через четыре точки — кубическую параболу. На этом основана *кубическая* сплайн-интерполяция табличных зависимостей. Она более сложна в компьютерной реализации, чем линейная интерполяция, но лишена «угловатых» недостатков, отмеченных выше. На сайте **http://communities.ptc.com/videos/1418** показана авторская анимация интерполяции кубическими сплайнами. Один кадр этой анимации отображен на рис. 14.13.

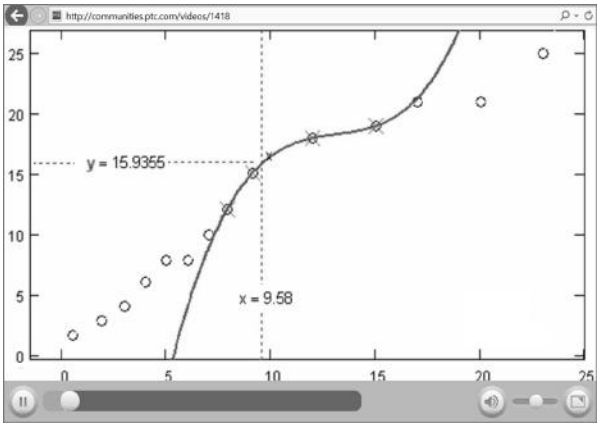


Рис. 14.13
Анимация сплайн-интерполяции

В среде Mathcad сплайн-интерполяцию можно реализовать с помощью функций `lspline`, `pspline` и `cspline`. Префиксы `l`, `p` и `c` отмечают, по какой зависимости будет вестись *экстраполяция* (нахождение значений воссозданной функции вне интервала дискретных значений вектора `X`). А именно: `l` (эль) — линейно с опорой на две крайние точки, `p` (пи) — по параболе с опорой на три крайние точки и `c` (си) — по кубической зависимости с опорой на четыре крайние точки.

Сплайн-интерполяция широко применяется в компьютерной графике, когда, например, необходимо через фиксированные точки какого-либо объекта провести гладкую линию или поверхность. Есть такая технология создания компьютерных мультфильмов. На человеке крепят светящиеся точки, человек двигается, а три координаты этих точек фиксируются компьютером. Затем компьютер проводит

через эти точки криволинейную гладкую поверхность или кусочки поверхности и воссоздает (оживляет, анимирует) движение виртуального человека.

Если интерполяция (от лат. *interpolatio* — изменение, подновление) обозначает нахождение промежуточных значений по ряду статистических данных, а математически выражается в построении функции, проходящей через заданные точки (узлы интерполяции), то экстраполяция (от лат. *extra* — сверх, вне и *polio* — выправляю, изменяю) — это процедура распространения (переноса) на будущее выводов, сделанных относительно какого-либо объекта на основе настоящих и (или) прошлых его состояний. В математике экстраполяция выражается продолжением ряда данных по определенным формулам или построением функции, описывающей поведение объекта в настоящем и предположительно описывающей поведение объекта в будущем.

Экстраполяцию в среде Mathcad можно реализовать и с помощью встроенной функции `predict` (предсказание), в которую заложен очень сложный алгоритм автокорреляции последовательных значений в скользящем окне. На рисунке 14.14 показано, как эта встроенная функция предсказывает значения функции пользователя `y(x)`: на первых точках значения заданной функции довольно хорошо предсказываются, а потом происходит срыв...

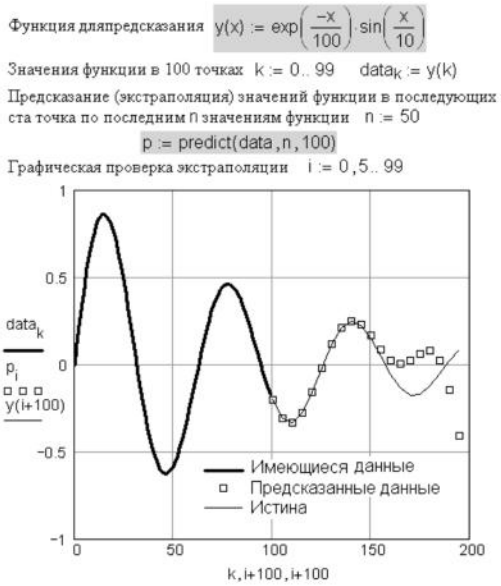


Рис. 14.14
Предсказание синуса в среде Mathcad

Пусть читатель попробует сам с помощью авторского сайта по адресу **http://twt.mpei.ac.ru/mas/worksheets/predict.mcd**, использующего функцию `predict`, предсказать температуру воздуха или курс доллара на ближайшую неделю, опираясь на данные прошедшей недели, месяца, года и т. д. Затем, когда будущее станет прошлым, можно оценить качество прогноза. Кстати о предсказании, вернее, о предсказателях. В СМИ нередко можно увидеть рекламу разного рода предсказателей, обычных и потомственных. Так вот, этим людями можно порекомендовать для начала предсказывать... курс доллара или евро, акций компаний и зарабатывать на операциях с ценными бумагами. Но что-то не слышно, чтобы какой-то предсказатель обогатился таким образом. На рисунке 14.15 показано, как функция `predict`, используя данные по курсу доллара к рублю за июнь 2015 г., «предсказывает» курс доллара на июль 2015 года.

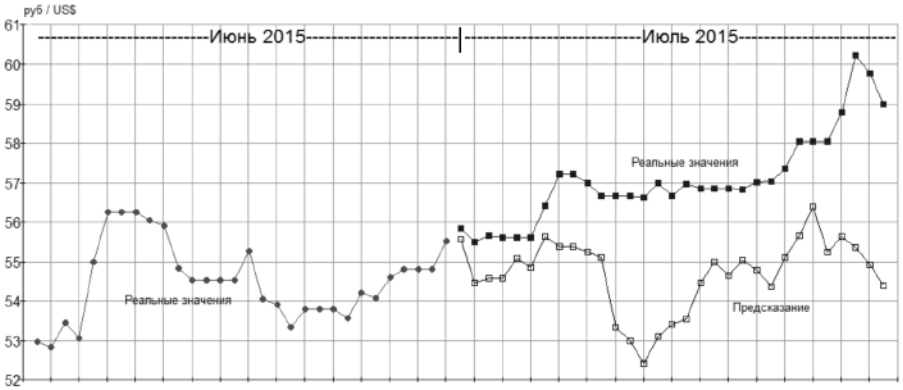


Рис. 14.15
Предсказание курса доллара в среде Mathcad

Хорошо бы публиковать в СМИ подобные кривые с данными предсказания и реальными данными для температуры воздуха, например. По ним можно было бы оценивать качество работы различных бюро прогнозов.

В среде Mathcad есть много функций, позволяющих обрабатывать дискретные случайные или закономерные данные — сглаживать их (аппроксимировать), строить по ним интерполяцию или экстраполяцию. На рисунке 14.16 показан сайт Интернета, созданный с помощью пакета Mathcad, на котором проводится аппроксимация табличных данных с помощью произвольной функции трех аргументов. На сайте показана работа с так называемой *логистической*[6] функцией. С ее помощью можно описать три основные фазы изменения многих объектов и явлений живой и неживой природы: возникновение, бурное развитие и умирание (затухание). Конкретный пример: количество паровозов[7] в мире сначала росло медленно, затем наблюдался бурный рост их производства, а потом их постепенно вытеснили тепловозы и электровозы...

В текстовое окно с меткой `f(x, a, b, c)`, показанное на рисунке 14.16, можно вставлять аппроксимирующие функции другого вида с не более чем тремя параметрами. В сам же пакет Mathcad встроены следующие функции с постфиксом `fit` (fitting[8]), предназначенные для такой работы:

`expfit (vx, vy, [vg]`[9]`)` — аппроксимация по формуле (зависимости) $a \cdot e^{bx} + c$ (мы эту функции позже будем использовать для решения конкретной практической задачи — см. рис. 14.25 на стр. 328);

[6] Логистическое уравнение, также известное, как уравнение Ферхюльста (по имени впервые сформулировавшего его бельгийского математика), изначально появилось при рассмотрении модели роста численности населения.

[7] Паровую машину мы анимировали в этюде 8.

[8] Этот суффикс или корень, кстати, присутствует в названии спортивно-развлекательного заведения Fitness Club — там посетители «сглаживают» свою фигуру (см. также сноску 3 на стр. 125).

[9] Аргумент `vg` — это вектор первых предположений при решении данной задачи регрессии. Векторы `vx` и `vy` хранят координаты узлов анализируемой выборки. Квадратные скобки означают необязательность данного параметра.

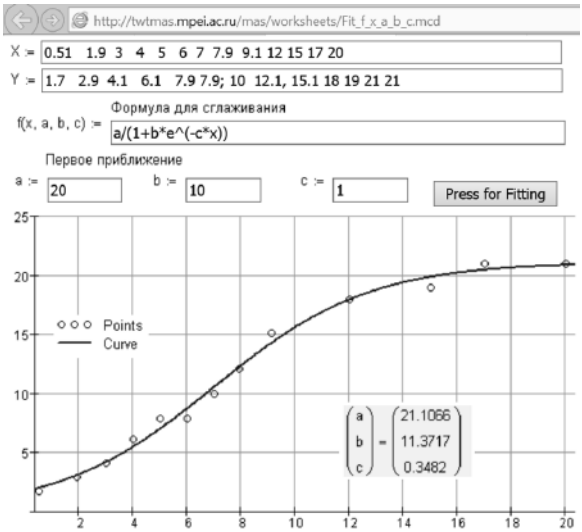


Рис. 14.16

Сглаживание табличной зависимости по произвольной формуле

`lgsfit (vx, vy, vg)` — $1 + b \cdot e^{-cx}$;

`lnfit (vx, vy)` — $a \cdot \ln(x) + b$;

`logfit (vx, vy, vg)` — $a \cdot \ln(x + b) + c$;

`medfit (vx, vy)` — $a + b \cdot x$ (альтернативный алгоритм, реализующий не минимизацию суммы квадратов отклонений (см. рис. 14.6 на стр. 313), а *медиан-медианную линейную регрессию* для расчета коэффициентов `a` и `b` (см. рис. 14.17);

`pwrfit (vx, vy, vg)` — $a \cdot x^b + c$;

`sinfit (vx, vy, vg)` — $a \cdot \sin(x + b) + c$;

`linfit (vx, vy, F)` — линейная (lin) комбинация элементарных функций;

`genfit (vx, vy, vg, F)` — регрессионный анализ в общем (gen–general) виде.

Из этого списка «фитинговых» функций несколько выпадает функция `medfit`, в имени которой сокращение med означает не *вид* аппроксимирующей функции, как у других функций в списке, а *метод* линейной аппроксимации. Функция `medfit` как и функция `line` (см. рис. 14.9), предназначена для решения задачи линейной регрессии. Но в функцию `medfit` заложен не метод наименьших квадратов, а метод медиан-медианной регрессии с минимизацией суммы абсолютных значений *медиан* ошибок в узлах[10]. На рисунке 14.17 показано сравнение регрессионного анализа параметров наших студентов, проведенного методом наименьших квадратов (более пологая прямая линия) и медиан-медианным методом.

Какая прямая на рис. 14.17 ближе к истине? Или спросим несколько иначе — в какой прямой меньше лжи (см. второе название этюда)? Ответ на этот вопрос[11], если, конечно, он вообще существует, зависит, в частности, от того, какое *распределение* заложено в анализируемые случайные величины. В среде Mathcad, кстати

[10] Медианой в статистике называют средний показатель в ряду чисел. Медиана для ряда из N чисел находится просто. Нужно поставить числа в возрастающем порядке. Если N является нечетным числом, то медиана равна центральному числу. Если N является четным числом, то медиана равна среднему арифметическому чисел центральной пары. Например, для ряда 1, 2, 4, 4, 6, 9, 12, 21 медиана равна 5 ((4+6)/2).

[11] Наша идея объективного подведения итогов конкурса красоты (см. выше) тут может «споткнуться» о споры о том, какой метод регрессионного анализа тут применять. Можно предложить победительницу конкурса определять одним методом (по одной прямой), а вице-победительницу — другим.

говоря, есть набор встроенных функций для генерации случайных величин по разным законам распределения.

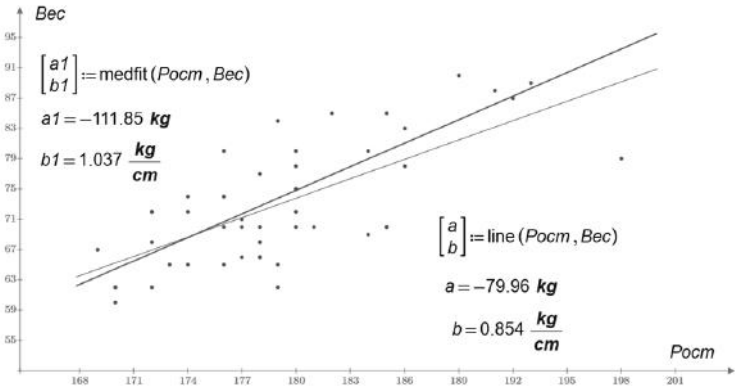


Рис. 14.17

*Сравнение результатов обработки данных медиан-медианной регрессией (использована функция **medfit**) и методом наименьших квадратов (**line**)*

Но нередки случаи, когда не наблюдается никакой корреляции данных, например, так, как это выявлено при одном исследовании статистической зависимости между баллами ЕГЭ по математике и результатами контрольной работы [70]. Диаграмма рассеяния (рис. 14.18) ясно дает понять, что остаточный уровень школьных математических знаний слабо зависит от полученных на ЕГЭ баллов, причем дальнейший регрессионный анализ данных не имеет смысла, поскольку регрессионная прямая будет расположена практически горизонтально.

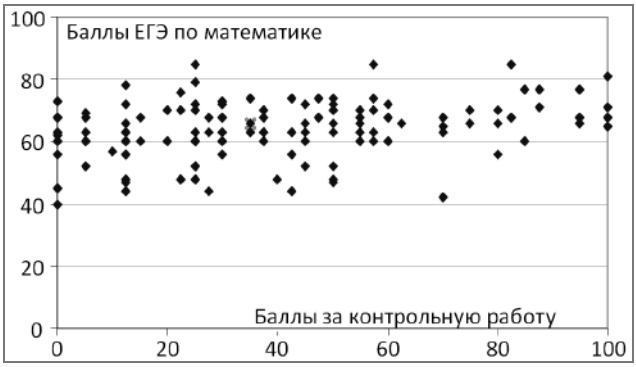


Рис. 14.18

Диаграмма рассеяния между баллами за контрольную работу и баллами ЕГЭ по математике

С другой стороны, предварительный анализ данных может помочь выявить и слабые корреляции. Так, например, была обнаружена, хотя и незначительная, но отрицательная корреляционная связь между временем, проводимым студентами за компьютером в целях развлечений, и успешностью сдачи ЕГЭ по математике [71]. Для студентов из небольших городов она равна –0.04, а для студентов из крупных городов она равна –0.25.

Мы видим, что, приступая к статистической обработке случайных величин, желательно предварительно построить графики и диаграммы для визуального анализа данных. Так, встроенная в Mathcad функция `histogram` позволяет построить *частотную характеристику* (гистограмму) выборки. На рисунке 14.19 гра-

фически показана частота появления того или иного значения роста у нашей полусотни студентов с точностью в один сантиметр.

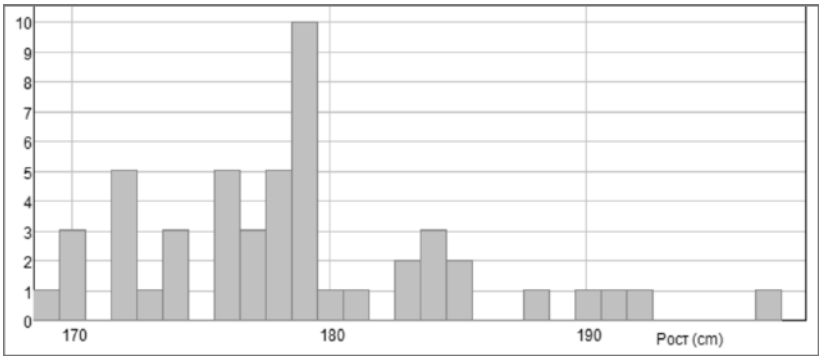


Рис. 14.19
Пример гистограммы в среде Mathcad

Так из гистограммы на рис. 14.17 видно, что больше всего студентов (10 человек) имеют рост 179 см. Но какую-то статистическую закономерность на рис. 14.17 увидеть довольно сложно. Если бы наша выборка студентов была бы намного больше (студенты всего вуза или всей страны), то мы бы увидели, что на гистограмме четко прорисовалась бы некая колоколообразная кривая — кривая *нормального распределения*: очень мало людей имеют очень низкий или очень высокий рост, а большое количество людей имеют некий средний (среднестатистический) рост.

Из рисунка 14.19 видно, что наша гистограмма немного сдвинута относительно нормального распределения. Но вспомним в который раз второе название нашего этюда: «Ложь, наглая ложь и статистика». В этом изречении скрыт и такой нюанс. Люди при опросах часто вольно или невольно слегка искажают информацию о себе. Можно предположить, что некоторые студенты, называя вслух свой рост, прибавляли к нему 1, 2 или даже 3 см. Скажем несколько мягче. Низкие студенты округляли свой рост до сантиметра в большую сторону (было, например, 156.3 cm — стало 157 см), а высокие проводили округление более правильно (было, например, 179.3 cm — стало 179 см). В таких опросах (при переписи населения, например) часто лукавят женщины: убавляют себе возраст и/или вес, приписывают себе несуществующего мужа и т. д.

В Интернете можно найти не очень качественную, но очень интересную фотографию «живой» гистограммы роста людей — рис. 14.20.

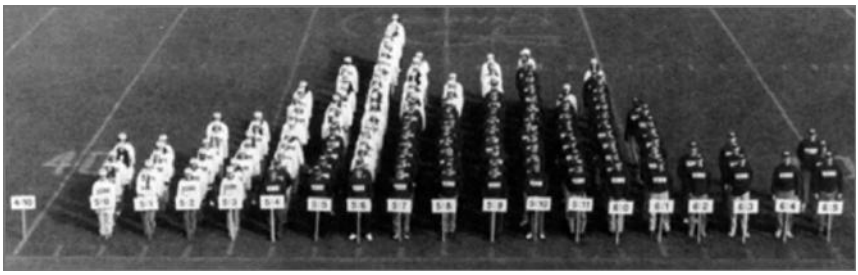


Рис. 14.20
Живая гистограмма роста человека

Группа игроков выстроилась на футбольном поле по росту, но не в одну шеренгу, а в колонну с разным числом людей в каждом ряду. Каждый ряд соответство-

вал определенному росту: пять футов (152.4 см), пять футов и один дюйм (154,94 см) и так далее до 6 футов и 5 дюймов (195,58 см). В таком «опросе» схитрить при ответе о собственном росте будет довольно сложно: завышение этого параметра на 2–3 см будет сразу заметно в соответствующем ряду, если только не встать на цыпочки. В диаграмме, показанной на рис. 14.21, в середине нет пустых рядов, какие мы наблюдали на рис. 14.19, потому что шаги на этих гистограммах разные — один сантиметр и один дюйм (2,52 см).

Если в США и Англии рост человека измеряют в футах и дюймах[12], то в старой России для этого брали аршины и вершки[13], но аршины (конкретно *два аршина*) по умолчанию не указывали. Дело в том, что рост взрослого человека очень редко был меньше двух аршин (примерно 142 см) или больше сажени (трех аршин — примерно 213 см). Читаем (и запомним — см. ниже) у Тургенева в повести «Муму»: «Из числа всей ее челяди самым замечательным лицом был дворник Герасим, мужчина двенадцати вершков роста, сложенный богатырем и глухонемой от рожденья». Несложно подсчитать «богатырский» рост Герасима с учетом недосказанных двух вершков — примерно 196 см (см. рис. 1.2 на стр. 16). На рисунке 14.21 построена гистограмма роста «авторских» студентов (см. рис. 14.5) с шагом не в один сантиметр или дюйм, а в один «старый добрый» русский вершок.

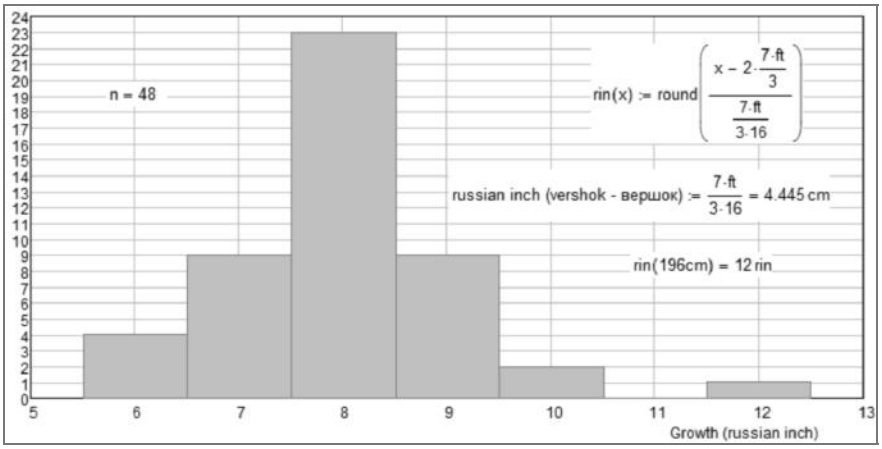


Рис. 14.21
«Вершковая» гистограмма роста студентов

На «вершковой» гистограмме (рис. 14.21) четко просматривается некий «статистический колокол» — кривая нормального распределения. Студенты по росту разбиваются на 5 условных групп: низкого роста (4 человека с шестивершковым ростом), скорее низкого, чем среднего роста (9 человек — 7 вершков), среднего роста (23 человека — 8 вершков), скорее высокого, чем среднего роста (9 человек — 9 вершков) и высокого роста (2 человека, 10 вершков). Один студент выпал из нашей «статистики» и оказался с ростом тургеневского Герасима в 12 вершков: исключение только подтверждает наше статистическое правило. Сравнивая ри-

[12] В одном из романов Агаты Кристи можно прочесть, что рост Эркуля Пуаро равнялся пяти футам и пяти дюймам. Такой же рост имел предок Форсайтов, описанный в знаменитой «Саге» Джона Голсуорси. Знание старинных и иностранных единиц измерения помогает почувствовать «дух» классической литературы. Это мы особо подчеркнули в этюде 6.

[13] Размер сажени долгое время был переменной величиной, но к середине XIX в. русскую сажень приравняли к семи английским футам (Семь футов под килем!). В сажени 3 аршина, а в аршине 16 вершков (недесятичная система исчислении). «Прелести» такой системы можно сполна почувствовать, читая книг Сергея Александровича Рачинского «1001 задача для умственного счета» — см. этюд 6.

сунки 14.19 и 14.21, можно сказать словами персонажа одного мультфильма: «А в вершках-то наша гистограмма гораздо красивее и статистичнее!» Кстати, здесь мы невольно затронули важнейший для обработки данных вопрос об установлении оптимального шага при работе с выборками. Ведь можно было бы группировать данные каким-то иным способом. Например, по признаку попадания значения роста на интервалы длиной в 1 метр, начиная с нуля (тогда гистограмма имела бы явно ложный вид прямоугольника).

На связь роста и веса человека влияет много факторов: на то она и корреляция, а не функция. Один из важных факторов — это возраст человека. Немногим людям удается сохранить свой вес таким, каким он был в молодости[14]. На вес человека также большое влияние оказывает его генетика (наследственность) и, конечно, образ жизни. У человека, как и у автомобиля (см. ниже) возраст определить несложно. А вот «пробег» человека, конкретно говоря, число шагов, которые он сделал за свою жизнь, тоже хорошо было бы измерять для более точного определения взаимозависимости веса и роста. Пробег замеряет спидометр автомобиля. Некоторые люди на прогулках или пробежках используют шагомер. Если читатель перейдет по ссылке, отмеченный в списке литературы [69], то он может узнать, как в среде Mathcad проводился более сложный регрессионный анализ — как определялась формула, по которой можно оценить стоимость подержанного автомобиля в зависимости от его возраста и пробега.

Автору тут вспоминается одна история, свидетелем которой он был в Западной Германии в 80-х годах прошлого века. Один немецкий профессор (автор был на стажировке в одном немецком университете) захотел быстро продать свой автомобиль за 2000 марок и выставил его по этой низкой цене на местном автомобильном «блошином» рынке. К нему довольно долго подходили потенциальные покупатели, осматривали машину, задавали вопросы, но все эти контакты оказывались безрезультатными. Но тут к ему подходит один довольно бойкий человек, покупает, как говорится, не глядя машину, меняет на ней... ценник и тут же на глазах изумленного профессора перепродает ее за 4000 марок... Этот перекупщик умел сходу правильно оценить стоимость подержанного автомобиля и понимал, что заниженная цена также плоха для торга, как и завышенная. Проецируя этот тезис на наших студентов, можно сказать, что слишком низкий рост также плох для жизни, как и слишком высокий...

Итак, автомобиль с его возрастом и пробегом!

Когда-то давно посетитель форума пакета Mathcad поместил на нем таблицу 14.1 и попросил форумчан подсказать, как из таблицы можно получить расчетную формулу для оценки стоимости подержанного автомобиля по его пробегу и возрасту.

Попытаемся сделать такой регрессионный анализ.

Таблица 14.1 перенесена в документ Mathcad Prime в виде трех векторов с размерностями времени, длины и стоимости, объединенных в таблицу — см. рис. 14.22.

Статистический анализ данных принято начинать с построения различного рода графических зависимостей, которые могут подсказать возможные пути обработки данных. На рисунках 14.23–14.25 можно видеть графики, построенные по точкам из «автомобильной» таблицы 14.1.

[14] «Когда хорошее настроение, что-то удалось, нет посторонних и помещение подходящее, хочется раскинуть руки, загудеть и побежать по кругу, изображая самолет на вираже... Давно я не совершал такой «полет», а тут «совершил». Жена посмотрела и сказала, что раньше, в молодости я был легкомоторным самолетом, а теперь стал... широкофюзеляжным» (www.anekdot.ru).

Таблица 14.1

Данные о стоимости подержанного автомобиля определенной марки

Возраст (лет)	Пробег (миль)	Цена (UD$)	Возраст (лет)	Пробег (миль)	Цена (UD$)	Возраст (лет)	Пробег (миль)	Цена (UD$)
11,5	88000	1195	13,5	120000	495	7,5	41000	3495
13,5	103000	750	11,5	101000	895	13,5	124000	850
10,5	82000	1295	3,5	39000	4995	10,5	77000	1595
10,5	65000	1495	10,5	78000	1295	6,5	56000	3495
12,5	97000	800	6,5	52000	2695	12,5	83000	895
10,5	70000	1495	9,5	84000	1995	9,5	67000	2495
8,5	51000	2295	4,5	39000	3995	4,5	38000	3990
10,5	80000	1495	4,5	46000	3675	6,5	43000	3400
9,5	79000	1995	12,5	92000	795	13,5	92000	795
6,5	57000	2695	11,5	108000	975	11,5	78000	1295

Область

$US\$:= ¤$

Возраст (*yr*)	*Пробег* (*mi*)	*Цена* (*US$*)
11.5	88000	1195
13.5	103000	750
10.5	82000	1295
10.5	65000	1495
12.5	97000	800
10.5	70000	1495
8.5	51000	2295

Рис. 14.22

Таблица Mathcad Prime с данными о подержанных автомобилях

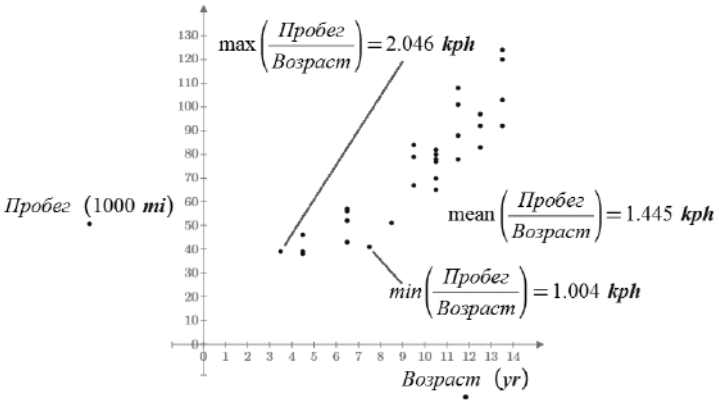


Рис. 14.23

Корреляция пробега автомобиля и его возраста

Если пробег автомобиля разделить на его возраст, то мы получим среднюю скорость автомобиля за время его эксплуатации. К этой дроби можно применить три встроенные функции `max`, `mean` и `min`, возвращающие минимальное, среднее и максимальное значения вектора. Из рисунка 14.23 видно, что корреляция пробега автомобиля и его возраста сосредоточена у прямой линии с наклоном 1.445 км/ч, проходящей через начало координат. В нашей выборке минимальная и максимальная скорости такие: 1.004 км/ч (7.5 лет и 41 000 миль пробега) и 2.046 км/ч (3.5 года и 39 000 миль). Если провести две прямые линии через начало координат и эти две точки, то получится некий «эксплуатационный» клин, внутри которого лежат наши 30 точек выборки. Отсюда вывод — если вам предлагают купить подержанный автомобиль, параметры которого не попадают в этот «клин» (1–2 км/ч), то нужно подумать, прежде чем приобретать такую машину, которую либо слишком интенсивно эксплуатировали (скорость больше 2 км/ч), либо она непонятно почему простаивала (скорость меньше 1 км/ч), либо у нее... подкручен спидометр. Возраст машины скрыть (подделать) намного трудней.

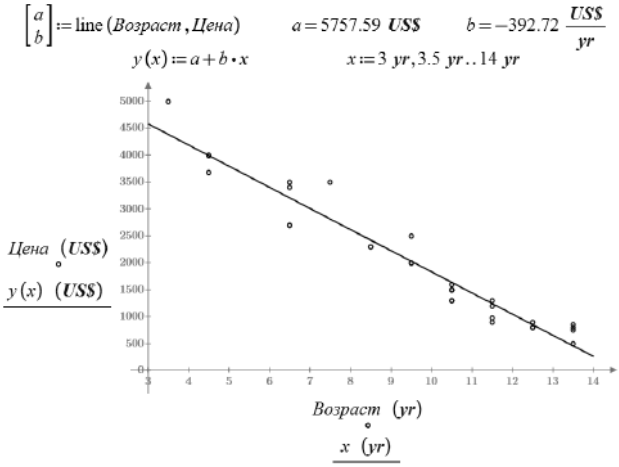


Рис. 14.24
Корреляция цены автомобиля и его возраста

Цена автомобиля от возраста зависит линейно `a + b x` — см. 14.24. Никакой другой более сложной зависимости из точек, разбросанных на рисунке 14.24, выудить невозможно. В нашем случае после каждого года пробега с цены машины нужно будет скидывать в среднем по 393 доллара. Цена же нового автомобиля составляет примерно 5750 долларов.

Зависимость цены автомобиля от пробега (рис. 14.25) более сложная — она меняется экспоненциально: $a \cdot e^{bx} + c$. При такой зависимости `(b < 0)` цена машины с пробегом никогда не упадет до нуля, чего не скажешь о зависимости цены от возраста (рис. 14.24), когда в районе 14–15 лет автомобиль становиться «бесценен» в одном из двух смыслов: либо у него теперь нулевая цена (наш смысл), либо автомобиль стал... «бесценным» музейным экспонатом.

Попробуем теперь создать нашу искомую функцию двух аргументов — возраста и пробега, по которой можно будет оценить стоимость подержанного автомобиля. Ее вид могут подсказать две отдельные функции одного аргумента (см. рис. 14.24 и 14.25). Искомая функция будет иметь четыре параметра (`a`, `b`, `c` и `d` — см. первую строку в расчете на рис. 14.26) и представлять собой комбинацию линейной и экспоненциальной зависимостей. Необходимо только рассчитать значения параметров `a`, `b`, `c` и `d`, при которых точки, зафиксированные в таблице 14.1 и на графиках 14.23–14.25, оказались бы вблизи уже не линии, а поверхности. Для

этого создается вспомогательная функция, возвращающая сумму квадратов отклонений точек от поверхности — см. второй оператор на рис. 14.26. Мы такую операцию уже проводили при регрессионном анализе роста и веса студентов (рис. 14.7).

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} := \mathrm{expfit}\left(Пробег, Цена, \begin{bmatrix} 10000\ US\$ \\ -0.001\ \frac{1}{mi} \\ 100\ US\$ \end{bmatrix}\right) \quad a = 12291.48\ US\$ \quad b = -2.991 \cdot 10^{-5} \frac{1}{mi} \quad c = 293.091\ US\$$$

$$y(x) := a \cdot e^{b \cdot x} + c \qquad x := 30000\ mi, 31000\ mi .. 130000\ mi$$

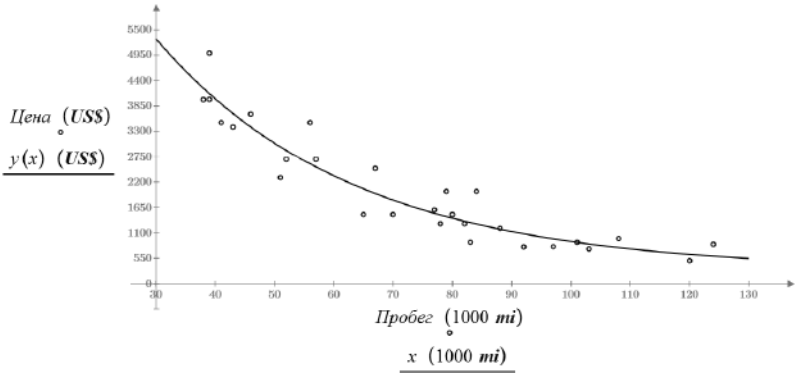


Рис. 14.25
Корреляция цены автомобиля и его пробега

$$z(Возраст, Пробег, a, b, c, d) := a + b \cdot Возраст + c \cdot e^{d \cdot Пробег}$$

$$\Sigma\Delta 2(a, b, c, d) := \sum_{i=1}^{\mathrm{last}(Цена)} \left(Цена_i - z\left(Возраст_i, Пробег_i, a, b, c, d\right)\right)^2$$

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} := \begin{bmatrix} 4000\ US\$ \\ -300\ \frac{US\$}{yr} \\ 2000\ US\$ \\ 0 \cdot \frac{1}{mi} \end{bmatrix} \qquad \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} := \mathbf{minimize}(\Sigma\Delta 2, a, b, c, d)$$

$$a = 4000\ US\$ \qquad b = -300\ \frac{US\$}{yr} \qquad c = 2000\ US\$ \qquad d = 0\ \frac{1}{mi}$$

Рис. 14.26
Неудачная попытка работы с функцией `minimize`

На рисунке 14.26 сделана попытка расчета параметров `a`, `b`, `c` и `d` через поиск минимума суммы квадратов отклонений точек от поверхности. Она оказалась неудачной — функция `minimize` вернула значения первого приближения. Дело в том, что в пакет Mathcad Prime был встроен новый алгоритм работы функции `minimize`. А новое нередко бывает недоработанным. Пришлось обратиться к функции `minerr`, которая уже давно встроена в Mathcad и в которой выявлены возможные ошибки и недоработки — см. рис. 14.27. Мы эту функцию использовали в этюде 4 и показывали, как ею можно заменить функцию `minimize` – см. рис. 4.17 на стр. 75.

Но и с функцией `minerr` пришлось повозится: давать первое предположение, получать ответ, видеть, что это еще не минимум и переносить (копировать) ответ на место первого предположения. Контроль же за качеством минимизации велся через построения четырех декартовых графиков с горизонтальными маркерами,

на которых один из четырех параметров задается переменной диапазона, а три остальные фиксируются как константы — см. рис. 14.28–14.31: в точке минимума функции, как известно, значения всех ее частных производных должны быть равны нулю.

Начальные приближения

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} := \begin{bmatrix} (4.454 \cdot 10^{3})\ ¤ \\ -(9.091 \cdot 10^{-6})\ \frac{¤}{s} \\ (1.255 \cdot 10^{4})\ ¤ \\ -(3.95 \cdot 10^{-8})\ \frac{1}{m} \end{bmatrix}$$

Copy

Ограничения

$$\Sigma\Delta 2(a, b, c, d) = 0\ US\$^{2}$$

Решатель

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} := \mathbf{minerr}(a, b, c, d) = \begin{bmatrix} (4.454 \cdot 10^{3})\ ¤ \\ -9.092 \cdot 10^{-6}\ \frac{¤}{s} \\ (1.256 \cdot 10^{4})\ ¤ \\ -3.95 \cdot 10^{-8}\ \frac{1}{m} \end{bmatrix}$$

$$a = 4453.81\ US\$ \qquad b = -286.91\ \frac{US\$}{yr}$$

$$c = 12555.641\ US\$ \qquad d = -6.358 \cdot 10^{-5}\ \frac{1}{mi}$$

Рис. 14.27
Удачная попытка работы с функцией `minerr`

$$\Delta := 1\% \qquad av := a \cdot (1-\Delta), a \cdot (1-\Delta) + \frac{a \cdot (1-\Delta)}{3000} .. a \cdot (1+\Delta)$$

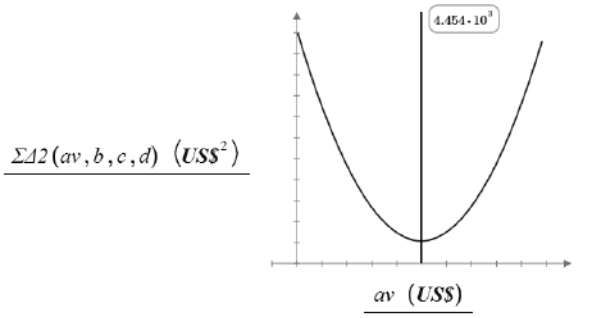


Рис. 14.28
Проверка точки минимума по параметру a

$$bv := b\cdot(1-\Delta), b\cdot(1-\Delta)+\frac{b\cdot(1-\Delta)}{3000}..b\cdot(1+\Delta)$$

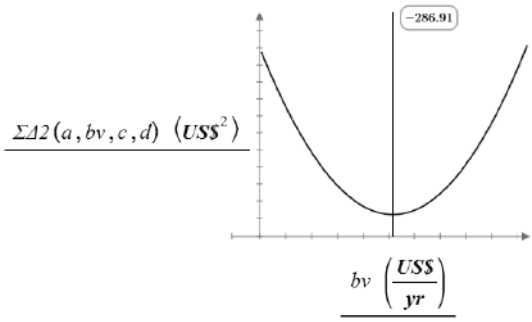


Рис. 14.29
Проверка точки минимума по параметру b

$$cv := c\cdot(1-\Delta), c\cdot(1-\Delta)+\frac{c\cdot(1-\Delta)}{3000}..c\cdot(1+\Delta)$$

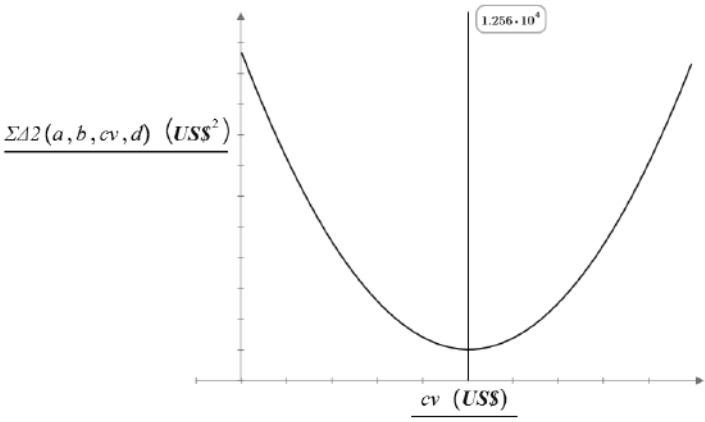


Рис. 14.30
Проверка точки минимума по параметру c

$$dv := d\cdot(1-\Delta), d\cdot(1-\Delta)+\frac{d\cdot(1-\Delta)}{3000}..d\cdot(1+\Delta)$$

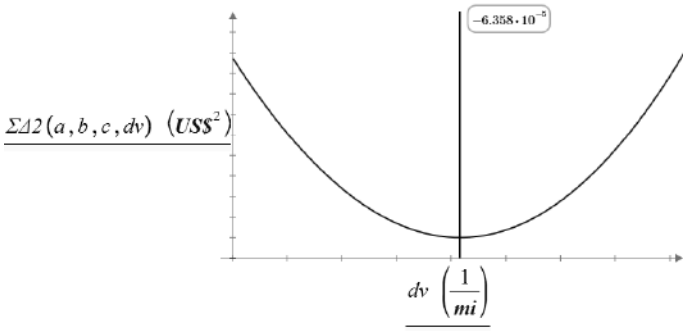


Рис. 14.31
Проверка точки минимума по параметру d

Честно говоря, было желание изъять из книги рисунок 14.26 и не рассказывать об этой недоработке Mathcad Prime. Тем более, следует ожидать, что к моменту выхода книги в свет появится новая версия этого пакета, где эта ошибка будет исправлена. Но рисунок был оставлен в книге и вот почему.

Нередки ситуации, когда пользователь программы видит, что в ней что-то не так и по умолчанию считает себя виновником ошибки, несмотря на то, что все было сделано вроде бы правильно. А ошибка сидит в самой программе. При таких сомнениях следует обратиться на форум Mathcad и узнать мнения других людей. Так было и сделано, и данная ошибка в функции `minimize` была зафиксирована и принята разработчиками. Кроме того, нужно помнить, что многие инструменты Mathcad и других математических программ продублированы. Это позволяет решать задачу разными способами, «уворачиваясь» от возможных недоработок.

Так или иначе, задача об оптимальных значениях параметров `a`, `b`, `c` и `d` регрессионного выражения для цены подержанного автомобиля была решена и найденную формулу можно использовать при оценке машин с пробегом — см. рис. 14.32, где вводятся значения переменных `МойВозраст` и `МойПробег`, рассчитывается цена автомобиля в долларах и в процентах от цены нового с нулевым пробегом и возрастом[15], а также показана данная точка среди точек статистической выборки.

$$МойВозраст := 9\ yr \qquad МойПробег := 110000\ km$$

$$z(МойВозраст, МойПробег, a, b, c, d) = 2034\ US\$$$

$$\frac{z(МойВозраст, МойПробег, a, b, c, d)}{z(0\ yr, 0\ mi, a, b, c, d)} = 12\%$$

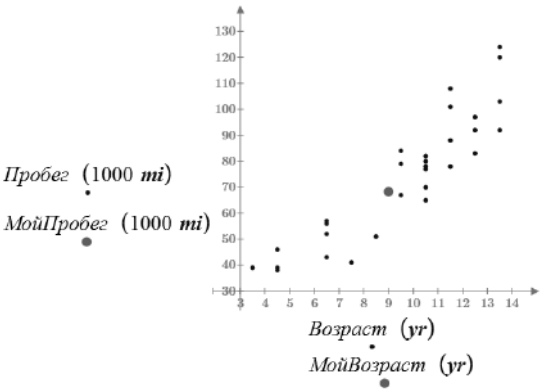


Рис. 14.32
Расчет цены подержанного автомобиля

Еще об одном сомнении. В расчете на рис. 14.27 был найден минимум функции, а на рисунках 14.28–14.31 показана графическая проверка этого решения. Но не ясно, является ли этот минимум глобальным и нет ли других значения параметров `a`, `b`, `c` и `d`, при которых сумма средних квадратичных отклонений будет еще ближе к нулю.

Чтобы развеять сомнения, нужно применить иные методы поиска минимума функции — например, генетический метод, авторский сайт которого показан на рис. 14.33. На этом сайте есть ссылка на описание метода. Этот метод (его реализация для Mathcad сделана В. Р. Сабаниным) позволяет найти глобальный минимум функции в заданной области ее аргументов. Так из рисунка видно, что функ-

[15] Тут нужно еще учесть тот факт, что сразу после продажи реальная цена автомобиля падает на 20–30%.

ция пользователя двух аргументов имеет два локальных минимума и глобальный минимум, который и был найден.

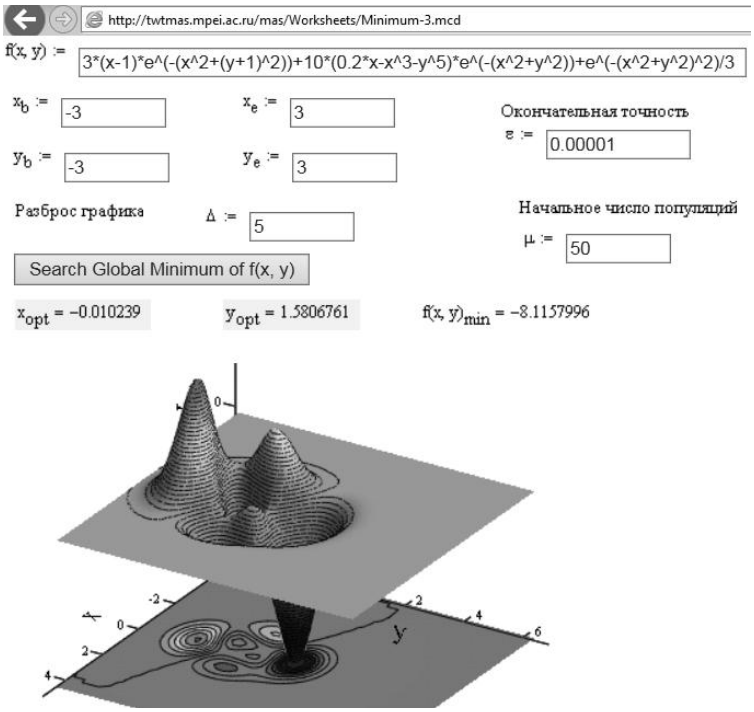


Рис. 14.33

Поиск глобального минимума по генетическому алгоритму

Статистическая обработка данных по цене подержанного автомобиля позволила создать Mathcad-сайт для оценки данной корреляции — см. рис. 14.34.

По графику на рис. 14.34 можно не только получить оценочную стоимость старого автомобиля, но и отсечь заведомо сомнительные варианты, зафиксированные на рис. 14.23 (параметры вне скоростного клина), когда пробег автомобиля не соответствует его возрасту. Mathcad-документ, по которому велись расчеты, можно «скачать» по адресу: **ftp://twt.mpei.ac.ru/ochkov/Auto**.

Один из авторов книги два раза приобретал автомобиль по схеме trade-in, когда в стоимость новой машины входит стоимость старой, сдаваемой на комиссию. Так вот, оба раза цена старого автомобиля, предлагаемая салоном, отличалась не более, чем на 5% от той, какая была рассчитана с помощью сайта, показанного на рис. 14.34.

А можно ли оценить в рублях не подержанный автомобиль, а «подержанного» человека по его возрасту и... пробегу, если под пробегом иметь в виду не число пройденный километров, а его деловые качества, которые приобретаются в результате «пробега»: учебы, работы по специальности, повышения квалификации, самообразования, решения разных жизненных и производственных (офисных) коллизий...?

Кому-то этот вопрос покажется несколько кощунственным — как можно человека измерять рублями, долларами, фунтами? Но тут можно упомянуть футболистов, многие из которых бегают по полю с «ценниками», где указаны суммы, какие нужно уплатить при переводе футболиста в другой клуб. У других специалистов высшей квалификации цена не прописана так явно, но она негласно фигури-

рует в досье и на переговорах по переходу, например, топ-менеджера из одной фирмы в другую.

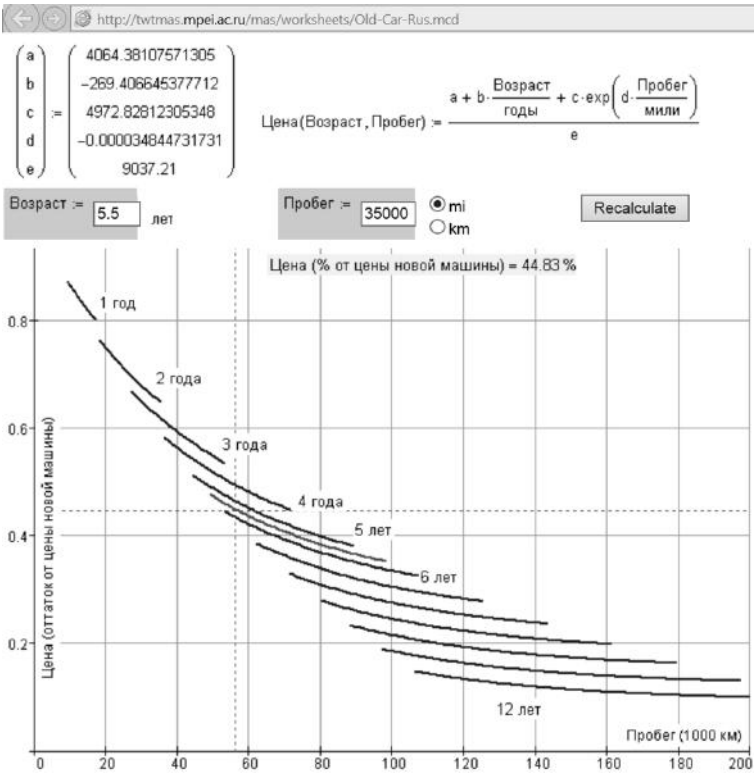


Рис. 14.34
Сайт по оценке стоимости подержанного автомобиля

Если же под стоимостью человека подразумевать его капитал, то тут можно наблюдать разные кривые в координатах «стоимость — возраст».

Одни люди начинают с нуля, а потом создают себе капитал. Другие же растрачивают средства, полученные по наследству, «скатываясь по наклону». Оптимальным же или самым «красивым» считается вариант, когда человек получает какой-то стартовый капитал и приумножает его, оставляя затем наследникам какую-то разумную сумму и потратив львиную долю на благотворительность, основав, например, новый университет или поддержав свою Alma Mater, свой родной город, поселок...

Но если говорить о цене человека, имея в виду не его денежный капитал, а его ценность как специалиста, то тут тоже можно видеть некую горбатую кривую зависимости этого основного параметра человека от его возраста. Пик (максимум) на этой кривой может приходиться на разный возраст у разных специалистов-профессионалов. У футболистов это 25–30 лет, у руководителя компании 40–60 и т. д.

Но вернемся к гистограммам, примеры которых показаны на рис. 14.18–14.20.

Давайте проведем глобальный (мысленный) статистический эксперимент и у всех взрослых людей на планете измерим какой-либо параметр: вес, рост (см. выше), ум и т. д. — словом все то, что можно измерить числом или оценить лингвис-

тическими критериями (гений, талант, очень умный, просто умный... и совсем, пардон, дурак).

Полученные точки превратим в кривые, где по оси X отложим параметр человека, а по оси Y — процент людей с данным параметром. При этом статистическую обработку проведем отдельно для мужчин и женщин. Что мы получим?

Статистический случай 1:

Рис. 14.35
Гистограмма роста мужчин и женщин

Кривые случая 1 (рис. 14.35) получаются для тех параметров человека, значение которых у мужчин больше, чем у женщин (рост, вес, сила мышц и т. д.). Это, как многие считают, связано с эволюцией — если среднестатистическая мужская особь крупнее женской, то новые поколения укрупняются. У пауков, например, самцы намного меньше самок, и пауки сейчас, слава богу, более мелкие, чем в доисторические времена. На рисунке 14.20 (стр. 323), кстати, игроки в белых одеждах — это, вероятно, женщины, а в черных — мужчины, что подтверждает нашу гипотезу на рисунке 14.35

Статистический случай 2:

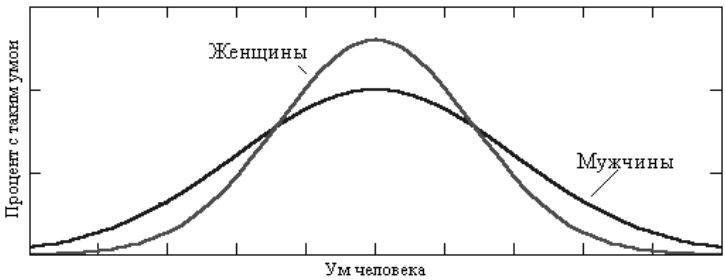


Рис. 14.36
Гистограмма интеллектуальных качеств мужчин и женщин

А так (случай 2: рис. 14.36) могут выглядеть кривые для параметров, значения которых у человека за последние несколько тысяч лет не менялись — ум, например. Многие вполне обоснованно полагают, что современный человек, если убрать налет образованности и культуры, ненамного умней древнего грека или египтянина. Среди мужчин гениев (нобелевских и прочих лауреатов, великих изобретателей, знаменитых писателей, художников, композиторов — см. правый край графиков на рис. 14.36) больше лишь потому, что и... дураков среди мужчин достаточно (левый край графика). Но средняя женщина *умнее* среднего мужчины — центр «женской» кривой приподнят за счет меньшего разброса по краям: площади фигур под графиками одинаковы — никого не обижая, будем считать, что Господь Бог или Природа (кто как для себя считает) одинаково наделили умом обе половины человечества.

С этим можно, конечно, поспорить, но... еще раз взглянем на второе название этюда.

Пакет Mathcad (как, впрочем, и другие математические пакеты) имеет богатый набор встроенных средств для статистической и прочей обработки массивов данных. Их освоение поможет решать учебные и производственные задачи и, в частности, отличать ложь (неверные или несколько искаженные исходные данные) от... статистики.

Задания читателям

1. Для проверки решения регрессионной задачи по стоимости подержанного автомобиля мы использовали четыре декартовых графика — см. 14.28–14.31 (стр. 329, 330). Можно данный минимум увидеть и на поверхностях, выбирая две переменные и фиксируя два параметра. Построить эти поверхности!

2. Применить генетический метод поиска глобального минимума (см. рис. 14.33) при регрессионном анализе цены подержанного автомобиля.

Этюд 15

ПОЙТИ ТУДА, ЗНАЯ КУДА, ИЛИ НОВЫЕ ШВЕЙЦАРСКИЕ ЧАСЫ

В этюде будет рассказано о необычном решении задачи коммивояжера и других задач оптимизации.

Математика: простые числа, разложение на множители, граф, минимум функции, задача коммивояжера, обратная задача коммивояжера, метод ближайшего соседа, уравнения в целых числах, интеграл, интегро-дифференциальное уравнение, матрица, задача о семи мостах, задача Штайнера, объемная задача Штайнера, транспортная задача, парадоксы оптимизации, финансовая пирамида.

Физика: часы, бионика, феромоны, мыльная пена.

Информатика: символьная математика, анимация, цикл с параметром, условный оператор, функции работы с матрицами.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127814**

Начнем с часов.

Современные *обычные* часы — это круглый диск со стрелками. По периметру такого диска расположены двенадцать арабских или римских цифр, образуя *циферблат*. Но нет на свете человека, которого бы оставили равнодушными *необычные часы,* как по форме, так и по содержанию. С момента изобретения часов люди не переставали менять не только их, устройство (солнечные, песочные, водяные, механические, электронные, атомные и т. д.), но и их внешний вид, преследуя при этом и практические, и декоративные цели. Если говорить о практических целях, то тут можно вспомнить нестандартные часы с 24-часовым циферблатом. Такие часы (хронометры) размещают на кораблях (морских, воздушных, космических), чтобы не спутать день с ночью. Есть часы с двумя циферблатами, показывающими путешественнику и местное и «родное» время. В гостиницах и офисах можно увидеть часы с несколькими циферблатами, вернее, несколько связанных часов, фиксирующих время в ключевых столицах мира: Tokyo, Moscow, Paris, London, New York и т. д.

Смешанные практические и декоративные функции несут так называемые *цифровые часы*, где время отмечается не стрелками («аналоговые» часы»), а числами: 14:21:37, например. Такие часы когда-то были и механическими, а сейчас они, как правило, электронные. Наиболее практичными считаются часы, где время отмечается стрелками (по стрелкам проще оценивать, сколько времени осталось до какого-то события), а календарная дата — числами.

Но основные «изыски и изощрения» в сфере *необычности* часов развернулись на чисто «декоративном фронте». В этом можно убедиться, если в Интернете сделать запрос по ключу «Необычные часы». Каких только часов там не увидишь?! Вершина дизайнерского аскетизма — вращающиеся у стенки две стрелки разной длины (спасибо и за это) без каких-либо цифр или других меток на циферблате, которого, кстати говоря, у таких часов совсем и нет. Другая крайность — перегруженность часов различными «художественными излишествами». Глядя на некоторые подобные часы, долго ломаешь голову, чтобы понять, какое время они показывают. Часто в этой «головоломке» и заключена вся дизайнерская суть таких «часов», доходящая иногда до абсурда. Читаем у Даниила Хармса — мастера литературного абсурда: «На дворе стоит старуха и держит в руках стенные часы. Я прохожу мимо старухи, останавливаюсь и спрашиваю ее: «Который час?» — Посмотрите, — говорит мне старуха. Я смотрю и вижу, что на часах нет стрелок. — Тут нет стрелок, — говорю я. Старуха смотрит на циферблат и говорит мне: — Сейчас без четверти три» [72].

Из всех часов необычной формы особо выделяются часы с неким *математическим смыслом*. Безусловно, математика присутствует во всех часах — стоит толь-

ко вспомнить о дифференциальном уравнении колебания маятника, балансира или кремневого кристалла, на которых основан ход современных часов [54]. В этюде 9 мы описали часы, показывающие не только время, но и суть второго закона Кеплера. В некоторых часах корпус делают прозрачным, чтобы была видна вся их внутренняя «механико-математическая сущность»[1]. Но мы сейчас поговорим о внешней, так сказать, математике часов.

Есть часы не с цифрами 1, 2, 3,..., 12, а с математическими выражениями (радикалами) $\sqrt{1}, \sqrt{4}, \sqrt{9}, \ldots, \sqrt{144}$ на циферблате. В этой «декоративности» можно отметить и некую «практичность» — такие часы будут уместны, например, в школьном кабинете математики: ученики, поглядывая на часы, будут запоминать таблицу квадратов натуральных чисел. На рис. 15.1 показаны авторские часы, где вместо цифр записаны математические формулы и выражения, по которым эти цифры можно вычислить в среде инженерного калькулятора Mathcad. Читатель по этим часам может не только узнать текущее время, но и оценить свои знания математики и пакета Mathcad — все ли формулы и обозначения чисел ему понятны [58]?

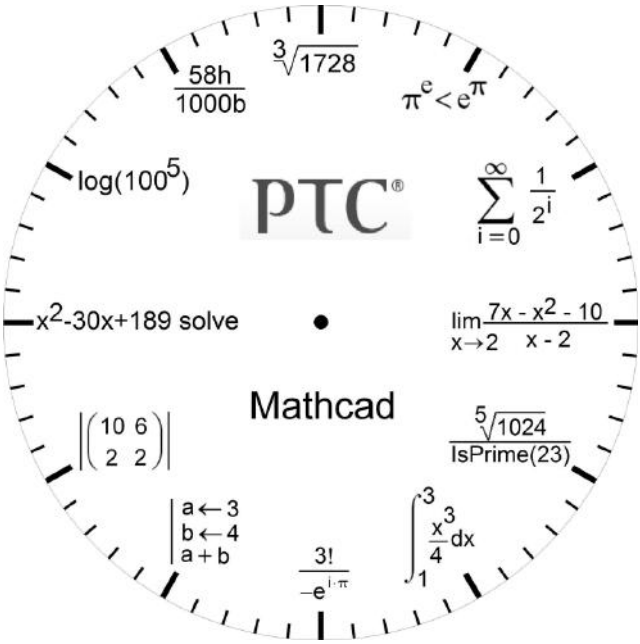


Рис. 15.1

Авторские часы с «математическим» циферблатом

Легко заметить, что вместо привычных натуральных чисел 1, 2, ..., 12, записанных цифрами от 0 до 9, на циферблат можно поместить те же самые числа, но записанные только с помощью цифр 1, 2 и 3, операций сложения, умножения и возведения в степень. Действительно: $4 = 2^2$; $5 = 2^1 + 3^1$; $6 = 2^1 \cdot 3^1$; $7 = 2^2 + 3^1$; $8 = 2^3$; $9 = 3^2$; $10 = 2^3 + 2^1$; $11 = 2^3 + 3^1$; $12 = 2^2 \cdot 3^1$.

Стоит напомнить, что двойка является единственным четным простым числом, а тройка — первым нечетным простым числом. Единица не считается простым числом, хотя и является ключевым в возникновении этого понятия. Евклид

[1] Хорошая идея по теме этюда 8: создать в среде Mathcad кинематическую схему часов, передающую колебание маятника или балансира часовой, минутной и секундной стрелкам.

определял простые числа так: «Простое число есть измеряемое только единицей». О каких же измерениях здесь идет речь? О самых привычных измерениях отрезка числовой оси от нуля до натурального числа посредством многократного откладывания отрезка единичной длины. Все натуральные числа, бóльшие единицы и не являющиеся простыми, можно помимо единицы «измерить», по крайней мере, еще одним числом. То есть можно откладывать отрезок, например, длины 2 (четные числа) или 5 (числа, кратные пяти). Такие числа называются составными. Говоря строго математически, простые числа не имеют других делителей, кроме единицы и самого себя. Саму единицу математики предпочитают не считать простым числом, еще и потому, что в этом случае многие важные теоремы теории чисел формулируются проще.

В обозначении часов на циферблате, что удивительно, участвуют ровно 6 (половина от 12) составных чисел: 4, 6, 8, 9, 10 и 12; 5 простых чисел и единица. Таким образом, простые числа составляют почти 42% чисел первой дюжины. Всегда ли они расположены так часто?

Для того чтобы это понять, математики ввели функцию $\pi(N)$ — количество простых чисел, не превосходящих N, и специальную величину $\frac{\pi(N)}{N}$, называемую средней плотностью простых чисел среди первых N натуральных чисел. Изучение таблиц простых чисел показывает, что с ростом N простые числа встречаются в среднем всё реже и реже (хотя все же чаще квадратов натуральных чисел). Эйлер доказал, что $\lim\limits_{N\to\infty}\frac{\pi(N)}{N}=0$, т. е. чем длиннее отрезок $[0,N]$, тем меньший процент от всех натуральных чисел на этом отрезке составляют простые числа. Но если предел плотности равен нулю, то, возможно, что рано или поздно все простые числа «закончатся»? Оказывается, что это невозможно. Теорему о бесконечном множестве простых чисел мы легко докажем с вами прямо сейчас.

Предположим, что нашлось самое большое (и поэтому последнее) простое число. Перемножим все известные простые числа, включая наибольшее, а к произведению добавим единицу. Если полученное таким способом число делить на каждое из известных простых чисел, то в остатке мы получим 1, т. е. новое число не делится ни на одно простое число. Поэтому оно не делится ни на одно составное число, т. к. каждое составное число, в свою очередь, является произведением некоторых простых чисел. Итак, мы нашли число, которое больше любого известного простого числа и делится только на единицу и на себя, т. е. само является простым, превосходящим все известные простые числа. Мы получили противоречие, а это означает, что наибольшего простого числа не существует.

Если вам захочется узнать, какие же простые числа лежат между двумя заданными натуральными числами, то вы сможете отыскать их, используя Mathcad-функцию `IsPrime` (см. «четыре часа» на рис. 15.1), определяющую, являются ли аргумент простым числом, и/или оператор `factor`, раскладывающий составное число на простые множители — см. рис. 15.2. Кстати, разложение большого числа на простые множители — это довольно сложная и длительная математическая операция. Одни из самых распространенных алгоритмов шифрования текстов основан на разложении числа на простые множители.

Из рис. 15.2 видно, что 1 не является ни простым числом, и, тем более, составным. Показаны также два первых простых числа с двумя и девятнадцатью цифрами-единицами. Следующие такие простые числа составлены из 23, 317, 1031 и т. д. единиц. Кстати, поиск таких «одноединичных» простых чисел («и т. д.») — это отдельная интересная задача.

$$\mathrm{IsPrime}(1) \rightarrow 0$$

$$11 \xrightarrow{factor} 11$$

$$11111111111111111 \xrightarrow{factor} 2071723 \cdot 5363222357$$

$$1111111111111111111 \xrightarrow{factor} 1111111111111111111$$

Рис. 15.2
Анализ числа «на простоту» в среде Mathcad

В оформление часов иногда закладывают сложные математические алгоритмы [58]. Одна корейская фирма выпускает часы (рис. 15.3) в виде решения *задачи коммивояжера*. Вот что о ней можно прочесть в Википедии (**www.wikipedia.org**): «Задача коммивояжера (разъездного торговца) является одной из самых известных задач комбинаторной оптимизации. Задача заключается в отыскании наивыгоднейшего маршрута, проходящего через указанные города с последующим возвратом в исходный город. В условиях задачи указываются критерий выгодности маршрута (кратчайший, самый дешевый и т. п.) и соответствующие данные о расстояниях, стоимости и т. п.». Коммивояжер при этом должен побывать в каждом городе только один раз. Задача коммивояжера имеет много практических приложений. Прежде всего, это оптимизация маршрутов — оптимальная логистика, как принято сейчас говорить. Читаем в романе Л. Н. Толстого «Воскресенье»: «Сообразив, куда прежде, куда после ехать, чтобы не возвращаться, Нехлюдов прежде всего направился в сенат» [73]. Классические коммивояжеры (коробейники, офени) сейчас практически перевелись, хотя на дверях многих офисов до сих пор можно увидеть табличку типа «Торговым агентам вход запрещен!» Задачу коммивояжера решает курьер, отправляющийся развозить заказы клиентов сетевого магазина, торгующего электроникой, лекарствами, книгами и прочим. Есть и не вполне обычные приложения задачи коммивояжера. Так, например, расшифровать геном живого организма невозможно без решения довольно сложной задачи коммивояжера.

Рис. 15.3
Часы-глобус с задачей коммивояжера

На глобусе часов с задачей коммивояжера (рис. 15.3) каждые сутки ровно в полночь световыми точками отмечаются 1440 населенных пунктов нашей планеты, выбранные случайным образом из почти двух миллионов, хранящихся в па-

мяти часов (см. ниже рис. 15.8). Внутри глобуса установлен миниатюрный планетарий, высвечивающий эти точки на поверхности глобуса. Число 1440 — это количество минут в сутках. Более крупными световыми точками на глобусе выделяются 24 узловых населенных пункта («города»). Это будут часы в сутках. Далее выбирается первый случайный город, в который помещается «коммивояжер» и которому дается задание обойти все населенные пункты, руководствуясь *алгоритмом ближайшего соседа* — из очередного города коммивояжер идет в ближайший город, который он еще не посетил. На цифровой панели этих часов высвечиваются две цифры: номер города, из которого вышел наш коммивояжер (0, 1... 23 — часы) и номер населенного пункта, который он только что прошел (0, 1... 59 — минуты). Пройденный маршрут на глобусе высвечивается линиями, соединяющими отдельные точки. Человек, смотрящий на такие часы, может не только узнать, какое сейчас время, но и увидеть, где сейчас находится коммивояжер, и оценить, куда он пойдет в следующую минуту. На панели часов помимо текущего времени высвечивается название пройденного населенного пункта, а из динамиков звучат типичные мелодии и песни данного региона Земли. Динамики часов могут, конечно, выполнять и свою основную функцию — транслировать радиопередачи. Глобус часов можно также использовать по прямому назначению — для изучения политической карты мира.

Другие подобные, но уже не настольные, а настенные часы висят на вокзале швейцарского города Люцерн. Швейцария — это страна часов и путешественников и такие часы там очень уместны. На этих часах (плоская плазменная панель) прорисован контур Швейцарии. Каждые сутки в полночь на этой карте высвечивается 1440 случайных городов, деревень и отдельно стоящих отелей этой альпийской страны. Человек, приехавший на вокзал в города Люцерн, может не только уточнить время по таким часам, но и при желании будет следить за передвижением «коммивояжера», угадывать, куда он повернет в следующую минуту. Это скрасит ожидание поезда, заставит задуматься о задаче коммивояжера, об оптимизации собственного путешествия по Швейцарии, Европе, миру...

На рис. 15.4 и 15.5 показано упрощенное графическое отображение «циферблата» таких часов. В прямоугольной области случайным образом размещаются 1440 квадратиков, 24 из которых накрываются кружечками, которые нумеруются числами от 0 до 23. В полночь из нулевого города (см. верхний левый угол на рис. 15.4 и 15.5) выходит «коммивояжер» и за 24 часа посещает все города. На рис. 15.4 показан «полдень» нашего путешественника (12:00:00), а на рис. 15.5 — минута до полуночи.

Анимацию этих необычных часов можно увидеть на авторском сайте **https://www.ptcusercommunity.com/message/153959**.

В задачу коммивояжера, схематично отображенную на рис. 15.4 и 15.5, заложен *алгоритм ближайшего соседа*: из очередного города коммивояжер идет ближайший город, который он еще не посещал. Алгоритм ближайшего соседа относится к разряду «жадных» алгоритмов. Из-за этой «жадности» нашему коммивояжеру приходится петлять, а в конце своего турне перескакивать в отдаленные города (см. длинные отрезки прямых линий на рис. 15.5), ранее опрометчиво пропущенные. Все это увеличивает общую длину маршрута.

Посетители вокзала города Люцерн, глядя на такие необычные часы, могут задумываться не только «о времени и пространстве», но и еще кое о чем...

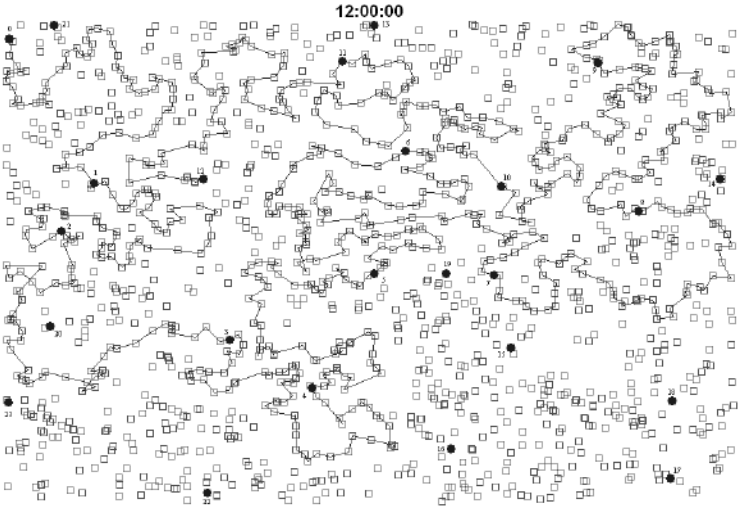


Рис. 15.4
Часы с задачей коммивояжера — полдень

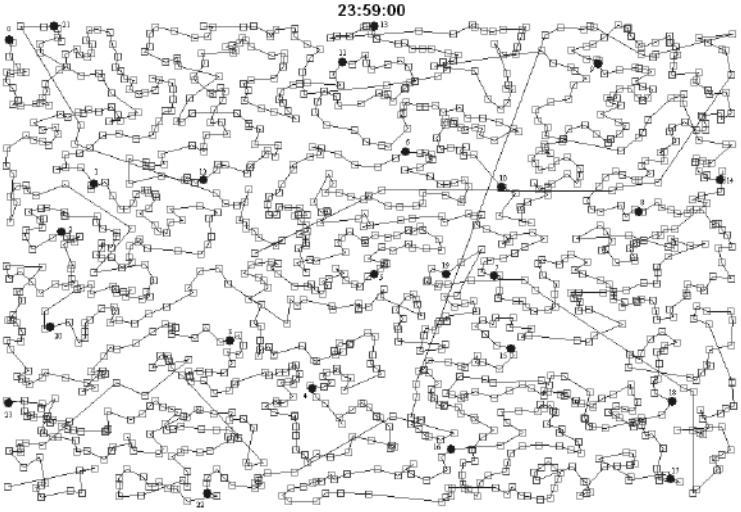


Рис. 15.5
Часы с задачей коммивояжера — без одной минуты полночь

Швейцария — страна не только часов и путешественников, но и банкиров[2], одно из качеств которых — это профессиональная «жадность», выражающаяся, в частности, в том, что банкиры стараются брать деньги у клиентов на хранение в банке под минимальные проценты, а давать кредиты под максимальную ставку. Профессиональная жадность заставляет банкиров раздавать кредиты даже тем, кто деньги может и не вернуть. Это одна из причин финансового кризиса, охватившего недавно почти весь мир, погубившего многие банки и поставившего некоторые страны на грань финансового краха[3]. Коммивояжер, пропуская нужный город и «жадно» направляясь к ближайшему, увеличивает свой маршрут. Бан-

[2] А еще шоколада и ножей. Но эта деталь лишняя для нашего повествования.

[3] Этим странам журналисты когда-то дали обидную аббревиатуру PIGS — поросята: Португалия (P), Ирландия (I), Греция (G) и Испания (S). Греция до сих пор не может выбраться из долговой ямы.

кир, гоняясь за сиюминутной прибылью, снижает общую прибыль банка. Банковское дело, связанное с кредитованием, опасно граничит с ростовщичеством. Еще одна неприглядная деятельность, похожая на банковскую, — это строительство финансовых пирамид, математическую модель одной из которых мы рассмотрим в конце этюда.

Алгоритм ближайшего соседа подкупает простотой своей реализации: задачу можно решить с помощью циркуля, карандаша и линейки. После того как все города будут соединены линиями, можно попытаться распутать некоторые петли, сократив тем самым общую длину маршрута. Но математики разработали более оптимальные алгоритмы решения задачи коммивояжера. Один из них подсказала живая природа. Люди заметили, что муравьи, найдя новый источник питания, сначала таскают пищу из него в свой муравейник случайными разбросанными маршрутами. Но постепенно разброс маршрутов стягивается в один единственный — кратчайший. Вот как об этом написано в той же Википедии: «В реальном мире муравьи первоначально ходят в случайном порядке и по нахождению продовольствия возвращаются в свою колонию, прокладывая феромонами[4] тропы. Если другие муравьи находят такие тропы, они, вероятнее всего, пойдут по ним. Вместо того чтобы отслеживать цепочку, они укрепляют её при возвращении, если в конечном итоге находят источник питания. Со временем феромонная тропа начинает испаряться, тем самым уменьшая свою привлекательную силу. Чем больше времени требуется для прохождения пути до цели и обратно, тем сильнее испарится феромонная тропа. На коротком пути, для сравнения, прохождение будет более быстрым и, как следствие, плотность феромонов остаётся высокой. Испарение феромонов также имеет свойство стремления к локально-оптимальному решению. Если бы феромоны не испарялись, то путь, выбранный первым, был бы самым привлекательным. В этом случае исследования пространственных решений были бы ограниченными. Таким образом, когда один муравей находит (например, короткий) путь от колонии до источника пищи, другие муравьи, скорее всего, пойдут по этому пути, и положительные отзывы в конечном итоге приводят всех муравьёв к одному, кратчайшему, пути».

На рис. 15.6 показан путь коммивояжера, проложенный с помощью муравьиного алгоритма по 16 862 населенным пунктам Италии [73]. Естественно, путешественник, пожелавший обойти самым коротким путем все «города и веси» какой-либо страны, далеко не всегда сможет по прямой линии перейти из одного населенного пункта в другой. Ему часто придется идти обходными путями по существующим дорогам и тропам. Недавно в Google появился специальный сервис, позволяющий оптимизировать маршруты передвижения не только из пункта *A* в пункт *B*, указанные на карте мира, но и по круговым маршрутам, используя при этом автомобиль, поезд, самолет и другие средства передвижения.

Решение задачи коммивояжера, как и в случае с оформлением часов, может совсем уходить от практики в сторону чистой декоративности. Есть направление в живописи под названием пуантилизм, когда художник пишет картину точечными мазками. Так мы писали портреты в этюде 5. На рис. 15.7 показан вариант знаменитой картины Леонардо да Винчи, где Мона Лиза[5] прорисована 100 000 точками, соединенными «тропой коммивояжера». Чисто декоративным можно также считать и путь коммивояжера через 1 904 711 населенных пунктов Земли — см. рис. 15.8.

[4] Феромо́ны (греч. φέρω — «нести» + ορμόνη — «побуждать, вызывать») — собирательное название веществ — продуктов внешней секреции, выделяемых некоторыми видами животных и обеспечивающих химическую коммуникацию между особями одного вида.

[5] «Бедная Лиза» — это не только повесть Карамзина, но и восклицания по поводу того, какие «художественные изыски» приходится терпеть этой знаменитой даме.

Рис. 15.6
Путь коммивояжера по Италии

Рис. 15.7
Задача коммивояжера и Мона Лиза

Но вернемся к вокзальным часам в швейцарском городе Люцерн.

Путешественник в отличие от коммивояжера часто намерено выбирает не наикратчайший маршрут, а маршрут, отвечающий каким-то другим критериям. Для путешественника предпочтительным маршрутом может быть даже и наидлиннейший. В связи с этим можно говорить об *обратной задаче коммивояжера*, о задаче нахождения самого длинного пути.

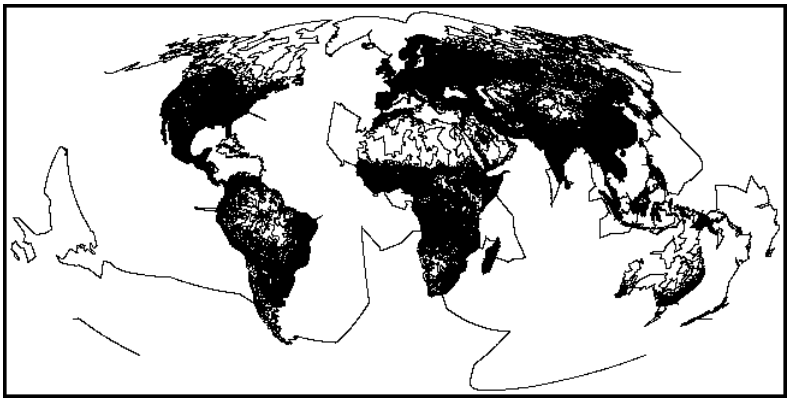


Рис. 15.8
Мировое турне коммивояжера

На рис. 15.9 показан путь коммивояжера через 979 городов и деревень Люксембурга при использовании жадного алгоритма для отыскания наикратчайшего маршрута. Анимация этого пути помещена на авторском сайте **https://www.ptcusercommunity.com/videos/1366**. Мы остановились на Люксембурге, а не на Швейцарии потому, что Швейцария не «влезет» в компьютер при реализации алгоритма ближайшего соседа — матрица `M` (см. ниже рис. 15.10) будет слишком велика. Но Люксембург — это тоже страна банкиров.

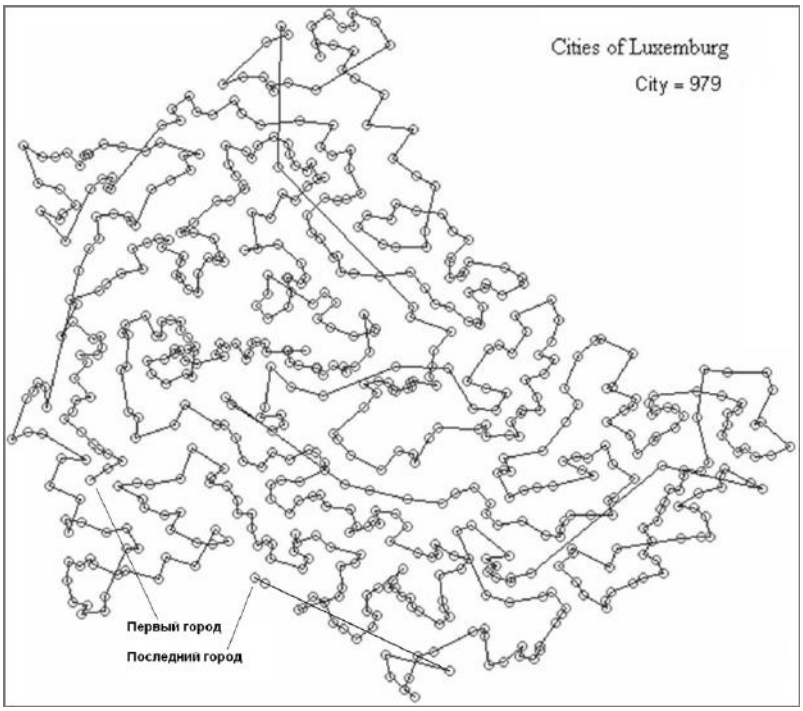


Рис. 15.9
Путь коммивояжера при использовании жадного алгоритма для отыскания наикратчайшего маршрута

Из рис. 15.9 (как, впрочем, и из рис. 15.4 и 15.5) видно, что выбранный маршрут не оптимальный, т. к. коммивояжеру, повторяем, из-за своей «жадности» приходится делать петли. Кроме того, несложно показать, что длина «жадного» маршрута сильно зависит от выбранного первого города. На рис. 15.9 первый и последний город отмечены. Путь, показанный на рис. 15.9, является оптимальным и в том смысле, что были перебраны все города в качестве стартового и зафиксирован город, из которого путь по жадному алгоритму оказался минимальным. Еще один признак неоптимальности пути коммивояжера, показанного на рис. 15.4, 15.5 и 15.9, состоит в том, что длины первой и второй половины пути сильно отличаются. На рисунке 15.10 показана Mathcad-программа отыскания пути коммивояжера методом ближайшего соседа. Программа возвращает вектор `way`, хранящий номера городов — порядок их обхода для минимизации маршрута.

```
way := | "Путь коммивояжера по "жадному" маршруту"
M ← | for  i ∈ 1 .. n
      |  for  j ∈ 1 .. n
      |    M_{i,j} ← if[i = j, ∞, √((X_i − X_j)^2 + (Y_i − Y_j)^2)]
      | M
way_1 ← ib
for  i ∈ 2 .. n
  | for  j ∈ 1 .. n
  |   s_j ← M_{(way_{i−1}, j)}
  | way_i ← match(min(s), s)_1
  | for  j ∈ 1 .. i − 1
  |   | M_{(way_j, way_i)} ← ∞   M_{(way_i, way_j)} ← ∞
return way
```

Рис. 15.10
Mathcad-программа метода ближайшего соседа

Исходными данными для программы являются вектора `X` и `Y`, с декартовыми координатами городов. Расчет начинается с формирования квадратной матрицы `M`, хранящей числа — расстояния между городами вне главной диагонали или «значение бесконечности» (∞ — сверхбольшое компьютерное число) на главной диагонали. «Значения бесконечности» будет также присваиваться элементам матрицы $M_{i,j}$ и $M_{j,i}$, если коммивояжер уже прошел путь от *i*-го до *j*-го города или наоборот. «Гвоздем» программы являются функции `min` и `match`. Функция `min` возвращает минимальный элемент своего аргумента-вектора, а функции `match` — место этого элемента в векторе. Функция `match` всегда возвращает вектор, даже если в нем всего один элемент. Поэтому при ее вызове в программе на рис. 15.10 использован оператор возврата первого элемента вектора, конвертирующий такой единичный вектор в скаляр.

Но если в программе функцию `min` заменить на функцию `max`, то она будет возвращать... максимальный путь коммивояжера, когда он из очередного города направляется не в ближайший, а самый дальний, в котором еще не был. Этот маршрут, вернее, его последний участок показан на рис. 15.11. Весь маршрут показать невозможно — получится сплошь заштрихованная страна Люксембург. На рисунке 15.11 заштрихована только ее центральная часть.

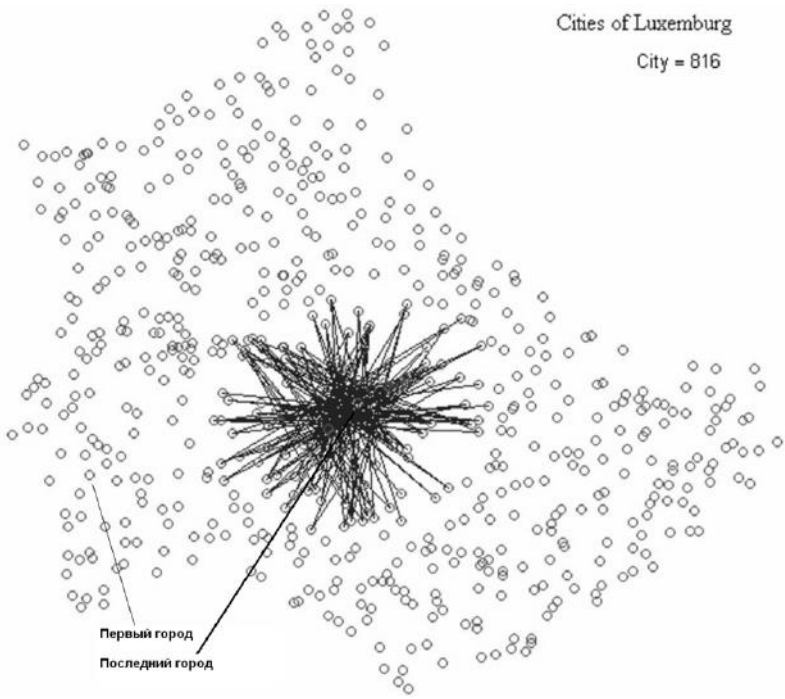


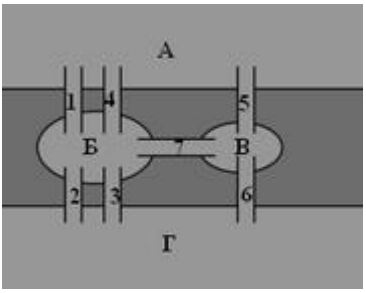
Рис. 15.11
Завершение пути коммивояжера по «щедрому» алгоритму
(см. анимацию здесь ***https://www.ptcusercommunity.com/videos/1464****)*

Назовем этот алгоритм «щедрым» и отметим, что его длина также зависит от выбранного начального города. На рис. 15.11 этот город тот же, какой был выбран при реализации «жадного» алгоритма (рис. 15.9), но можно опять перебором выбрать город, путь коммивояжера из которого по «щедрому» алгоритму окажется самым длинным. Эту задачу мы поручаем читателю, отсылая его за нужными программами и другими данными на сайт — **http://twt.mpei.ac.ru/ ochkov/SalesMan**. Еще более длинный маршрут можно получить, если, например, применить вышеописанный «муравьиный» алгоритм к матрице, хранящей обратные расстояния между городами.

Первой наиболее известной задачей оптимизации маршрута считается задача о семи мостах Кенигсберга (ныне город Калининград). Издавна среди жителей этого города была распространена такая загадка: как пройти по всем мостам, не проходя ни по одному из них дважды? Решить эту задачу как теоретически, так и практически, долго никому не удавалось, однако не удавалось и доказать, что это невозможно. В 1736 г. задача о семи мостах заинтересовала выдающегося математика, члена Петербургской академии наук Леонарда Эйлера. Он смог найти общие правила, пользуясь которыми легко определить, можно ли пройти по всем мостам любого города, не проходя дважды ни по одному из них. Оказалось, что в случае семи мостов Кенигсберга это невозможно.

Эйлер построил граф, в котором мостам соответствуют линии (ребра графа), а частям города — точки (вершины графа). В ходе решения этой задачи ученый выявил и доказал следующее.

1. Число нечетных вершин (вершин, к которым ведет нечетное число ребер) графа всегда четно. Не может существовать граф, который имел бы нечетное число нечетных вершин.



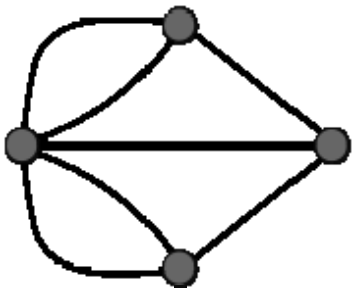


Рис. 15.12
Упрощенная схема и граф мостов Кенигсберга[6]

2. Если все вершины графа четные, то можно, не отрывая карандаша от бумаги, начертить граф, начинающийся и заканчивающийся в одной и той же любой наперед выбранной вершине.

3. Граф с более чем двумя нечетными вершинами невозможно начертить одним росчерком.

Граф кенигсбергских мостов имел четыре нечетные вершины, следовательно, невозможно было пройти по всем мостам, не проходя ни по одному из них дважды.

Созданная Эйлером (благодаря этой задаче) теория графов нашла очень широкое применение: например, ее используют при изучении транспортных и коммуникационных систем, в частности, для маршрутизации данных в Интернете. Заметим, что и задача коммивояжера имеет решение потому, что ее граф (см. рис. 15.9 и 15.11) имеет все вершины четные (вход в город по одной дороге, выход — по другой, т. е. ровно два ребра). Этот граф подчиняется второму утверждению Эйлера.

Но наличие решения вовсе не означает, что это решение будет восприниматься как естественное и интуитивно понятное. Например, задача коммивояжера содержит парадокс: часто не стоит идти к ближайшему городу!

А вот еще три задачи, схожие с задачей коммивояжера и тоже имеющие элементы парадоксальности.

Задача 1. Вы находитесь в точке *A* (см. схему на рис. 15.13) на границе пашни (вверху) и газона (внизу). Начинается гроза, и вам нужно быстро добежать до дома, находящемуся в точке *B*[7]. Скорость вашего бега по пашне 5 км/ч, а по газону 10 км/ч. Как вы побежите в дом? Первое, что приходит на ум, тем более, в стрессовой ситуации — это бежать по прямой от точки *A* до точки *B*. Но оптимальный маршрут — это некий «ход конем» (см. рис. 10.6 на стр. 246): сначала нужно побежать по газону со скоростью 10 км/ч, а потом свернуть на вспаханное поле, где ваша скорость ниже. В правой части рисунка 15.13 показан график изменения времени такого спасительного бега в зависимости от координаты поворота с газона на пашню. На графике четко видна точка минимума в районе 40 метров.

[6] **http://dic.academic.ru/dic.nsf/ruwiki/12707**

[7] Другой вариант этой задачи. Вы неспеша идете у берега моря по кромке широкого пляжа. Впереди правее/левее в метрах 100 вы видите свободную скамейку в метрах 30 от моря, на которую желаете присесть. К скамейке нужно подойти как можно быстрее, а то ее займут раньше вас. Как вы пойдете или побежите к скамейке, если у моря песок плотный, позволяющий перемещаться быстро, а на самом пляже — рыхлый, тормозящий передвижению? В этой задаче оптимизации будет еще одно ограничение — двигаться нужно так, чтобы не было заметно, что вы стараетесь занять скамейку раньше кого-то другого.

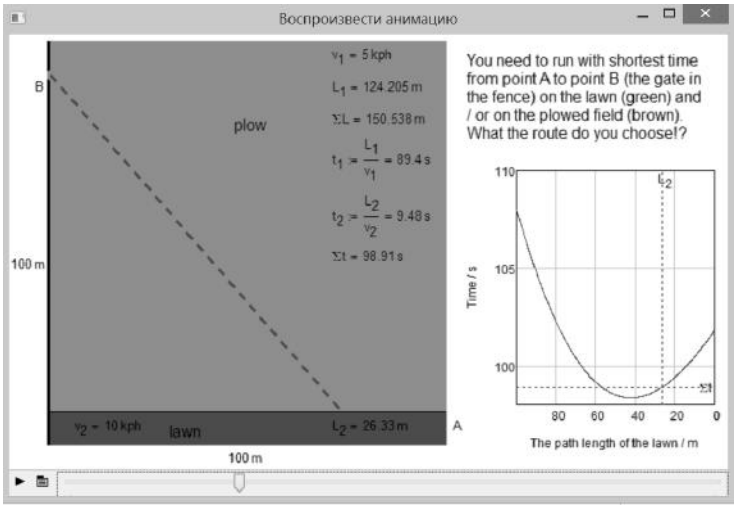


Рис. 15.13

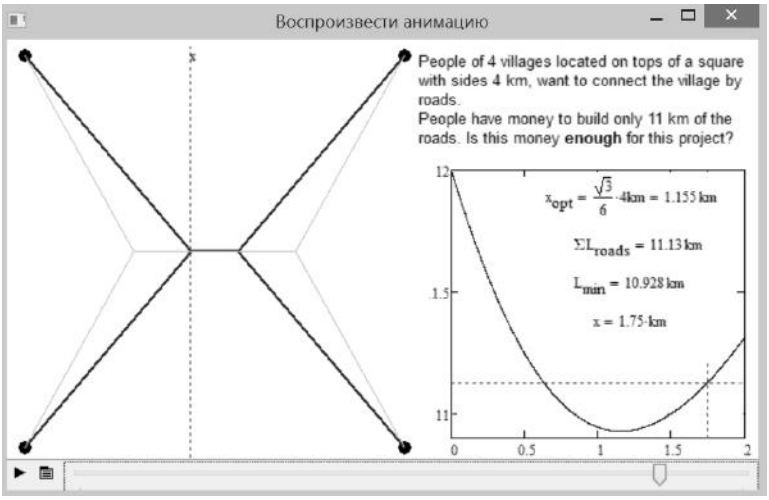
Оптимизация бега от грозы (см. анимацию здесь ***https://www.ptcusercommunity.com/videos/2284****)*

Но представим себе, что около дома есть заасфальтированная дорожка до газона и дальше вниз (см. левый край рис. 15.13), по которой можно бежать со скоростью еще большей, чем при беге по газону — например, 15 км/ч (см. **http://frs24.ru/st/skorost-bega-cheloveka**). Тогда для сокращения времени в пути можно бежать по пашне не прямо к дому, а к асфальтированной дорожке, а потом по ней в дом. В этом случае мы будем иметь уже два параметра оптимизации — координата точки схода с газона на пашню и координата схода с пашни на дорожку. А может быть, вообще не нужно бежать по пашне, а только по газону и по дорожке? Какая в этом случае минимальная скорость бега по дорожке должна быть? Эту усложненную задачу мы просим решить самим читателям.

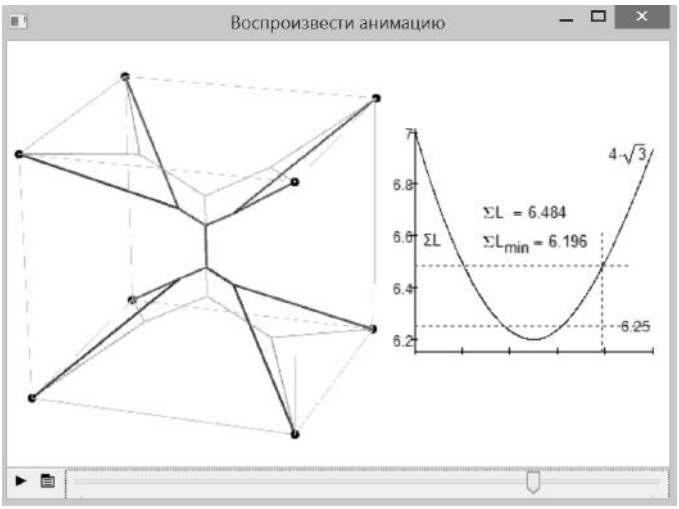
Но вернемся к коммивояжеру. На рисунке 15.9 показано, что этот странствующий торговец шел между городами (точками) строго по прямой. Но так идти в реальном путешествии практически невозможно — каждый отрезок пути нужно оптимизировать, учитывая различные ограничения. Пример такой оптимизации был показан на рис. 15.13.

Задача 2. Точки, показанные на рис. 15.9, нужно не просто обойти, предлагая свой товар, а соединить дорожками так, чтобы их общая длина была минимальной. Эта задача называется задачей Штейнера (Steiner) по имени математика, ее впервые сформулировавшего. Она на порядок сложнее задачи коммивояжера. Но, тем не менее, мы ее поставим не только для плоскости (рис. 15.14*a*), но для пространства (рис. 15.14*b*). Если нужно соединить линией с минимальной длиной вершины квадрата, то многие тут соединяют эти четыре точки двумя отрезками прямых, проходящих через центр квадрата. Но это не оптимальное решение — см. рис. 15.14*a*. В задаче Штейнера в одну точку могут сходиться только три прямых линии с углом 120° друг к другу (звезда Мерседеса).

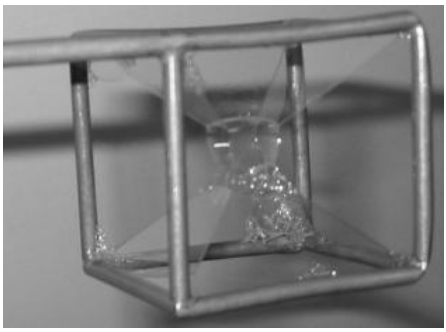
А вот объемный вариант задачи: под землей живут кроты в норках, расположенных на вершинах куба, которые решили соединить свои жилища так, чтобы длина ходов была минимальной. Как их прокопать? На рис. 5.14*b* показан кадр анимации решения этой задачи с отображением графика изменения суммарной длины ходов (целевая функция) в зависимости от их конфигурации. Оптимальный вариант также подразумевает, что «в одну точку могут сходиться только три прямых линии с углом 120° друг к другу».



a



b



c

Рис. 15.14

Задача Штейнера:

a — на плоскости; *b* — в объеме; *c* — физическая модель.

Задачу Штейнера обычно иллюстрируют... мыльной пеной (см. фото на рис. 15.14*c*), которая вследствие поверхностного натяжения минимизирует свою поверхность.

Практический вариант задачи Штейнера можно видеть на авторской анимации по адресу **https://www.ptcusercommunity.com/videos/2099**: имеется два дома, к которым нужно при минимальной суммарной длине труб подвести газ. Дома расположены на одной стороне от газовой магистрали. Тут могут быть такие решения: a) можно подводить отдельные трубы к каждому из домов перпендикулярно магистрали, b) можно подвести трубу к первому дому, а потом сделать от него отвод ко второму дому, а можно (c) где-то вне дома установить тройник. Для двух домов эта задача решается и анимируется довольно просто. А для трех, четырех и большего числа домов задача резко усложняется. Общей методики решения задачи Штейнера пока не найдено. Читатель может попробовать нарисовать на бумаге несколько точек и соединить отрезками прямых, имея под рукой треугольник с углом 120°. Потом измерить длины этих отрезков, просуммировать их и сделать выводы о минимизации этой суммы. А еще лучше — попытаться решить эту задачу на компьютере.

В этюде 9 мы экспериментировали с пятиконечной звездой, запуская с ее вершин спутники (см. рис. 9.5). На рис. 15.15 показан последний кадр анимации изменения конфигурации отрезков прямых с минимальной суммарной длиной, соединяющие пять точек, которые блуждая (предыдущие кадры анимации) стремятся к вершинам пятиконечной звезды. Эта звезда здесь оказалась не случайно — книга писалась во время празднования 70-летия Победы в Великой Отечественной войне.

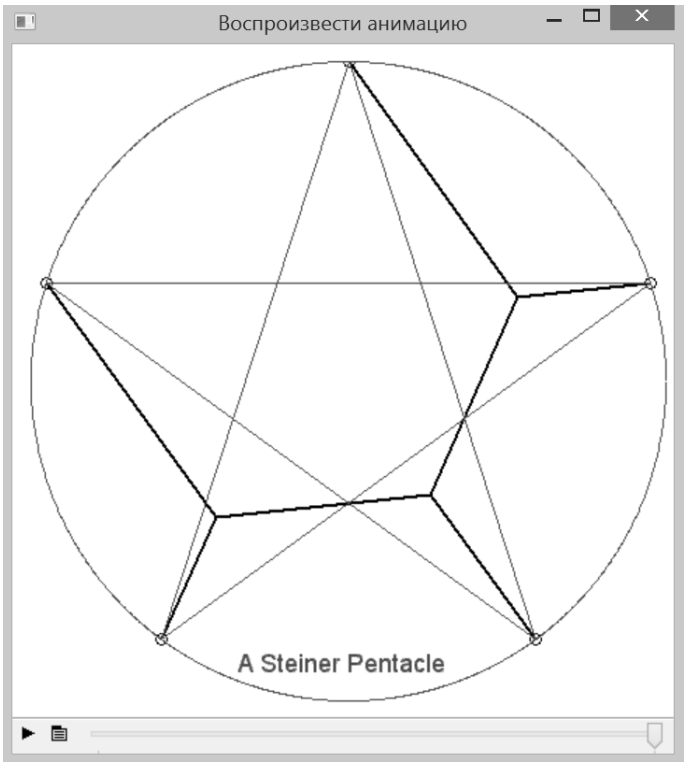


Рис. 15.15
Задача Штейнера для пятиконечной звезды

Задача 3. Имеются две шахты, где добывают уголь, и две ТЭЦ, где этот уголь сжигают «с колес», т. е. без складирования. Первая шахта «выдает на-гора» 50 т угля в сутки, а вторая — 70. Первая электростанция сжигает 40 т угля в сутки, а вторая — 80. Перевозка угля с первой шахты на первую электростанцию стоит 1200 руб. за тонну, с первой шахты на вторую электростанцию — 1600, со второй шахты на первую электростанцию — 800 и со второй шахты на вторую электростанцию — 1000 руб. за тонну. Нужно так организовать перевозку угля, чтобы ее стоимость была минимальна. На рис. 15.16 показаны условия этой транспортной задачи и ее «ручное» решение: самый дешевый маршрут (вторая шахта — первая ТЭЦ: 800 руб. за тонну перевезенного угля) нагружается полностью (см. первое слагаемое в выражении внизу рис. 15.16), а остатки угля «разбрасываются» по остальным маршрутам.

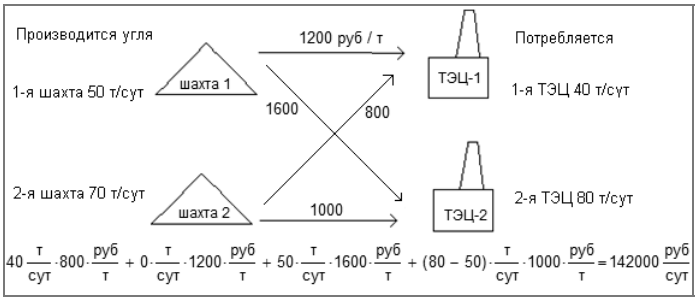


Рис. 15.16

Схема и ручное неверное решение транспортной задачи

На рисунке 15.17 показано автоматическое, а не ручное решение данной задачи стандартными средствами Mathcad 15 — формируется целевая функция (СП — стоимость перевозки), дается первое приближение (мы взяли значения, найденные при ручном решении задачи, показанном на рис. 15.16), за ключевым словом `Given` формируются ограничения, а функция `Minimize` находит значения неизвестных, минимизирующих целевую функцию при выполнении ограничений. Автоматическое решение задачи оказалось *парадоксальным* — по самому дешевому маршруту ничего не нужно возить (см. сноску на стр. 276).

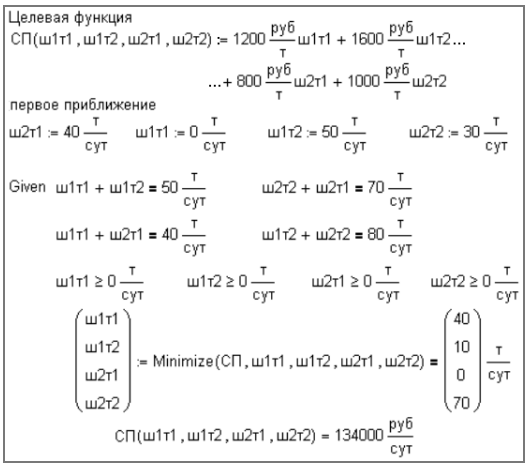


Рис. 15.17

Автоматическое правильное решение транспортной задачи

На авторском форуме **https://www.ptcusercommunity.com/groups/optimisation-with-mathcad** помещены анимации и других задач оптимизации.

А теперь давайте вернемся к банкирам, профессиональная жадность которых была нами ранее отмечена.

Банковское дело при слишком высокой плате за кредит превращается в ростовщичество. С другой стороны, слишком высокий процент по вкладу тоже не приветствуется — за такими «банковскими продуктами» может скрываться элементарная финансовая пирамида. Вот ее простейшая математическая модель — см. рис. 15.18.

Правильная банковская система как на *трех* китах покоится на *трех* числах. Первое число N_1 — плата за кредит. Взял в банке сто рублей — будь любезен в конце года верни $100 + N_1$ рублей. Второе число N_2 — процент по вкладу. Положил в банк сто рублей — получи в конце года $100 + N_2$ рублей. Разница между первым и вторым числом $(N_1 > N_2)$ заставляет банки прибыльно работать. Третье число N_3, подпирающее снизу два предыдущих и заставляющее людей нести деньги в банк, — это величина инфляции. В нормальной экономической ситуации низкий уровень инфляции и не очень высокая плата за кредит держат в узких рамках процент по вкладу:

$N_1 > N_2 > N_3$.

Если же инфляция высокая, то многие люди, забывая о ненормальности такой ситуации, легко верят в 20, 30, 50 и более процентов годовых по вкладу (ведь величина N_2 должна быть больше величины N_3) и ложатся в основание очередной финансовой пирамиды. Есть и менее наивные люди, понимающие, что пирамида — это особый род игры, где нужно уметь «*вовремя смыться*».

Итак, строим финансовую пирамиду (рис. 15.18).

В городе миллион жителей (переменная N), семь человек из которых в первый день (D = 1) купили по одной акции по цене 100 руб. Затем число этих людей будет увеличиваться (вектор NK). Если люди сдадут эту акцию через 50 дней (переменная Время), то они получат уже по 200 руб. за каждую сданную акцию. Эту динамику (курс продажи и курс покупки акций) определяют две функции K(D) и P(D). В городе возникает некий ажиотаж, который зафиксирован переменной K_a. Из кассы каждый день забирается определенная сумма денег (переменная Расход) на аренду помещений, рекламу, взятки чиновникам и пр., а также 7% дохода организаторов пирамиды (переменная Доход). Более-менее сложный оператор задачи, показанной на рис. 15.18, — это определение числа людей, купивших акции в какой-то день. Эта величина пропорциональна (с коэффициентом K_a) числу людей, еще не купивших акций, умноженному на число людей, уже купивших акции. Эта зависимость определяет форму двух волн покупателей и продавцов акций, изображенных на рис. 15.18.

На последнем графике на рис. 15.18 показано изменение суммы денег в кассе организаторов финансовой пирамиды. На 164-й день пирамиду нужно закрывать — прекращать выкуп акций и... уезжать куда-нибудь из этого города.

Мы же никуда не уезжаем, остаемся у своего компьютера и, собираясь вкладывать деньги в какое-то сомнительное предприятие, сначала просчитаем, что из этого может выйти. Так мы легко можем вернуть и даже приумножить деньги, потраченные на приобретение компьютера и легального программного обеспечения для него.

Если модель финансовой пирамиды предельно упростить, то ее решение можно свести к решению дифференциально-интегрального уравнения [10] — см. рис. 15.18, где также виден первоначальный рост доходов организаторов, а потом их спад.

N	Расход	Время	K_a	Доход	M_1	NK_1	SNK_1	MMM_1
(чел)	$\left(\frac{тыс_руб}{сут}\right)$	(сут)		(1%)	(млн_руб)	(чел)		
10^6	300	50	10^{-7}	7	70	7	NK_1	M_1

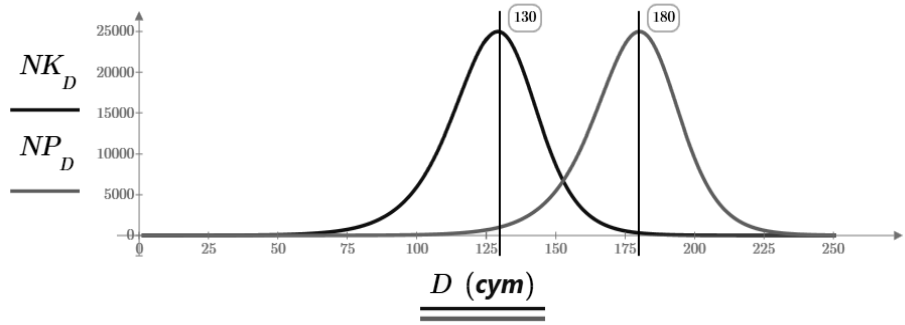
$$K(D) := 100\ руб + 2 \cdot \frac{руб}{сут} \cdot (D-1) \qquad P(D) := 105\ руб + 2 \cdot \frac{руб}{сут} \cdot (D-1)$$

$$D := 1 \cdot сут, 2 \cdot сут .. 280 \cdot сут \qquad \begin{bmatrix} NK_{D+1} \\ SNK_{D+1} \end{bmatrix} := \begin{bmatrix} K_a \cdot (N - SNK_D) \cdot SNK_D \\ SNK_D + NK_D \end{bmatrix}$$

$$NP_{D+1} := \text{if}\left(D \le Время, 0, NK_{D-Время}\right)$$

$$M_{D+1} := M_D + NK_D \cdot P(D) - NP_D \cdot K(D) - Расход - \text{if}\left(M_D > 0, Доход \cdot M_D, 0\right)$$

$$MMM_{D+1} := MMM_D + Доход \cdot M_D \qquad Max := \max(MMM) = 337.836\ млн_руб$$



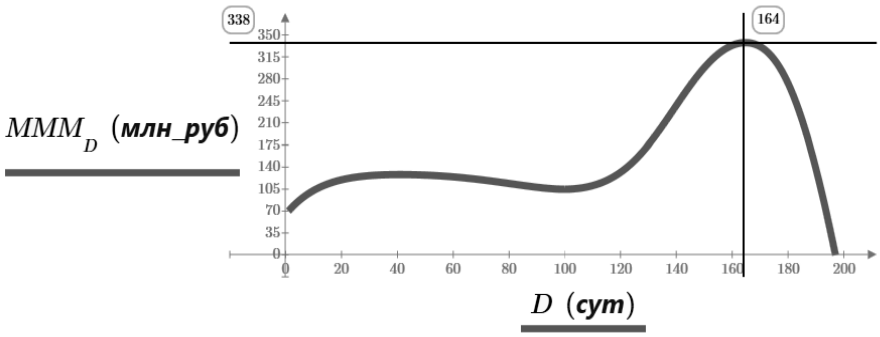


Рис. 15.18
Математическая модель финансовой пирамиды

Первый автор книги имел опыт участия в своеобразной финансовой пирамиде. Но очень давний, очень невинный и без особых последствий. В школьные годы

кем-то в классе было предложено не тратить выдаваемые родителями ежедневные обеденные 20 копеек, а складывать их и отдавать все разом поочередно каждому участнику этой финансовой кампании. Тут просматривалась старая как мир детская мечта: «*Вот если бы каждый взрослый дал бы мне по копеечке. Он бы от этого не обеднел, а у меня бы оказался целый капитал*». Но в классе вышло так, что участники складчины, получившие первыми деньги, из игры выбывали, она потихоньку глохла и кто-то (а среди них был и автор этой книги) оставался с носом.

Число жителей в городе $N := 1000000$

Коэффициент ажиотажа $KA := 10^{-7}$

Число людей, купивших акций в первый день $NK_1 := 7$

y(x) - число акций, купленных в момент времени t

Given

$$y(1) = NK_1 \qquad y'(t) = \left[KA\cdot\left(N - \int_1^t y(t)\,dt\right) - KA\cdot\int_1^t y(t)\,dt\right]\cdot y(t)$$

y := Odesolve(t, 365)

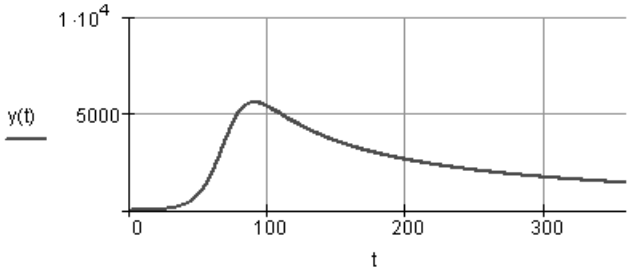


Рис. 15.19
Упрощенная математическая модель финансовой пирамиды (Mathcad 11)

Задания читателям

1. Придумать операторы для чисел 13, 14, 15, ..., 24, которые можно разместить на циферблате часов с суточным, а не с полусуточным (см. рис. 15.1) циклом.

2. Найти в справочниках или Интернете описание решения задачи коммивояжера муравьиным алгоритмом или методом отжига (an anneal — есть приложение к старым версиям Mathcad с одноименной функцией, решающей эту задачу — см. рис. 15.20).

3. Создать в среде Mathcad функцию, возвращающую сумму денег на банковском вкладе со сложным процентом. Такая функция, правда, уже есть в списке встроенных финансовой группы Mathcad. Проверить с помощью созданной функции утверждение из рассказа Чехова «Грач»: «Ведь ты пойми, дура: один рубль, положенный в банк по 5 сложных процентов, обращается: через 283 года в миллион! Высчитай-ка!»

ORIGIN := 1 n := 70 X := runif(n, 0, 100) Y := runif(n, 0, 100)

Путь := anneal(X, Y) $Путь_{n+1} := Путь_1$ i := 1.. n + 1

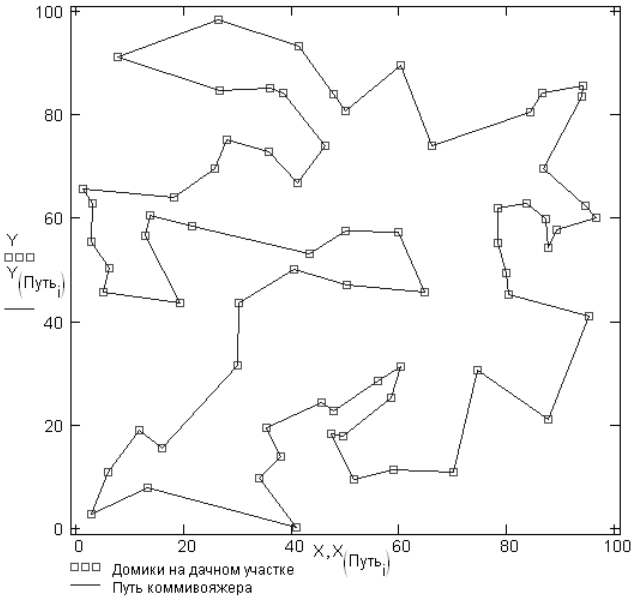


Рис. 15.20
Решение задачи коммивояжера с помощью функции anneal

Этюд 16

ОРЕЛ — РЕШКА, ИЛИ ТРЕХСТОРОННЯЯ ДУЭЛЬ В МОНТЕ-КАРЛО

В этюде будет рассказано о решении на компьютере задач теории вероятности, о методе статистических испытаний.

Математика: метод Монте-Карло, число π, погрешность вычислений, вероятность, нечеткие множества.

Физика: подбрасывание монетки и игральной кости.

Информатика: случайные и псевдослучайные числа, цикл с параметром, цикл с предпроверкой.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127813**

Есть такой старый анекдот. Студент просыпается утром, бросает монетку и загадывает: «Выпадет орел — посплю еще на правом боку, выпадет решка — лягу на левый бок, встанет монетка на ребро — пойду в институт, а зависнет в воздухе — займусь курсовой работой[1]».

Шутки шутками, но нам часто приходится подбрасывать монетку — реальную или виртуальную, чтобы случайным образом выбрать одну возможность из двух равновероятных. Футбольный судья, к примеру, перед матчем подбрасывает монетку, чтобы определить, какой команде отдать те или иные ворота. Шахматист перед игрой зажимает в одной руке белую пешку, а в другой — черную и предлагает сопернику выбрать «одну возможность из двух равновероятных» — определить, кто будет играть белыми фигурами, а кто — черными. Тасуя колоду карт, перемешивая костяшки домино, раскручивая рулетку, мешая в мешочке бочонки лото, мы опять же отдаем себя на волю случая. А от этого, от воли случая, может зависеть очень многое... Даже сама жизнь человека, если вспомнить название этой статьи. А почему не простая (двухсторонняя), а необычная (трехсторонняя) дуэль, да еще и в Монте-Карло?! Дело в том, что задача о трехсторонней дуэли описана во многих книгах, например в [75, 76], где приводится одно из частных решений этой задачи, полученное методом логических рассуждений. Мы же рассмотрим более полное решение этой задачи на компьютере методом Монте-Карло (методом статистических испытаний): смоделируем одиночную дуэль, проведем ее достаточно большое количество раз и для каждого участника подсчитаем число побед в этих дуэлях. Если эти числа поделить на общее число дуэлей, то результатом и будет искомая вероятность побед.

Но начнем мы с простых задач, бескомпьютерное решение которых известно.

Давайте вспомним нашего нерадивого студента из анекдота, с помощью компьютера подбросим монетку много-много раз и подсчитаем, сколько раз выпадет орел, а сколько — решка. На рисунке 16.1 показана соответствующая программа для компьютера[2] — функция `ОрелИлиРешка` с аргументом `n` (число бросаний монетки), возвращающая вектор с двумя элементами, хранящими вероятность выпадения орла или решки.

[1] «Вывешу задание по курсовой работе на форуме PTC Community и буду ждать, пока мне работу кто-нибудь не сделает!» — см. рассуждения на эту тему в [51].

[2] Она написана в среде математической программы Mathcad Prime. Ее можно бесплатно скачать с сайта **http://www.ptc.com/product/mathcad/free-trial**.

```
ОрелИлиРешка(n) := | Орел ← 0
                   | Решка ← 0
                   | for i ∈ 1..n
                   |   | if rnd(1) < 0.5
                   |   |   | Орел ← Орел + 1
                   |   | else
                   |   |   | Решка ← Решка + 1
                   | [Орел]
                   | [Решка] / n

ОрелИлиРешка(10000000) = [49.95552%]
                         [50.04448%]
```

Рис. 16.1
Программа «Орел — Решка»

Ядро программы, показанной на рис. 16.1, встроенная в Mathcad функция `rnd`, возвращающая случайное число[3] в интервале от нуля до значения аргумента функции `rnd` — в нашем случае до единицы. Если (`if`) функция `rnd(1)` вернет число меньше `0.5`, то будем считать, что выпал орел, иначе (`else`) выпала решка. Остается только в цикле `for` бросаний монетки подсчитывать, что выпало: `Орел ← Орел + 1` или `Решка ← Решка + 1`.

Можно, конечно, спросить, почему в программе на рис. 16.1 стоит `rnd(1) < 0.5`, а не `rnd(1) ≤ 0.5`. Замена оператора «меньше» на оператор «меньше или равно» ничего не изменит в расчете, т. к. выполнение условия `rnd(1) = 0.5` равносильно падению монетки… на ребро: вероятность генерацией функцией `rnd(1)` величины `0.5` практически равна нулю. Это можно проверить, заменив оператор «меньше» на оператор «равно» в программе на рис. 16.1 и подсчитав, сколько раз выпадет такой «орел» при достаточно большом числе бросаний монетки. Равенство же функции `rnd(1)` нулю или единице вообще (теоретически) невозможно («зависание монетки в воздухе»), т. к. функция `rnd(1)` генерирует случайные числа в интервале (0, 1), а не на отрезке [0, 1].

Если же нужно, чтобы выпадали целые случайные числа не в двух вариантах `0 – 1` (орел — решка, чет — нечет, да — нет и т. д.), а, например, в шести вариантах, то в руки берется не монетка, а игральная кость — кубик, с пронумерованными шестью гранями. Целые случайные числа от 1 до 36 генерирует рулетка[4], а числа от 1 до 90 — мешок с «бочонками» для игры в лото, когда из него вынимают первый боченок.

На рисунке 16.2 показана программа, с помощью которой методом Монте-Карло вычисляется вероятность выпадения чисел 2, 3, 4, …,12 при бросании двух игральных костей. В столбце справа вычислена теоретическая (классическая) вероятность такого выпадения, которая для удобства сравнения представлена также и в процентах (что, вообще говоря, не принято в теории вероятностей).

[3] Вернее, псевдослучайное число. Дело в том, что функция `rnd` при каждом новом ее вызове возвращает один и тот же ряд случайных чисел. При бросании реальной монетки так никогда не бывает, но при бросании виртуальной монетки на компьютере так специально запрограммировано, чтобы можно было отлаживать программы. В различных языках программирования есть специальные инструменты для получения и истинно случайных чисел. В этюде 5 мы рассмотрели один нестандартный способ получения истинно случайных чисел — см. рис. 5.15 на стр. 118.

[4] Европейская рулетка. Другие варианты этого «колеса фортуны» могут иметь другое число делений. Европейскую рулетку иногда называют «Чертовым колесом» из-за того, что сумма всех чисел на ней равняется 666. Но скорее потому, что эта азартная игра разорила многих людей, не знающих меры в игре. Недаром говорят: «Хочешь выиграть в казино — купи его!»

$$\mathbf{ORIGIN} := 2$$

$$\text{ДвеКости}(n) := \left| \begin{array}{l} \text{for } i \in 2..12 \\ \quad S_i \leftarrow 0 \\ \text{for } i \in 1..n \\ \quad \mathit{Кость1} \leftarrow \text{ceil}(\text{rnd}(6)) \\ \quad \mathit{Кость2} \leftarrow \text{ceil}(\text{rnd}(6)) \\ \quad S_{\mathit{Кость1}+\mathit{Кость2}} \leftarrow S_{\mathit{Кость1}+\mathit{Кость2}} + 1 \\ \dfrac{S}{n} \end{array} \right.$$

$$\text{ДвеКости}(10^7) = \begin{bmatrix} 2.77461\% \\ 5.55332\% \\ 8.32831\% \\ 11.11674\% \\ 13.8837\% \\ 16.67906\% \\ 13.89141\% \\ 11.10639\% \\ 8.34383\% \\ 5.5534\% \\ 2.76923\% \end{bmatrix} \qquad \begin{array}{l} \dfrac{1}{36} = 2.77778\% \\ \dfrac{2}{36} = 5.55556\% \\ \dfrac{3}{36} = 8.33333\% \\ \dfrac{4}{36} = 11.11111\% \\ \dfrac{5}{36} = 13.88889\% \\ \dfrac{6}{36} = 16.66667\% \end{array}$$

Рис. 16.2
Игра в кости

Центральным элементом программы, показанной на рис. 16.2, также является функция `rnd`, которая возвращает случайные вещественные (нецелые) числа в интервале от нуля до 6 (аргумент функции `rnd`). Другая встроенная в Mathcad функция — функция `ceil`[5] округляет эти числа до наибольшего целого: было `0.3` стало `1`, было `5.7`, стало `6` и т. д. Вложение функции `rnd` в функцию `ceil` (`ceil(rnd(6))`) позволяет нам смоделировать бросание одной игральной кости — генерацию целых случайных чисел из множества 1, 2, 3, 4, 5 и 6.

В программе на рис. 16.2 десять миллионов раз (как и в задаче о монетке) бросаются две игральные кости и подсчитывается, сколько раз выпало 2, 3, 4, 5, ... или 12. Эти выпадения суммируются в векторе с именем `s`, первый элемент которого имеет нумерацию два (это определяется оператором `ORIGIN:= 2`), а последний — 12. Это единственная программа книги, где системная переменная `ORIGIN` равна не нулю и не единице (читайте об этом в этюде 2).

Проанализируем полученные результаты. На рисунке 16.2 показаны вероятности выпадения двойки 2 (2.77...%), тройки (5.55...%) и т. д. после десяти миллионов бросаний двух костей. Эти вероятности можно подсчитать и без компьютера: одна кость имеет 6 граней, а две кости 36 (6^2) «граней». Двойка может выпасть только в одной комбинации: 1 + 1, тройка в двух комбинациях: 1 + 2 и 2 + 1, четверка в трех: 1 + 3, 2 + 2 и 3 + 1, пятерка в четырех: 1 + 4, 2 + 3, 3 + 2 и 4 + 1, шестерка в пяти: 1 + 5, 2 + 4, 3 + 3, 4 + 2 и 5 + 1, семерка в шести: 1 + 6, 2 + 5, 3 + 4, 4 + 3, 5 + 2 и 6 + 1, восьмерка в пяти: 2 + 6, 3 + 5, 4 + 4, 5 + 3 и 6 + 2, девятка в четырех: 3 + 6, 4 + 5, 5 + 4 и 6 + 3, десятка в трех: 4 + 6, 5 + 5 и 6 + 4, одиннадцать в двух: 5 + 6 и 6 + 5 и, наконец, двенадцать (как и двойка) в одной комбинации: 6 + 6.

[5] Ceil по-английски — это потолок. У нецелых чисел есть и «пол». Функция `floor` (пол, «напарница» функции `ceil`) округляет числа до наименьшего целого — было 1.2, стало 1, было минус 4.5 — стало минус 5 и т. д.

Отсюда видно, что при бросании двух игральных костей наибольшая вероятность выпадения у семерки (одна шестая), а наименьшая — у двойки и двенадцати (одна тридцать шестая). Об этом уже давно знали или догадывались опытные игроки в кости и выбирали правильную стратегию игры. И хотя значения классической вероятности выпадения, например, шести и восьми очков совпадают, компьютерный эксперимент дает не совпадающие результаты. И это вполне понятно почему.

Запуская на компьютере программы, показанные на рис. 16.1 и 16.2, мы фактически не моделировали бросание монетки или двух игральных костей, а... проверяли качество генератора случайных чисел программы Mathcad — добротность функции `rnd`[6]. Это «потребительское качество» генератора случайных чисел можно оценить и визуально. На рисунке 16.3 показано определение в среде Mathcad Prime методом Монте-Карло значения числа π (отношения длины окружности к ее диаметру). Задача решается так — берется квадрат, в который случайным образом «бросаются» точки и подсчитывается, сколько точек попало в круг, вписанный в данный квадрат. Отношение числа точек, попавших в круг, к общему числу точек при достаточно большом числе бросании должно стремиться к отношению площади круга ($\pi d^2/4$) к площади квадрата (d^2)[7], то есть к величине $\pi / 4$.

В примере с кругом и квадратом работает уже геометрическая вероятность: если внутри одного геометрического объекта расположен другой геометрический объект, то вероятность попадания в меньший объект случайно брошенной точки считается равной отношению меры меньшего объекта к мере большего объекта. Под мерой понимают: в случае линии — длину, в случае плоской фигуры — площадь, в случае тела — объем.

Как-то студентам было дано задание вычислить площадь фигуры со сложной конфигурацией границы двумя способами: прямым интегрированием (точное вычисление) и методом Монте-Карло (приближенное вычисление), используя таблицу случайных чисел для определения координат «падающих» на фигуру и мимо фигуры точек. У одного студента такой таблицы под рукой не оказалось. Тогда он взял фломастер и с закрытыми глазами стал ставить точки на рисунок. Для чистоты эксперимента он попросил сделать это же самое и своего соседа. В итоге задание было выполнено правильно, относительная погрешность вычисления площади методом статистических испытаний оказалась равной примерно 2%.

На рисунке 16.3 показан Mathcad-расчет числа π методом Монте-Карло. Здесь задействована функция `runif` — модификация функции `rnd`. Функция `runif` возвращает вектор с `n` элементами, содержащий случайные числа в интервале, заданным вторым и третьим аргументом функции `runif`. При решении задачи о приближенном значении числа π мы сгенерировали два вектора `X` и `Y` — координаты `n` точек, брошенных случайным образом в квадрат со стороной 2. Остается только подсчитать число точек (`No`), попавших в круг, выделить координаты этих точек в вектора `Xo` и `Yo` и нарисовать их на графике. Из графика видно, что точки равномерно «размазаны» по квадрату, что свидетельствует о хорошем качестве генератора случайных чисел, встроенного в Mathcad. На сайте **http://communities.ptc.com/videos/1480** можно увидеть анимацию процесса определения числа π методом Монте-Карло. Она была опубликована авторами на 14 марта неважно какого года. Дело в том, что в этот день во многих странах отмечают как день математика (3.14 — март, четырнадцатый день). В этот день в школах

[6] В среде Mathcad есть другие функции, генерирующие случайные числа на заданном интервале не только равномерно, но и по другим законам распределения — по закону нормального распределения, например, график которого обычно рисуют в виде колокола или холма: в центре интервала вероятность наступления некоего события самая высокая, а на краях интервала — низкая.

[7] Так иногда оценивают площадь облаков на фотографии участка земли, сделанной из космоса: тыкают случайным образом иголкой в картинку и подсчитывают число попаданий в облако. Так же на топографической карте можно вычислить площадь леса или поляны, насыпав на карту, например, пшено или гречку и пересчитав зерна.

проводятся специальные математические уроки, рассказывается о важности математики в учебном процессе и жизни. Кстати, 6 февраля кое-где отмечают как день химика: 2.06 — это начало числа Авогадро (альтернативная дата — 23 октября, если учитывать не мантиссу, а показатель степени этой константы 10^{23}). Но у нас в стране день химика (профессиональный праздник) — это последнее воскресенье мая. А на какой день можно назначить день физика? Можно на девятое августа (9.8 — ускорение свободного падения), но в этот день, как правило, все физики (школьные и вузовские преподаватели, например) в отпуске, а можно и 8 сентября (если прочесть это число в обратном порядке, как и приближение числа π). Но лучше 6 июня (6.6) — начало гравитационной постоянной, постоянной Планка и... летней экзаменационной сессии! День теплотехника [2] можно отмечать 19 апреля, вспоминая то, что в калории 4 целых 19 сотых джоуля (см. рис. 16.4).

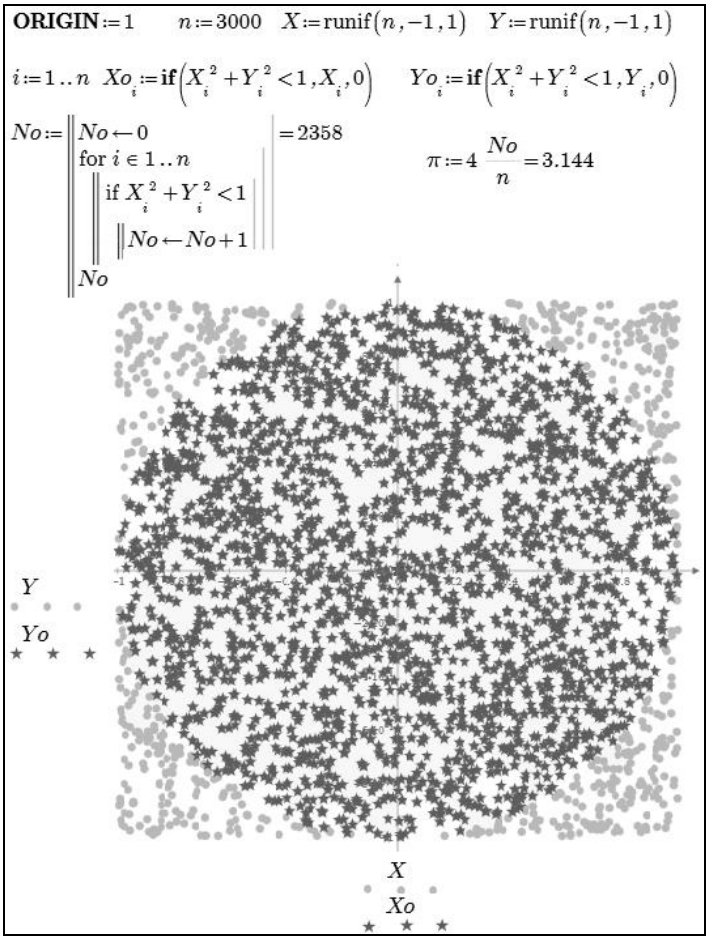


Рис. 16.3
Число π и Монте-Карло

Все эти константы встроены в Mathcad — см. рис. 16.4.

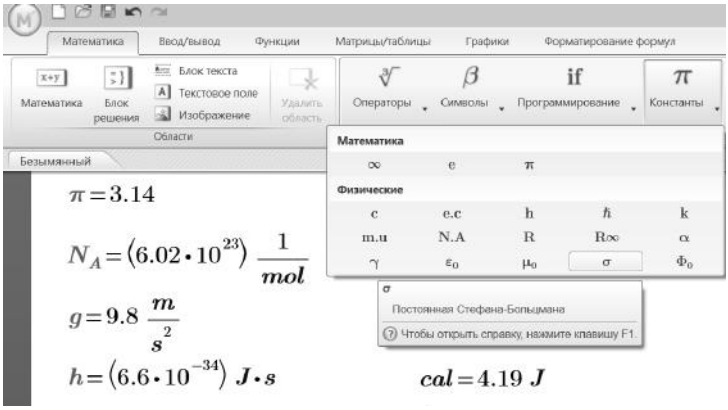


Рис. 16.4
Встроенные праздничные константы Mathcad

Но оставим константы и вернемся к случайным числам.

Функция `rnd`, использованная нами для подбрасывания монетки и игральной кости, также поможет нам решить на компьютере задачу о трехсторонней дуэли. Вот ее описание из книги [76].

Сэм, Билл и Джон договорились сразиться на дуэли втроем по следующим правилам:

- жеребьевка определяет, кто стреляет первым, вторым и третьим;
- дуэлянты располагаются на одинаковых расстояниях друг от друга (по углам равностороннего треугольника);
- обмениваются выстрелами по очереди, определенной жребием, пока двое не будут убиты;
- очередной стреляющий может стрелять в любого из живых.

Известно, что Сэм — снайпер и никогда не промахивается с данной дистанции, Билл поражает мишень в 80% случаев, а Джон — в 50%. Какова наилучшая стратегия для каждого из участников и каковы вероятности их выживания, если они следуют оптимальным стратегиям?

В этой дуэли у Сэма (снайпер) и Билла (хороший стрелок) могут быть две тактики поведения: случайная, когда стреляющий ничего не знает о меткости соперников и целит в первого подвернувшегося — загадывает цель и подбрасывает монетку (см. рис. 16.1). Вторая тактика (бей в меткого), когда дуэлянту известно о том, кто как стреляет, и он метит в соперника с наивысшими стрелковыми качествами в надежде остаться tête-a-tête с наихудшим стрелком.

Джон может следовать еще одной, третьей, «хитрой» тактике. Чтобы получить наивысшие шансы выйти победителем из дуэли, он должен намеренно стрелять мимо цели, пока двое его соперников живы. Ведь очередной стреляющий, если он будет придерживается тактики «бей в меткого», будет бить не в Джона, а в другого противника. После того как Сэм или Билл будет убит, Джон должен показать все, на что он способен. В такой ситуации его шансы выжить составляют 50%, если он остался наедине с Сэмом, и более 50% — если с Биллом. И в итоге шансы выжить у Джона будут выше, чем у Сэма и Билла.

Давайте смоделируем эту трехстороннюю дуэль на компьютере.

Участник дуэли, прежде чем выстрелить, должен, во-первых, зафиксировать («взять на мушку») самого меткого соперника, в которого нужно стрелять, придерживаясь тактики «бей в меткого». Для этого предназначена функция `СамыйМеткий` (рис. 16.5), возвращающая номер противника с наивысшими стрелковыми качествами. В ней перебором всех участников дуэли (цикл `for`) учитываются, естест-

венно, только живые противники (`Статус`$_{Цель}$ = «жив») и не сам стреляющий (`Цель` ≠ `Стрелок`).

```
СамыйМеткий(Меткость, Статус, Стрелок) := | "Определение самого меткого стрелка"
                                          | ВысшаяМеткость ← 0
                                          | for Цель ∈ 0..2
                                          |   if Статус_Цель = "жив"
                                          |     if Цель ≠ Стрелок
                                          |       if Меткость_Цель > ВысшаяМеткость
                                          |         СамыйМеткий ← Цель
                                          |         ВысшаяМеткость ← Меткость_Цель
                                          | СамыйМеткий
```

Рис. 16.5
Функция «Самый меткий»

Участник дуэли, придерживающийся хитрой тактики, перед выстрелом должен определить, сколько противников стреляют лучше его. Эту работу выполняет функция `Меткие` (рис. 16.6). В нее заложен такой же алгоритм перебора противников, как и в функции `СамыйМеткий`.

```
Меткие(Меткость, Статус, Стрелок) := | "Определение числа противников, стреляющих лучше"
                                     | ЧислоМетких ← 0
                                     | for Цель ∈ 0..2
                                     |   if Статус_Цель = "жив"
                                     |     if Цель ≠ Стрелок
                                     |       if Меткость_Цель > Меткость_Стрелок
                                     |         ЧислоМетких ← ЧислоМетких + 1
                                     | ЧислоМетких
```

Рис. 16.6
Функция «Меткие»

Функции `СамыйМеткий` и `Меткие` в качестве аргументов имеют вектор `Меткость`, вектор `Статус` и скаляр `Стрелок`.

Функция `Победитель` (рис. 16.7) возвращает номер победителя в одиночной дуэли. При этом учитывается меткость и тактика каждого участника дуэли (два вектора-аргумента функции `Победитель`).

Что происходит в функции `Победитель`?

В начале дуэли все участники живы: все три элемента вектора `Статус` принимают значение «жив», а переменная `Убийство` (счетчик числа убийств) обнуляется. Далее проводится жеребьевка: определяется первый стреляющий[8] (переменная `Стрелок`) и направление очередности выстрелов. Если переменная Очередь будет равна единице, то очередность идет в таком направлении … 0 → 1 → 2 → 0 → 1 → 2…(…Сэм → Билл → Джон → Сэм → Билл → Джон…), если минус единице — в таком …0 → 2 → 1 → 0 → 2 → 1 (…Сэм → Джон → Билл → Сэм → Джон → Билл…).

Математическая модель дуэли опирается на цикл с выходом из середины (`while 1… return…`): дуэль продолжается до тех пор, пока не будут сделаны два результативных выстрела — пока переменная `Убийство` не станет равной двум. В тело цикла `while` вложено три ветви, определяемые тактикой очередного стреляющего:

[8] Тут бы пригодилась монетка (см. рис. 16.1), но не с двумя, а с тремя сторонами. Такой монетки в природе нет, но мы ее смоделировали оператором `floor(rnd(3))`. Кстати, «монетка» с четырьмя сторонами — это тетраэдр, а «монетка» с шестью сторонами-гранями — это игральная кость (см. рис. 16.2).

«случайная», «бей в меткого» и «хитрая». В каждой ветви расчета определяется `Цель` — либо случайный, либо самый меткий противник, которого убивают ($\text{Статус}_{\text{Цель}} \leftarrow$ "убит"), если, во-первых, не промахиваются $\text{Меткость}_{\text{Стрелок}} > \text{rnd}(1)$ и, во-вторых, не стреляют намеренно в воздух. Второе имеет место при хитрой тактике стреляющего ($\text{Тактика}_{\text{Стрелок}}$ = "хитрая") и если метких противников более одного.

```
Победитель(Меткость, Тактика) :=  "Моделирование трехсторонней дуэли"
                                  [ Статус ← [ "жив" ]      Убийство ← 0 ]
                                                [ "жив" ]
                                                [ "жив" ]
                                  [ Стрелок ← floor(rnd(3))      Очередь ← if(rnd(1) > 0.5, 1, −1) ]
                                  while 1
                                    if Тактика_Стрелок = "случайная"
                                      while 1
                                        Цель ← floor(rnd(3))
                                        if Цель ≠ Стрелок
                                          if Статус_Цель = "жив"
                                            break
                                      if Меткость_Стрелок > rnd(1)
                                        [ Статус_Цель ← "убит"  Убийство ← Убийство + 1  if Убийство = 2 ]
                                                                                          return Стрелок
                                    if Тактика_Стрелок = "бей в меткого"
                                      Цель ← СамыйМеткий(Меткость, Статус, Стрелок)
                                      if Меткость_Стрелок > rnd(1)
                                        [ Статус_Цель ← "убит"  Убийство ← Убийство + 1  if Убийство = 2 ]
                                                                                          return Стрелок
                                    if Тактика_Стрелок = "хитрая"
                                      Цель ← СамыйМеткий(Меткость, Статус, Стрелок)
                                      if Меткие(Меткость, Статус, Стрелок) < 2
                                        if Меткость_Стрелок > rnd(1)
                                          [ Статус_Цель ← "убит"  Убийство ← Убийство + 1  if Убийство = 2 ]
                                                                                            return Стрелок
                                    while 1
                                      Стрелок ← Стрелок + Очередь
                                      [ if Стрелок > 2 | if Стрелок < 0 | if Статус_Стрелок = "жив" ]
                                        Стрелок ← 0    | Стрелок ← 2   |   break
```

Рис. 16.7
Функция «Победитель»

Определение следующего стреляющего ведется также в цикле с выходом из середины (`while 1 ... break`): цикл прерывается, когда, перебирая очередь, отмеченную выше (...0 → 1 → 2 → 0 → 1 → 2.. или ...0 → 2 → 1 → 0 → 2 → 1...), «натыкаются» на живого участника.

Функция `Победитель` возвращает непредсказуемое целочисленное значение 0, 1 или 2, так как в ней вызывается уже рассмотренная нами встроенная в Mathcad функция `rnd`, которая возвращает псевдослучайное число в интервале от нуля до значения аргумента функции `rnd`. Этот аргумент у нас равен либо единице (случайный выбор очередности выстрелов и имитация выстрела с заданной вероятностью попадания, пропорциональной меткости стреляющего), либо трем (случайный выбор первого стреляющего.

Функция `ВероятностьПобеды` (рис. 16.8) возвращает вектор, элементы которого — это отношение числа побед каждого участника дуэли к общему количеству дуэлей (третий аргумент функции `ВероятностьПобеды`; два первых аргумента-вектора — это параметры дуэлянтов: их меткость и тактика), то есть вероятность победы.

$$\text{ВероятностьПобеды}(\textit{Меткость}, \textit{Тактика}, \textit{ЧислоДуэлей}) := \left| \begin{array}{l} \text{“Проведение серии дуэлей”} \\ \textit{Победы} \leftarrow \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ \text{for } \textit{Дуэль} \in 1..\textit{ЧислоДуэлей} \\ \quad \left| \begin{array}{l} \textit{Победитель} \leftarrow \text{Победитель}(\textit{Меткость}, \textit{Тактика}) \\ \textit{Победы}_{\textit{Победитель}} \leftarrow \textit{Победы}_{\textit{Победитель}} + 1 \end{array} \right. \\ \dfrac{\textit{Победы}}{\textit{ЧислоДуэлей}} \end{array} \right.$$

Очередность выстрелов случайная

$$\text{ВероятностьПобеды}\left(\begin{bmatrix} 1 \\ 0.8 \\ 0.5 \end{bmatrix}, \begin{bmatrix} \text{“бей в меткого”} \\ \text{“бей в меткого”} \\ \text{“хитрая”} \end{bmatrix}, 10^6\right) = \begin{bmatrix} 29.92\% \\ 17.81\% \\ 52.27\% \end{bmatrix}$$

Теория

$$\begin{bmatrix} 30\% \\ 17.778\% \\ 52.222\% \end{bmatrix}$$

Очередность выстрелов Сэм - Билл - Джон. Первым стреляет Сэм.

$$\text{ВероятностьПобеды}\left(\begin{bmatrix} 1 \\ 0.8 \\ 0.5 \end{bmatrix}, \begin{bmatrix} \text{“бей в меткого”} \\ \text{“бей в меткого”} \\ \text{“хитрая”} \end{bmatrix}, 10^6\right) = \begin{bmatrix} 50.02\% \\ 0 \\ 49.98\% \end{bmatrix}$$

Очередность выстрелов Билл - Джон - Сэм. Первым стреляет Билл.

$$\text{ВероятностьПобеды}\left(\begin{bmatrix} 1 \\ 0.8 \\ 0.5 \end{bmatrix}, \begin{bmatrix} \text{“бей в меткого”} \\ \text{“бей в меткого”} \\ \text{“хитрая”} \end{bmatrix}, 10^6\right) = \begin{bmatrix} 9.99\% \\ 35.53\% \\ 54.47\% \end{bmatrix}$$

Рис. 16.8
Проведение серии дуэлей

Теперь, когда все необходимые функции сформированы, можно проводить *статистические испытания* (см. рис. 16.8) и фиксировать вероятности побед участников дуэли, исходя из их меткости и тактики. Если увеличивать число дуэлей, то, набравшись терпения, можно получить результат, близкий к теоретическому.

Задача о трехсторонней дуэли, как мы уже отметили, приводится во многих книгах. И что интересно — она там решается не совсем верно. Априори считается, что в этой дуэли самый слабый стрелок Джон имеет наихудшие шансы выжить. Но если он немного подумает (хитрая тактика), то вероятность выйти победителем у него становится самой высокой (52.222...%).

Но наше решение (см. рис. 16.8) говорит о том, что у Джона и так самые высокие шансы выжить (44–46%). Начиная хитрить, он мало чего выигрывает, но подводит Билла.

Откуда такая ошибка в постановке задачи? Дело в том, что у дуэлянтов есть еще одна тактика. Если участники дуэли ничего не знают о стрелковых качествах соперников, то они бьют в первого попавшегося. Здесь вероятность побед можно подсчитать сразу без компьютера: Сэм — 43.48% (1 / (1 + 0.8 + 0.5)), Билл — 34.78% (0.8 / (1 + 0.8 + 0.5)) и Джон — 21.74% (0.5 / (1 + 0.8 + 0.5)) или 100 – 43.48 – 34.78). Наша программа может обсчитать и этот случай.

Задачу подправили, теперь ее можно развить.

Подсчитанная нами вероятность побед относится к ситуации, когда еще не проводилась жеребьевка по очередности выстрелов: переменная `Очередь` у нас случайно равна либо единица, либо минус единица.

Но после жеребьевки шансы Сэма и Билла резко меняются. Дела Билла становятся совсем уж плохи (10–12%), если Джон после своего намеренного промаха

передает право выстрела не ему (`Очередь = -1`), а Сэму (`Очередь = 1`). И наоборот: Сэм может потерять свои 30%, если после намеренного промаха Джона Билл будет стрелять в Сэма. У Джона вероятность победы (52.2 (2)%) не зависит от очередности выстрелов.

Можно придумать и проанализировать четвертую тактику ведения дуэли: Билл и Джон *сговариваются* целить в Сэма, убить его, раз он такой меткий, а уж потом выяснять отношения между собой. Инициатором такого сговора, как понимает читатель, скорее всего, будет Билл. Джон пойдет на него, если не смоделирует дуэль на компьютере и не узнает, что из этого может получиться.

Еще одно задание читателю: доработать программу имитации дуэли так, чтобы она была пригодна для дуэли с любым числом участников.

Наша модель — не такая уж оторванная от жизни. Дуэли в чистом виде сейчас, к счастью, не проводятся. Но, кстати, можно дуэль проводить в формате игры пейнтбол, когда соперники стреляют друг в друга шариками с краской. Какое-то подобие дуэли со сговором участников наблюдается на рынках, включая финансовые. Кровь там не льется, но случаются инфаркты, лопаются компании, банки, разоряются люди и даже целые страны (Греция, Испания, Италия, Кипр, Исландия и т. д., если говорить о современной ситуации).

Теория игр, тактика поведения участников — это не только интересная, но и очень полезная штука. Недаром в 1998 г. лауреатами Нобелевской премии по экономике стали ученые, применившие теорию игр к анализу работы финансовых бирж.

И все-таки сама модель чересчур искусственна. Что такое меткость дуэлянта и как ее определить? Проводить реальные статистические испытания? Кстати, именно они и проводятся при опытных стрельбах из различных видов испытываемого оружия. Есть очень известная задача о вероятности крушения самолета при попадании пуль в различные части его корпуса. Самая высокая вероятность крушения, указанная там, — при попадании пуль в крыло. Человек, далекий от авиации, этой ситуации понять не может: ну, что может сделать маленькая дырочка в крыле?! А специалист понимает сразу: ведь в крыльях размещены топливные баки! Так и с выстрелом в человека на дуэли — нужно не просто попасть и ранить, нужно попасть в жизненно важный орган (и это не обязательно сердце).

К тому же одно дело стрелять по мишеням, а другое — целить в живого человека. На дуэлях, как правило, не убивают наповал, а ранят с различной степенью тяжести. Подстреленный дуэлянт, если хватало сил и злости, стрелял в противника (дуэль Пушкина и Дантеса, например). Попытки «приземлить» задачу о дуэлях неизбежно потребуют привлечения аппарата *теории нечетких множеств* (ТНМ — см. этюд 12).

Меткость дуэлянта — величина нечеткая, «размытая». Никто и нигде не измеряет ее числами, а только оценивает категориями (лингвистическими константами): «мазила», «хороший стрелок», «снайпер» и т. д. Статус дуэлянта — это никакая не булева переменная. Вспомним «консилиум врачей» у лежащего без чувств Буратино: «Пациент скорее мертв, чем жив», — «Нет, пациент скорее жив, чем мертв»...

Задание читателям

Смоделировать и решить на компьютере задачу о поединке более трех дуэлянтов.

Этюд 17

ЛЕТЯТ ГУСИ, ИЛИ ЗАДАЧА О РЫБАКАХ И РЫБКЕ

В этюде будет рассказано, как компьютер помог по-новому решить старинную английскую народную загадку.

Математика: уравнения в целых числах, система трех линейных уравнений.
Физика: рыбалка.
Информатика: цикл с предпроверкой, цикл с параметром, флаг (признак) в программе.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127812**

Есть такая «русская народная» задача-загадка: «Летит гусь. Навстречу ему — стая гусей. «Здравствуйте, 100 гусей», — говорит он им. Они отвечают: «Нас не 100 гусей; вот если бы нас было столько, сколько сейчас, да еще столько, да еще полстолька и четверть-столько, да еще ты, вот тогда нас было бы 100 гусей». Сколько гусей летит в стае?» Имеется в виду, что эту задачу нужно решить в уме, мобилизовав свои способности к устному счету. Подобные задачи мы рассматривали в этюде 6. Но в настоящее время при решении подобных задач все чаще и чаще используют калькулятор или даже компьютер [77, 78].

На рисунке 17.1 показано решение этой задачи с использованием команды `solve` (решить уравнение аналитически) математической программы Mathcad. Человеку достаточно только составить алгебраическое уравнение `(x + x + x/2 + + x/4 + 1 = 100)`, ввести его в компьютер и приказать программе выполнить нужную команду — решить данное уравнение[1] и получить ответ — 36 гусей.

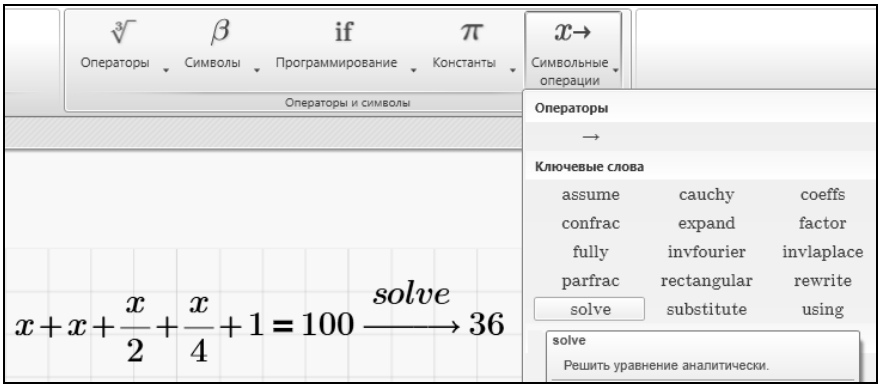


Рис. 17.1
Решение задачи о гусях в среде Mathcad Prime

Подобные задачи-загадки есть у всех народов мира. Английским аналогом задачи о гусях можно считать задачу о рыбаках и рыбке: «Три рыбака легли спать, не поделив улова. Проснувшийся ночью первый рыбак решил уйти, взяв свою долю. Но число рыб не делилось на три. Тогда он выбросил одну рыбу, а из остатка забрал треть. Второй и третий рыбаки поступили аналогичным образом. Спрашивается, какое наименьшее количество рыб может удовлетворить условию задачи».

По условию задачи составим уравнения, считая все неизвестные натуральными числами $x-1=3y$, $2y-1=3z$, $2z-1=3t$, причем x не кратно трем (исходное

[1] В нашем «гусином» уравнении только одна переменная `x`. Но если переменных в уравнении две и более, то за ключевым словом `solve` нужно будет через запятую указать, по какой переменной (неизвестной) решается данное уравнение.

число рыб нельзя было разделить на три). Мы получили систему трех линейных уравнений с четырьмя натуральными неизвестными. Найти ответ у этой задачи через составление и решение алгебраического уравнения, как это показано на рис. 17.1, так просто не получится. Во-первых, нужно найти не просто решение, а *целочисленное* решение. В задаче о гусях условие было подобрано так, чтобы получилось именно целочисленное решение — «целых» 36 гусей, а не «гусей с половинками». Половинки и прочие доли часто получаются при неверном решении целочисленных задач. Вспомним, как Виктор Перестукин из мультфильма «Страна невыученных уроков» решил задачу по арифметике и получил... полтора землекопа.

Во-вторых, полученная система имеет бесконечно много решений.

А в-третьих, нас интересует решение, при котором число x будет наименьшим.

Можно пойти следующим путем. Если уравнения записать в виде $x=3y+1$, $2y=3z+1$, $2z=3t+1$, то мы увидим, что t должно быть нечетным числом, т. е. $t=2k-1$ при $k\geq 1$. Но тогда $z=3k-1$ и $2y=9k-2$, что может быть лишь при четном k. Теперь если положить $k=2n$ при $n\geq 1$, то можно выразить все неизвестные: $x=27n-2$, $y=9n-1$, $z=6n-1$, $t=3n-1$. Очевидно, что наименьшее значение x равно 25, причем оно получается при $n=1$.

Но это решение могут получить лишь те, кто догадается применить свойства четности. А если мы про четность забыли, то попробуем решить систему методом перебора по переменной $t=3n-1$, решая при каждом фиксированном значении $t\in\{1,2,3,\ldots\}$ систему трех линейных уравнений уже с тремя неизвестными

$$\begin{cases} x-3y=1, \\ 2y-3z=1, \\ 2z=3t+1. \end{cases}$$

Заметим, что с ростом t увеличиваются все неизвестные, а мы ищем наименьшее число x.

При каждом значении t эта система имеет единственное решение, которое легко получить. Это мы и предоставляем сделать читателю.

Намного сложнее решить аналогичную задачу, если рыбаков не три, а больше. Ведь тогда и число неизвестных увеличится!

Но есть еще один способ решения задачи. И он не требует составления уравнений. Мало того, он позволяет решить задачу при любом числе рыбаков.

Когда-то давно один из авторов этой статьи вел занятия со школьниками по программированию и поручил им составить компьютерную программу решения задачи о рыбаках и рыбке по несложному алгоритму: задается первое приближение к решению (30 рыб, например) и далее проверяется, удовлетворяет ли это число условию задачи. Если нет, то предположение уменьшается на единицу, а сама проверка повторяется до тех пор, пока не будет выполняться условие задачи.

В языках программирования этот несложный алгоритм реализуется циклом с проверкой условия задачи и вложенным в него циклом перебора трех рыбаков, забирающих свою долю улова. На рисунке 17.2 показано, как этот алгоритм записывается на языке программирования, встроенном в пакет Mathcad Prime. Программа реализована в виде функции пользователя с именем `Рыбы` с двумя аргументами: `Ответ` — первое приближение к ответу и `рыбаки` — число рыбаков, участвующих в дележе улова[2].

[2] Читатели могут усложнить задачу — ввести, например, еще один аргумент: сколько рыб выбрасывает или, наоборот, подлавливает каждый рыбак перед своим уходом «по-английски», не попрощавшись...

```
Рыбы(Ответ, рыбаки) := | "Задача о рыбаках и рыбке"
                       | Ответ ← Ответ + 1
                       | Поделили ← "нет"
                       | while Поделили = "нет"
                       |   | Ответ ← Ответ − 1
                       |   | Улов ← Ответ
                       |   | for рыбак ∈ 1..рыбаки
                       |   |   | Улов ← Улов − 1 − (Улов − 1)/рыбаки
                       |   |   | Поделили ← if(Улов ∈ ℤ, "да", "нет")
                       | Ответ

Рыбы(30, 3) = 25        Рыбы(24, 3) = −2        Рыбы(−3, 3) = −29
Рыбы(300, 4) = 253      Рыбы(252, 4) = −3       Рыбы(−4, 4) = −259
Рыбы(4000, 5) = 3121    Рыбы(3120, 5) = −4      Рыбы(−5, 5) = −3129
Рыбы(50000, 6) = 46651  Рыбы(46650, 6) = −5     Рыбы(−6, 6) = −46661
Рыбы(900000, 7) = 823537 Рыбы(823536, 7) = −6   Рыбы(−7, 7) = −823549
```

Рис. 17.2
Решение задачи о рыбаках и рыбке в среде Mathcad Prime

Алгоритм программы несложный: в переменную `Поделили` заносятся слова «`да`» или «`нет`» в зависимости от того, будет ли переменная `Улов` (число рыб, остающееся после ухода очередного рыбака) принадлежать (см. оператор ∈ на рис. 17.2) множеству целых чисел `Z`[3]. И все это, повторяем, скомпоновано в цикл `while`, в который вложен цикл `for`.

На рисунке 17.2 также показаны результаты вызова функции `Рыбы` при разных начальных приближениях и разном числе рыбаков[4]. Предыстория этих ответов такова. Школьники в вышеупомянутой группе по изучению программирования в задаче о трех рыбаках взяли в качестве первого приближения 30 рыб и получили традиционный английский ответ 25 рыб: первый рыбак взял 8 рыб и оставил 16 рыб, второй рыбак взял 5 рыб и оставил 10 рыб и, наконец, третий рыбак взял 3 рыбы и оставил 6 рыб. Все по-честному! Такое решение давали многие поколения английских детей и взрослых, пока не появился... Поль Дирак. Этот английский физик прославился не только тем, что придуман античастицы, но и тем, что додумался до... антирыб. Поль Дирак сказал, что правильное решение задачи о рыбаках и рыбке не 25 рыб, а... минус 2 рыбы: выбрасываем из улова одну — получаем минус три рыбы, забираем треть, оставляем минус 2 рыбы и так до бесконечности. Для проверки решения Дирака школьники взяли в качестве первого приближения 24 рыбы и получили ответ Дирака — минус две рыбы. А один школьник не поленился[5] и ввел еще одно первое приближение — минус 3 рыбы и получил... еще один ответ — минус 29 рыб: выбрасываем одну — получаем минус 30 рыб, забираем треть, оставляем минус 20 рыб, выбрасываем одну — получаем минус 21 рыбы, оставляем минус 14 и, наконец, выбрасываем одну — получаем минус 15 рыб. Так школьник поправил самого Поля Дирака! Дирак оказался неправ: у этой задачи есть и другие решения и их бесконечное множество. В традиционной постановке

[3] В среде Mathcad Prime кроме множества `Z` (целые числа) определены еще три множества чисел: `C` — комплексные числа, `Q` — рациональные числа и `R` — вещественные числа. Об этом писалось в этюде 12 — см. рисунок 12.1 на стр. 273.

[4] Для трех рыбаков ответ можно найти и без компьютера, а для четырех и большего числа рыбаков это сделать весьма затруднительно.

[5] Не поленился, руководствуясь правилом: «Много ты, компьютер, о себе воображаешь! Попробуем, проглотишь ли ты вот это!»

задачи по умолчанию имелось в виду, что число рыб — это целое *положительное*[6] число. Дирак же перешагнул ноль и открыл некий ящик Пандоры, лишив задачу единственного ответа. Поль Дирак, давая свой ответ, подразумевал в постановке задачи нахождение минимального *по модулю* числа рыб. А этот «модуль» (абсолютное значение) подразумевает в ответе эти самые антирыбы.

Интересно, что было время, когда отрицательные числа отвергались математиками как ненужные, посторонние и бесполезные. Даже сам Франсуа Виет свою знаменитую теорему, устанавливающую связь коэффициентов многочлена с его корнями и опубликованную им в 1591 г., доказывал исключительно при условии, что все корни многочлена положительны. Отрицательных чисел Виет не признавал, хоть он был родоначальником алгебраических символьных вычислений и преобразований. Итальянские математики XVI в. — Тарталья, Кардано, Бомбелли — одни из первых, кто учитывал помимо положительных еще и отрицательные корни многочленов.

Итак, в среде Mathcad нет инструментов целочисленного решения алгебраических уравнений. Но они есть в другой математической программе — Maple.

Задача о рыбаках и рыбке

Осталось рыб после ухода первого рыбака

$$Ост1 := n - 1 - \frac{n-1}{3}$$

$$\frac{2}{3}n - \frac{2}{3} \quad (1)$$

Осталось рыб после ухода второго рыбака

$$Ост2 := Ост1 - 1 - \frac{Ост1 - 1}{3}$$

$$\frac{4}{9}n - \frac{10}{9} \quad (2)$$

Осталось рыб после ухода третьего рыбака

$$Ост3 := Ост2 - 1 - \frac{Ост2 - 1}{3}$$

$$\frac{8}{27}n - \frac{38}{27} \quad (3)$$

isolve(Ост3 = m)

$$\{m = 6 + 8_Z1, n = 25 + 27_Z1\} \quad (4)$$

Рис. 17.3

Решение задачи о рыбаках и рыбке в среде Maple

На рисунке 17.3 показано решение задачи о рыбаках и рыбке в среде другой популярной математической программы Maple, где сформулированы условия задачи (действия 1 – 3) и отдана команда найти значение `m` и `n`, при которых остаток улова после его дележа был бы целым числом (действие 4). Для этого в среде Maple была отдана команда не просто решить (`solve`), а решить целочисленно — `isolve` (i — integer, целочисленный).

Программа Maple выдала ответ не в виде конкретного числа, а виде бесконечного ряда целых чисел, где переменная `_Z1` означает уже рассмотренное нами в программе на рис. 17.2 множество всех целых чисел `Z` ...–3, –2, –1, 0, 1, 2, 3... Если их подставить в ответ (см. рис. 17.3), то получится требуемое множество ответов для `m` (остаток рыб после ухода третьего рыбака): `m = …-9, -5` (ответ школьника), `-1` (ответ Поля Дирака), `3` (традиционный английский ответ), `7, 11, 15…` и `n` (начальное число рыб в улове) `= …-56, -29` (ответ школьника), `-2` (ответ Поля Дирака), `25` (традиционный английский ответ), `52, 79, 106…`

[6] Древние люди знали только положительные числа — работали только с правой частью числового ряда, не имея понятия об отрицательных числах. Отголоски этой «древности» можно усмотреть и в фольклоре — в наших задачах о гусях и рыбах.

В решении, показанном на рис. 17.2, задавалось начальное приближение, заведомо *большее* ответа. Потом от этого числа *отнималась* единица в каждом такте выполнения программного цикла. Можно изменить направление приближения к ответу — задать начальное приближение, заведомо *меньшее* ответа и *прибавлять* единицу в цикле. Но в этом случае антирыб мы вряд ли получили.

Задания читателям

1. Выудить из книги или Интернета старинную задачу и решить ее с помощью Mathcad.

2. Создать программу для задачи с любым числом рыбаков, выбрасывающих при дележе любое количество пойманных рыб.

Этюд 18

БУКВЕННАЯ ГОЛОВОЛОМКА, ИЛИ НА КОМПЬЮТЕР НАДЕЙСЯ, А САМ НЕ ПЛОШАЙ!

В этюде будет рассказано о том, как можно сочетать свой интеллект с возможностями компьютера.

Математика: десятичная запись числа, система уравнений в целых числах.
Физика: географическая карта.
Информатика: вложенный цикл с параметром, условный оператор, матрицы, выдача результата.

Сайт этюда: **https://www.ptcusercommunity.com/thread/127811**

Когда-то давно в журнале «Квант» был опубликован такой буквенный ребус:

$$\begin{array}{r} \text{USA} \\ +\ \text{USSR} \\ \hline \text{PEACE} \end{array}$$

Рис. 18.1
Буквенная головоломка[1]

Предлагалось определить, какие цифры стоят за этими латинскими буквами[2]. При этом, естественно, одинаковым буквам должны соответствовать одинаковые цифры, а разным — разные. Ясно, что буквой P (крайняя слева в сумме) может быть только единица, а буквой U — девятка. Раз так, то буква E хранит ноль $(1 + 9 = 10)$[3]. Можно и дальше продолжить такие логически-арифметические рассуждения («мозговую атаку» на задачу) и полностью решить этот буквенный ребус — найти, какие цифры соответствуют остальным неизвестным буквам S, A, C и R. Но есть резон привлечь к этой работе компьютер. Почему!? Во-первых, можно перефразировать[4] второе название этого этюда книги и сказать: «На себя надейся, а к компьютеру обращайся!» То есть опять же — «не плошай!» «Оплошать» же можно так — найти только одно решение из множества возможных. А такой пример мы приведем ниже. Во-вторых, занимательная информатика[5] подразумевает использование компьютера[6] для решения занимательных задач с помощью именно компьютера. Можно программно подставить в наши неразгаданные буквы числа от 2 до 8 (числа 0, 1, и 9 уже заняты — см. выше), перебрать все варианты и вы-

[1] Один мальчик пришел домой с очередного урока английского языка и сказал родителям: «Теперь я знаю, как по-английски будет «Миру-мир!» — «Пису-пис!», чем очень рассмешил маму с папой. В те (советские) времена холодной войны этот лозунг (слоган, как говорят теперь) был очень популярен, вернее, был очень пропагандистски навязчив. Кстати, все ли знают, какой «мир» прописан в названии романа Л. Н. Толстого «Война и мир»? До революции это теперешнее одно слово «мир» писалось поразному: миръ (общество) и міръ (покой, не война) [3].

[2] Есть подобные головоломки с подвохом, где одна буква C — латинская, а другая — русская.

[3] Эта головоломка перекликается с этюдом 3, где мы рассматривали шифр, основанный на замене символов.

[4] Название этюда — это уже перефразированная известная полуатеистическая пословица «На Бога надейся, а сам не плошай!» Так что мы делаем перефразировку, так сказать, в квадрате.

[5] Некоторые этюды книги, как следует из списка литературы, опубликованы в рубрике «Занимательная информатика» журнала «Информатика в школе».

[6] Честно говоря, информатика сама по себе существует безо всяких компьютеров. Просто у нас, да и во всем мире информатика, вернее, информационные технологии часто ассоциируются именно с компьютерными технологиями.

дать «на печать» ответ. У нас, кстати, остались неразгаданные не четыре, три буквы, т. к. $A + R = 10$ — см. правый столбец букв на рис. 18.1, где $E = 0$. Можно, повторяем, продолжить «мозговую атаку» на задачу, но тут стоит остановиться и поручить эту работу компьютеру: составление программы для него — это тоже неплохая головоломка для ума[7].

На рисунке 18.2 показана программа, написанная в среде математического пакета Mathcad Prime [79], решающая наш буквенный ребус. В программе заложено три цикла `for`, перебирающие в буквах `A`, `C` и `S` (параметры цикла) числа от 2 до 8. Если при таком переборе выполняется равенство `USA + USSR = PEACE`, то ответ (числа, хранящиеся в наших семи переменных) записываются n-й строкой матрицы с именем `M`. В нулевую же строку матрицы `M` операторам первой строки программы, показанной на рис. 18.2, записывается «шапка» (заголовок) этой матрицы. Эта строка программы заодно будет служить и ее названием.

```
M := | M^(0) ← ["#" "U" "S" "A" "+" "U" "S" "S" "R" "=" "P" "E" "A" "C" "E"]
     | [n ← 1  U ← 9  P ← 1  E ← 0]
     | for A ∈ 2..8
     |   | R ← 10 − A
     |   | for C ∈ 2..8
     |   |   | for S ∈ 2..8
     |   |   |   | if 100 U + 10 S + A + 1000 U + 100 S + 10 S + R = 10000 P + 1000 E + 100 A + 10 C + E
     |   |   |   |   | M^(n) ← [n U S A "+" U S S R "=" P E A C E]
     |   |   |   |   | n ← n + 1
     | M
```

Рис. 18.2

Программа решения буквенной головоломки

На рисунке 18.3 показан результат выполнения программы на рис.18.2. Она выдала нам четыре варианта ответов, но только первый из них (932 + 9338 = = 10 270) верен. В остальных ответах `S = A`, что не соответствует условиям задачи, и что ясно видно из простого просмотра (визуального анализа) элементов матрицы `M`. Можно, конечно, вставить в программу, показанную на рис. 18.2, еще один условный оператор `if S ≠ A`, но мы этого делать не будем — ответ и так ясен и из рисунка 18.3.

Ребус USA + USSR = PEACE устарел[8]. Во-первых, уже нет такой страны USSR/СССР, а во-вторых, решение этого ребуса описано во многих занимательных книгах и даже фигурирует в Интернете — стоит только отдать соответствующий запрос (usa + ussr = peace) в каком-либо поисковике.

[7] Примерно так была решена задача о четырех красках на политической карте: в математике теорема о четырёх красках утверждает, что всякую расположенную на сфере карту можно раскрасить четырьмя красками так, чтобы любые две области, имеющие общий участок границы, были раскрашены в разные цвета. Люди «мозговой атакой» доказали, что это так для почти всех случаев конфигурации карты. Остальные случаи «разъяснил» компьютер. Но и сейчас часто в первоапрельских выпусках научно-популярных журналов помещают контурные географические карты, которые якобы нельзя раскрасить четырьмя красками.

[8] Как сказать. Если Россия — правопреемница СССР, то теперешняя ее ссора с Америкой из-за Украины дает новую жизнь этой головоломке.

$$M=\begin{bmatrix} \text{“}\#\text{”} & \text{“}U\text{”} & \text{“}S\text{”} & \text{“}A\text{”} & \text{“}+\text{”} & \text{“}U\text{”} & \text{“}S\text{”} & \text{“}S\text{”} & \text{“}R\text{”} & \text{“}=\text{”} & \text{“}P\text{”} & \text{“}E\text{”} & \text{“}A\text{”} & \text{“}C\text{”} & \text{“}E\text{”} \\ 1 & 9 & 3 & 2 & \text{“}+\text{”} & 9 & 3 & 3 & 8 & \text{“}=\text{”} & 1 & 0 & 2 & 7 & 0 \\ 2 & 9 & 6 & 6 & \text{“}+\text{”} & 9 & 6 & 6 & 4 & \text{“}=\text{”} & 1 & 0 & 6 & 3 & 0 \\ 3 & 9 & 7 & 7 & \text{“}+\text{”} & 9 & 7 & 7 & 3 & \text{“}=\text{”} & 1 & 0 & 7 & 5 & 0 \\ 4 & 9 & 8 & 8 & \text{“}+\text{”} & 9 & 8 & 8 & 2 & \text{“}=\text{”} & 1 & 0 & 8 & 7 & 0 \end{bmatrix}$$

$$932+9338=10270$$

Рис. 18.3
Решение буквенной головоломки

Более современной и еще нерешенной (если не считать публикации [79]) задачей[9] можно считать задачу USA + IRAN = PEACE. Нерешенной и в прямом и в переносном смыслах. В переносном смысле, учитывающим нынешнее состояние американо-иранских отношений.

```
M:= | M^0 ← ["#" "U" "S" "A" "+" "I" "R" "A" "N" "=" "P" "E" "A" "C" "E"]
    | [n ← 1  I ← 9  P ← 1  E ← 0]
    | for A ∈ 2..8
    |   | N ← 10 − A
    |   | for C ∈ 2..8
    |   |   | for S ∈ 2..8
    |   |   |   | for U ∈ 2..8
    |   |   |   |   | for R ∈ 2..8
    |   |   |   |   |   | if 100 U + 10 S + A + 1000 I + 100 R + 10 A + N = 10000 P + 1000 E + 100 A + 10 C + E
    |   |   |   |   |   |   | if U≠S∧A≠S∧N≠R∧U≠R∧S≠R∧C≠R∧N≠A∧N≠U∧C≠U
    |   |   |   |   |   |   |   | M^n ← [n U S A "+" I R A N "=" P E A C E]
    |   |   |   |   |   |   |   | n ← n + 1
    | M
```

Рис. 18.4
Программа решения новой буквенной головоломки

На рисунке 18.4 показана Mathcad-программа, решающую эту новую задачу. Она отличается от программы на рис. 18.2 количественно, но не качественно: в новой программе записано не три, а пять вложенных циклов `for`. Кроме того, в программу вставлен дополнительный оператор `if`, отсекающий неверные решения, когда разным буквам соответствует одно и тоже число. Без этого условного оператора программа выдаст уже не 4 (см. рис. 18.3), 75 (!) решений, анализировать которые «вручную» («вглазную») и отбраковывать неверные результаты будет весьма затруднительно. Поступить тут можно так. Вывести «на печать» матрицу `M` со всеми 75 решениями, просматривать их и если окажется, что в какой-то строке разным буквам соответствует одно число, то в программу, показанную на рис. 18.4, нужно будет вставить дополнительный условный оператор `if U ≠ S`, например. После этого еще раз запустить программу и вывести «на печать» укороченную (отфильтрованную) матрицу `M`. Если снова окажется, что другая пара букв не будет соответствовать условию задачи, то нужно будет расширить оператор `if`: было `if U ≠ S`, стало `if U ≠ S ^ A = S` (оператор с символом ^ — это булев оператор `И (AND)` — логическое умножение — см. этюд 11). Эту операцию (фильтрование матрицы удалением из нее ненужных строк) нужно будет продолжить до тех пор, пока в ней не останутся только правильные решения — см. рис. 18.5.

[9] Это можно проверить в том же Интернете.

$$M=\begin{bmatrix} \text{“}\#\text{”} & \text{“}U\text{”} & \text{“}S\text{”} & \text{“}A\text{”} & \text{“}+\text{”} & \text{“}I\text{”} & \text{“}R\text{”} & \text{“}A\text{”} & \text{“}N\text{”} & \text{“}=\text{”} & \text{“}P\text{”} & \text{“}E\text{”} & \text{“}A\text{”} & \text{“}C\text{”} & \text{“}E\text{”} \\ 1 & 5 & 3 & 2 & \text{“}+\text{”} & 9 & 7 & 2 & 8 & \text{“}=\text{”} & 1 & 0 & 2 & 6 & 0 \\ 2 & 7 & 3 & 2 & \text{“}+\text{”} & 9 & 5 & 2 & 8 & \text{“}=\text{”} & 1 & 0 & 2 & 6 & 0 \\ 3 & 5 & 2 & 3 & \text{“}+\text{”} & 9 & 8 & 3 & 7 & \text{“}=\text{”} & 1 & 0 & 3 & 6 & 0 \\ 4 & 8 & 2 & 3 & \text{“}+\text{”} & 9 & 5 & 3 & 7 & \text{“}=\text{”} & 1 & 0 & 3 & 6 & 0 \\ 5 & 5 & 7 & 4 & \text{“}+\text{”} & 9 & 8 & 4 & 6 & \text{“}=\text{”} & 1 & 0 & 4 & 2 & 0 \\ 6 & 8 & 7 & 4 & \text{“}+\text{”} & 9 & 5 & 4 & 6 & \text{“}=\text{”} & 1 & 0 & 4 & 2 & 0 \\ 7 & 7 & 5 & 6 & \text{“}+\text{”} & 9 & 8 & 6 & 4 & \text{“}=\text{”} & 1 & 0 & 6 & 2 & 0 \\ 8 & 8 & 5 & 6 & \text{“}+\text{”} & 9 & 7 & 6 & 4 & \text{“}=\text{”} & 1 & 0 & 6 & 2 & 0 \end{bmatrix} \begin{matrix} 532 + 9728 = 10260 \\ 732 + 9528 = 10260 \\ 523 + 9837 = 10360 \\ 823 + 9537 = 10360 \\ 574 + 9846 = 10420 \\ 874 + 9546 = 10420 \\ 756 + 9864 = 10620 \\ 856 + 9764 = 10620 \end{matrix}$$

Рис. 18.5
Решения новой буквенной головоломки

Новый резус имеет уже не одно, а восемь решений! Что и требовалось доказать! А что требовалось доказать?! А то, что ребус USA + USSR = PEACE можно решить и без компьютера, а новый ребус USA + IRAN = PEACE без компьютера решить весьма затруднительно. Вернее, решить-то можно (найти в уме один вариант ответа), но не будет полной уверенности в том, что найденное «мозговой атакой» решение единственное. Но и без собственных рассуждений (без «мозговой атаки», подсказывающей, что `I = 9`, `P = 1`, `A + N = E` или `A + N = E + 10`) эту задачу решить будет довольно трудно: в ребусе 9 разных букв. Это значит, что лобовая компьютерная атака на задачу потребует девяти, а не пяти вложенных циклов, каждый из которых нужно будет «прогнать» десять раз. Такая программа будет выполняться довольно долго, а выданная ею матрица `M` будет объемной, и ее придется очень долго и нудно фильтровать, доводя до вида, показанного на рис. 18.5.

Но если совсем не отвергать математику, то есть возможность помочь компьютеру. Вспоминая правила десятичной записи чисел и используя очевидные значения `I = 9` и `P = 1`, мы можем составить уравнение

$$100U + 10S + A + 9000 + 100R + 10A + N = 10000 + 1000E + 100A + 10C + E.$$

Упростив его, получим: $1000(E + 1 + 100(A - U - R) + 10(C - S - A) + E - A - N = 0$. Так как $E - A - N$ делится на 10, то $E - A - N = 10x$, причем $|x| \le 1$ (так как все переменные неотрицательные и не больше девяти). Но тогда, разделив на 10, мы имеем $100(E + 1) + 10(A - U - R) + C - S - A + x = 0$. Так как $C - S - A + x$ делится на 10, то $C - S - A + x = 10y$ и $|y| \le 2$. Снова разделив на 10, получим $10(E + 1) + A - U - R + y = 0$. Повторив процедуру еще раз, приходим к уравнению $E + 1 + z = 0$, где $A - U - R + y = 10z$ и $|z| \le 2$. Итак, мы получили систему уравнений в целых числах:

$$E = -1 - z \ge 0;$$

$$A + N = z - 10x + 1;$$

$$S + A - C = x - 10y;$$

$$U + R - A = y - 10z.$$

Кроме того, $0 \le E \le 9$, $0 \le A \le 9$, $0 \le N \le 9$, $0 \le C \le 9$, $0 \le S \le 9$, $0 \le R \le 9$ и $0 < U \le 9$.

Эту систему можно решить перебором.

А в качестве задания пусть читатель с помощью компьютера попробует доказать, что ребус: «УЧЕНИК» + «УЧЕБНИК» = «ИНТЕРНЕТ» решения не имеет!

Есть, кстати, приложение решения буквенных головоломок для «яблочных устройств» [80].

Вывод, который, надеемся, читатель сделал после решения описанных в этом этюде и многих других задач.

Решая задачу (школьную, вузовскую, инженерную, научную, житейскую), нужно уметь сочетать свой собственный интеллект с возможностями компьютера. А компьютер сейчас встраивают куда только можно. Не за горами время, когда

компьютер будут встраивать прямо в... мозг. Сначала животных, а потом и человека. Это позволит быстро и эффективно решать любые задачи, встречающиеся на жизненном пути.

Этим очень важным выводом мы и заканчиваем книгу, добавляя сюда уместный пример в виде цитаты из Википедии (мы привели в этюде ребус с названиями стран (см. рис. 18.4), а теперь поговорим о задаче раскраски стран на географической карте, при решении которой сочетался интеллект человека, точность и быстродействие компьютера — см. сноску на стр. 372).

«В математике **теорема о четырёх красках** утверждает, что всякую карту можно раскрасить четырьмя красками так, чтобы любые две области, имеющие общий участок границы, были раскрашены в разные цвета. Эта теорема была сформулирована Фрэнсисом Гутри в 1852 г., однако доказать ее долгое время не удавалось. В течение этого времени было предпринято множество попыток как доказательства, так и опровержения, и эта задача носила название *проблемы четырёх красок*.

Теорема о четырёх красках была доказана в 1976 г. Кеннетом Аппелем и Вольфгангом Хакеном из Иллинойского университета. **Это была первая крупная математическая теорема, доказанная с помощью компьютера**. Первым шагом доказательства была демонстрация того, что существует определенный набор из 1936 карт, ни одна из которых не может содержать карту меньшего размера, которая опровергала бы теорему. Аппель и Хакен использовали специальную компьютерную программу, чтобы доказать это свойство для каждой из 1936 карт. Доказательство этого факта заняло сотни страниц. После этого Аппель и Хакен пришли к выводу, что не существует наименьшего контрпримера к теореме, потому что иначе он должен бы содержать, хотя не содержит, какую-нибудь из этих 1936 карт. Это противоречие говорит о том, что вообще не существует контрпримера. Изначально доказательство не было принято всеми математиками, поскольку его невозможно было проверить вручную. В дальнейшем оно получило более широкое признание, хотя у некоторых долгое время оставались сомнения».

И до сих пор некоторые научно-популярные журналы в первоапрельских номерах (см. конец этюда 13) публикуют замысловатую контурную карту, которую якобы нельзя правильно закрасить четырьмя красками. И многие на этой шутке «покупаются», берут четыре разноцветных карандаша (реальных или виртуальных) и начинают... опровергать утверждение журнала...

Один из авторов этой книги также каждое первое апреля помещает Mathcad-шутку на сайте PTC Community. Пример одной такой шутки **https://www.ptcusercommunity.com/message/301856**.

Задания читателям

1. Решить составлением программы ребусы, выложенные на сайте **http://rebus1.com**. Там, кстати, под первым номером стоит ребус USA + USSR = PEACE. Вот другие ребусы с этого сайта: УДАР + УДАР = = ДРАКА, SEND + MORE = MONEY.

2. Решить числовой ребус без или с компьютером — это задача для любителя. Профессионалы же сами составляют новые ребусы без или с компьютером. Попробовать себя на этом поприще!

3. Попробовать применить компьютер к другим видам числовых головоломок — к судоку, например. Кстати, на сайте:

http://twt.mpei.ac.ru/MCS/Worksheets/Sudoku.xmcd

расположена Mathcad-программа решения этой головоломки.

4. Придумать хорошую Mathcad-шутку для 1 апреля.

Этюд 19

СЕМЬ ВЫЧИСЛИТЕЛЬНЫХ КРИВЫХ, ИЛИ ВЕЛОСИПЕД АПОЛЛОНИЯ И ГИТАРА АПОЛЛОНА

В этюде будет рассказано о методах построения плоских кривых, связанных в том числе и с вычислительными операторами сложения (эллипс), вычитания (гипербола), умножения (овал Кассини) и деления (окружность Аполлония). Исследованы свойства кривых по трем другим (остальным) вычислительным операторам с двумя операндами: остаток от деления, возведение в степень и логарифм по заданному основанию. Подвергнута ревизии теория размерных величин — обоснована возможность и необходимость работы с размерными величинами в показателе степени.

Математика: плоские кривые второго порядка, фокальное и директориальное свойство гиперболы и эллипса, решение системы двух нелинейных алгебраических уравнений, вектор, декартов и полярный график.

Физика: траектории движения небесных тел.

Информатика: символьная математика, компьютерная плоская графика, вложенные циклы с параметром, условный оператор.

Сайт этюда: **https://www.ptcusercommunity.com/message/469394**

Бросая в воду камешки, смотри на круги, ими образуемые;
иначе такое бросание будет пустою забавою.

Козьма Прутков. Плоды раздумья

В школьные годы первого автора книги многие мальчики мечтали стать космонавтами. В те времена Советский Союз «жил» космосом: запускались спутники Земли, Луну облетали зонды, человек поднимался в космос и т. д. Других особых научно-технических достижений мирового уровня у страны не было. Все это вызывало большой интерес к теории космических полетов, в частности к небесной механике. Мы это довольно подробно описали в этюде 9. На школьных уроках математики, физики, астрономии учителя рассказали, что спутники (естественные и искусственные) вращаются вокруг планет по круговым и эллиптическим орбитам и о том, как просто можно нарисовать этот самый *эллипс* — ненамного сложней, чем окружность (частный случай эллипса). Первый автор статьи после такого урока пришел домой, вбил в стену комнаты два гвоздя, привязал к ним веревочку, натянул ее карандашом и нарисовал на обоях половинку эллипса — см. рис. 9.40 на стр. 238. Вторую половинку нарисовать не удалось — мать, повторяем, увидела эти физико-математические «художества» и дала сыну нагоняй...

Давайте попробуем еще раз дорисовать этот эллипс, но не на стене комнаты, а на экране компьютера в среде пакета Mathcad 15 — см. рис. 19.1.

Можно взять алгебраическое выражение эллипса и использовать его для построения этой замкнутой кривой. А можно поступить иначе: вспомнить, что эллипс — это геометрическое место точек на плоскости, *сумма* расстояний от которых до двух других фиксированных точек (фокусов эллипса) постоянна. Эллипс и некоторые другие «вычислительные» кривые, о которых будет рассказано ниже, строились в среде Mathcad так. Решалась система двух нелинейных алгебраических уравнений — находилось аналитическое выражение для координат точек, отстоящих от двух фокусов (их декартовы координаты x_{F1} - y_{F1} и x_{F2} - y_{F2}) на расстояниях L (на графике Black — отрезок черного цвета) и L2 (Blue — отрезок синего цвета; сам же эллипс у нас красный). В систему уравнений заложено «эллиптическое» равенство: L + L2 = a (при построении других «вычислительных» кривых мы будем менять вид этого равенства). У нашей системы два решения по двум неизвестным x и y, которые формируют матрицу с двумя строками (два решения) и двумя столбцами (две неизвестные системы). По этой матрице формируются четы-

ре пользовательские функции с аргументом L и с именами X1, X2, Y1 и Y2, по которым параметрически и поточечно строится сам эллипс. Переменная (параметр) l (эль) — это фиксированное значение длины из переменной области L (range variable), при которой в эллипс вписываются два отрезка, сумма которых постоянна и равна заданному значению переменной a (6 м — см. рис. 19.1: запомним это, точнее, то, что переменная a равна не просто шести, а именно шести метрам). Авторская методика расчета, показанная на рис. 19.1, позволяет строить довольно сложные кривые без какого-либо предварительного вывода заложенных в кривые аналитических зависимостей. В этой методике сочетаются аналитические (символьные) и численные методы решения задачи.

$$L + L2 = a \text{ solve}, L2 \rightarrow a - L$$

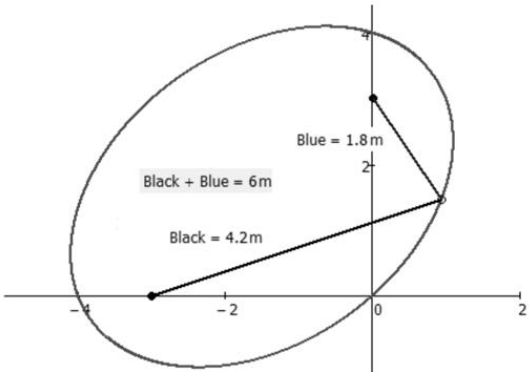
$$x_{F1} := -3m \qquad x_{F2} := 0m \qquad y_{F1} := 0m \qquad y_{F2} := 3m \qquad a := 6m$$

$$\text{Coord}(L) := \begin{bmatrix} (a-L)^2 = (x-x_{F2})^2 + (y-y_{F2})^2 \\ L^2 = (x-x_{F1})^2 + (y-y_{F1})^2 \end{bmatrix} \text{solve}, \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} L - \frac{9\cdot m}{2} + \frac{\sqrt{12\cdot L\cdot m - 2\cdot L^2 - 9\cdot m^2}}{2} & L - \frac{3\cdot m}{2} - \frac{\sqrt{12\cdot L\cdot m - 2\cdot L^2 - 9\cdot m^2}}{2} \\ L - \frac{9\cdot m}{2} - \frac{\sqrt{12\cdot L\cdot m - 2\cdot L^2 - 9\cdot m^2}}{2} & L - \frac{3\cdot m}{2} + \frac{\sqrt{12\cdot L\cdot m - 2\cdot L^2 - 9\cdot m^2}}{2} \end{pmatrix}$$

$$X1(L) := \text{Coord}(L)_{1,1} \qquad Y1(L) := \text{Coord}(L)_{1,2} \qquad X2(L) := \text{Coord}(L)_{2,1} \qquad Y2(L) := \text{Coord}(L)_{2,2}$$

$$L := 0mm, \frac{a}{70000} .. a \qquad FRAME := 200 \qquad l := 0.8m + \frac{5.9m - 0.8m}{300} FRAME = 4.2m$$

$$Black := \sqrt{(Y1(l) - y_{F1})^2 + (X1(l) - x_{F1})^2} = 4.2m \qquad Blue := \sqrt{(Y1(l) - y_{F2})^2 + (X1(l) - x_{F2})^2} = 1.8m$$



Y1(L) =

	1
10247	-0.62-0.03i
10248	-0.62-0.03i
10249	-0.62-0.02i
10250	-0.62-0.02i
10251	-0.62-0.02i
10252	$-0.62-6.91i\cdot 10^{-3}$
10253	-0.63
10254	-0.64
10255	-0.64
10256	-0.65
10257	-0.65
10258	-0.65
10259	...

m

Рис. 19.1

Эллипс — сложение (анимация https://www.ptcusercommunity.com/videos/2089)

Переменная l (эль) в нашем расчете на рис. 19.1 привязана к системной переменной FRAME, которая управляет *анимацией* в среде Mathcad [44]. Меняя значения переменной FRAME от нуля до, например, трехсот (кадры анимации), можно показывать, как... карандаш, удерживаемый веревочкой, рисует эллипс — см. начало статьи и анимацию здесь **https://www.ptcusercommunity.com/videos/2089**. Анимация рисования эллипса и других замкнутых кривых, описываемых в статье, расположена также на сайте статьи. Там же можно скачать соответствующие расчетные документы пакета Mathcad.

На рисунке 19.1 показана также часть распечатки значений, которые выдает функция Y1 при заданных дискретных значениях переменной L. Там есть действительные и комплексные числа. Последние получаются в том случае, если значения переменной L (аргумент функции Y1) не позволяют строить эллипс (L > a, например). Графика пакета Mathcad игнорирует эти значения и строит эллипс без проблем. Такой особенности нет у трехмерной графики Mathcad, что создает пользователям ряд неудобств. Методы построения графиков, предложенные в статье,

годятся и для построения замкнутых поверхностей, являющихся развитием рассматриваемых кривых: эллипсоидов, гиперболоидов и т. д.

Эллипс на рис. 19.1 можно рассматривать как орбиту вращения одного небесного тела вокруг другого. Это вытекает из аналитического решения соответствующего дифференциального уравнения, учитывающего силы, действующие на небесные тела — см. этюд 9. Во времена развития небесной механики, как раздела математики и физики были попытки аналитического решения задачи о трех и более небесных телах, подчиняющихся закону всемирного тяготения. Но общего аналитического решения так и не было найдено — были найдены решения только для некоторых частных случаев. Один из них описан в этюде 9 — см. хоровод трех планет на рис. 9.4 (стр. 201). В этих поисках взоры исследователей обращались к *эллипсам с более чем двумя фокусами* — к центрам планет и спутников. Такие замкнутые кривые называют n-эллипсами или по имени человека, впервые их исследовавшего, — кривыми Чирнхауза (см. **https://en.wikipedia.org/wiki/N-ellipse**). Этот философ, математик и экспериментатор считается одним из изобретателей европейского белого фарфора, который в начале XVIII в. стали производить в саксонском городке Мейсен недалеко от Дрездена. Можно предложить этой порцелановой фабрике, которая успешно работает и поныне, изготовить в честь Эренфрида Вальтера фон Чирнхауза (1651–1708) сувенирную фарфоровую тарелку с формой и рисунком, показанными на рис. 19.2.

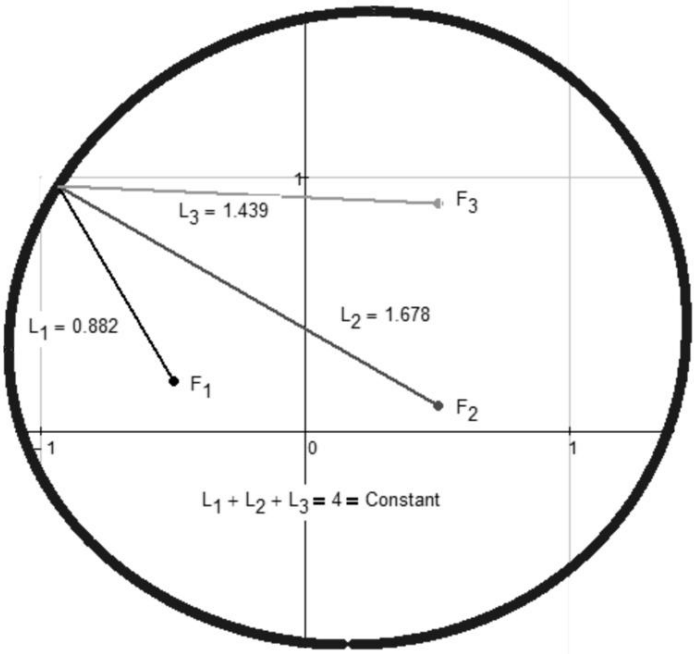


Рис. 19.2

Эллипс с тремя фокусами (анимация https://www.ptcusercommunity.com/videos/7045)

На этой тарелке яйцеобразной формы (а эти кривые еще называют и яйцеобразными[1]) помимо трех фокусов и трех отрезков, сумма которых остается постоянной при рисовании голубой каемки тарелки, можно поместить и другую ин-

[1] Можно «вбить несколько гвоздей в трехмерное пространство», «привязать к ним веревочку» и очертить некое тело в виде настоящего яйца. За работу, читатель!

формацию[2]. Мы, в свою очередь, можем предложить еще один трехфокусный эллипс, который несложно нарисовать не только на экране компьютера, воспользовавшись методом на рис. 19.1, но и на... тех же обоях комнаты, вбив в стенку уже не два, а три гвоздя-фокуса (F1, F2 и F3), привязав веревочку к двум гвоздям (F1 и F2) и перекинув ее особым образом через грифель карандаша и третий гвоздь (F3) — см. рис. 19.3.

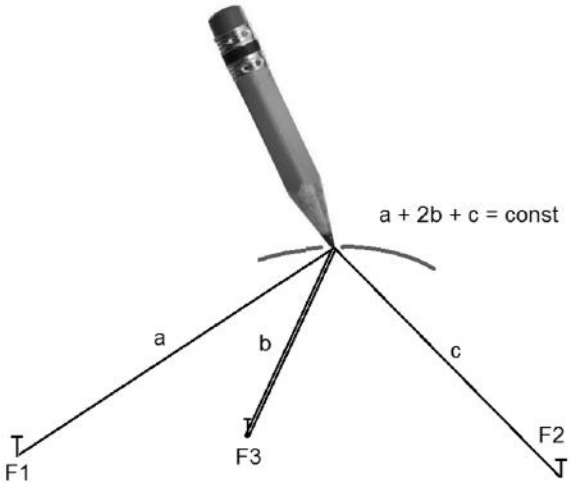


Рис. 19.3

Рисование трехфокусного эллипса (анимация https://www.ptcusercommunity.com/videos/7060)

Можно справится в Интернете, исследованы ли уже кривые, подобные той, малая часть которой показана на рис. 19.3, найдены ли соответствующие аналитические выражения для декартовых и полярных координат. Этой интересной работой при желании может заняться читатель, увеличивая число фокусов и изменяя порядок перекидывания веревочки через них. Мы же идем дальше.

Если вместо суммы длин отрезков использовать *разность*, то будет построен не эллипс, а другая плоская кривая второго порядка — *гипербола*: см. рис. 19.4. Она имеет две ветви. Одна из них соответствует разности Black-Blue, а вторая — разности Blue-Black. Объединить эти две разности позволяет оператор абсолютного значения (модуля). Но если не опираться на модуль, а работать с двумя разностями, то допустимо присваивать параметру a и отрицательные значения. В случае с гиперболой работа с отрицательными значениями параметра a ничего не изменит. Но при рассмотрении «логарифмической» кривой, например, (см. ниже) отрицательные значения параметра a могут расширить форму исследуемой кривой.

Гиперболу, вернее одну ее ветвь, мы получили, когда рассчитывали траекторию движения кометы (рис. 9.18 на стр. 218) и профиль линзы (рис. 20.4 на стр. 402). Гиперболу можно также построить, опираясь не на два фокуса, а на один фокус и директрису — см. рис. 20.12 на стр. 412.

[2] Сетка графика на тарелке напоминает... тюремную решетку. Это будет намеком на то, что настоящим изобретателем европейского фарфора был не аристократ Чирнхауз, а алхимик Иоган Бёттгер, которого Чирнхауз держал под арестом в «шарашке», созданной специально под этот «фарфоровый проект». Чирнхауз же был директором этого закрытого учреждения — комендантом крепости, где располагался этот «почтовый ящик». Такая практика ведения инновационных научно-технических разработок два с лишним века спустя широко использовалась сталинским режимом. Из-за этого мы зачастую не знаем имен истинных изобретателей и авторов разработок, а помним и чествуем только «директоров шарашек».

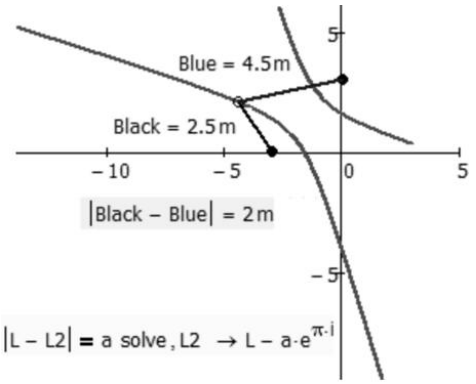


Рис. 19.4

Гипербола — вычитание (анимация https://www.ptcusercommunity.com/videos/7035)

Если же работать с двумя фокусами и не с суммой (рис. 19.1) или разностью (рис. 19.4), а с третьей вычислительной операцией — с *произведением* значений отрезков `Black` и `Blue`, то можно получить так называемый *овал (овалы) Кассини* — см. рис. 19.5.

Овалом (деформированным эллипсом[3]) замкнутые кривые на рис. 19.5 можно назвать весьма условно. Уменьшение значения параметра `a` (произведения длины отрезка `Black` на длину отрезка `Blue`) при фиксированных координатах фокусов приводит к тому, что у этого «овала» появляется «талия» (см. центральную кривую на рис. 19.5), которая при дальнейшем уменьшении значения параметра `a` «рвет» эту плоскую фигуру на две половинки. Анимация этого процесса напоминает деление живой клетки. Если и дальше уменьшать значение параметра `a`, то овалы уменьшатся до точек (`a = 0`). Было время, когда полагали, что спутники вращаются вокруг планет не по эллиптическим орбитам (рис. 19.1), а по орбитам, подобным той, какая показана вверху рис. 19.5. Вернее, велись научные дискуссии по этому поводу, в которых активно участвовал и сам Кассини со своим овалом. Но в конце концов было доказано, что тут «работает» эллипс. Другая же возможная траектория движения небесных тел — это одна из ветвей гиперболы, показанной на рис. 19.4.

У овала Кассини, как и у эллипса, может быть более двух полюсов. Такие замкнутые кривые условно (см. ниже) называют лемнискатами («увитыми лентами»). Наиболее известна лемниската Бернулли, которая является частным случаем овала Кассини. У этого овала «нулевой объем талии», который имеет место, если произведение длин отрезков `Black` и `Blue` равно квадрату половины расстояния между фокусами.

На рис. 19.6 можно видеть эволюцию лемнискаты с тремя фиксированными фокусами при изменении значения параметра `a`.

Анимация, четыре кадра которой показаны на рис. 19.6, похожа на то, как на неровную поверхность наливают воду, формирующую отдельные лужи, которые постепенно сливаются в одну. Если же эту анимацию запустить в обратном порядке, то получится картина высыхания лужи. Подобные контуры можно наблюдать у... высыхающего Аральского моря, например.

Располагая разное количество фокусов в различных точках плоскости и задавая разное значение параметра-произведения `a`, можно получать довольно забавные картинки. На рисунке 19.7 показаны три этапа трансформации овалов Кассини, обрисовывающих... покемона с девятью фокусами: уши, глаза, руки, ноги и

[3] Овалом иногда (в англоязычной математической литературе) называют любую замкнутую плоскую кривую. В русском же языке овал — это нечто сугубо гладкое и выпуклое.

пупок. Фокусы-глаза дополнительно обведены кружочками. Этого «виртуального зверя» можно поймать, не только бегая как угорелый со смартфоном по городам и весям, но и изучая алгебру, геометрию и информатику…

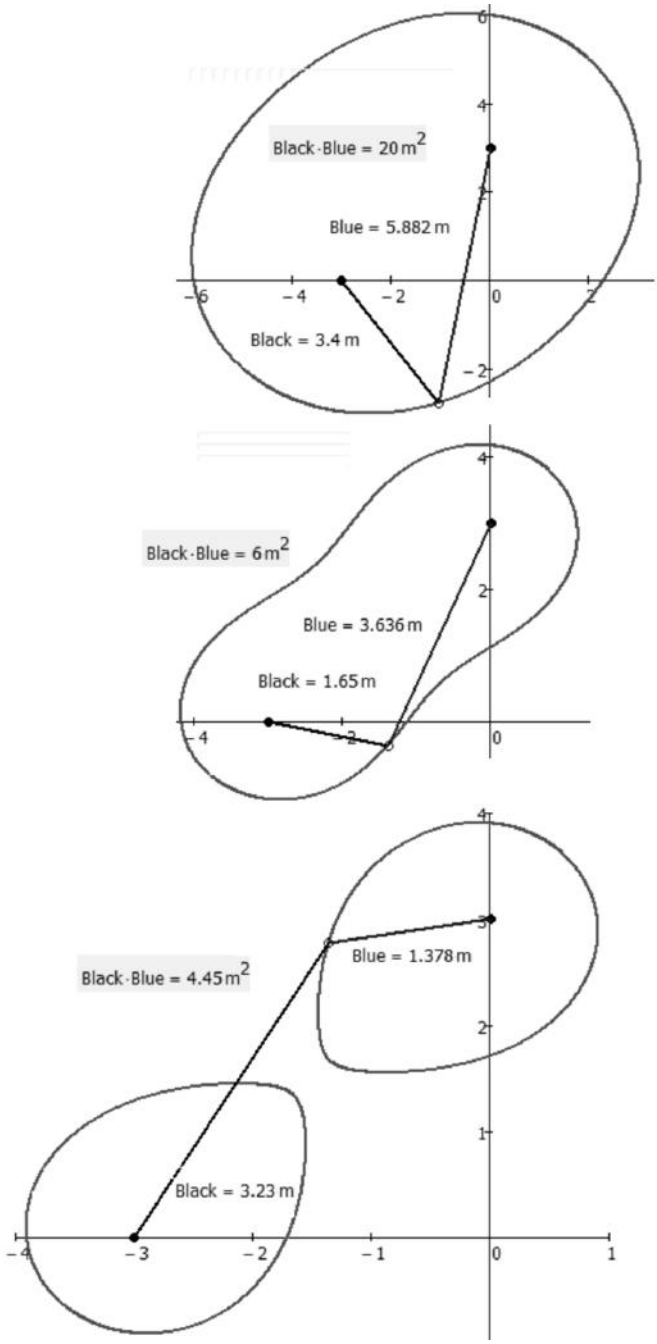


Рис. 19.5
Овал Кассини (Cassini) — умножение (анимация ***https://www.ptcusercommunity.com/videos/7036****)*

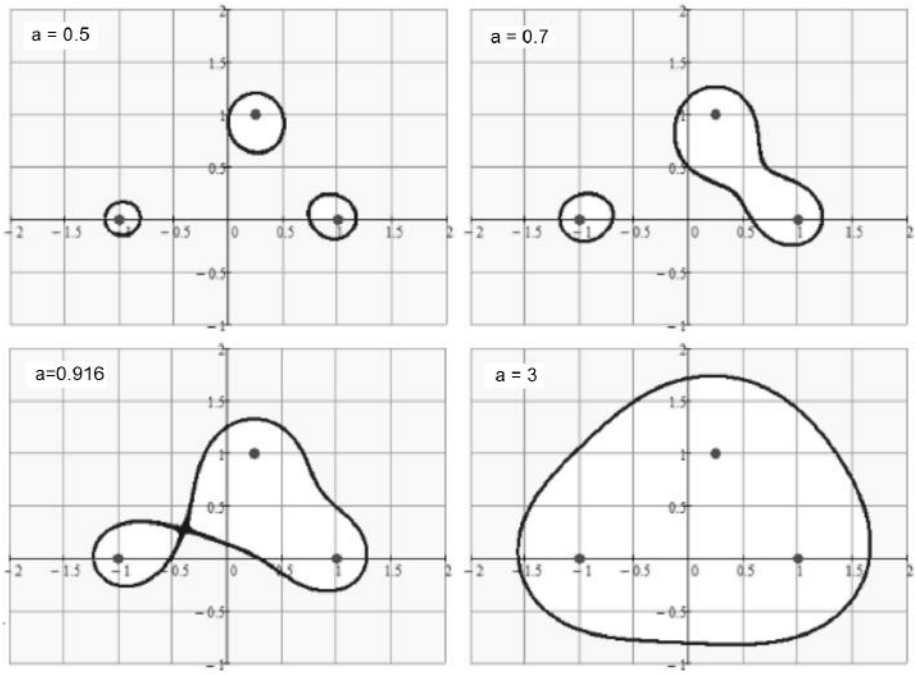


Рис. 19.6
Трехфокусная лемниската
(анимацию с пятью фокусами можно увидеть здесь ***https://www.ptcusercommunity.com/videos/7061****)*

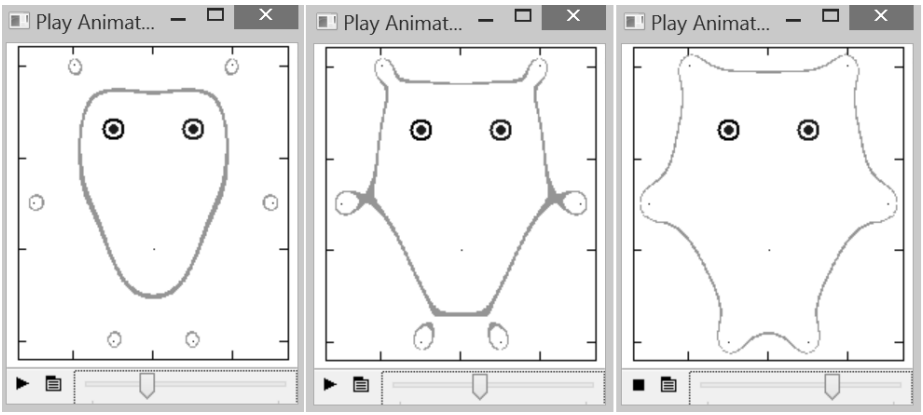


Рис. 19.7
Покемон Кассини (анимация https://www.ptcusercommunity.com/thread/137953)

Описание лемнискаты как многофокусного овала Кассини можно найти в Википедии. Правда, только в ее русскоязычном секторе. В других секторах (английском, немецком и др.) такого описания нет, что вызывает сомнения в таком толковании термина лемниската. Обычно, повторяем, говорят только о лемнискатах с двумя фокусами особого вида — лемниската Бернулли, лемниската Бута и др.

Если фокусы эллипса сместить в одну точку, то получится окружность. Но окружность можно построить и с двумя отделенными фокусами, используя не сложение отрезков, а их *деление* — см. рис. 19.8.

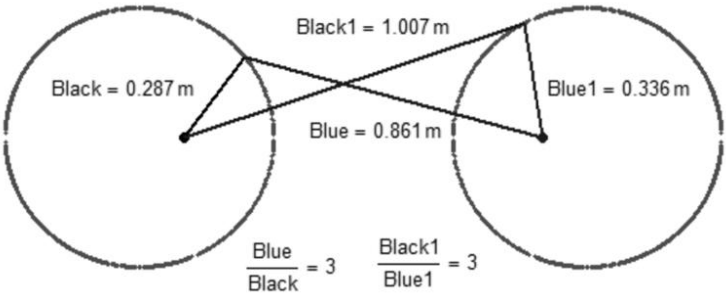


Рис. 19.8
Окружности Аполлония — деление (анимация https://www.ptcusercommunity.com/videos/7040)

Можно дать такое определение окружности. Окружность — это частный случай эллипса, при построении которого нужно опираться не на сумму отрезков (см. рис. 19.1), а на их деление (рис. 19.8). Такие окружности имеют имя — *окружности Аполлония*[4]. Окружность Аполлония, как и гипербола, имеет две «ветви», показанные на рис. 19.8. Эти две ветви-окружности получаются из двух дробей: прямой `Black/Blue` и обратной `Blue/Black`. Изображение окружностей Аполлония, на которое наложены их формирующие отрезки `Black` и `Blue`, напоминает... велосипед с рамой и двумя колесами: у Аполлона есть колесница (или квадрига — см. фронтон Большого театра в Москве или сторублевую купюру), а у Аполлония... велосипед — a bicycle, двухколесник, «двухокружник», показанный на рис. 19.8 и вынесенный в название этюда. Рисунок 19.8 позволяет также утверждать, что у окружности есть центр и есть фокус и эти две точки не совпадают.

Кривые, показанные на рис. 19.1, 19.4, 19.5 и 19.7 (эллипс, гипербола, овал Кассини и окружность Аполлония), довольно хорошо исследованы [86–88]. Но давайте зайдем в еще не исследованную область, огороженную некими «красными флажками» — теорией размерных величин [89].

Арифметические операторы сложения, вычитания, умножения и деления, задействованные в решениях на рис. 19.1, 19.4, 19.5 и 19.7, имеют два операнда. В среде Mathcad 15 панель Calculator (см. рис. 19.9) кроме этих четырех операторов с двумя операндами содержит еще три других: *возведение в степень*, *логарифм с произвольным основанием* и *корень n-ой степени*. Кнопки ввода в расчет этих семи операторов[5] на рис. 19.9 выделены прямоугольниками.

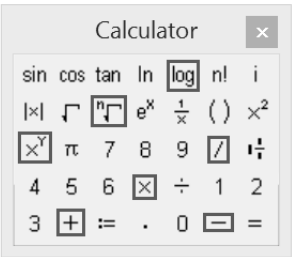


Рис. 19.9
Панель калькулятора Mathcad 15

[4] Этот древнегреческий математик, кстати, придумал и термин «гипербола».

[5] Семь не только красивое число. Эта «красивость» имеет и некое физическое обоснование — вспомним семь цветов видимого спектра света, семь нот европейской гаммы, семь базовых единиц СИ, семь дней недели и т. д. Еще можно упомянуть семь чудес Света, семь древних мудрецов и т. д. и т. п. Еще одно красивое число — тройка — см. http://twt.mpei.ac.ru/ochkov/Izvest_Art/art.htm.

Возникла крамольная идея пропустить через расчет, показанный на рис. 19.1, и остальные три вычислительные оператора элементарной математики: степень, логарифм и корень, получив *великолепную семерку вычислительных кривых*. Но тут возникла одна проблема, вызвавшая бурную дискуссию у пользователей Mathcad — см. **https://www.ptcusercommunity.com/thread/137385**.

Дело в том, что при построении кривых, показанных на рис. 19.1, 19.4, 19.5 и 19.7, была задействована *единица длины* метр. Обычно такие графики строят в безразмерных координатах — длина и все тут: никаких метров, сантиметров, футов или дюймов... Но пакет Mathcad оборудован инструментарием физических величин [4], которым было бы грешно не воспользоваться, например, для контроля правильности ввода формул. Ввел пользователь, например, не ту степень у переменной — расчет без метров это «проглотит»[6] и выдаст ложное решение, а с метром прервет расчет и выдаст сообщение об ошибке. В наших расчетах на рис. 19.1, 19.4 и 19.5 параметр a был равен шести метрам (рис. 19.1 — эллипс), двум метрам (рис. 19.4 — гипербола) и 20, 6 и 0.45 квадратным метрам (рис. 19.5 — овалы Кассини). В расчете на рис. 19.7 параметр a был безразмерным, вернее, имел размерность метр, деленный на метр, но отрезки прямых, по которым строилась окружность Аполлония, по-прежнему имели «метровую» размерность.

Если же мы захотим построить замкнутую кривую с характеристикой Black в степени Blue, равной параметру a, то в среде Mathcad от единиц длины придется отказаться, т. к. расчет будет прерываться сообщением об ошибке — *показатель степени не может быть размерной величиной*. А собственно, почему мы не можем возводить что-то в размерную степень? Обычно тут следуют возражения в том плане, что нет задач, требующих этого. Но мы сейчас покажем одну такую задачу!

Пусть два метра, возведенные в степень три метра, дадут в ответе 8 с размерностью, суть которой нужно еще обсудить. Пока положим, что эта единица измерения имеет символ m^m. Но это не метр в степени метр, а некий новый символ, отображающий данную метрическую закономерность[7]. Основное возражение против размерной степени сводится, повторяем, к тому, что нет в природе таких физических величин. Но вот наша очередная замкнутая кривая, кривая, где $Black^{Blue} = a$, подсказывает, что это совсем не так и что мы можем что-то измерять единицей, где метр возводится в метр.

Кривая, показанная на рис. 19.10, похожа на овал Кассини с той лишь разницей, что две половинки новой кривой не симметричны, что четко видно в нижней части рис. 19.10. Эту несимметричность можно рассматривать как графическую иллюстрацию положения о том, что перестановка местами операндов оператора возведения в степень меняет результат. Следствием этого положения является и то, что замкнутая кривая на рис. 19.10 имеет две ветви: одну для выражения $Black^{Blue} = a$, а вторую — для выражения $Blue^{Black} = a$. Изменение формы этих двух ветвей при изменении значения параметра a показано в анимации здесь **https://www.ptcusercommunity.com/thread/137466**. Верхняя замкнутая кривая на рис. 19.10 напоминает... деку гитары, что дает основание дать ей соответствующее имя «Гитара» или «Гитара Аполлона» (см. название этюда). В наше время Аполлон играл бы не на лире, а на гитаре.

---

[6] Один участник дискуссии об вычислительных кривых Mathcad опубликовал кривую, построенную по безразмерному расчету: в нем разность a - L и сама переменная L (см. третью строку на рис. 19.1) по ошибке не были возведены во вторую степень. Ложный график был построен, но другой посетитель форума заметил эту ошибку. Но лучше, конечно, чтобы сам пакет замечал такие ошибки и не выдавал ложные решения. Это бы случилось, если б единицы измерения не были отключены.

[7] Ту нет ничего необычного. В математике и в математических пакетах некоторые операторы и символы нередко трактуются двояко. Выражение x^T, например, в зависимости от ситуации может означать и переменную x в степени T и транспонирование матрицы с именем x. Так что вполне будет обоснованным толковать m^m (или cm^{cm}, m^{mm}, ft^{in} и т. д.) не только как единицу длины в степени единицы длины, а как нечто иное, делающее правильные пересчеты физической величины с такой размерностью.

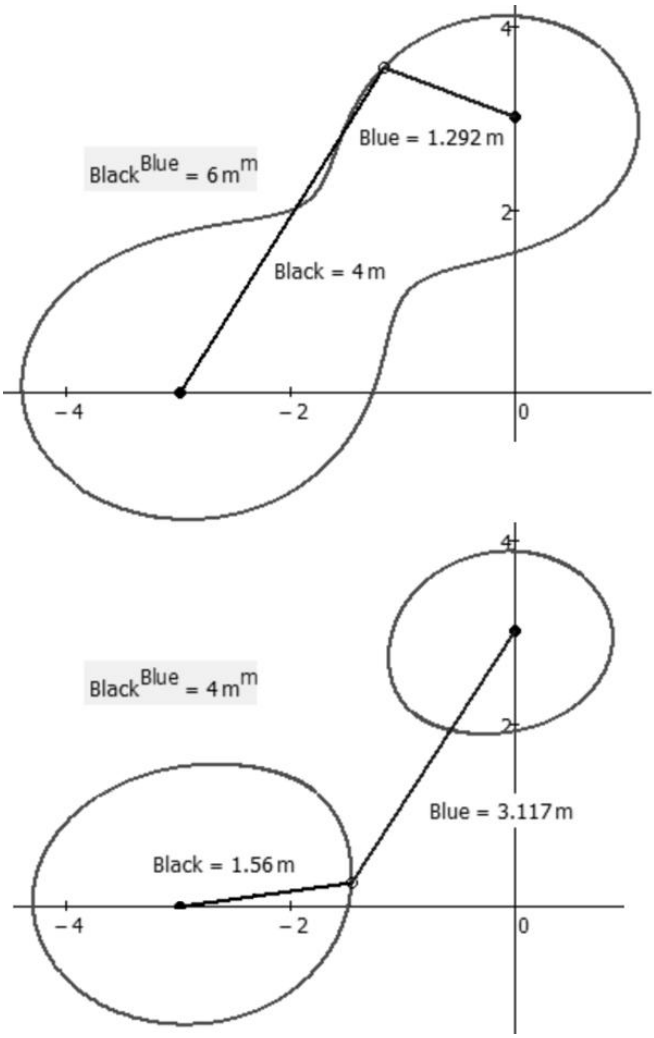


Рис. 19.10
Кривая возведения в степень (анимация https://www.ptcusercommunity.com/thread/137466)

Можно попытаться «засунуть» метр не только в показатель степени, но и в логарифм. Тем более уже были прецеденты взятия логарифма размерной величины — см. рис. 19.11, где показано вычисление разности логарифмов двух размерных величин — давлений. А такая операция часто проводится в термодинамике [2].

$$log(10\ atm) - log(1\ bar) = 1.006$$

Рис. 19.11
Логарифм размерной величины

Фокус расчета на рис. 19.11 в том, что не показано промежуточное вычисление логарифма дроби. Но этот пример тут не совсем уместен. Так что вернемся к нашим замкнутым кривым.

На рисунке 19.12 показана кривая, отвечающая условию `log(Black, Blue) = a`. Тут у семерки (`a = 7`) совсем уж экзотическая единица измерения `log(m, m)`, которую также еще нужно уметь правильно толковать (см. выше).

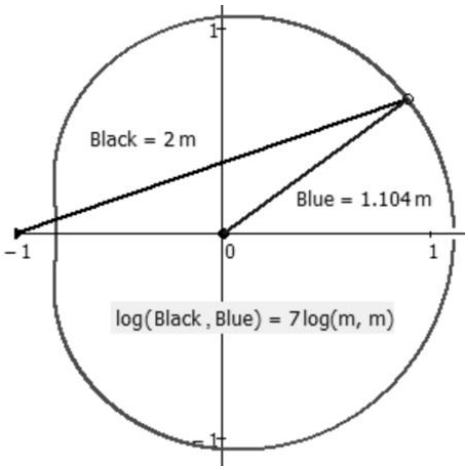


Рис. 19.12
Кривая логарифма (анимация https://www.ptcusercommunity.com/videos/7042)

Кривая, показанная на рис. 19.12, также имеет две ветви: одну для выражения `log(Black, Blue) = a`, а вторая — для выражения `log(Blue, Black) = a`. Изменение формы этих двух ветвей при изменении значения параметра `a` показано в анимации здесь **https://www.ptcusercommunity.com/thread/137465**. Как и в случае с гиперболой, параметр `a` может иметь и отрицательные значения.

И наконец, на рис. 19.13 показана седьмая кривая, отвечающая седьмому двухоперандному вычислительному оператору из панели, показанной на рис. 19.9 — взятию n-го корня числа. Но этот оператор подразумевает в среде Mathcad, что n — это целое число, большее единицы. Поэтому мы немного схитрим и заменим корень на степень с обратным показателем: $\mathtt{Black}^{1/\mathtt{Blue}}$ и $\mathtt{Blue}^{1/\mathtt{Black}}$.

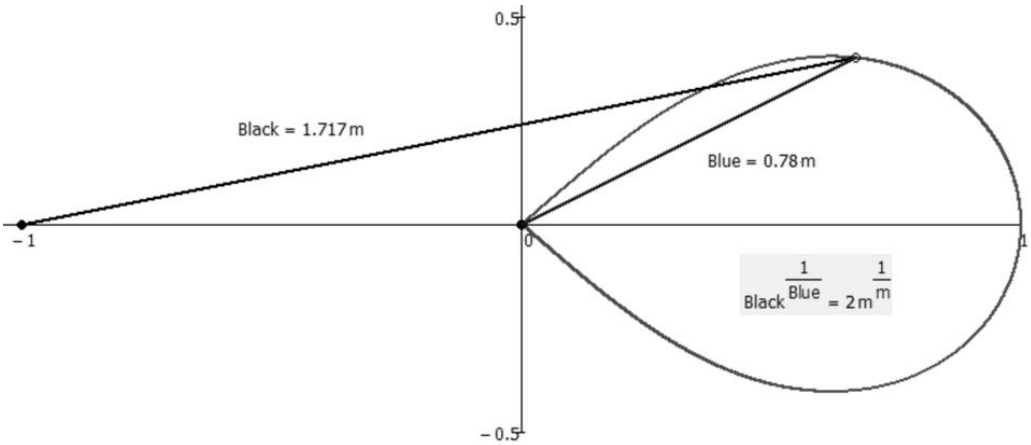


Рис. 19.13
Кривая корня n-й степени (анимация https://www.ptcusercommunity.com/videos/7043)

Кривая, показанная на рис. 19.13, также имеет две ветви: одну для выражения $\mathtt{Black}^{1/\mathtt{Blue}} = \mathtt{a}$ и вторую для выражения $\mathtt{Blue}^{1/\mathtt{Black}} = \mathtt{a}$. Изменение формы этих двух ветвей при изменении значения параметра a показано в анимации здесь **https://www.ptcusercommunity.com/thread/137464**.

Попытки авторов найти в литературе или Интернете подобные «незаконнорожденные» (с точки зрения теории размерностей) кривые (рис. 19.9, 19.12 и 19.13) не увенчались успехом. Может быть, читателям это удастся. Эти кривые, повторяем, требуют более глубокого математического анализа — нахождения алгебраических выражений, их описывающих, построения квадратур и проч., как это сделано для эллипса, гиперболы, овалов Кассини и окружностей Аполлония.

Начало такого анализа может быть таким. Если, например, нашу экспоненциальную кривую на рис. 19.9 определить через систему уравнений:

$$\mathrm{eq}(a,b,c) = \begin{bmatrix} b^2 = x^2 + y^2 \\ a^2 = (x-c)^2 + y^2 \end{bmatrix}, \quad \text{где} \quad b = \frac{\ln(\kappa)}{\ln(f)},$$

то она на может быть параметризирована так:

$$x = \frac{c^2 \cdot \ln(a)^2 - a^2 \cdot \ln(a)^2 + \ln(\kappa)^2}{2 \cdot c \cdot \ln(f)^2} \quad \text{и} \quad y = \frac{\sqrt{\Delta}}{2 \cdot c \cdot \ln(f)^2},$$

где

$$\Delta = \ln(a^a \cdot a^c \cdot \kappa) \cdot \ln\left(\frac{a^a \cdot \kappa}{a^c}\right) \cdot \ln\left(\frac{\kappa}{a^a \cdot a^c}\right) \cdot \ln\left(\frac{a^a}{a^c \cdot \kappa}\right).$$

В полярных координатах это будет выглядеть так:

$$\cos(\varphi) = \frac{\rho^2 + c^2 - \kappa^{2/\rho}}{2 \cdot \rho \cdot c}.$$

Обсуждение этого решения можно найти на сайте **https://www.ptcusercommunity.com/thread/137356**.

Вопрос о том, какую размерность и физический смысл должна иметь длина, возведенная в степень длины, остается открытым. Тем не менее, не дожидаясь окончательного решения этой метрологической проблемы, первый автор книги собирается на своем дачном участке соорудить реальную, «физическую» клумбу по размерам, отображенным на рис. 19.14[8]. Эта клумба будет иметь реальные *физические* параметры: расстояние между фокусами, длину периметра, площадь и... наш параметр a — длину в степени длины. Физическая величина есть, а единицы ее измерения пока нет. Единица $\mathbb{m}^{\mathbb{m}}$, о которой говорилось выше, вызывает споры и сомнения, требует более детального токования. Но какая-то единица измерения тут должна быть, чтобы, в частности, вести пересчеты.

Представим себе, что кто-то в США захочет соорудить себе такую же клумбу, имея в руках рулетку, но не с *метрами и сантиметрами* на делениях, а с *футами и дюймами*. На какое другое число вместо четырех (см. рис. 19.14) тут нужно будет опираться? Это число также должно иметь размерность. Вопрос о ее виде и сути мы, повторяем, оставляем открытым. Более того, можно попытаться соорудить

[8] Клумба может быть размечена так. Вбиваются два колышка, фиксирующие фокусы этого «степенного овала Кассини с талией». Далее на первой рулетке отмеряется отрезок Black в метрах и вычисляется значение отрезка Blue тоже в метрах по формуле ln(4)/ln(Black). На второй рулетке отмеривается полученное расстояние в метрах и вбивается первый колышек, намечающий контуры будущей клумбы. Операция повторяется с новым значением длины отрезка Black, пока контур клумбы, отмечаемый колышками, не замкнется. Так, кстати говоря, сам «глупый» пакет Mathcad поточечно строил наши овалы и другие кривые. Человек же без компьютера в таких случаях начинает анализировать зависимости, искать асимптоты, нули и другие особые точки на графике и только после этого рисует в уме или на бумаге кривые качественно, а не количественно.

клумбу, где $Black + Blue^2 = a$, но это будет уже явным нарушением закона размерных величин: нельзя складывать величину с ее квадратом. Наше же нарушение этого закона по отношению к «степенному овалу Кассини» (рис. 19.9 и 19.13) не так явно. А вот другая сторона этой проблемы. В задачниках по математике можно увидеть такого рода задачи: решить квадратное уравнение типа $2x^2 + 3x - 4 = 0$. При этом ни у кого не возникает мысль, что нельзя складывать квадрат величины с самой величиной. Тут имеется в виду, что коэффициенты при x^2 и x (двойка и тройка) содержат что-то такое, связанное с пересчетом размерностей, что позволяет делать такое «незаконное» сложение. Выражение $Black + Blue^2$, по которому нужно будет построить кривую, тоже может содержать некие единичные коэффициенты с размерностями, позволяющими складывать величину (длину) и квадрат величины (площадь).

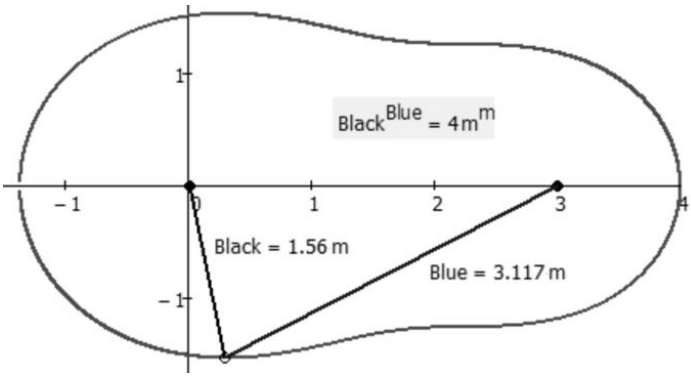


Рис. 19.14
Экспоненциальная клумба

Вторая тема этюда такая: современные вычислительные средства позволяют отойти от изощренных и непонятных для многих алгоритмов построения кривых и вернуться к... истокам.

Вспомним, что такое окружность, эллипс, парабола, гипербола и проч., о чем было рассказано выше. Это геометрическое место точек на плоскости, отвечающих какому-то заданному признаку. Давайте же, не мудрствуя лукаво (а это мы делали, создавая расчет на рис. 19.1), просто-напросто запрограммируем это определение — просканируем ограниченную область плоскости, где могут находится эти кривые, и отметим точки, отвечающие заданному признаку. Мы уже прибегали к этому методу, строя график сложной функции в виде плетеной салфетки — см. рис. 4.44 на стр. 97. На рисунке 19.15 показано построение замкнутой кривой с тремя фокусами, расстояния от которых до кривой отвечает такому условию: $|Black + Blue - Green| = a$.

В расчете на рисунке 19.15 область на плоскости от заданных значений x_1 до x_2 и от y_1 до y_2 разбивается на отдельные точки, из которых собираются в двух векторах X и Y и выводятся на график только те, какие отвечают заданному свойству кривой. Но не по условию «равно», а по условию «примерно равно» — мы задачу решаем не аналитически, а численно и, следовательно, приближенно. Как говорил один киноперсонаж: «Это не эстетично, зато дешево, надежно и практично». Более «эстетичный» вариант расчета, использующий предварительный анализ зависимостей, и символьное (аналитическое) решение системы уравнений показан на первом рисунке этюда.

На рисунке 19.15 мы получили некий гибрид эллипса и гиперболы — некую *эллипсогиперболу*, у которой от эллипса осталась замкнутость кривой, а от гиперболы — две ветви, одна из которых находится внутри другой. Эллипсогиперболой

эту кривую можно назвать и потому, что при ее построении использовалась и сумма (эллипс) и разность (гипербола). На сайте **https://www.ptcusercommunity.com/message/458540** можно увидеть в анимации, как эта кривая превращается в эллипс или гиперболу при плавном изменении ее параметров в ту или иную сторону.

$\approx(a, b) := |a - b| < 0.0001 \quad ORIRIN := 1 \quad n := 5000$

$x_1 := -2.2 \quad x_2 := 0.2 \quad y_1 := -1.2 \quad y_2 := 1.2 \quad a := 0.2$

$x_{F1} := -1 \quad y_{F1} := 0 \quad x_{F2} := 0 \quad y_{F2} := 0 \quad x_{F3} := 0 \quad y_{F3} := -1$

```
(X)
(Y) := | i ← 0
       | for  x ∈ x1, x1 + (x2 − x1)/n .. x2
       |   for  y ∈ y1, y1 + (y2 − y1)/n .. y2
       |     | Black ← √((x − xF1)² + (y − yF1)²)
       |     | Blue ← √((x − xF2)² + (y − yF2)²)
       |     | Green ← √((x − xF3)² + (y − yF3)²)
       |     | if |Black + Blue − Green| ≈ a
       |     |   | i ← i + 1
       |     |   | Xi ← x
       |     |   | Yi ← y
       | (X)
       | (Y)
```

$i := 80 \qquad Black := \sqrt{(X_i - x_{F1})^2 + (Y_i - y_{F1})^2} = 0.664$

$Blue := \sqrt{(X_i - x_{F2})^2 + (Y_i - y_{F2})^2} = 1.603 \qquad Green := \sqrt{(X_i - x_{F3})^2 + (Y_i - y_{F3})^2} = 2.067$

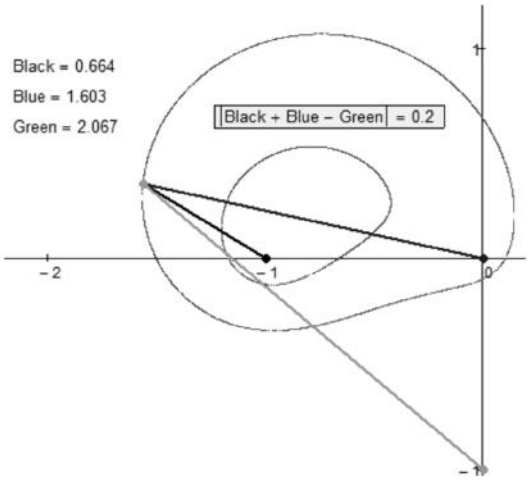


Рис. 19.15

*Универсальный метод рисования кривых по заданным условиям (анимация **https://www.ptcusercommunity.com/thread/137779**)*

«Практичность» же расчета на рисунке 19.15 состоит в том, что его несложно преобразовать так, чтобы можно было работать с любым числом фокусов и любым равенством[9], задающим вид кривой. Это позволит создавать очень занимательные и поучительные кривые не только в «статике», но и в «кинематике» — в анимации. Что мы и сделали: пример на рис. 19.16.

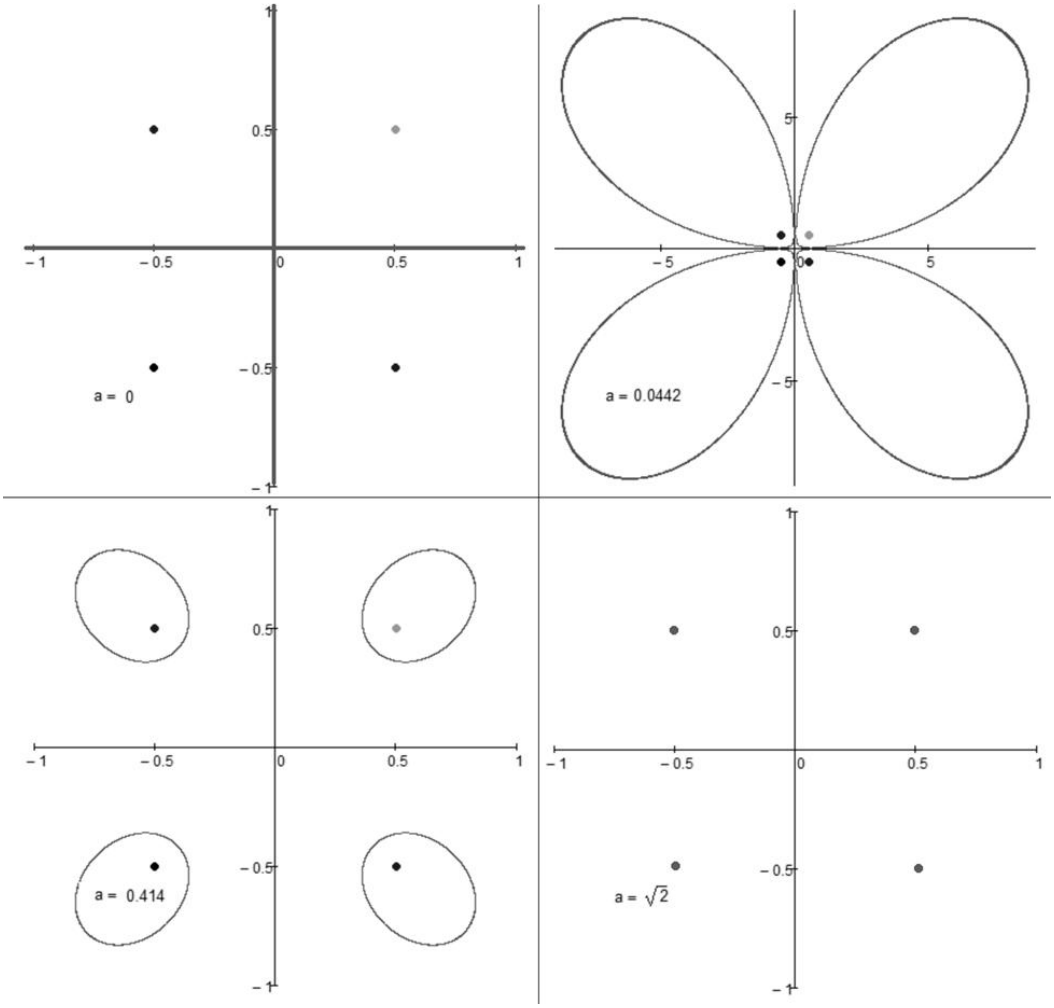


Рис. 19.16

Четырехлистник (анимация https://www.ptcusercommunity.com/thread/137971)

Кривые на рис. 19.16 имеют четыре фокуса и отвечают равенству `|L1 - - L2 + L3 - L4| = a`, которое опять же роднит их и с эллипсом (сумма) и с гиперболой (разность). При `a = 0` мы имеем четыре ветви «гиперболы», которые слились с осями графика. При значении `a` больше нуля эта гипербола-крест сворачивается в четыре овала. При значении `a`, равном корню из двух, листочки этого четырехлистника сжимаются в точки.

Интересно также отображать наши кривые не только в декартовых, но и в полярных координатах. Если говорить о полярном графике, то тут в среде Mathcad

[9] Использование вместо равенств неравенств, кстати говоря, позволит строить не линии, а фигуры.

также когда-то были перейдены некие «красные флажки»: в последних версиях этого пакета стало возможным отображать и отрицательные значения координаты-радиуса, что является довольно спорным моментом с точки зрения классической математики. Обычно полярные графики используют там, где аргумент меняется «по кругу» — от нуля до 360 угловых градусов. Типичный пример на рис. 19.17, где отображена самая простая «круговая», вернее, периодическая функция — синус.

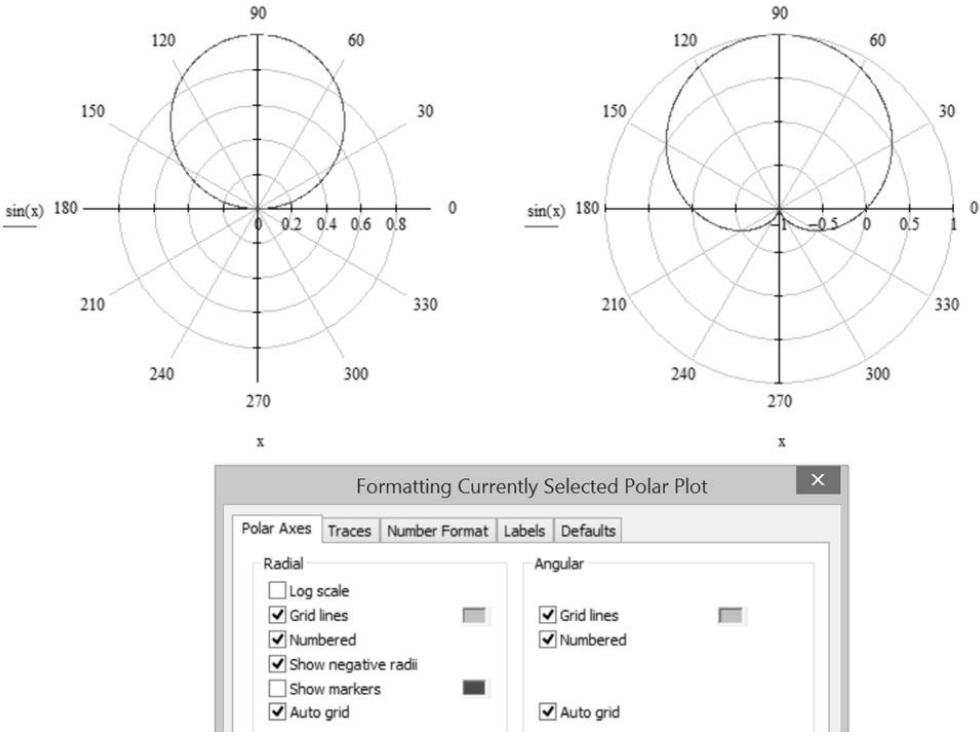


Рис. 19.17

Синус в полярных координатах: слева — только с положительными радиусами-координатами, справа — и с отрицательными радиусами-координатами

В окне форматирования полярного графика пакета Mathcad появилась галочка Show negative radii (Показывать отрицательные радиусы), включение которой меняет вид графика синуса с окружности (это фактически не синус, а абсолютное значение синуса) на некое перевернутое сердечко (синус без модуля).

Если попытаться построить в полярных координатах нашу самую простую «вычислительную» кривую — окружность (см. ее декартов график на рис. 19.7), включив при этом опцию отрицательных радиусов, то можно получить довольно интересные графики — см. рис. 19.18.

Если же на полярном графике отключить опцию показа отрицательных радиусов, то сердечки и овалы (красные линии) превратятся в окружности, а их криволинейные радиусы (синие линии) выпрямятся, как это отображено на декартовом графике. Кроме того, пропадет вторая «фантомная» окружность. А почему фантомная?! У гиперболы, как известно, две ветви — см. рис. 19.4. Можно предположить, что и у окружности, эллипса и у других «вычислительных» кривых тоже имеется некомплексная (действительная) вторая ветвь, которую можно увидеть только на полярном графике, «перепрыгнув через красные флажки» — допустив

отрицательные значения координаты-радиуса. Мы не зря упомянули здесь комплексные значения — рассмотренные в этюде вычислительные операторы и функции, в частности, степень и логарифм имеют в комплексной области довольно сложные графики с несколькими ветвями, с периодичностью и проч.

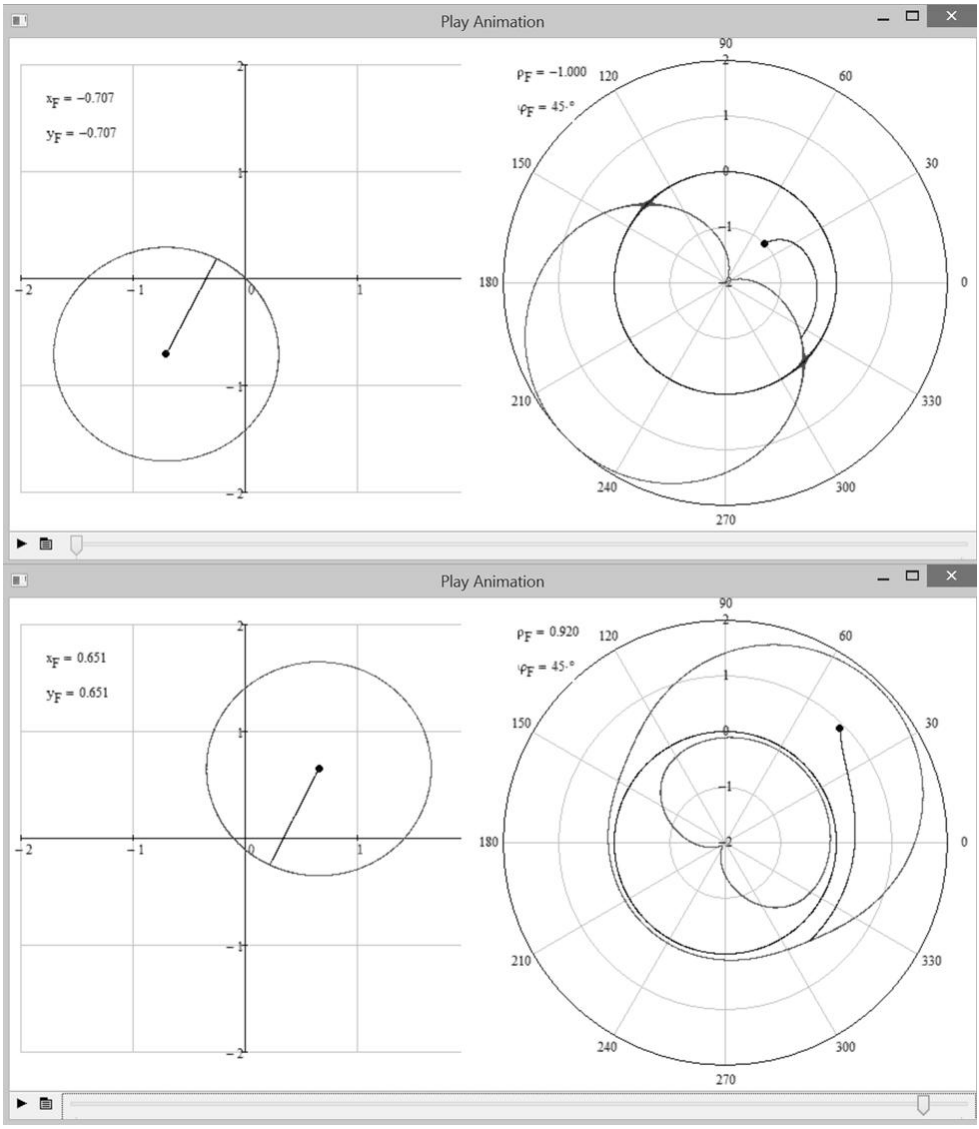


Рис. 19.18

Окружность с разными координатами центра на декартовом и полярном графиках

Оператор взятия корня n-й степени (см. рис. 19.13), честно говоря, в данный этюд вставлен не совсем обосновано, т. к. этот оператор является модификацией оператора возведения в степень (рис. 19.9). Поэтому взоры авторов обратились к еще одному важному арифметическому оператору с двумя операндами *остаток от деления*. В среде Mathcad и во многих других программных средах такая функция имеет имя **mod** (деление по модулю). Она, в отличие, например, от возве-

дения в степень (рис. 9), логарифма (рис. 11) и корня (рис. 12), не нарушает общепринятых правил работы с размерными величинами: **mod(1.5 m, 0.4 m) = 0.3 m**. Проверка: **(1.5 m – 0.3 m) / 0.4 m = 3** (целое число). Уравнение **mod(x, y) = a** невозможно решить аналитически относительно переменной **x** или **y**. Поэтому данную кривую нельзя нарисовать по методике, показанной на рис. 19.1. Тут придется «тупо» сканировать, чтобы построить кривые, отвечающие равенству **mod(Black, Blue) = a** (первая ветвь) и **mod(Blue, Black) = a** (вторая ветвь) — см. рис. 19.19 (четыре кадра анимации) и 19.20 (отдельные ветви выделены цветом).

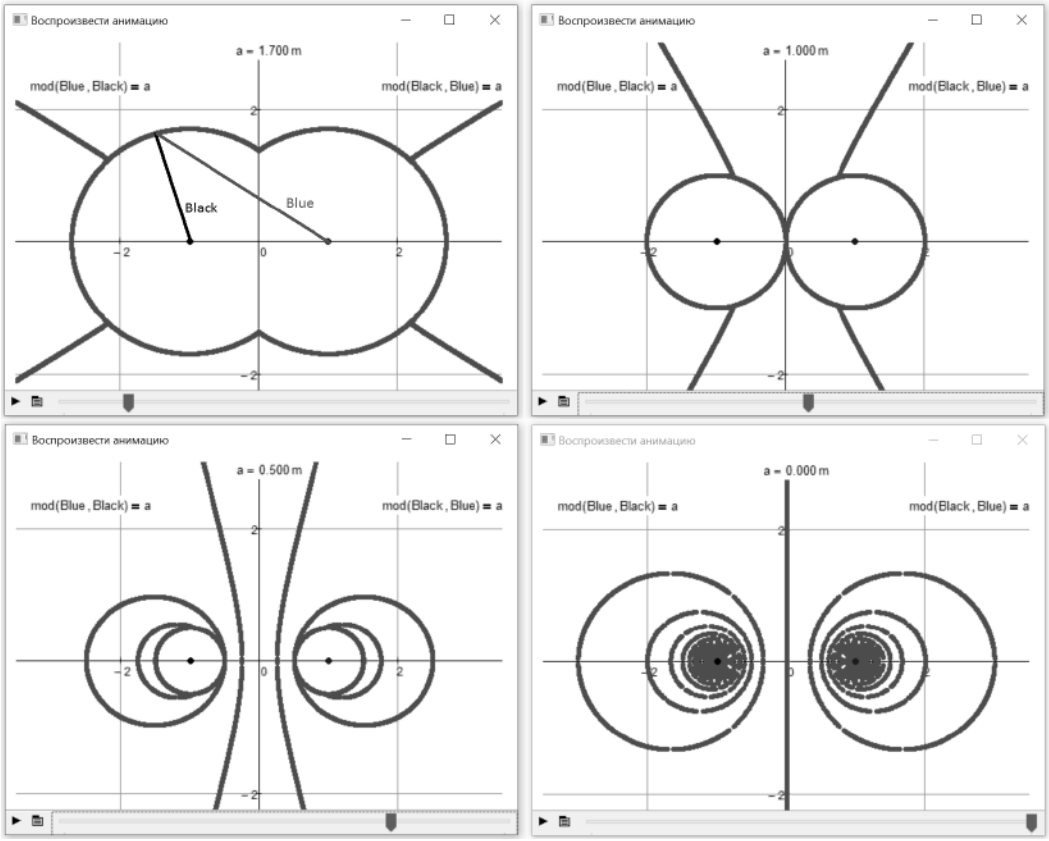


Рис. 19.19

Кривые остатка от деления (анимация https://www.ptcusercommunity.com/message/468690)

При `a = 0` кривые функции деления без остатка превращаются во множество окружностей Аполлония — колес велосипеда, отмеченного в названии этюда.

«Великолепная семерка» вычислительных кривых может выглядеть так: (1) сложение (эллипс), (2) вычитание (гипербола), (3) умножение (овал Кассини), (4) деление (окружность Аполлония), (5) остаток от деления (?), (6) возведение в степень (?) и (7) логарифм по заданному основанию (?). Две последние кривые можно считать кандидатами в «партию вычислительных кривых» из-за нерешенного «метрического вопроса». Но если возведение в степень еще можно как-то оставить в списке вычислительных кривых, то логарифм тут совсем уж не уместен. Попробуем заменить его еще чем-то.

Если вычитание можно считать операцией, обратной сложению, а деление — операцией, обратной умножению, то гиперболу можно считать кривой, «обратной» эллипсу, а окружности Аполлония — кривыми, «обратными» овалу Касси-

ни. На рисунке 19.21 можно увидеть кривые, «обратные» кривым операции деления без остатка.

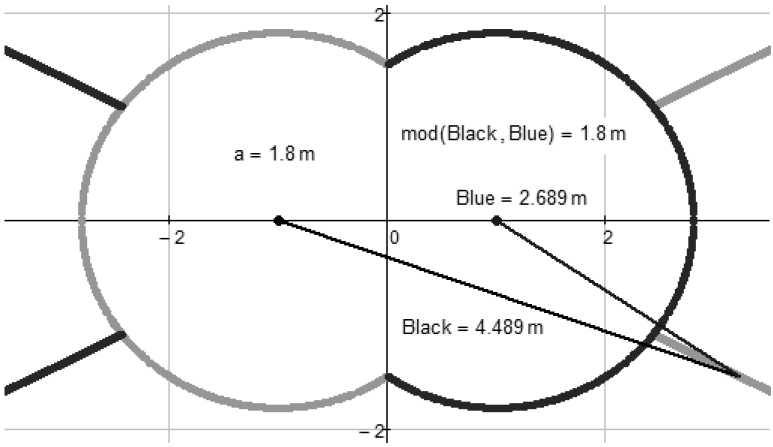


Рис. 19.20
Две ветви кривой остатка от деления, выделенные цветом

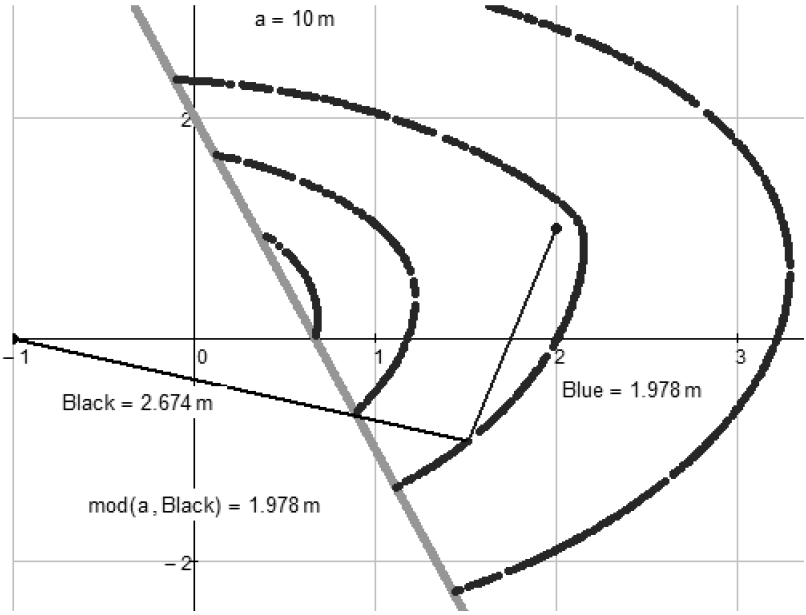


Рис. 19.21
Кривые, «обратные» операции деления без остатка

Авторы благодарны Вернеру Эксингеру (Werner Exinger) и Альваро Диасу (Alvaro Diaz) за участие в обсуждении этюда и создании некоторых расчетов и анимаций к ним.

ДИВЕРТИСМЕНТ. ЖУК ВТОРОГО ПОРЯДКА

Если спросить школьников или студентов, что такое *парабола*, то почти все скажут, что это линия, отображающая на графике квадратное уравнение. И только единицы вспомнят, что парабола — это геометрическое место точек на плоско-

сти с определенным свойством. Такая же ситуация с гиперболой. Все помнят ее простейшую «школьную» формулу `y = 1/x`, но забывают о геометрическом месте точек. С *окружностью* ситуация обратная. Все знают про точки на плоскости, равноудаленные от центра окружности, но мало кто сходу напишет алгебраическое выражение, по которому строится окружность на декартовом графике.

Давайте рассмотрим «геометрическое», а не «алгебраическое» определение параболы, гиперболы, а также «сплюснутой» окружности (эллипса) и построим их графики, опираясь не на алгебраические выражения, а на «поточечные» определения.

Начнем с параболы. Если на плоскости провести прямую линию и поставить около нее точку, то парабола будет состоять из точек, равноудаленных от прямой линии (от *директрисы* параболы) и от заданной точки (от *фокуса* параболы). Такое поточечное построение параболы несложно сделать на бумаге, взяв в руки карандаш, циркуль и угольник: раздвигаем циркуль на определенный угол (рис. 19.22), ставим его иглу в точке-фокусе и чертим дугу окружности. Затем с помощью угольника проводим под прямым углом от директрисы к дуге прямую линию так, чтобы на дуге окружности поставить точку, равноудаленную от фокуса и директрисы. Далее меняем угол раскрытия циркуля и повторяем все сначала, проставляя на графике все новые и новые точки будущей параболы.

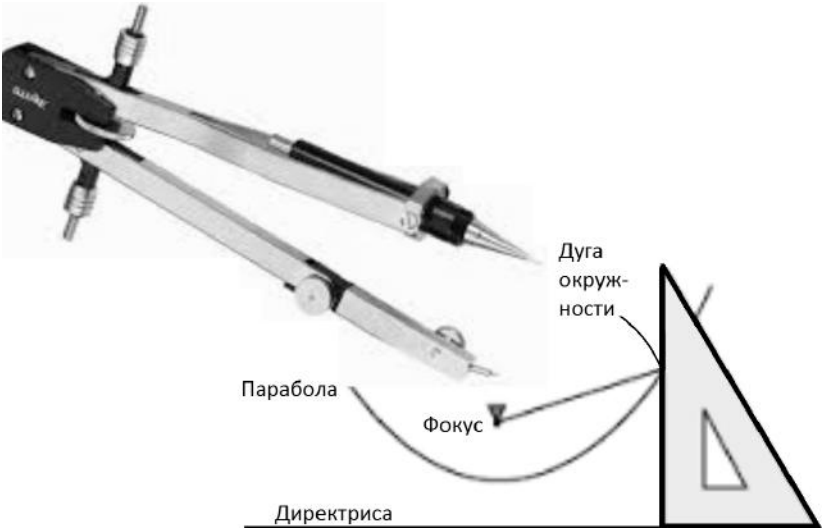


Рис. 19.22
Построение параболы циркулем и угольником

Но современные школьники и студенты, увы, постепенно отучаются работать с реальными чертежными инструментами, так как все можно начертить на экране компьютера, а затем перенести начерченное на бумагу принтера или даже «вылепить» модель на 3D-принтере.

По методике, показанной на рис. 19.22, можно построить также и гиперболу с эллипсом. Нужно будет только отрезки, откладываемые циркулем и линейкой, делать не равными, а отличающимися на некую константу, называемую *эксцентриситетом* (ε) На рисунке 19.23 показано построение гиперболы, параболы и эллипса сканированием (см. рис. 19.15). У эллипса $\varepsilon < 1$, у параболы $\varepsilon = 1$, а у гиперболы $\varepsilon > 1$.

В этюде 3 мы упоминали жука, но не простого, а золотого, который помог герою рассказа Эдгара По найти клад. У нас на рис. 19.23 получился жук, и тоже не простой, а... второго порядка, составленный из эллипса, параболы и двух ветвей

гиперболы — из кривых второго порядка, исследованных в дивертисменте 1 этюда 9 (стр. 214). К эксцентриситету гиперболы мы еще вернемся в этюде 20, когда будем рассматривать оптическое свойство этой кривой.

$\approx(a,b) := |a-b| < 0.001 \qquad n := 3000$

$x_1 := -1 \qquad x_2 := 1 \qquad y_1 := 0 \qquad y_2 := 1.6 \qquad x_F := 0 \qquad y_F := 1 \qquad Dir := 0.5$

$$\begin{pmatrix} X \\ Y \end{pmatrix} := \left| \begin{array}{l} \text{for } x \in x_1, x_1 + \dfrac{x_2 - x_1}{n} .. x_2 \\ \quad \text{for } y \in y_1, y_1 + \dfrac{y_2 - y_1}{n} .. y_2 \\ \quad\quad \left| \begin{array}{l} \left[L1 \leftarrow \sqrt{(x - x_F)^2 + (y - y_F)^2} \quad L2 \leftarrow y - Dir \right] \\ \left(X_i \leftarrow x \quad Y_i \leftarrow y \quad i \leftarrow i + 1 \right) \text{ if } \left| \begin{array}{l} \text{Ellipse} \\ \varepsilon \leftarrow 0.53 \\ L1 \approx (\varepsilon \cdot L2) \end{array} \right. \vee \left| \begin{array}{l} \text{Parabola} \\ \varepsilon \leftarrow 1 \\ L1 \approx (\varepsilon \cdot L2) \end{array} \right. \vee \left| \begin{array}{l} \text{Hyperbola} \\ \varepsilon \leftarrow 3 \\ L1 \approx |\varepsilon \cdot L2| \end{array} \right. \end{array} \right. \\ \begin{pmatrix} X \\ Y \end{pmatrix} \end{array} \right.$$

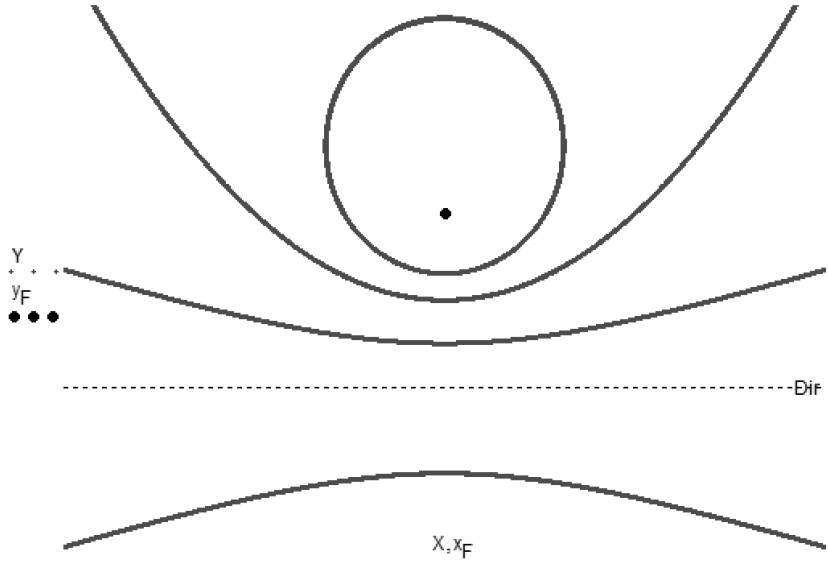


Рис. 19.23
Эллипс, парабола и две ветви гиперболы: жук второго порядка
*(анимация: **https://www.ptcusercommunity.com/thread/140084**)*

Задания читателям

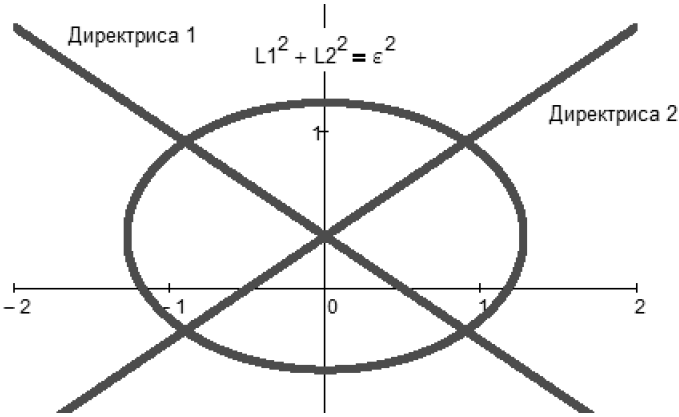
1. Вывести аналитические выражения для новых «вычислительных кривых», рассмотренных в этюде.

2. Определите координаты центров двух окружностей (колес велосипеда Аполлония), показанных на рис. 19.8.

3. Велосипед Аполлония (см. рис. 19.8) имеет втулки колес, размещенные не в их центрах, а в фокусах окружностей. Какой формы должна быть велодорожка, чтобы такой велосипед двигался строго горизонтально? Смотри на рис. 7.19 (стр. 156) такую дорожку для велосипеда с квадратными колесами.

4. Построить полярные графики других «вычислительных» кривых, описанных в этюде. В практической пользе этой работы можно сомневаться, но графики будут получаться весьма занимательными. А это главное в математическом образовании с использованием вычислительных пакетов. См. также эпиграф этюда.

5. Нарисуйте на плоскости две прямые линии (две директрисы) и еще одну (исследуемую) линию, точки которой находятся в определенных отношениях к директрисам. Пример на рисунке внизу. Исследуйте полученную кривую! А могут ли директрисы быть не прямыми, а кривыми?



6. Список вычислительных кривы можно дополнять. Читатель сам или со своими студентами может поэкспериментировать в этом направлении. В частности, найти седьмую «законную» по отношению к закону размерностей вычислительную кривую. А пока у нас их только шесть: сложение и «обратное» сложение (вычитание), умножение и «обратное» умножение (деление), деление без остатка и «обратное» деление без остатка.

7. Сконструируйте реальный циркуль, который бы рисовал не окружности, а эллипсы, овалы Кассини, отрезки гипербол, параболу и прочие интересные кривые. Детали такого циркуля распечатайте на 3D-принтере и соберите из них этот чертежный прибор.

Этюд 20

ПУТЕШЕСТВИЕ ОТ ДОМА К ШКОЛЕ ПО МАРШРУТУ ФЕРМА, ИЛИ ВТОРОЕ ОПТИЧЕСКОЕ СВОЙСТВО ГИПЕРБОЛЫ

В этюде рассмотрено решение задач оптимизации применительно к проблемам отражения и преломления лучей света. Представлено численное и аналитическое решение алгебраических уравнений линзы и фокусирующего зеркала. Показано, как современные программные средства могут легко и изящно решать оптические задачи. Подчеркнуто оптическое свойство гиперболы, базирующееся не на законе отражения, а на законе преломления света.

Математика: решение уравнения, оптимизация, плоские кривые второго порядка, теорема Ферма.

Физика: расстояние, время, скорость, геометрическая оптика, преломление и отражение света, принцип Ферма, принцип таутохронизма, оптический путь луча света.

Информатика и программирование: численное решение задачи оптимизации, численное решение алгебраического уравнения, компьютерные инструменты аналитических преобразований (символьная математика), компьютерная графика плоских кривых, вложенный цикл с параметром.

Сайт этюда: **https://www.ptcusercommunity.com/message/469479**

Дворник приколачивал к дому флаг.
Прохожий спросил, что это значит.
Дворник ответил: «Это значит, что в городе праздник».
«А какой это праздник?» — спросил прохожий.
«А праздник такой, что наш математик нашел
еще одно оптическое свойство гиперболы!» — ответил дворник.
Прохожий, устыженный своим незнанием, растворился в воздухе.

Даниил Хармс. «Праздник», новая редакция

Школьник или студент волен добраться из дома или общежития до учебного заведения разными путями. Он может идти по улице, может пробираться через дворы и закоулки, может проехать часть дороги на общественном транспорте и т. д. Кратчайшее расстояние между двумя точками, как известно, — прямая линия. Но далеко не всегда можно и нужно перемещаться по прямой. Даже если нет явных препятствий в виде зданий, заборов и прочих преград, то перемещение по прямой даже в чистом поле не всегда оказывается самым оптимальным, если под оптимизацией понимать не сокращение расстояний, а минимизацию времени в пути. Ведь ученики, опаздывая в школу, думают не о расстоянии, а о времени.

Давайте не будем голословными и для начала решим такую простенькую задачу: человек находится на открытой местности в точке A (рис. 20.1). Сгущаются тучи, гремит гром, вот-вот начнется гроза с ливнем... Нужно как можно быстрее добраться до точки B, где расположено некое укрытие: дом, беседка, навес... Как человеку бежать, если он стоит на газоне, перед домиком асфальт, а путь пересекает вспаханная полоса пашни? Мы уже решали что-то подобное в этюде 15 — см. рис. 15.13 на стр. 348. Но там был только газон и пашня, но не было асфальта.

Это типичная задача *оптимизации*, где *целевой функцией* будет время перемещения от точки A к точке B, а *переменными* оптимизации — абсциссы точек C и D. Ординаты точек C и D заданы и отмечают прямые линии раздела участков «газон — пашня» и «пашня — асфальт», которые параллельны оси абсцисс. Скорость бега по газону (г), вспаханному полю (п) и асфальту (а) тоже заданы и, естественно, разные: $v_а > v_г > v_п$. Заданы также координаты начальной и конечной точек A и B.

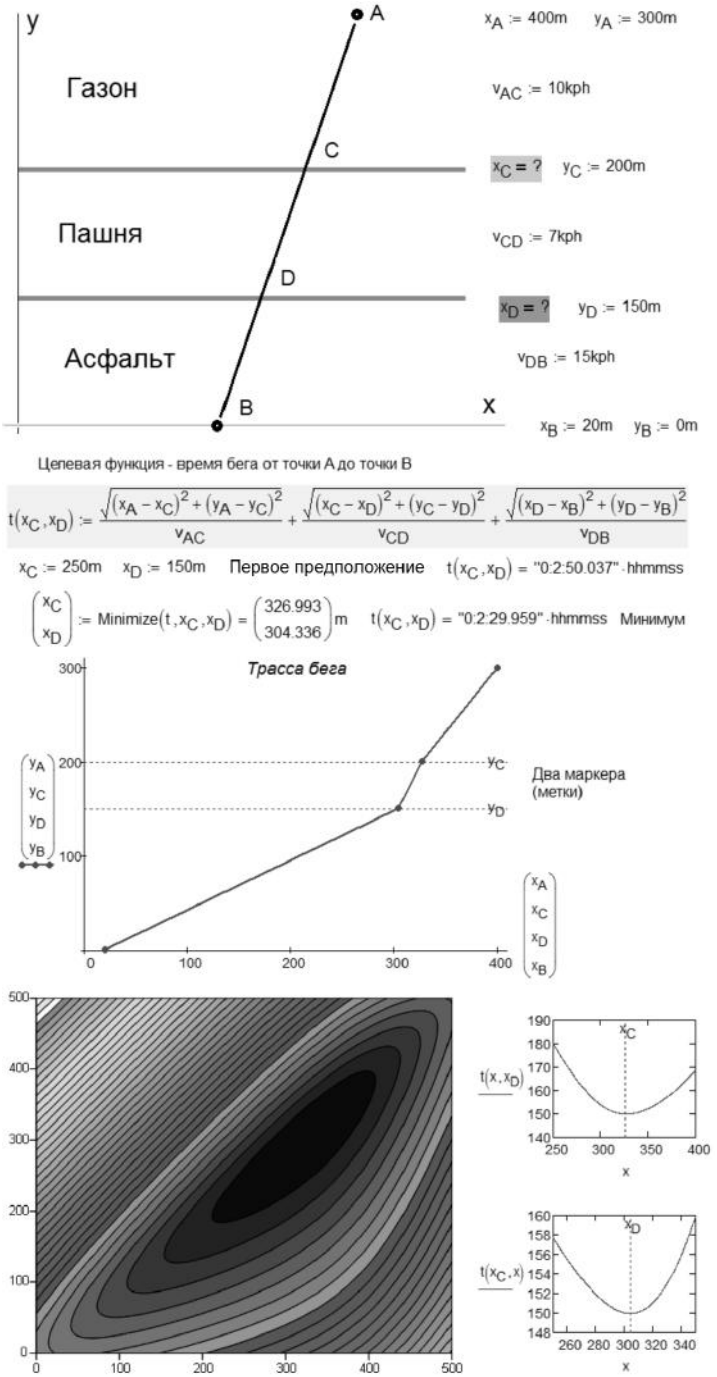


Рис. 20.1

Задача о беге по газону, пашне и асфальту: оптимизация

На рисунке 20.1 показано решение этой задачи с использованием встроенной функции `Minimize`. У нее переменное число аргументов. В нашей задаче их три:

имя целевой функции t (time) и имена двух переменных оптимизации x_C и x_D. Функция Minimize по особому численному алгоритму (см. этюды 4 и 13) будет автоматически менять значения переменных оптимизации x_C и x_D так, чтобы целевая функция приняла минимальное значение. Значения переменных x_C и x_D при проведении поиска изменяются от некоторых исходных (первое предположение — см. рис 20.1), которые задаются самим пользователем, исходя из «физики» задачи. При значениях x_C и x_D, взятых для первого предположения, время движения равно 2 мин 50 с, а минимизация с помощью функции Minimize дает значение 2 мин 31 с. Но возникают сомнения, минимум ли это, нет ли более оптимального маршрута бега по трем ломаным отрезкам прямой? Ответ проверяется построением контурного графика и двух сечений поверхности целевой функции в точке минимума: все верно — минимум найден: он окружен линиями одного уровня и расположен в точках минимума двух кривых сечений по двум переменным оптимизации. Кривые сечения тут лишние, но они будут необходимы, когда переменных оптимизации будет больше двух: человек, например, бежит по газону, пашне, болоту и потом уже по асфальту (три, а не две границы раздела фаз, три переменные оптимизации). Мы уже рисовали кривые сечений функции четырех аргументов, когда определяли стоимость автомобиля с пробегом — см. рис. 14.28–14.31 на стр. 329 и 330.

График в центре рисунка 20.1 с ломаной кривой напоминает... «луч света в темном царстве» и заставляет вспомнить о знаменитом принципе геометрической оптики, сформулированном в 1662 г. великим Ферма: свет от точки A к точке B проходит по такому пути, чтобы время его движения было минимально. Этот путь называется *оптическим путем*. Отсюда и изломы прямых линий, определяемые скоростями света в разных средах на границе раздела. Если одну скорость поделить на другую (смежную), то мы получим так называемый *относительный коэффициент преломления* n, численно равный отношению синуса угла падения к синусу угла преломления (*закон Снеллиуса*).

На рисунке 20.2 показан другой подход решению задачи о беге по газону, пашне и асфальту, сводящийся к поиску не минимума функции двух аргументов, а корня системы двух нелинейных алгебраических уравнений: отношение синусов углов на двух границах раздела равно отношению значений скоростей бега по обе стороны от границы. Синус угла — это отношение длины противолежащего катета к гипотенузе, высчитываемое по теореме Пифагора.

Функция Find, задействованная в решении на рис. 20.2, опять же особым численным алгоритмом с опорой на первое предположение так будет менять значения своих аргументов, чтобы система уравнений, записанная после ключевого слова Given (дано), превратилась в систему тождеств. Проверка правильности решения, как и на рис. 20.1, ведется графически: две кривые, соответствующие двум уравнениям системы, пересекаются в искомой точке (см. нижний график на рис. 20.2). График же строился методом сканирования, который мы более подробно (с показом программы) описали в этюде 19 — см. рис. 19.15 на стр. 389. Метод сканирования показан также ниже на рис. 20.12.

Примечание. Скорость света в атмосфере и, следовательно, коэффициент преломления зависит от плотности воздуха, его температуры. Из-за этого лучи света на небе могут, как школьники, «путешествовать» по весьма замысловатым маршрутам. Отсюда и миражи, какие можно видеть в пустыне или над гладью моря — см. рис. 20.3, где наблюдатель на берегу видит парусное судно («Летучий голландец») не на поверхности воды, а высоко в небе, де еще и перевернутым.

Рассмотрим финальный момент путешествия из дома в школу.

Школьники входят в ворота школы и направляются к ее дверям, что напоминает действие... *линзы*, фокусирующей свет. Давайте смоделируем этот оптический прибор, опираясь еще на один важный принцип геометрической оптики, непосредственно вытекающий из принципа Ферма, — на принцип *таутохронизма*: оптические пути лучей света от точечного источника до его изображения одинако-

вы, и свет затрачивает одно и то же время на прохождение этих *оптических путей*. Пакет Mathcad элегантно и просто может решить и такую более сложную задачу о линзе — см. рис. 20.4.

Рис. 20.2

Задача о беге по газону, пашне и асфальту: решение системы двух «оптических» уравнений

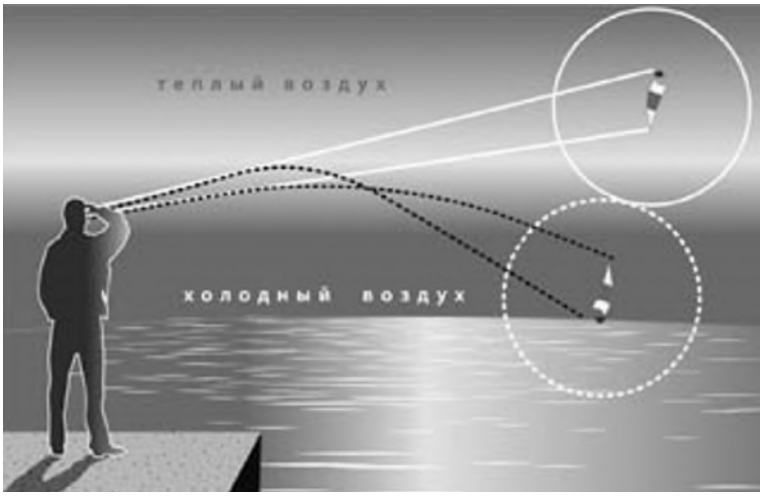
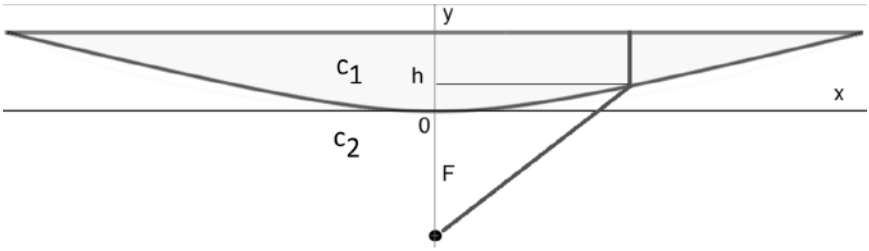


Рис. 20.3
Физическое объяснение миража (взято из Интернета)



$$\frac{h}{c_1}+\frac{F}{c_2}=\frac{h-y}{c_1}+\frac{\sqrt{(F+y)^2+x^2}}{c_2}\xrightarrow[\text{solve},F]{\text{substitute},\,c_2=n\cdot c_1}-\frac{x^2-n^2\cdot y^2+y^2}{2\cdot y-2\cdot n\cdot y}$$

$$F=\frac{x^2-n^2\cdot y^2+y^2}{2\cdot n\cdot y-2\cdot y}\qquad \text{или}\qquad F\cdot(2\cdot n\cdot y-2\cdot y)=x^2-n^2\cdot y^2+y^2$$

$$2\,F\cdot(n-1)\cdot y-x^2+(n^2-1)\cdot y^2=0$$

$$a_{11}\cdot x^2+a_{12}\cdot x\cdot y+a_{22}\cdot y^2+2\,a_1\cdot x+2\,a_2\cdot y+a_0=0$$

$$n:=2\qquad F:=1$$

$$a_{11}:=-1\qquad a_{12}:=0\qquad a_{22}:=n^2-1\qquad a_1:=0\qquad a_2:=F\cdot(n-1)\qquad a_0:=0$$

$$D:=\left\|\begin{bmatrix}a_{11} & a_{12}\\ a_{12} & a_{22}\end{bmatrix}\right\|=-3\qquad D<0\qquad \Delta:=\left\|\begin{bmatrix}a_{11} & a_{12} & a_1\\ a_{12} & a_{22} & a_2\\ a_1 & a_2 & a_0\end{bmatrix}\right\|=1\qquad \Delta\neq 0\qquad \mathit{Hyperbola}$$

Рис. 20.4
Вывод выражения для линзы с опорой на принцип таутохронизма

Плоско-выпуклая линза (см. схему на рис. 20.4) имеет в центре толщину `h` и *фокусное расстояние* `F`. Линза сделана из прозрачного материала с относительным коэффициентом преломления n. Спрашивается: какую форму должна иметь нижняя поверхность линзы, чтобы параллельный пучок света сошелся в *фокусе*? Левая часть уравнения на рис. 20.4 под схемой линзы, которое решается относительно переменной `F`, — это время прохождения луча света по оси `X` через стекло со

скоростью c_1 (первое слагаемое) и через воздух со скоростью c_2 (второе слагаемое). Правая же часть — это время прохождения луча света уже не через центр линзы, а в произвольном месте, отстоящем от центра на расстоянии x. Тут также два слагаемых для стекла и воздуха. Решение преобразуется так, чтобы можно было вычленить коэффициенты a_{11}, a_{12}, a_{22}, a_1, a_2 и a_0, формирующие выражение для уравнения второго порядка. По этим коэффициентам высчитывается инварианта D и определитель Δ, значения которых указывают на то, что мы получили *гиперболу*, а не параболу или эллипс: значение F больше нуля (1), а значение n больше единицы (2). Гиперболу как плоскую кривую второго порядка мы, кстати, уже исследовали с помощью инварианта и определителя в этюде 9, когда рассматривали пролет кометы — см. рис. 9.18 на стр. 218.

В задаче на рисунке 20.4 пакет Mathcad помог нам решить исходное уравнение линзы относительно переменной F и рассчитать инварианту D и определитель Δ. Остальное (преобразование уравнения к стандартному виду плоской кривой второго порядка) делалось вручную. Такое сочетание усилий компьютера и человека — это обычная практика работы в среде математических программ, позволяющая оптимальным образом решать задачи.

Известно, что гипербола обладает таким «отражательным» оптическим свойством: луч, исходящий из источника света, находящегося в фокусе первой ветви гиперболы, после отражения от второй ветви гиперболы направляется так, как будто он исходит из фокуса второй ветви гиперболы — см. рис. 20.5, поясняющий это. (На этом рисунке изображение гиперболы повернуто на 90 градусов и сдвинуто начало координат. Этого требует каноническая форма записи уравнения гиперболы.)

$$a := 3 \qquad b := 2 \qquad y(x) := \frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \xrightarrow{solve,\,y} \begin{bmatrix} \dfrac{2\cdot\sqrt{x-3}\cdot\sqrt{x+3}}{3} \\ -\dfrac{2\cdot\sqrt{x-3}\cdot\sqrt{x+3}}{3} \end{bmatrix}$$

$$c := \sqrt{a^2+b^2} = 3.606$$

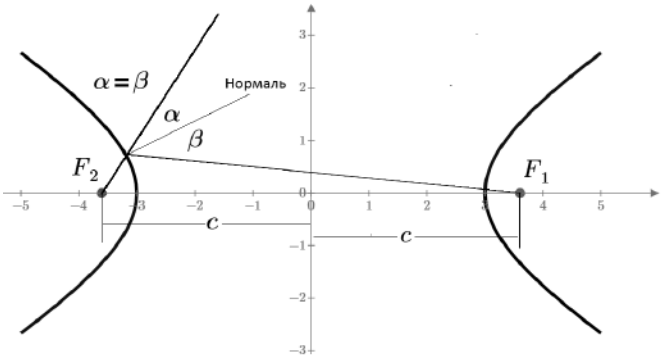


Рис. 20.5
Отражательное свойство гиперболы

На рисунке 20.5 показано построение двух ветвей гиперболы по ее каноническому уравнению, включающему два коэффициента a и b, по которым высчитывается значение переменной c, по которому определяются положения фокусов F_1 и F_2. Луч из точки F_1 (фокус правой ветви гиперболы) после отражения от левой ветви гиперболы направлен так, как будто он исходит из точки F_2 (фокус левой ветви гиперболы).

Выкладки на рис. 20.4 дают основание говорить и о «преломляющем» оптическом свойстве гиперболы: параллельный пучок света, пройдя через плоско-выпуклую линзу, сойдется в точке, если в выпуклой поверхности линзы «сидит» гипербола. Наша линза не просто плоско-выпуклая, а плоскогиперболическая.

Примечание. «Отражательное» свойство гиперболы (рис. 20.5) прописано почти во всех справочниках (бумажных и электронных), где дается информация об этой кривой второго порядка. А вот о «преломляющем» свойстве гиперболы (рис. 20.4) эти справочники молчат. Это можно объяснить тем, что закон отражения света прост и понятен — угол падения равен углу отражения (см. ниже). С преломлением же света не так все просто, и мы уже об этом писали в этюде 7 на стр. 174. Повторим это еще раз. Коэффициент преломления света n мы рассматривали как константу, не зависящую ни от длины волны луча света (явление хроматической аберрации), ни от интенсивности светового потока, ни от положения луча в преломляющей субстанции. Но тут впору вспомнить стеклянную призму, которая разлагает белый свет на цветовые составляющие и помогает, например, определять состав вещества методами спектроскопии. Это позволило, например, найти гелий сначала на Солнце, а только потом в атмосфере Земли. Нужно всегда помнить о том, что реальные линзы сводят пучок света не в точку, а в некий радужный сгусток световой энергии, с которым играют солнечными днями мальчишки, выжигая на дереве всякие фигурки. Герои романа Жюль Верна «Таинственный остров», например, добыли огонь с помощью двух стекол, снятых с часов. Эту импровизированную линзу они наполнили водой и скрепили ее края глиной. Так было сделано настоящее зажигательное стекло, которое сосредоточило лучи солнца на охапке сухого мха и воспламенило его. Почему мало упоминаются гиперболы в их связи с линзами? Дело тут в огромных несферических аберрациях (хроматическая аберрация, коматическая аберрация и др.) в таких линзах при больших диаметрах («апертурах») линз. При малых же диаметрах линз они аппроксимируются первым квадратичным членом и линзы имеют форму сферы.

В этюде 7 мы решали задачу о линзе через численное решение дифференциального уравнения — см. рис. 7.37 на стр. 174. На рисунке 20.6 эта задача дополнена формулой аналитического решения.

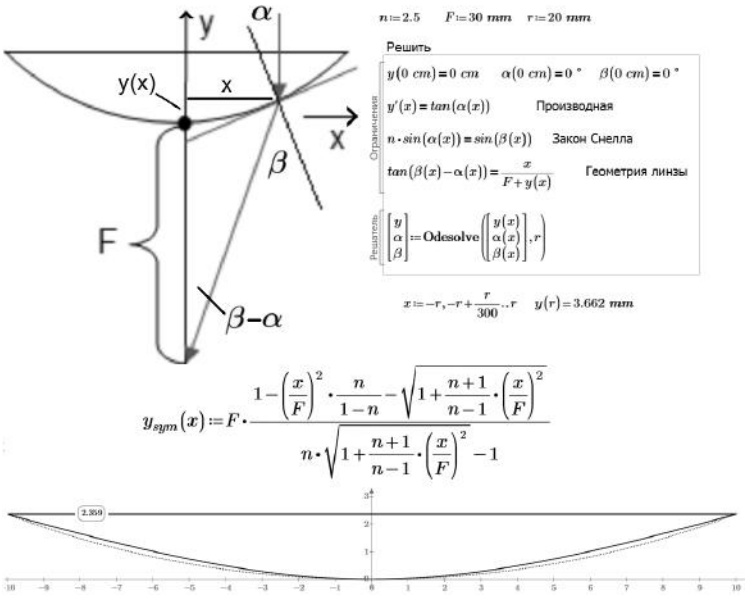


Рис. 20.6

Численное и аналитическое решение задачи о линзе

Сплошная кривая на графике рис. 20.6 — это не одна кривая, а целых три, которые слились в одну. Первая кривая y(x) отображает численное решение задачи посредством встроенной в Mathcad функции Odesolve, вторая — формулу (гиперболу), найденную решением на рис. 20.4, а третья $y(x)_{sym}$ — аналитическое (символьное) решение системы дифференциального-алгебраических уравнения, показанной на рис. 20.6. Эту «страшную» формулу для функции $y_{sym}(x)$ вывел пользователь Mathcad из Голландии — родины великого Христиана Гюйгенса, внесшего большой вклад и в развитие оптики. У этого человека была старая добрая 11-я версия Mathcad с символьным движком от Maple, а не от MuPAD, что и позволило решить задачу. Ну и, конечно, помог тут интеллект и сноровка голландца. А дело было так. Необходимо было найти асимптоты сплошной, «численной» кривой на рис. 20.6, т. е. найти конус, в который будет вписываться линза при бесконечном увеличении ее диаметра при фиксированном фокусе. Еще, кстати, не было известно, что это гипербола. Нахождение предела выражения y(x)/x оказалось невозможным, т. к. «численная» функция y(x) определена только в оговоренном диапазоне от центра линзы до ее края. Да и вообще, функция y(x) — это не «настоящая» функция, а некая псевдофункция, созданная интерполяцией табличных значений, сгенерированных численным методом решения обыкновенного дифференциального уравнения. Задача о конусе линзы была вывешена на форуме пользователей Mathcad. На запрос и откликнулся Лук Мекес, который решил задачу кардинально — нашел аналитическое решение, а у полученного выражения определил требуемые асимптоты: см. **https://www.ptcusercommunity.com/thread/130129**. Формулы на рис. 20.4 и 20.6 отличаются видом, но это одна и та же зависимость, в чем мы уже убедились, построив совпадающие графики.

Примечание. Плоскую сторону линзы мы намерено повернули вверх (см. рис. 20.4) для упрощения задачи. На сайте книги читатель найдет решение и для линзы, повернутой выпуклой стороной вверх — к параллельному пучку лучей света. Так обычно и поступают, выжигая, например, в солнечный день на дереве с помощью плоско-выпуклой линзы [82]. Но при таком положении линзы нужно будет рассматривать преломление луча света не один, а два раза — на границах «воздух — стекло» и «стекло — воздух»[1]. В случае, показанном на рисунке 20.4, преломление света происходит только на нижней поверхности.

И еще один уточняющий момент. Мы рассматриваем не объемную, а плоскую задачу. Форма линзы — это поверхность, полученная вращением найденной плоской кривой вокруг оси ординат, что часто забывают отметить.

Линза, которая должна собрать параллельный пучок света в фокусе, как правило, не имеет сферической формы, она, как говорят оптики, *асферическая* (асфера — это не-сфера). Асферические линзы обычно изготавливаются так: вращающуюся сферическую заготовку шлифуют до получения нужной асферической формы. Но тем не менее почти вся школьная и вузовская оптика, как часть учебной дисциплины «Физика», пользуется допущением, что линзы имеют сферические поверхности, а сам процесс обучения направлен не на освоение базовых понятий оптики, основанных на общих физических законах, а на вывод или заучивание готовых формул для расчета фокусных расстояний, диоптрий, коэффициентов увеличения/уменьшения, аберраций и прочих сугубо профессиональных оптических характеристик и явлений. Вернее, так. Школьная физика просто рассматривает априори такую поверхность линз, которая обеспечивает схождение параксиального пучка в точке, называемой фокусом. Задача нахождения формы этой поверхности для широкого пучка просто не ставится. И это понятно: когда развивалась наука «Геометрическая оптика» и создавались соответствующие

[1] В прошлом веке во многих семьях были линзы с четырьмя границами раздела сред «воздух — стекло (вернее, оргстекло)», «воздух — жидкость», «жидкость — стекло» и «стекло — воздух». Их ставили перед телевизорами, экраны которых в те времена были чуть больше почтовой открытки (см. **https://www.ptcusercommunity.com/thread/130193**). Наполняли же эти линзы водой или глицерином, у которого коэффициент преломления выше.

учебные курсы, не было удобных, простых и доступных средств решения уравнений — алгебраических и дифференциальных. Теперь же они есть, и это позволяет менять методику и содержание учебной дисциплины «Оптика».

Оптический принцип Ферма можно приложить и к другому важному закону — *закону отражения света*. На рисунке 20.7 показано аналитическое решение задачи о минимуме времени прохождения луча света от точки 1 к отражающей поверхности (точка 2) и далее к точке 3. Создается целевая функция t с аргументом x (горизонтальный отступ точки 2 от точки 1), у которой ищется значение аргумента, при котором производная равна нулю. Ведь если функция гладкая и непрерывная, то в точке ее минимума производная равна нулю. Выкладки на рис. 20.7 показывают, что минимум функции t(x) имеет место тогда, когда тангенс угла падения α равен тангенсу угла отражения β. Следовательно, и угол α равен углу β. Кстати, отражение и преломление «идут рука об руку»: свет, падая, например, на стеклянную поверхность частично отражается, а частично уходит с преломлением вглубь стекла, и соотношение этих частей зависит от угла падения.

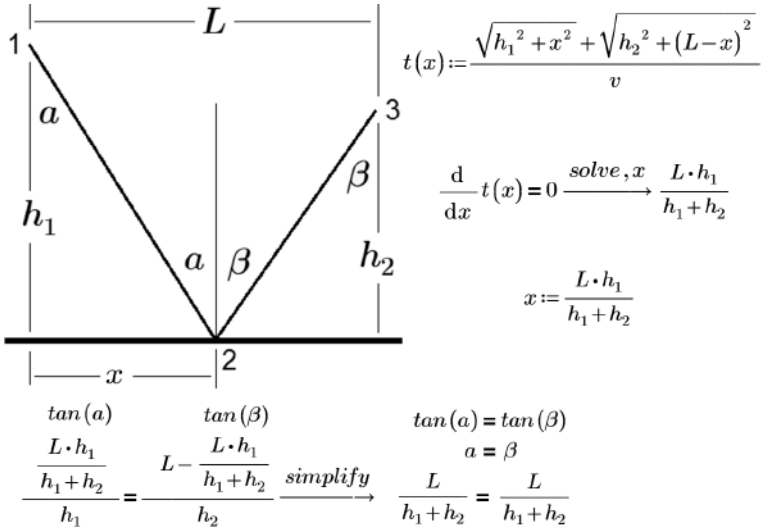


Рис. 20.7

Доказательство закона отражения света из принципа Ферма

Один из изобретателей лазера, нобелевский лауреат Чарльз Таунс признался, что на создание этого устройства его вдохновил английский перевод романа Алексея Толстого «Гиперболоид инженера Гарина», о котором мы уже упоминали на стр. 171. Название этого перевода «The Garin Death Ray». В Википедии можно прочесть, что «более правильное название устройства Гарина — параболоид. Толстой знал об этом, однако выбрал слово «гиперболоид» из-за более внушительного звучания». Но в этом устройстве могли быть не только зеркала, но и линзы — плоскогиперболические, например. Тогда название романа и устройства вполне оправдано. Подробно об этом можно прочесть здесь **http://www.numbernautics.ru/artefact-real/838-giperboloid-inzhenera-garina-byl-i-mify.html**.

Гиперболоид, вернее гиперболу, обрисовывающую нижний край линзы, мы уже смоделировали. Давайте смоделируем и параболоид — параболическое вогнутое зеркало (см. рис. 20.8), используя уже упомянутый принцип таутохронизма.

На рисунке 20.8 фактически повторены выкладки, показанные на рис. 20.4, но не для фокусирующей плоско-выпуклой линзы, а для фокусирующего вогнутого зеркала. Выкладки на рис. 20.8 выдали не гиперболу, а параболу, у которой

есть и другое неоптическое геометрическое толкование. Если на плоскости нарисовать прямую и поставить возле нее точку, то парабола — это геометрическое место точек, равноудаленных от заданной прямой (ее называют *директрисой* параболы) и заданной точкой (ее называют фокусом *параболы*). Гиперболу, параболу и не рассмотренный в этом дивертисменте эллипс называют *коническими кривыми*, так как они получаются при рассечении прямого кругового конуса плоскостью. По эллиптическим, параболическим и гиперболическим орбитам, кстати, движутся небесные тела (см. этюд 9), за которыми мы следим с помощью зеркальных и линзовых телескопов. Преобразованию этих кривых к каноническому виду, у которого не шесть (см. выше рис. 20.4 и 20.8), а всего лишь два (гипербола и эллипс) или даже один (парабола, окружность как частный случай эллипса) коэффициент, уделяется много внимания в вузовской математике.

$$F+h=\sqrt{(F-y)^2+x^2}+h-y \xrightarrow{solve,\,y} \frac{x^2}{4\cdot F}$$

$$y-\frac{1}{4\,F}\cdot x^2=0 \qquad F:=1$$

$$a_{11}:=-\frac{1}{4\,F} \quad a_{12}:=0 \quad a_{22}:=0 \quad a_1:=0 \quad a_2:=\frac{1}{2} \quad a_0:=0$$

$$D:=\left|\begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix}\right|=0 \quad D=0 \quad \Delta:=\left|\begin{bmatrix} a_{11} & a_{12} & a_1 \\ a_{12} & a_{22} & a_2 \\ a_1 & a_2 & a_0 \end{bmatrix}\right|=0.063 \quad \Delta\neq 0$$

$$f(x,y):=a_{11}\cdot x^2+a_{12}\cdot x\cdot y+a_{22}\cdot y^2+2\,a_1\cdot x+2\,a_2\cdot y+a_0$$

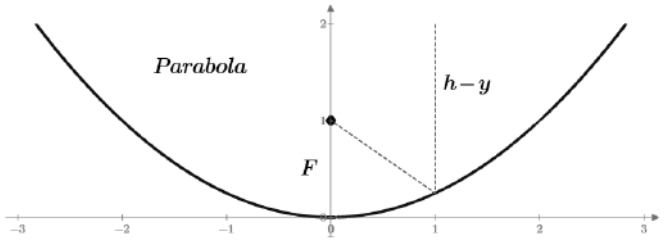


Рис. 20.8
Вывод уравнения параболического зеркала

Примечание. Параболу, вернее параболоид (поверхность, полученную вращением параболы вокруг оси, перпендикулярной директрисе и проходящей через фокус), можно увидеть не только на стенах домов, где смотрят спутниковое TV, но и... зимой на ледяных горках. Дело в том, что когда-то давно одному заводу, выпускающему такие антенны для нужд военных, поручили изготавливать и конверсионную продукцию. Руководство завода недолго думая решило прикрепить к этой тарелке две кожаные петли. Получилась отличная ледянка. А вот еще один возможный товар народного потребления с «параболой внутри», идея еще одного «артобъекта»: стеклянная или фарфоровая ваза для фруктов в виде параболоида, с круглой подставкой-директрисой и ножкой, пронизывающей в центре параболоид и заканчивающейся в фокусе ручкой. Такая ваза будет хорошо смотреться в доме математика — будет о чем поговорить с гостями за столом. Ее можно расписать не традиционными узорами, а математическими выкладками, связанными с параболой.

Форму фокусирующего зеркала, как и форму линзы, можно также найти через решение не алгебраического, а дифференциального уравнения — см. рис. 7.34–7.36 на стр. 172.

Оптика присутствует не только в учебниках и задачники по физике, но и в художественной литературе. Вспомним не только Жюль Верна (см. выше), но пуш-

кинское «Всё хлопает. Онегин входит, / Идет меж кресел по ногам, / Двойной лорнет скосясь наводит / На ложи незнакомых дам». Или рассказ Василия Шукшина «Микроскоп», а также басню Крылова «Мартышка и очки». Да что там далеко ходить! Глаза, через которые человек получает основную долю информации об окружающем мире (читает, например, эту книгу), это ни что иное, как совершеннейшие оптические приборы, которые мы часто подправляем и усиливаем рукотворными оптическими устройствами: моноклем, лорнетом, пенсне, очками, подзорной трубой, биноклем, перископом, микроскопом, телескопом и т. д. Сочетая математику, физику и литературу (базовые школьные предметы) с современными информационными технологиями, можно успешно и, главное, с интересом решать довольно сложные оптические задачи, попутно изучая законы физики.

Современные компьютерные средства позволяют отказаться от многих допущений и упрощений и более точно рассчитывать оптические приборы. Это можно делать не только с помощью специализированных программ для расчета оптических систем (TracePro, OPTIS, LightTools и др.), но и в среде универсальной математической программы Mathcad, а также с помощью сайтов Интернета и специалистов, работающих на профессиональных форумах. А начинать изучение оптики можно и нужно не с заучивания готовых формул, зачастую малопонятных из-за принятых в них допущений и упрощений, о которых умалчивают, а с генерирования на компьютере базовых положений оптики, переходя затем к упрощенным формулам. Что мы и попытались сделать в этом дивертисменте.

Авторы благодарны преподавателю физики МЭИ С. Д. Федорович и преподавателю физики лицея № 1502 при МЭИ А. В. Соколову за помощь при написании данного этюда.

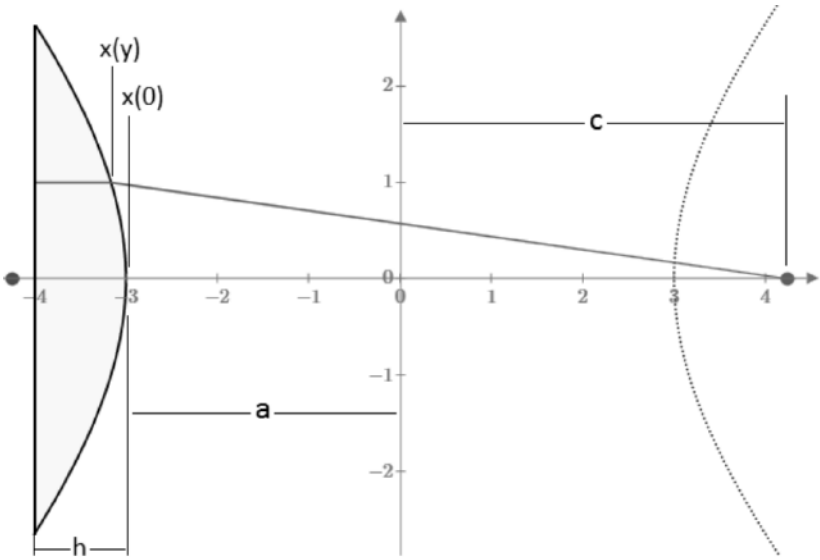
ДИВЕРТИСМЕНТ 1. ЛУПА ДЛЯ ЧТЕНИЯ ФИЗИКО-МАТЕМАТИЧЕСКИХ СПРАВОЧНИКОВ

В данном этюде мы показали, что плосковыпуклая линза должна быть плоско-гиперболической, чтобы она собирала параллельный пучок света в оптическую точку-фокус — см. рис. 20.4. Но у гиперболы есть пара собственных геометрических фокусов, и мы это обсуждали в этюде 19 — см. рис. 19.4 на стр. 380. Как соотносятся между собой эти два фокуса гиперболы — оптический линзы и один из геометрических? У параболы-зеркала эти фокусы совпадают — см. рис. 20.8 на стр. 407. А могут ли они совпасть в гиперболе-линзе?

Давайте решим такую задачу. Гипербола задана коэффициентами a и b, входящими в каноническое уравнение $x^2 / a^2 - y^2 / b^2 = 1$. Левая ветвь гиперболы (см. рис. 20.9 — мы линзу повернули на 90 градусов) образует выпуклую поверхность линзы толщиной h. Спрашивается, какой коэффициент преломления должен быть у материала такой линзы, чтобы оптический фокус совпал с геометрическим фокусом гиперболы?

На рисунке 20.9 показана попытка решения этой новой задачи с опорой на тот же принцип таутохронизма. Уравнение времени на прохождение светом двух оптических путей взято из рис. 20.4, но сама линза, повторяем, повернута на 90 градусов, чтобы можно было работать с канонической формой записи гиперболы: $x^2 / a^2 - y^2 / b^2 = 1$. В уравнение на рис. 20.9 вставлено фокусное расстояние, равное сумме двух важных параметров гиперболы $a + c$. Уравнение после соответствующих замен переменных решается относительно переменной n — коэффициента преломления материала линзы. Получена формула, в которую входит переменная y, которая в принципе не должна влиять на значение n. Численное исследование функции $n(a, b)$ показывает, что коэффициент преломления материала нашей линзы не зависит от значения y, а у линзы, образованной равнобочной гиперболой, где $a = b$, этот коэффициент равен **квадратному корню из двух**. При расчетах по формуле для $n(a, b)$ значение y можно принимать за единицу или за единичный метр, если учитываются единицы измерения в формуле. Но дальней-

ший анализ формулы на рис. 20.9 показал, что она совпадает численно с формулой для еще одной важной характеристики гиперболы — *эксцентриситет* `ε = c / a`. На рисунке 20.10 показаны две совпадающие поверхности, построенные по функциям `n(a, b)` и `ε(a, b)`.



$$\frac{x^2}{a^2}-\frac{y^2}{b^2}=1 \xrightarrow{solve,x} \begin{bmatrix} \frac{a\cdot\sqrt{b^2+y^2}}{b} \\ -\frac{a\cdot\sqrt{b^2+y^2}}{b} \end{bmatrix} \qquad x(y)=-\frac{a\cdot\sqrt{b^2+y^2}}{b} \qquad x(0)=-a \qquad c=\sqrt{a^2+b^2}$$

$$\frac{h}{c_1}+\frac{a+c}{c_2}=\frac{h-\left(-a+\frac{a\cdot\sqrt{b^2+y^2}}{b}\right)}{c_1}+\frac{\sqrt{y^2+\left(a+c+\left(-a+\frac{a\cdot\sqrt{b^2+y^2}}{b}\right)\right)^2}}{c_2} \xrightarrow[\substack{substitute,c_2=n\cdot c_1 \\ substitute,c=\sqrt{a^2+b^2} \\ solve,n \\ simplify,\max}]{} \cdots$$

$$\cdots \rightarrow \frac{b\cdot\left(a+\sqrt{a^2+b^2}-\sqrt{2\cdot a^2+b^2+y^2+\frac{a^2}{b^2}\cdot y^2+\frac{2\cdot a}{b}\cdot\sqrt{a^2+b^2}\cdot\sqrt{b^2+y^2}}\right)}{a\cdot\left(b-\sqrt{b^2+y^2}\right)}$$

$$y:=\begin{bmatrix}1\\2\\3\end{bmatrix}$$

$$n(a,b):=\frac{b\cdot\left(a+\sqrt{a^2+b^2}-\sqrt{2\cdot a^2+b^2+y^2+\frac{a^2}{b^2}\cdot y^2+\frac{2\cdot a}{b}\cdot\sqrt{a^2+b^2}\cdot\sqrt{b^2+y^2}}\right)}{a\cdot\left(b-\sqrt{b^2+y^2}\right)}$$

$$n(4,4)=\begin{bmatrix}1.414\\1.414\\1.414\end{bmatrix} \qquad n(2,3)=\begin{bmatrix}1.803\\1.803\\1.803\end{bmatrix} \qquad n(3,2)=\begin{bmatrix}1.202\\1.202\\1.202\end{bmatrix}$$

Рис. 20.9

Коэффициент преломления линзы с совпадающими оптическим и геометрическим фокусами

Исходя из расчетов на рис. 20.9 и 20.10, можно еще раз и точнее сформулировать *второе оптическое свойство гиперболы*. **Параллельный пучок света, пройдя через плоско-выпуклую линзу, сойдется в одной точке, если выпуклую поверхность линзы определяет одна из ветвей гиперболы. Если при этом коэффициент преломления материала линзы численно совпадает с эксцентриситетом гипербо-**

лы, то *оптический* фокус такой плоскогиперболической линзы совпадет с одним из *геометрических* фокусов гиперболы.

$$n(a,b) := \left\| \begin{array}{l} y \leftarrow 1 \\ \dfrac{b\cdot\left(a+\sqrt{a^2+b^2}-\sqrt{2\cdot a^2+b^2+y^2+\dfrac{a^2}{b^2}\cdot y^2+\dfrac{2\cdot a}{b}\cdot\sqrt{a^2+b^2}\cdot\sqrt{b^2+y^2}}\right)}{a\cdot\left(b-\sqrt{b^2+y^2}\right)} \end{array} \right.$$

$$\varepsilon(a,b) := \sqrt{1+\frac{b^2}{a^2}}$$

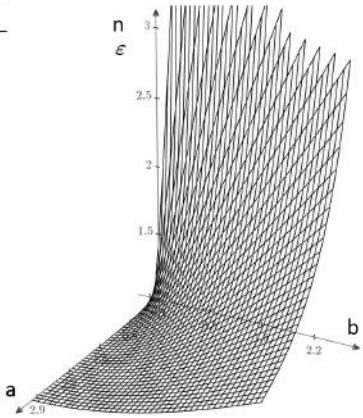


Рис. 20.10
Коэффициент преломления и эксцентриситет гиперболы

Рисунок 20.11 поясняет это новое «преломляющее» свойство гиперболы.

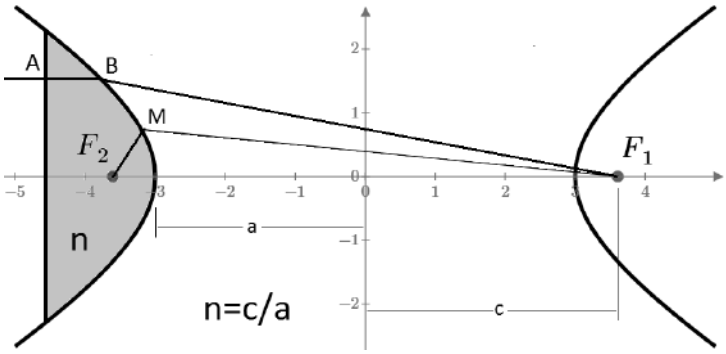


Рис. 20.11
Второе оптическое свойство гиперболы

На сайте этюда читатель найдет строгое доказательство второго оптического свойства гиперболы. Оно получено так. Гипотеза была размещена в Интернете, а один аноним их Харькова доказал ее правильность. В этюде 4 на стр. 103 и 104 мы писали о том, что немного опоздали с открытием новой математической константы. Надеемся, что в случае с новым оптическим свойством гиперболы мы стали пионерами.

Одно из определений гиперболы такое. Гипербола — это геометрическое место точек M евклидовой плоскости, для которых абсолютное значение разности расстояний от M до двух выделенных точек F_1 и F_2 (называемых фокусами гиперболы) постоянно. Мы в этюде 19 строили гиперболу по такому ее *фокальному* определению — см. рис. 19.4 на стр. 380. Из рисунка 20.11 можно понять, что разность

отрезков F_2–M и F_1–M формирует первую (левую) ветвь, а разность отрезков F_1–M и F_2–M — вторую (правую) ветвь гиперболы. Если левая ветвь гиперболы формирует одну из поверхностей плосковыпуклой линзы, материал которой имеет коэффициент преломления `n`, равный эксцентриситету гиперболы `c / a`, то параллельный пучок света (один его луч проходит через точки A–B) сойдется в точку F_1, совпадающую с фокусом правой ветви гиперболы.

Эксцентриситет (отношение величины `c` к величине `a`) — это важный и определяющий параметр плоских кривых второго порядка. Если он больше единицы, то мы имеем гиперболу (см. рис. 20.11, например, больше единицы, кстати, и коэффициент преломления света[2]), равен единице — параболу, а меньше единицы — эллипс. На рисунке 19.22 на стр. 395 было показано построение этих трех кривых методом сканирования, описанным в этюде 19. При этом задается один фокус и директриса (директориальное, а не фокальное определение кривых).

От «лишней» переменной `y` в функции `n(a, b)` на рис. 20.10 можно избавится разными путями. Самый простой, и мы уже его отметили, — присвоить переменной `y` единичное значение, вернее значение единичного метра, если задача будет решаться с использованием единиц измерения. Но... Тут впору вспомнить начало этюда 9 с пассажем на тему «Что бы было, если б у Ньютона был компьютер». Один из ответов на этот риторический вопрос таков. Если бы у Ньютона был компьютер, то... никакого дифференциального исчисления не было бы. А было бы нагромождение решенных и якобы решенных задач, множество таблиц и кривых на дисплеях компьютеров, из которых какие-либо выводы, а тем более обобщающие законы, получить было бы чрезвычайно трудно — «голь (люди без компьютера) на выдумки хитра». Если б у нас не было компьютера, то мы бы смогли как-то иначе и изящнее доказать, что в линзе запрятан эксцентриситет ее гиперболы. Компьютер же выдал «страшные» формулы, в которых трудно узреть какой-то физический смысл.

Мы в этой книге уже не раз предлагали идеи предметов с запрятанным математическим смыслом. Вот еще одна такая идея. Можно изготовить «равнобочную» плоскогиперболическую линзу из оптического стекла с составом, обеспечивающим коэффициент преломления, равный корню из двух (вариант — использовать неравнобочную гиперболу с эксцентриситетом, равным коэффициенту преломления стекла или пластика линзы). Эту линзу вставить в оправу, на футляре которой выгравированы вышеприведенные выкладки и пояснения к ним. Получится лупа для математика, который может ею пользоваться при чтении физических и математических справочников с мелким шрифтом.

ДИВЕРТИСМЕНТ 2. ТЕОРЕМА ФЕРМА И MATHCAD

Французский математик Пьер де Ферма (1601–1665) известен в первую очередь не вышеописанным принципом геометрической оптики, а своей *Великой теоремой*. На рисунке 20.12 она записана не традиционными словами и формулами, а... программой Mathcad.

В программе на рисунке 20.12 четыре вложенных цикла. Мы уже использовали этот прием, когда в этюде 18 решали буквенные головоломки (см. рис. 18.2 на стр. 372 и рис. 18.4 на стр. 373). Теорема Ферма гласит, что для любого натурального числа `n(n > 2)` уравнение $a^n + b^n = c^n$ не имеет решений в целых ненулевых числах `a`, `b` и `c`. Это и выдала программа на рис. 20.13 — шапка таблицы-ответа есть, а чисел под шапкой нет!

Но, как понимает читатель, программу на рис. 20.12 исполнить невозможно теоретически и практически. Но если числу `N` присвоить не бесконечность (для

[2] Физики обсуждают вопрос, есть ли материал с коэффициентом преломления меньше единицы или по крайней мере равным единице. Как будет вести себя лупа, сделанная из такого материала?

численной математики Mathcad она равна 10^{307}), а какое-либо конечное число, то пакет Mathcad… опровергнет теорему Ферма — см. рис. 20.13.

```
N := ∞

Ans ← ["a" "b" "c" "n"]                                  = ["a" "b" "c" "n"]
for a ∈ −N..N
  for b ∈ −N..N
    for c ∈ −N..N
      for n ∈ 3..N
        if a≠0∧b≠0∧c≠0∧a^n + b^n = c^n
          Ans ← stack(Ans, [a b c n])
Ans
```

Рис. 20.12
Великая теорема Ферма

```
N := 20

Ans ← ["a" "b" "c" "n"]                                  =  0      ⋮
for a ∈ −N..N                                              ⋮      ⋮
  for b ∈ −N..N                                         1349  20  2   20  19
    for c ∈ −N..N                                       1350  20  2   20  20
      for n ∈ 3..N                                      1351  20  3  −20  20
        if a≠0∧b≠0∧c≠0∧a^n + b^n = c^n                  1352  20  3   20  20
          Ans ← stack(Ans, [a b c n])
Ans
```

Рис. 20.13
«Опровержение» Великой теоремы Ферма

Ответы, показанные на рис. 10.13, тривиальные — сразу видно, что они неверны, и это связано с точностью вычислений в среде Mathcad. Но в Интернете можно найти и не такие явные «ляпы». Один из них приводится в Википедии: $1782^{12} + 1841^{12} = 1922^{12}$.

Калькулятор с точностью не более 9 значащих цифр подтверждает это равенство:

$$1782^{12} + 1841^{12} = 254\,121\,026 \cdot 10^{31};$$

$$1922^{12} = 254\,121\,026 \cdot 10^{31}.$$

Но если такую проверку провести в среде Mathcad с более высокой точностью, то Ферма будет реабилитирован — см. рис. 20.14.

```
1782^12 + 1841^12 =  254121025861459000000000000000000000000000

1922^12 =  254121025931480000000000000000000000000000

1782^12 + 1841^12 → 254121025861458917628866995814242852665 7

1922^12 → 254121025931480141081927864964365156761 6
```

Рис. 20.14
Реабилитация Ферма

Из рисунка 20.13 видно, что равенство $1782^{12} + 1841^{12} = 1922^{12}$ неверное. Равенство «верно» при точности до 13 знаков численной математики Mathcad (оператор =). Символьная же математика (оператор →) дает тут абсолютную точность. «Слабость» численной математики наглядна видна на рисунке 20.14. Но, как говорят про женщин: «Их сила в их слабости». Тоже можно сказать и о численной математике, когда нужна не только точность, но и быстрота вычислений.

Теорема Ферма тут упомянута неслучайно. Эту теорему Ферма поместил без доказательства на полях одной книги. Мы тоже на «полях» данной книге помещаем описание преломляющего свойства гиперболы без полного доказательства.

ДИВЕРТИСМЕНТ 3. МАТРИЧНАЯ ОПТИКА

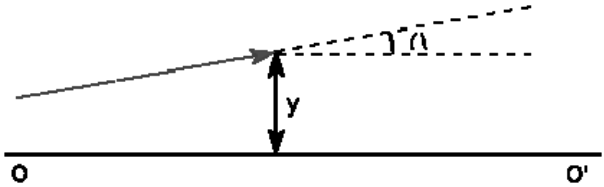
Реальная электромагнитная волна, каковой и является свет, крайне непроста и капризна в своем описании, и на начальном этапе расчета оптических задач приходится отказаться от многих особенностей световых волн. Первым приближением в описании световых процессов является геометрическая оптика. В геометрической оптике распространение световой волны моделируется при помощи лучей — линий, вдоль которых происходит перемещение световых волн. При этом не учитываются волновые явления света (дифракция и интерференция). Далее, при описании прохождения лучей через оптическую систему, как правило, учитываются лишь меридианные лучи, т. е. лучи, которые мы изображаем в плоскости рисунка или чертежа оптической системы. Также не учитываются и нелинейные эффекты, связанные, например, с зависимостью показателя преломления вещества от интенсивности излучения. Несмотря на все указанные ограничения, расчеты все равно остаются очень сложными, и связано это в первую очередь с так называемым законом Снеллиуса (или Снелля), это закон преломления света:

$$n_1 \cdot \sin(\alpha) = n_2 \cdot \sin(\beta).$$

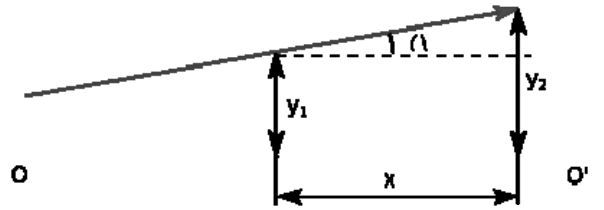
«Неприятная» расчетная особенность этого закона состоит в том, что он нелинейным образом (через синус угла) связывает углы падения и преломления света. Но расчеты существенно упрощаются в случае, когда углы падения и преломления света малы. В этом случае можно использовать приближенные соотношения: $\sin\alpha \approx \mathrm{tg}\alpha \approx \alpha$, если угол измерен в радианах. Такое часто использующееся приближение называется приближением параксиальной оптики или просто параксиальным приближением. Для удобства восприятия при графическом построении изображений в этом случае принято не соблюдать реальных значений углов, а изображать их преувеличенно большими.

Для расчета линейных оптических схем в параксиальном приближении удобно использовать методы матричной оптики, особенно в тех случаях, когда расчеты можно выполнить при помощи математических программ. Подробное описание методов матричной оптики можно найти в книге А. Джеррарда, Д. М. Бёрча «Введение в матричную оптику».

Суть метода матричной оптики состоит в описании светового луча заданием двух его параметров: высоты над оптической осью «y» и приведенного угла «v», равного произведению показателя преломления среды на угол между лучом и оптической осью $v = n \cdot \alpha$. Введенный таким образом приведенный угол обладает интересной и важной особенностью — он не меняется при преломлении на плоской поверхности, перпендикулярной оптической оси. Действительно, по закону Снеллиуса, $n_1\sin(\alpha) = n_2\sin(\beta)$. В приближении малых углов $\sin(\alpha) = \alpha$. Поэтому $n_1\alpha = n_2\beta$. Т. е. $v_1 = v_2$.



Выведем матрицу распространения луча в свободном пространстве.



$$y_2 = y_1 + x \cdot \mathrm{tg}(\alpha) \approx y_1 + x \cdot \alpha = y_1 + \frac{x \cdot n \cdot \alpha}{n} = y_1 + \frac{x}{n} \cdot v_1.$$

При преобразовании учтено, что в параксиальном приближении

$$\mathrm{tg}(\alpha) \approx \alpha.$$

Так как угол прохождения луча в свободном пространстве не меняется, то

$$v_2 = v_1.$$

Система уравнений

$$\begin{cases} y_2 = y_1 + \dfrac{x}{n} \cdot v_1; \\ v_2 = v_1 \end{cases}$$

эквивалентна записи в матричной форме:

$$\begin{pmatrix} y_2 \\ v_2 \end{pmatrix} = \begin{pmatrix} 1 & \dfrac{x}{n} \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ v_1 \end{pmatrix}.$$

Поскольку

$$y_2 = 1 \cdot y_1 + \frac{x}{n} \cdot v_1;$$
$$v_2 = 0 \cdot y_1 + 1 \cdot v_1,$$

в этом случае распространение луча в свободном пространстве описывается матрицей

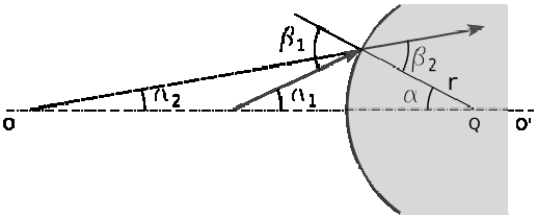
$$\begin{pmatrix} 1 & \dfrac{x}{n} \\ 0 & 1 \end{pmatrix}. \qquad (1)$$

Рассмотрим прохождения луча через сферическую преломляющую поверхность.

Поверхность имеет радиус r, показатель преломления среды слева от сферической поверхности равен n_1, справа n_2.

Точка преломления луча не меняет свои координаты, поэтому

$$y_2 = y_1.$$



При преломлении выполняется закон Снеллиуса в параксиальном приближении

$$n_1 \cdot \sin(\beta_1) = n_2 \cdot \sin(\beta_2);$$
$$n_1 \cdot \beta_1 \approx n_2 \cdot \beta_2.$$

По теореме о внешнем угле треугольника

$$\beta_1 = \alpha_1 + \alpha = \alpha_1 + \frac{y_1}{r};$$

$$\beta_2 = \alpha_2 + \alpha = \alpha_2 + \frac{y_1}{r}.$$

Поэтому

$$n_1 \cdot \left(\alpha_1 + \frac{y_1}{r}\right) = n_2 \cdot \left(\alpha_2 + \frac{y_1}{r}\right)$$

или

$$v_1 + n_1 \cdot \frac{y_1}{r} = v_2 + n_2 \cdot \frac{y_1}{r};$$

$$v_2 = v_1 + n_1 \cdot \frac{y_1}{r} - n_2 \cdot \frac{y_1}{r} = v_1 - (n_2 - n_1) \cdot \frac{y_1}{r}.$$

То есть верно соотношение

$$\begin{pmatrix} y_2 \\ v_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -\dfrac{(n_2 - n_1)}{r} & 1 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ v_1 \end{pmatrix}.$$

В этом случае прохождение луча через преломляющую сферическую поверхность описывается матрицей

$$\begin{pmatrix} 1 & 0 \\ -\dfrac{(n_2 - n_1)}{r} & 1 \end{pmatrix}. \qquad (2)$$

Пример 1. Рассмотрим прохождение луча через тонкую двояковыпуклую линзу с радиусами поверхностей r_1 и r_2, показатель преломления вещества линзы равен n. В этом случае преобразования луча линзой опишется произведением двух матриц:

$$\begin{pmatrix} 1 & 0 \\ -\dfrac{(1-n)}{r_2} & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ -\dfrac{(n-1)}{r_1} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -(n-1) \cdot \left(\dfrac{1}{r_1} - \dfrac{1}{r_2}\right) & 1 \end{pmatrix}.$$

Так как для фокусного расстояния тонкой линзы верно соотношение

$$\frac{1}{F} = (n-1) \cdot \left(\frac{1}{r_1} - \frac{1}{r_2}\right),$$

то можно сделать вывод, что прохождение луча через линзу с фокусным расстоянием F описывается матрицей

$$\begin{pmatrix} 1 & 0 \\ -\dfrac{1}{F} & 1 \end{pmatrix}. \qquad (3)$$

Матрицы (1), (2) и (3) являются основными расчетными матрицами для всех приведенных ниже примеров.

При прохождении луча последовательно через несколько оптических элементов описывается результирующей матрицей R, равной произведению соответствующих матриц оптических элементов:

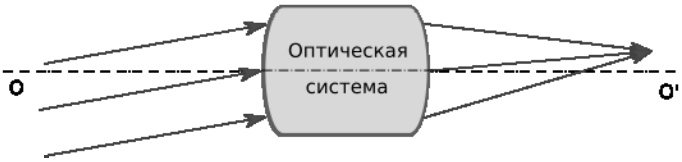
$$R = \ldots \cdot \begin{pmatrix} A'' & B'' \\ C'' & D'' \end{pmatrix} \cdot \begin{pmatrix} A' & B' \\ C' & D' \end{pmatrix} \cdot \begin{pmatrix} A & B \\ C & D \end{pmatrix}.$$

Анализ матрицы преобразования лучей вида:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

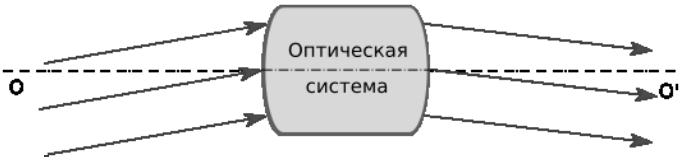
показывает, что

1) в случае $A = 0$ матрица соответствует фокусировке параллельного пучка лучей в фокальной плоскости



2) в случае $B = 0$ матрица соответствует фокусировке лучей, выходящих из точки перед оптической системой в точке после оптической системы

3) в случае $C = 0$ матрица соответствует преобразованию параллельного пучка лучей в другой параллельный пучок

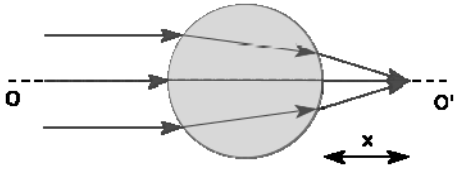


4) в случае $D = 0$ матрица соответствует преобразованию лучей, выходящих из точки в фокальной плоскости перед оптической системой в параллельный пучок лучей.

Рассмотрим применение матричных методов при решении следующих задач.

Пример 2. Найти положение фокуса стеклянного шарика радиуса r с показателем преломления n.

Координату расположения фокуса обозначим x.



Для расчета составляем матрицы преломления первой сферической поверхностью

$$\begin{pmatrix} 1 & 0 \\ -\dfrac{(n-1)}{r} & 1 \end{pmatrix}.$$

Перемещения луча внутри шарика

$$\begin{pmatrix} 1 & \dfrac{2r}{n} \\ 0 & 1 \end{pmatrix}.$$

Преломления второй сферической поверхностью

$$\begin{pmatrix} 1 & 0 \\ -\dfrac{(1-n)}{-r} & 1 \end{pmatrix}.$$

Ход луча после преломления

$$\begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix}.$$

Результирующая матрица преобразований:

$$R = \begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ -\dfrac{(1-n)}{-r} & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & \dfrac{2r}{n} \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ -\dfrac{(n-1)}{r} & 1 \end{pmatrix}.$$

Выполняем преобразования в Mathcad

$$\begin{bmatrix} 1 & \dfrac{x}{1} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ -\dfrac{(1-n)}{-r} & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ -\dfrac{(n-1)}{r} & 1 \end{bmatrix} \rightarrow$$

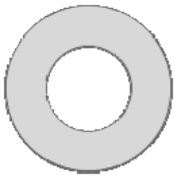
$$\rightarrow \begin{bmatrix} 1 - \dfrac{(n-1)\cdot\left(x - \dfrac{2\cdot r\cdot\left(\dfrac{x\cdot(n-1)}{r} - 1\right)}{n}\right)}{r} - \dfrac{x\cdot(n-1)}{r} & x - \dfrac{2\cdot r\cdot\left(\dfrac{x\cdot(n-1)}{r} - 1\right)}{n} \\ \dfrac{\left(\dfrac{2\cdot(n-1)}{n} - 1\right)\cdot(n-1)}{r} - \dfrac{n-1}{r} & 1 - \dfrac{2\cdot(n-1)}{n} \end{bmatrix}.$$

Из условия фокусировки лучей следует, что элемент A итоговой матрицы равен 0

$$1-\frac{(n-1)\cdot\left(x-\frac{2\cdot r\cdot\left(\frac{x\cdot(n-1)}{r}-1\right)}{n}\right)}{r}-\frac{x\cdot(n-1)}{r}\xrightarrow{\text{solve},x}\frac{2\cdot r-n\cdot r}{2\cdot n-2}.$$

Фокус расположен на расстоянии $x=\frac{r(2-n)}{2(n-1)}$ от поверхности шарика. Интересно, что при $n = 2$ фокусировка происходит на внутренней поверхности шарика. На этом основано действие сферического катафота.

Пример 3. Найти положение фокуса толстостенной стеклянной сферы с внутренним радиусом $r_1 = 3$ см, внешним радиусом $r_2 = 6$ см и показателем преломления $n = 1.5$.



Координату расположения фокуса обозначим x.

Для расчета составляем матрицы преломления первой сферической поверхностью

$$\begin{pmatrix} 1 & 0 \\ -\frac{(n-1)}{r_2} & 1 \end{pmatrix}.$$

Перемещения луча внутри сферического слоя

$$\begin{pmatrix} 1 & \frac{r_2-r_1}{n} \\ 0 & 1 \end{pmatrix}.$$

Преломления второй сферической поверхностью

$$\begin{pmatrix} 1 & 0 \\ -\frac{(1-n)}{r_1} & 1 \end{pmatrix}.$$

Перемещения луча внутри воздушной полости

$$\begin{pmatrix} 1 & \frac{2r_1}{1} \\ 0 & 1 \end{pmatrix}.$$

Следующее преломление

$$\begin{pmatrix} 1 & 0 \\ -\frac{(n-1)}{-r_1} & 1 \end{pmatrix}.$$

Перемещения луча внутри сферического слоя

$$\begin{pmatrix} 1 & \frac{r_2-r_1}{n} \\ 0 & 1 \end{pmatrix}.$$

Следующее преломление

$$\begin{pmatrix} 1 & 0 \\ -\dfrac{(1-n)}{-r_2} & 1 \end{pmatrix}.$$

Ход луча после преломления

$$\begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix}.$$

Результирующая матрица преобразований:

$$r = \begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ -\dfrac{(1-n)}{-r_2} & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & \dfrac{r_2 - r_1}{n} \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ -\dfrac{(n-1)}{-r_1} & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & \dfrac{2r_1}{1} \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ -\dfrac{(1-n)}{r_1} & 1 \end{pmatrix} \times$$

$$\times \begin{pmatrix} 1 & \dfrac{r_2 - r_1}{n} \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ -\dfrac{(n-1)}{r_2} & 1 \end{pmatrix}.$$

Выполняем преобразования в Mathcad. Для упрощения выполним преобразования численно:

```
r1:= 3 cm      r2:= 6 cm      n:= 1.5 cm
```

$$\begin{bmatrix} 1 & \dfrac{x}{1} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ -\dfrac{(1-n)}{-r_2} & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & \dfrac{r_2 - r1}{n} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ -\dfrac{(n-1)}{-r_1} & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & \dfrac{2 \cdot r_1}{1} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ -\dfrac{(1-n)}{r_1} & 1 \end{bmatrix} \times$$

$$\times \begin{bmatrix} 1 & \dfrac{r_2 - r_1}{n} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ -\dfrac{(n-1)}{r_2} & 1 \end{bmatrix} \xrightarrow{\text{simplif } y}$$

$$\xrightarrow{\text{simplif } y} \begin{bmatrix} \dfrac{0.1111111111111111111 \cdot x}{cm} + 1.6666666666666666667 & 1.6666666666666666667 \cdot x + 16.0 \cdot cm \\ \dfrac{0.1111111111111111111}{cm} & 1.666666666666666667 \end{bmatrix}.$$

Из условия фокусировки следует, что элемент A матрицы преобразований равен 0

$$\frac{0.1111111111111111111 \cdot x}{cm} + 1.6666666666666666667 \xrightarrow{\text{solve}} -15.0 \cdot cm.$$

Результат: толстослойная сферическая оболочка действует как рассеивающая линза, фокус которой расположен на расстоянии 15 см от последней сферической границы (или на расстоянии 3 см перед сферой).

Дивертисмент 3 написан А. В. Соколовым — учителем физики Московского лицея № 1502 при МЭИ.

Задание читателям

1. Найти прозрачный материал (природный или искусственный) с коэффициентом преломления наиболее близким к корню из двух. Или корню из трех. Есть

ли какой-то физико-математический смысл коэффициента преломления, равного корню не из двух, а из трех?

2. Определите параметры плоско-выпуклой линзы, в которую заложена равнобочная гипербола, через решение не алгебраического, а дифференциального уравнения.

3. Нарисовать в среде Mathcad конус, рассечь его плоскостью, двигать в анимации эту плоскость и получить гиперболу, параболу и эллипс.

4. Предложить новое равенство типа $1782^{12} + 1841^{12} = 1922^{12}$, которое можно назвать ложным только с помощью символьной, а не численной математики Mathcad (см. рис. 20.15).

5. «Поиграть» на компьютере с равенством $a^n + b^n + c^n = d^n$.

Этюд 21

ОПТИМИЗАЦИЯ ПОТЕНЦИАЛЬНОЙ ЭНЕРГИИ, ИЛИ ЕЩЕ РАЗ ПРО ЦЕПЬ

В этюде рассмотрено решение задачи о цепной линии на основе анализа потенциальной энергии механической системы в среде Mathcad. Предложены две лабораторные работы на стыке физики, математики (геометрии) и информатики.

Математика: цепная функция, оптимизация с ограничениями, декартов график, суммирование элементов.

Физика: потенциальная энергия, нить с грузиками, центр масс, центр тяжести.

Информатика: недостатки встроенных инструментов Mathcad, текстовый индекс и индекс вектора, побитовое представление рисунка.

Сайт этюда: **https://www.ptcusercommunity.com/message/469478**

Задачу о канатной дороге (о цепной функции — см. этюд 7) можно решить и по-другому, сведя ее к задаче оптимизации.

Давайте проведем такой физико-математический компьютерный эксперимент. Закрепим грузики одного веса на нити так, чтобы получились некие бусы или четки — см. рис. 21.1. Посмотрим, как все это будет провисать, если эту нагруженную нить на концах закрепить в двух точках. Грузики будут стараться опуститься как можно ниже так, чтобы их суммарная потенциальная энергия стала минимальной. Препятствовать же падению грузиков на пол будет нить, длины отрезков которой от грузика к грузику должны оставаться постоянными (нить у нас нерастяжимая и невесомая, вернее, ее вес несравним с весом грузиков). Тут вырисовывается типичная задача *оптимизации* с *ограничениями*, где *целевой функцией* будет суммарная потенциальная энергия грузиков, которую нужно минимизировать, ограничения — постоянство расстояний между грузиками, а *переменными оптимизации* — координаты грузиков. На рисунке 21.1 показано решение этой задачи с помощью пакета Mathcad.

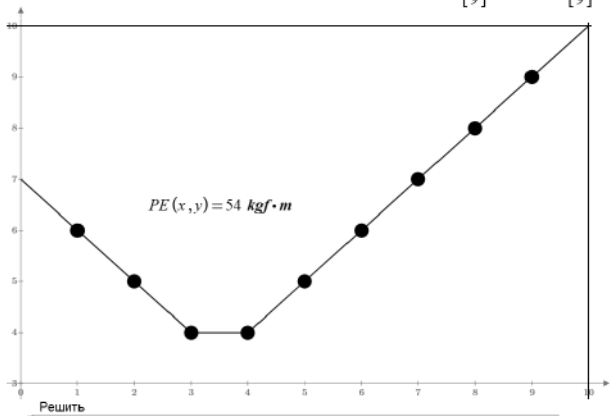
Итак, вычислительный эксперимент! Берется нерастяжимая и невесомая (в сравнении с грузиками) нить длиной `S` (15 м, почему не сантиметров — см. ниже), на которую нанизываются и закрепляются на равных расстояниях друг от друга 9 бусинок (`n`) массой 1 кг (`mass`) каждая. Почему тут не граммы — опять же см. ниже. Один конец нити подвешивается на высоте 7 м (y_0), а второй на высоте 10 м (y_{10}). Расстояние по горизонтали между точками подвеса равно 10 м (x_{10}). Внимание! Нулевой индекс тут означает не нулевой элемент векторов, а просто текстовый индекс — часть имени переменной, сдвинутый вниз. Этот нюанс мы еще отметим.

Как провиснут эти «бусы»? Задачу мы будем решать численно, а не аналитически. Поэтому всем исходным величинам `n`, `S`, `mass`, x_0, y_0, x_{10} и y_{10} мы задали конкретные значения (9, 15 м, 1 кг, 0 м, 7 м, 10 м и 10 м) — см. первую строку расчета на рис. 21.1. Расстояния тут измеряются метрами, а не сантиметрами (что было бы естественнее для бус), потому что, во-первых, это не принципиально, а во-вторых, точность нашего расчета по умолчанию равна 0.001, т. е. 1 мм, если говорить о размерной задаче, где метр — это базовая единица длины. Переход на сантиметры потребует нарушения умолчания — изменения точности расчета с 0.001 на 0.00001, например. Многие забывают это сделать и получают неверный, вернее недостаточно точный ответ.

$x_0 := 0\ m \qquad y_0 := 7\ m \qquad S := 15\ m \qquad n := 9 \qquad mass := 1\ kg \qquad x_{10} := 10\ m \qquad y_{10} := 10\ m$

$$\Delta s := \frac{S}{n+1} = 1.5\ m \qquad PE(x,y) := g \cdot mass \cdot \sum_{i=1}^{9} y_i$$

Первое приближение к решению $x := \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{bmatrix} m \qquad y := \begin{bmatrix} 6 \\ 5 \\ 4 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{bmatrix} m$



Решить

Ограничения

$$\Delta s = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} \qquad \text{Длина первого звена цепи}$$

$$\Delta s = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \qquad \Delta s = \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2}$$

$$\Delta s = \sqrt{(x_3 - x_4)^2 + (y_3 - y_4)^2} \qquad \Delta s = \sqrt{(x_4 - x_5)^2 + (y_4 - y_5)^2}$$

$$\Delta s = \sqrt{(x_5 - x_6)^2 + (y_5 - y_6)^2} \qquad \Delta s = \sqrt{(x_6 - x_7)^2 + (y_6 - y_7)^2}$$

$$\Delta s = \sqrt{(x_7 - x_8)^2 + (y_7 - y_8)^2} \qquad \Delta s = \sqrt{(x_8 - x_9)^2 + (y_8 - y_9)^2}$$

$$\Delta s = \sqrt{(x_9 - x_{10})^2 + (y_9 - y_{10})^2} \qquad \text{Длина последнего звена цепи}$$

Математика

Блок решения

Блок решения (Ctrl+1)

Решатель

$$\begin{bmatrix} x \\ y \end{bmatrix} := \mathbf{Minimize}(PE, x, y) \qquad \text{Минимизация потенциальной энергии}$$

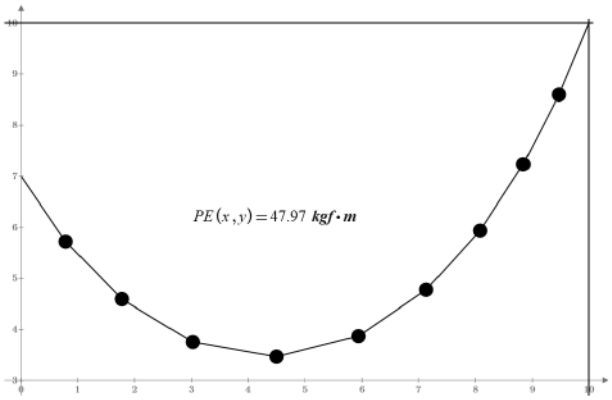


Рис. 21.1

Решение задачи о провисающих на нити грузиках

Потенциальная энергия — это произведение веса груза на его высоту. В расчете она выражается через функцию с именем PE (potential energy). Из этой функции можно убрать константы g (ускорение свободного падения) и mass (масса каждого грузика). Решение при этом не изменится. Отсюда вывод, который мы уже делали в этюде 7: тяжелая металлическая цепь и легкая веревочка будут провисать одинаково и на Земле, и на Луне, если у них совпадают длины и координаты точек подвеса. Но мы специально оставляем величины g и mass в расчете, чтобы можно было, например, рассчитать геометрию цепи с грузиками разной массы (бусы, где центральная бусинка самая крупная, а остальные плавно уменьшаются к периферии) или цепь с подвешенным массивным грузом (канатная подвесная дорога — см. рис. 7.7 на стр. 146). А такую канатную дорогу в будущем можно соорудить и на Марсе, например, где иное («не земное») значение ускорения свободного падения. Эти решения можно найти на сайте книги.

Вторая особенность целевой функции PE связана уже не с физикой задачи, а с особенностями ее решения в среде Mathcad. Функция PE имеет два аргумента-вектора x и y, а зависит только от одного — от y. Такой записи с лишним (фиктивным) аргументом требует сам пакет Mathcad: если оптимизация ведется по двум переменным, то и целевая функция должна иметь тоже две переменные-аргумента — ни больше ни меньше. Поэтому пришлось ввести в функцию PE дополнительный аргумент x[1]. Но фактически оптимизация у нас будет вестись не по двум переменным, а по 18 — по девяти значениям в векторе x и по девяти значениям в векторе y. Задача получается довольно сложная. Следствием этого стало то, что Решатель пакета Mathcad 15 с ней не справился, но новый Решатель под названием Artelys Knitro (**https://www.artelys.com/en/optimization-tools/knitro**), встроенный в пакет Mathcad Prime, задачу решил — см. обсуждение этой проблемы на форуме **https://www.ptcusercommunity.com/thread/136739**.

Еще один вычислительный, вернее, оформительский нюанс задачи. В расчете на рис. 21.1 переменные x_0, y_0, x_{10} и y_{10} — это, как уже отмечено, не элементы векторов x и y, а скалярные переменные с индексами 0 и 10. Но в расчете есть и одноименные векторы x и y, хранящие координаты девяти грузиков. На сайте этюда можно найти решение для 25 грузиков. Прием перемешивания двух видов индексов довольно сомнительный — он часто путает даже опытных пользователей Mathcad.

После ввода исходных данных и целевой функции необходимо дать первое числовое предположение для переменных оптимизации: для двух векторов x и y — координат центров грузиков. Здесь нужно помочь компьютеру — дать такие первые приближения, чтобы они были близки к реальности — см. первый график на рис. 21.1. В центре графика показан подсчет потенциальной энергии грузиков (54 кг-силы на метр) при таком их начальном расположении. Нужно сделать так, чтобы эта энергия стала минимальной, но выполнялись бы ограничения по геометрии цепи. Это делает блок Решить, показанный на рис. 21.1 сразу после первого графика, где в области ограничений записаны 10 равенств: длины отдельных участков нити, соединяющих грузики в цепь (в нитку бус), должны оставаться равными Δs. Функция Minimize по особому численному алгоритму меняет значения элементов в векторах x и y так, чтобы функция PE приняла минимальное значение, а ограничения бы выполнялись. Нижний график на рис. 21.1 показывает, что наша задача вроде решена: наши грузики провисли правильной цепью, а их потенциальная энергия снизилась с 54 до 47.97 кг-силы на метр. Проверить выполнение ограничений несложно. Достаточно вывести значения 10 выражений с квадратными корнями, записанные в области ограничений решателя Mathcad. Все они должны быть равны значению 1.5 м (Δs) с точностью до миллиметра. Труднее по-

[1] В пакет Mathcad встроена функция с именем time, которая имеет один фиктивный аргумент. Эта функция возвращает машинное время компьютера. Мы ее использовали для оценки времени счета — см., например, в этюде 4 рис. 4.43 на стр. 96, и воспользуемся еще ниже.

казать, что энергия нашей механической системы достигла «дна» потенциальной ямы. Это можно сделать, вспомнив, что «нет ничего практичней хорошей теории»: нужно вспомнить (см. этюд 7), что такое цепная функция и какая ее формула — см. рис. 21.2.

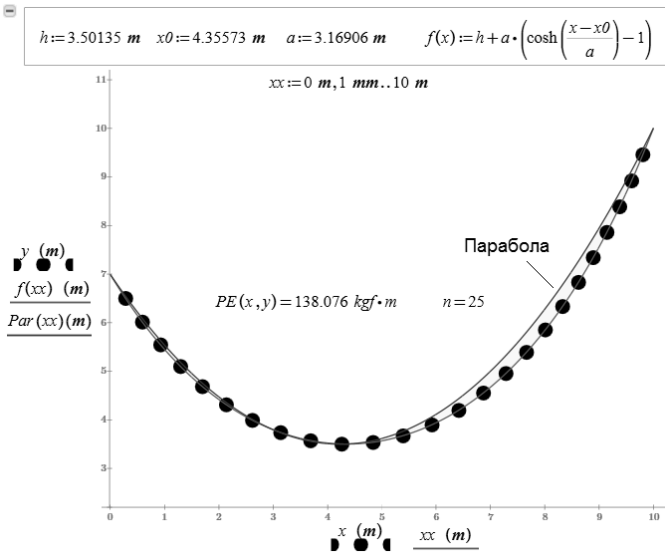


Рис. 21.2

Проверка модели нити с грузиками: наложение цепной линии на грузики

В этюде 7 на рисунке 7.3 (стр. 142) показан авторский сайт Интернета, созданный по технологии Mathcad Calculation Server, когда можно считать в среде Mathcad без установки на компьютер самого пакета. Достаточно, чтобы компьютер имел выход в Интернет. На сайте нужно ввести значение h_1, h_2, L и S (7, 10, 10 и 15 м), нажать кнопку Recalculate и получить ответ — значение параметров цепной функции x_0, h и a, кривая которой проходит через две заданные точки и имеет заданную длину. Сама же цепная функция включает в себя гиперболический косинус cosh, который, в свою очередь, состоит из двух экспонент. Дополнительно на сайте прорисовывается сама цепная линию, в формуле которой h — это ордината нижней точки цепи, а x_0 — ее абсцисса. Задача сводится к решению системы трех нелинейных алгебраических уравнений блоком Given-Find пакета Mathcad 15. Само же уравнение цепной линии, как уже было отмечено в этюде 7, выводится через решение дифференциального уравнения, полученного в результате анализа сил, действующих на элементарный участок цепи — см., например, эти аналитические выкладки здесь **www.math24.ru/уравнение-цепной-линии.html**. Оценка правильности решения, основанного на анализе потенциальной энергии механической системы, отображена на рис. 21.2: грузики (их уже не 9, а 24) «пронизывают» цепную линию. На рисунке 21.2 дополнительно прорисована парабола, проходящая через точки подвеса нити и через ее нижнюю точку. Видно, что цепь провисает не по параболе (а так считал даже Галилей — и это мы уже отметили в этюде 7), но близко к параболе.

Примечание. Параметр a в уравнении цепной линии, как мы уже отмечали ранее, характеризует ее крутизну и характер. Если значение a уменьшить от некоего положительного значения (см. рис. 21.2) к почти нулю, то цепь будет стремиться к прямой линии (к натянутой струне), а потом (при отрицательных значениях параметра a) превратится в... арку. Такую арку, кстати, можно получить, если в расчете на рис. 21.1 функцию Minimize заменить на функцию Maximize — см. рис. 21.3, где грузики превратятся

в... воздушные шарики, наполненные гелием. Здесь потенциальная энергия шариков не минимизируется, а максимизируется.

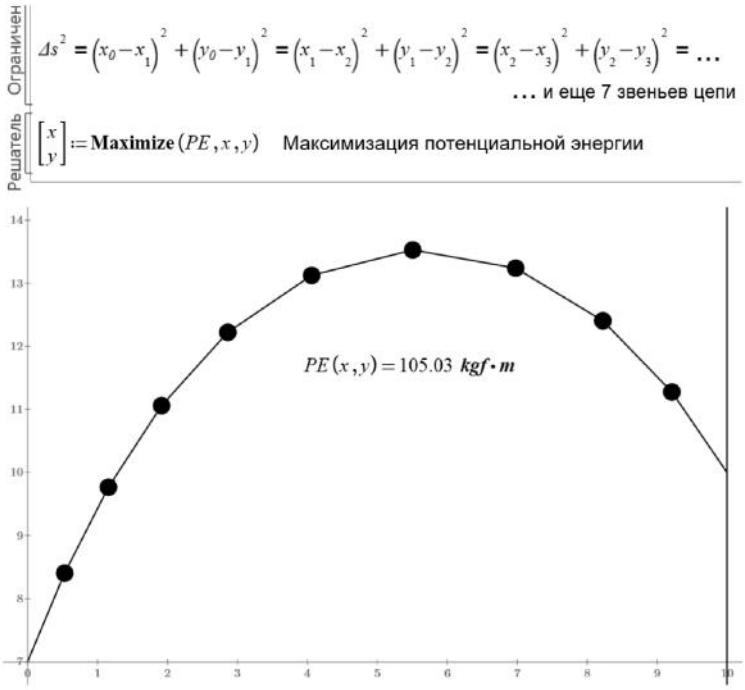


Рис. 21.3
Арка — перевернутая цепь

В решении на рис. 21.3 мы имеем только одно ограничение в виде не отдельных уравнений, а в виде... цепи операторов равенства. Цепь, так сказать, решается с помощью цепи. Кроме того, был убран квадратный корень, что ускорило расчет, не меняя его суть.

Примечание. При формировании целевой функции (см. рис. 21.1) мы использовали фактически элемент программирования — цикл с параметром i в операторе суммирования. Такой прием желательно было бы задействовать и при формировании ограничений. Но пакет Mathcad этого не допускает. Приходится либо записывать отдельные уравнения-ограничения (см. рис. 21.1), число которых равно числу отрезков нити, либо писать цепочку равенств в виде одного ограничения (см. рис. 21.3). На сайте этюда эта цепочка равенств прописана в виде... провисающей цепи.

Изложенное позволяет предложить такую интересную школьную или вузовскую лабораторную работу, объединяющую физику, математику и информатику.

Берется нить, на которой крепятся грузики одного или разного веса на одинаковых или разных расстояниях друг от друга. Эти «бусы» подвешивается за два конца нити. Все это фотографируется на камеру и отправляется на компьютер, где оцифровывается — определяются координаты всех узловых точек цепи. Далее решается описанная нами задача оптимизации с ограничениями. Тут можно также учесть и вес нити, заменив ее на реальную цепочку, принимая ее не за ломаную прямую (рис. 21.1 и 21.3) или отрезки цепной функции. Заканчивается эта работа наложением графика провисания «бус» на фотографию реальной механической системы с грузиками и анализом полученных результатов. Нерастяжимые нити можно заменить на растяжимые резинки и решить данную задачу с учетом потенциальной энергии растяжения и с учетом закона Гука или какой-то другой, уже нелинейной закономерности.

ДИВЕРТИСМЕНТ. ЦЕНТР ТЯЖЕСТИ

Есть простой способ определения центра тяжести плоской фигуры, опирающийся на минимизацию потенциальной энергии. В плоской фигуре на краях делаются два отверстия, через которые булавкой фигура, сделанная из плотного картона, последовательно подвешивается на доске. К булавке прикрепляется отвес. При этом на картонной фигуре отмечаются карандашом прямые линии отвеса — см. рис. 21.4.

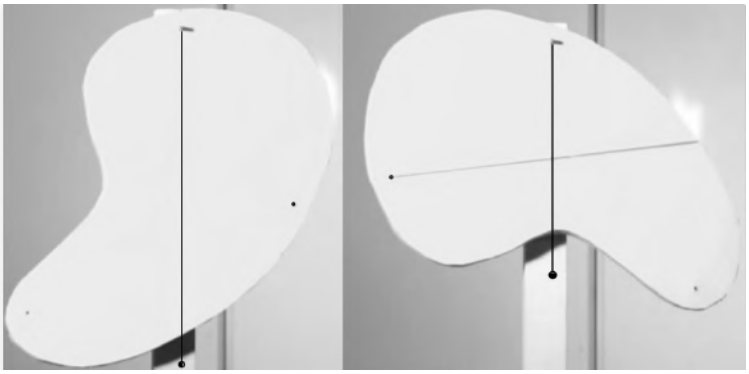


Рис. 21.4
Экспериментальное определение центра тяжести плоской фигуры

Рисунок 21.4 — это два кадра анимации, которую можно найти в Википедии в статье с названием, повторяющим название нашего дивертисмента. В данной анимации можно видеть, как подвешенная на булавке картонка сначала покачивается из стороны в сторону, как маятник, а затем замирает в положении, при котором ее потенциальная энергия будет минимальна. Так и наша нить с грузиками (см. выше) после ее подвеса будет сначала колебаться, а затем застынет, зафиксировавшись в некой потенциальной яме. Кстати, хорошее задание для читателей: смоделировать на компьютере колебание такого необычного маятника, нить которого закреплена на двух концах и у которого несколько грузиков. Тут колебания могут быть и продольными, и поперечными.

В картонке, показанной на рис. 21.4, можно проделать еще одно, третье отверстие и подвесить картонку с отвесом через него (это третье отверстие, кстати, видно на рис. 21.4). Третий отрезок прямой пройдет через точку пересечения двух первых отрезков. Эта точка будет отмечать *центр тяжести* нашей плоской фигуры. Плоскую фигуру можно так вырезать из картона, чтобы центр тяжести оказался вне контура. В этом случае нужно не проводить прямые линии, а закрепить нить на картонке двумя клеевыми каплями, отрезав затем (после засыхания клея) грузик отвеса.

Так можно доказать, что точка на пересечении двух или трех и более отрезков — следов от отвеса является центром тяжести, вернее центром масс. Эти два центра совпадают, если поле силы тяжести однородно, что можно допустить на поверхности Земли для фигур не слишком большого размера. Об этом нюансе рассказано в вышеотмеченной статье Википедии, и мы еще его коснемся в этюде 23.

Давайте снимем нашу фигуру с доски, отсканируем ее, а полученный рисунок отправим в компьютер, превратив его в матрицу[2]. Такую операцию мы проделывали в этюде 5, когда брали из Интернета портрет Че Гевары — см. рис. 5.3

[2] Когда Иисуса сняли с креста и положили на плащаницу, то его фигура тоже была «отсканирована» на ткани. Эта христианская реликвия сейчас хранится в одном из соборов Турина. Такая вот, как кому-то может показаться, несколько кощунственная аналогия.

(стр. 107) и рис. 5.4 (стр. 108). На рисунке 21.5 показаны такие действия в среде Mathcad 15.

M :=

$n := cols(M) = 350$

$m := rows(M) = 385$

$M := \frac{M}{255}$

M =

	1	2	3	4	5	6	7	8	9	10
321	0	0	0	1	1	1	1	1	1	1
322	0	1	1	1	1	1	1	1	1	1
323	0	1	1	1	1	1	1	1	1	1
324	0	1	1	1	1	1	1	1	1	1
325	0	1	1	1	1	1	1	1	1	1
326	0	1	1	1	1	1	1	1	1	1
327	0	1	1	1	1	1	1	1	1	1
328	0	1	1	1	1	1	1	1	1	1
329	0	1	1	1	1	1	1	1	1	1
330	0	1	1	1	1	1	1	1	1	1
331	0	1	1	1	1	1	1	1	1	1
332	0	1	1	1	1	1	1	1	1	1
333	0	1	1	1	1	1	1	1	1	1

Рис. 21.5
Превращение подопытной картонки в матрицу

На рисунке 21.5 отображено, как в Mathcad-документ вставлен объект «Рисунок» из панели инструментов «Матрица», который отформатирован на ввод в переменную `M` рисунка из буфера обмена. А до этого в графическом редакторе Paint скан нашей картонки был подредактирован и сохранен как черно-белый рисунок: белое — это сама фигура, а черное — все, что находится вне фигуры. Белые биты растрового рисунка имеют значения 255, но мы изменили их на единицу, что видно на рис. 21.5, где выведена часть матрицы M с левым краем картонки.

У полученной матрицы `M` можно определить координату (номер строки и номер столбца), равноудаленную от координат остальных элементов матрицы. Это делается в программе, показанной на рис. 21.6.

В расчете формируется матрица `S`, равная по размеру матрице `M`, но хранящая не нули и единицы, а значения некой отдаленности данной ячейки матрицы от остальных элементов матрицы. Тут задействована теорема Пифагора о гипотенузе и катетах, но без использования квадратного корня, что сделано для ускорения расчета. А о скорости расчета пришлось думать — программа на рис. 21.6 имеет четыре вложенных цикла с параметром, и эти параметры меняются от 1 до 350 (столбцы матрицы) и от 1 до 385 (строки матрицы). Компьютер одного из авторов книги создавал эту матрицу более 11 часов, что было подсчитано через функцию `time`. Другая встроенная в Mathcad функция `match` (ее мы использовали в этюде 15, когда решали задачу коммивояжера — см. рис. 15.10 на стр. 345) позволила определить центр масс — координату матрицы `S` с минимальным значением.

Задача о центре масс была выложена на форуме пользователей Mathcad (см. **https://www.ptcusercommunity.com/thread/140868**) с просьбой найти более быстрый способ нахождения центра матрицы, пардон, центра масс плоской фигуры. Решение нашел американец Марк Гейз (Mark Gase). Оно отображено на рис. 21.6 в виде двойных сумм. Но наше «одиннадцатичасовое решение» не пропало — оно помогло раскрасить нашу картонку: см. рис. 21.7.

```
evening := time

S := | for  i ∈ 1 .. m
     |   for  j ∈ 1 .. n
     |     S_{i,j} ← | s ← 0
     |               | for  i1 ∈ 1 .. m
     |               |   for  j1 ∈ 1 .. n
     |               |     s ← s + M_{i1,j1}·[(i − i1)^2 + (j − j1)^2]
     |               | s
     | S
```

$$\text{morning} := \text{time} \qquad \frac{\text{morning} - \text{evening}}{3600} = 11.617 \quad \text{hr}$$

$$\min(S) = 1.376 \times 10^9 \qquad \max(S) = 7.453 \times 10^9$$

$$\text{Mass} := \sum_{i=1}^{m} \sum_{j=1}^{n} M_{i,j} = 83430$$

$$\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} (i \cdot M_{i,j})}{\text{Mass}} = 190$$

$$\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} (j \cdot M_{i,j})}{\text{Mass}} = 187 \qquad \text{match}(\min(S), S) = \left[\begin{pmatrix} 190 \\ 187 \end{pmatrix}\right]$$

Рис. 21.6
Определение «центра масс» матрицы

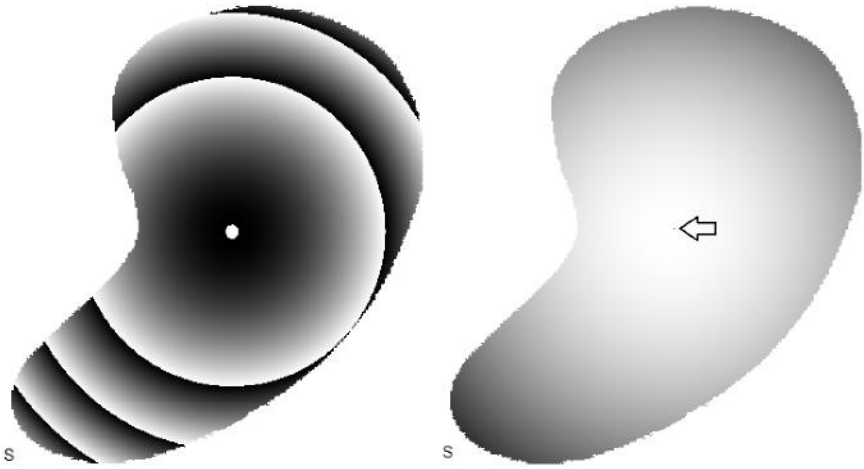


Рис. 21.7
Раскрашенные картонки с отмеченным центром масс

Если через белую точку на полосатой картонке слева пропустить иголку и воткнуть ее в доску, то картонка займет нейтральное положение: если ее раскручивать, то она будет останавливаться в разных позициях, а не в определенных, пока-

занных на рис. 21.4. Это будет дополнительным подтверждением правильности нашего решения. Такую «раскрутку», кстати, проводят с колесами машин, когда центруют их.

Центр масс — это очень важный параметр при расчете остойчивости судов, и мы к нему еще вернемся в этюде 23.

Задание читателям

1. Смоделировать провисание бус в случае, если силы притяжения не параллельны, а сходятся в точку (неоднородное гравитационное поле): длинная цепь с грузиками подвешена, например, в космическом околоземном пространстве.

2. Бусы, показанные на рис. 21.1 и 21.2, нанизаны на растяжимую нить (резинку). Как они провиснут?

3. Определить дату с днем недели и точное время, когда функция `time`, упомянутая в сноске 1 на стр. 423, возвращала нулевое значение.

4. Придумайте другие (цветные) схемы раскраски картонок с рассчитанным центром масс (рис. 23.7).

5. В Пуэрто-Рико в кратере потухшего вулкана сооружен крупнейший в мире радиотелескоп в виде круглого гамака или провисшего батута, над которым подвешен приемник радиосигналов. Гамак без специальных решений провиснет по цепной линии, а для телескопа нужна парабола — см. рис. 21.2. Какие это могут быть спецрешения? Грузики? Канаты с переменным сечением? Рассчитайте их!

6. Кстати о канате с переменным сечением. Какое должно быть оно, чтобы канат провис по параболе, гиперболе, по дуге окружности или эллипса?

7. Какое должно быть переменное сечение провисающего каната, чтобы удельная сила его разрыва была постоянной? По какой «формуле» будет провисать такой канат?

Этюд 22

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ, ИЛИ ПО ПОРЯДКУ СТАНОВИСЬ!

В этюде рассмотрены вопросы последовательного и параллельного вычисления, а также создания программ сортировки вектора и анимации этой процедуры в среде Mathcad.

Математика: вектор.

Физика: хаотичное движение, упорядочивание.

Информатика: алгоритмы сортировки, параллельные вычисления, цикл с параметром, цикл с предпроверкой, анимация, скорость выполнения программного кода.

Сайт этюда: **https://www.ptcusercommunity.com/message/469477**

Как-то раз первый автор книги разговорился в Интернете с одной своей бывшей одноклассницей. Вспоминали хорошее и плохое школьных лет. Эта одноклассница призналась, что для нее одними из самых неприятных моментов в школе были... построения на уроках физкультуры. Девочки и мальчики строились в спортзале по росту и вставали друг перед другом в две шеренги в стиле «белый верх — черный низ»: в белых маечках и черных трусиках. Эта одноклассница была самая высокая в классе и довольно полная, пардон, крупная. В обычной жизни и в обычной одежде это было не так заметно, но в спортивном зале она оказывалась во главе шеренги полуголая (по стандартам тех лет) и как бы... у позорного столба. Мальчики ее разглядывали, а потом дразнили дылдой. Автор также вспомнил неприятный момент таких построений: у него был друг довольно низкого роста, который всегда оказывался в конце «мужской» шеренги, и на него из женской половины презрительно поглядывала девочка, которая ему нравилась[1].

Фотографию построения людей по росту можно видеть в этюде 14 на рис. 14.20 (стр. 323). Там построившиеся разбились на группы по росту с шагом в один дюйм и образовали живую гистограмму.

Да, нужно признать, что процедура построения по росту в школе или в армии несколько дискриминационная. В армии к тому же низкорослые солдаты оказываются в конце марширующей колоны и «глотают пыль». Можно предложить альтернативный порядок построения[2], например, по возрасту, цвету глаз, волос или по какому-то другому атрибуту школьников или солдат, не имеющему признаков дискриминации. Но, так или иначе, тут нужно будет использовать процедуру *сортировки*, которая часто становится темой школьных и вузовских уроков по информатики. Операция вроде бы рутинная, разобранная «по косточкам», но давайте внесем в нее что-то новенькое.

Разработано множество алгоритмов сортировки. В Википедии можно найти такие их названия: сортировка пузырьком, сортировка перемешиванием, гномья

---

[1] Говорят, что наш великий полководец Александр Суворов однажды выстроил своих крепостных по росту в две шеренги: неженатые парни с одной стороны, незамужние девки — с другой. Так сложились пары, которым он повелел идти в церковь венчаться. Мнение новобрачных никого не интересовало. (**https://lenta.ru/articles/2016/04/26/russian_slavery**).

[2] А вообще без построения по какому-либо признаку тут не обойтись: школьники или солдаты будут спорить и толкаться, стараться встать ближе к правому (приоритетному) краю шеренги или, наоборот, «затеряться в середине толпы». Можно предложить строить школьников в том порядке, в каком они по алфавиту записаны в классном журнале. Но тут также можно узреть элементы дискриминации: кого чаще вызывают к доске: Абрамова, Яковлева или, наоборот, какого-нибудь Иванова-Петрова-Сидорова из середины списка. Кстати, к сортировке мы очень часто прибегаем, выстраивая файлы на экране компьютера по имени, типу, размеру или времени создания. Эти значения «в шеренге файлов» могут либо возрастать, либо убывать после повторного щелчка по имени атрибута.

сортировка, сортировка вставками, сортировка слиянием, сортировка с помощью двоичного дерева, сортировка Timsort, сортировка подсчётом, блочная сортировка, поразрядная сортировка, сортировка выбором, сортировка Шелла, сортировка расчёской, пирамидальная сортировка, плавная сортировка, быстрая сортировка, интроспективная сортировка, терпеливая сортировка, рекурсивная сортировки, глупая сортировка, блинная сортировка, сортировка перестановкой...[3]

Эти алгоритмы реализованы на разных языках программирования. Попробуем и мы это сделать, но в среде пакета Mathcad, который в настоящее время широко используется в школах и вузах[4].

Основа многих алгоритмов сортировки — это обмен численных значений у двух переменных. В языке программирования BASIC, например, эту процедуру выполняет оператор `swap(a, b)`. Но в математическом пакете Mathcad этого оператора нет, но его нетрудно реализовать другими средствами.

На рисунке 22.1 можно видеть два способа реализации оператора `swap(a, b)` в среде Mathcad: традиционный («треугольный» — с использованием вспомогательной переменной) и «экзотический», о котором многие пользователи Mathcad и не догадываются — не через последовательное присваивание, а в некоем параллельном режиме, когда операторы присваивания записаны в матричном виде и выполняются независимо друг от друга. Эта технология обсуждалась на сайте пользователей Mathcad **https://www.ptcusercommunity.com/message/448295**.

Рис. 22.1
Два способа перестановки значений переменных в среде Mathcad

Если под рукой есть оператор перестановки `swap(a, b)` в явном («треугольном») или в матричном виде, то несложно написать в среде Mathcad функцию-программу (рис. 22.2), реализующую, наверно, самый примитивный алгоритм сортировки, суть которого такова.

Школьники произвольно выстраиваются в ряд (у нас на рис. 22.2 это делает функция `runif`, возвращающая вектор с 10 элементами и со случайными числами

[3] В студенческие годы первый автор подрабатывал на овощной базе. Самой непыльной работой там считалась... сортировка лимонов. В те времена лимоны были единственными фруктами, какие продавались не по весу, а поштучно (к чаю) за 25, 30, 35, 40 и 45 копеек или что-то в этом роде: чем лимон крупнее, тем он, естественно, дороже. На овощных базах их сортировали так — просовывали в круглые дырки с разным диаметром на специальном сортировочном столе. Лимоны падали в отдельные ящики, которые потом развозили по разным магазинам, чтобы не было пересортицы. Автор же в те годы, глядя на эти ящики с отсортированными лимонами, вспоминал... вузовский курс математической статистики с его нормальным распределением. Анекдот тех времен: «Где подрабатываешь?» — «На овощной базе» — «Устаешь?» — «Очень!» — «Ящики таскаешь?» — «Нет, лимоны сортирую — голова устает!»

[4] В пакете Mathcad есть, кстати, функция `sort` для этой работы, но мы пока забудем о ее существовании. Кроме того, встроенные в Mathcad функции `csort` и `rsort` позволяют сортировать матрицы по столбцам (`c`) или рядам (`r`), задавая ключ сортировки — номер столбца или строки матрицы. На сайте **https://www.ptcusercommunity.com/videos/6867** размещена анимация сортировки матрицы, когда значение ее элементов отображается цветом.

в интервал от 150 до 200 — возможный рост человека в сантиметрах), а учитель физкультуры (физрук) проходит вдоль шеренги слева-направо (вложенный цикл `for`) и меняет местами пары школьников, если левый школьник (V_{i-1}) окажется выше правого (V_i). Эта процедура выполняется до тех пор, пока (цикл `while`) очередной проход физрука вдоль шеренги не выявит беспорядка, и переменная `Flag` (признак окончания сортировки) останется равной нулю.

```
MySort(V) := | Flag ← 1
             | while Flag
             |   | Flag ← 0
             |   | for i ∈ 1..last(V)
             |   |   | if V_{i-1} > V_i
             |   |   |   | [V_i  V_{i-1}] ← [V_{i-1}  V_i]
             |   |   |   | Flag ← 1
             | V
```

$$V := \mathrm{runif}(10, 150, 200) = \begin{bmatrix} 163 \\ 150 \\ 190 \\ 161 \\ 178 \\ 156 \\ 188 \\ 177 \\ 172 \\ 185 \end{bmatrix} \qquad MySort(V) = \begin{bmatrix} 150 \\ 156 \\ 161 \\ 163 \\ 172 \\ 177 \\ 178 \\ 185 \\ 188 \\ 190 \end{bmatrix}$$

Рис. 22.2
Наипростейшая сортировка вектора (Mathcad Prime)

Пакет Mathcad 15 имеет удобные средства анимации (см. этюд 8), которые можно применить и для визуализации процесса сортировки. На рисунке 22.3 показано, как в программу на рис. 22.2 были вставлены три дополнительных оператора (они в рамках), позволившие показать сортировку вектора в анимации: столбики на диаграмме последовательно меняются местами, выстраиваясь в правильную шеренгу. При этом очередной «высокий столбик» перемещается вправо, как бы толкаясь и передавая эстафету столбику с более высоким ростом. Эта анимация выложена на сайте **https://www.ptcusercommunity.com/videos/6795**.

Управляет анимацией системная переменная `FRAME`, которая при создании анимации автоматически будет менять свое значение от 0 (рисуется исходный вектор в виде хаотических столбиков) до 999 (отсортированный вектор, часть которого показана на рис. 22.3). Перед созданием анимации можно вручную менять значение системной переменной `FRAME` и видеть изменения на графике — на столбчатой диаграмме. Так программа отлаживается перед анимацией. В программу на рис. 22.3 вставлен счетчик перестановок соседних элементов векторов `n`. Если он становится равным номеру кадра анимации `FRAME`, то программа оператором `return` возвращает частично отсортированный вектор, который рисуется на графике в виде столбчатой диаграммы.

Характер анимации, один кадр которой показан на рис. 22.3, можно кардинально поменять, если в конце плеча оператора `if` (после оператора `Flag ← 1`) записать оператор `break`. Он прерывает выполнение цикла `for` и передает управление программой в его конец: физрук, переставив местами очередную пару школь-

ников, не пойдет дальше с проверками вдоль шеренги, а вернется в ее конец[5] и перейдет к циклу while. С оператором break отсортированные элементы вектора будут группироваться не на правом (рис. 22.3), а на левом фланге шеренги, но конечный результат будет тот же: слева окажутся самые низкие, а справа самые высокие. Эта анимация размещена на сайте **https://www.ptcusercommunity.com/videos/6824.**

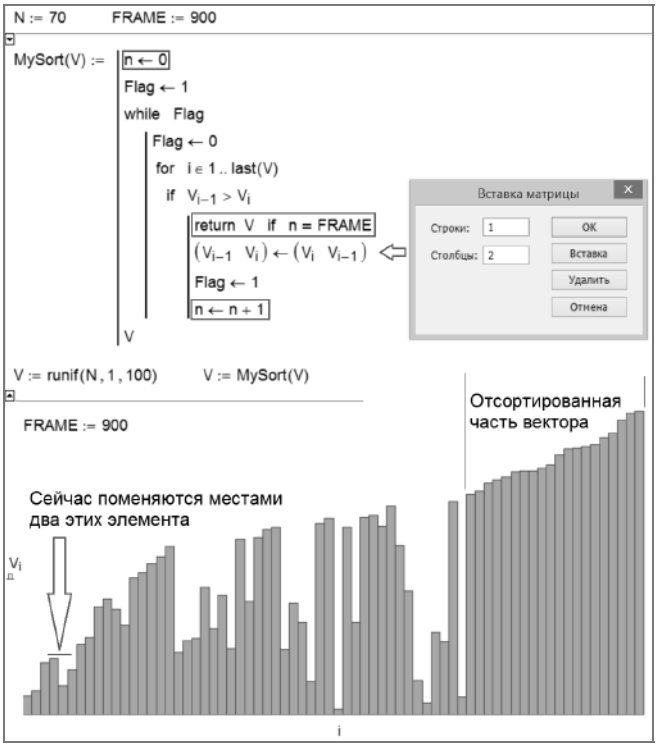


Рис. 22.3

Создание анимация сортировки по простейшему алгоритму (Mathcad 15)

Программа сортировки на рис. 22.3 имеет переменное число выполнения вложенных циклов while-for, и это число (оно хранится в переменной n) зависит от вида исходного вектора. Если этой программе «подсунуть» уже отсортированный вектор, то вложенный цикл while-for будет выполнен минимальное число раз, равное размеру вектора (у нас на рис. 22.3 — это семьдесят раз, как указано первым оператором). Если же этой программе дать уже отсортированный, но «перевернутый» вектор (этот «переворот» делает встроенная в Mathcad функция reverse), то вложенный цикл for будет «глупо»[6] выполняться максимальное число раз, переставляя пары элементов вектора. Эта анимация размещена здесь на сайте

[5] В среде Mathcad, как и в языке Паскаль, есть три «прерывающих» оператора: return, break и continue. Первый из них прерывает выполнение всей программы и возвращает из нее свой операнд (см. рис. 22.3), второй оператор (он без операнда) прерывает выполнение цикла for или while и передает управление программой в его конец. Третий оператор тоже прерывает выполнение цикла, но передает управление в его начало. Кстати, мало кто даже из англичан знает, что означает break в слове завтрак — a breakfast. A fast — это по-английски пост (в смысле, не послание в Интернете, а ограничение в питании), a breakfast — это прерывание поста — разговение, которое имеет место на Пасху или в Рождество.

[6] Если «глупой» обезьяне дать очищенный банан, то она попытается его еще раз очистить. Так и наша программа на рис. 22.3 будет еще раз пытаться отсортировать уже отсортированный вектор.

https://www.ptcusercommunity.com/thread/135862 как некий добавок к вопросу на форуме Mathcad по «перевертыванию» векторов.

Но в среде Mathcad несложно создать очень простую программу, сортирующую любой исходный вектор одним циклом. При этом число выполнения тела цикла в такой программе будет равно числу элементов в исходном векторе — см. рис. 22.4.

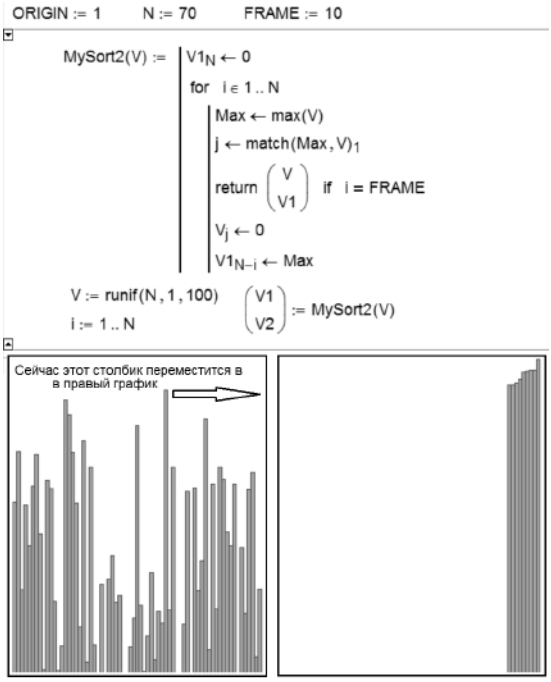


Рис. 22.4

Создание анимации сортировки по второму алгоритму (Mathcad 15)

Алгоритм в программе на рис. 22.4 таков: физрук оценивает взглядом неотсортированную шеренгу школьников, находит самого высокого и отправляет его на левый фланг отдельно формируемой шеренги. Эта операция повторяется `N` раз, где `N` — число элементов в исходном векторе. На столбчатых диаграммах рис. 22.4 видно, что из исходного вектора в формируемый вектор перенесено 9 «высокорослых» элементов (см. пустые места в левой диаграмме), и очередного переноса ожидает очередной «высокорослый» элемент. Программа на рис. 22.4 исполняет этот простой алгоритм, работая с двумя специфическими функциями пакета Mathcad:

`max` — определение максимального элемента массива (в нашем случае — вектора);

`match` — определение номера или номеров элементов массива с заданным значением (эта функция выдает вектор с одним или более элементами, поэтому-то в программе на рис. 22.4 эта функция записана с единичным индексом). Эти две удобные функции мы уже использовали при решении задачи коммивояжера — см. в этюде 15 рис. 15.10 (стр. 345).

Примечание. Самый «левый» элемент вектора в среде Mathcad по умолчанию имеет нулевой номер. Это умолчание присутствовало в программах на рис. 22.2 и 22.3. Но в программе на рис. 22.4 это умолчание нарушено: второй важной системной переменной Mathcad — переменной `ORIGIN` — присваивается единица. После этого нумерация векто-

ров будет начинаться с единицы[7]. Но в программировании тут более привычен нуль (см. этюд 2). И еще. Алгоритм, реализованный в программе на рис. 22.4, применим к исходному вектору с положительными элементами. Но его несложно доработать и для вектора с любыми числовыми элементами.

Можно развить алгоритм, реализованной в программе на рис. 22.4, — выбирать из неотсортированной части вектора не только максимальный, но и минимальный элементы и отправлять его на левый фланг формируемого нового вектора.

Анимация, создание которой отображено на рис. 22.4, расположена по адресу **https://www.ptcusercommunity.com/videos/6837.**

Читатели могут попробовать реализовать в среде Mathcad и, главное, анимировать алгоритмы сортировки, описанные в Википедии (см. выше) или свои собственные. При этом можно использовать встроенную в Mathcad функцию `time` для оценки скорости работы тех или иных пользовательских функций сортировки по сравнению со встроенной функцией `sort`. При этом нужно иметь в виду, что скоростные изощренные алгоритмы сортировки создавались во времена тихоходных вычислительных машин, когда время счета было одним из главных лимитирующих факторов. Теперь же право на жизнь имеют и медленные алгоритмы, у которых их суть лежит на поверхности — как в наших программах.

Есть такая наука бионика. Она позволяет решать технические проблемы, наблюдая за поведением живых организмов. Например, один из самых оптимальных алгоритмов решения задачи коммивояжера подсказали... муравьи, притаскивающие грузы в муравейник наикратчайшим маршрутом. И об этом подробнее будет рассказано в этюде 14. Мы, создавая алгоритм сортировки, тоже можем обратиться к «живым организмам» — построить в спортзале шеренгу школьников, например, по алфавиту, а затем приказать им перестроится по росту. Все это можно записать на видеокамеру, а затем с помощью замедленного просмотра на мониторе компьютера проанализировать и попытаться воссоздать данный «живой» алгоритм самосортировки. Если сортировку, проводимую физруком (см. выше), можно реализовать на однопроцессорной машине, то самосортировка школьников потребует уже многопроцессорной системы. При этом физрук может вмешиваться в процесс такой сортировки, контролировать его. Тут на память приходят реальные технические объекты или объекты живой природы. В современном самолете есть и центральный компьютер, и локальные вычислительные устройства (контроллеры), один из которых, например, регулирует подачу топлива в двигатель, прекращает подачу топлива при нештатной ситуации. У человека есть «центральный» мозг, но человек отдергивает руку от горячего предмета рефлекторно, без задействования «умственных способностей». Все это повышает надежность работы сложных технических и живых систем, повышает их управляемость.

И вторая, более простая тема для исследования. Как изменится характер сортировки, если в конце плеча альтернативы программы на рис. 22.2 поставить не оператор `break`, а оператор `continue`? И вообще, стоит ли вставлять их в программу? Как это будет влиять на скорость сортировки исходных векторов с разным распределением векторов «по росту»?

ДИВЕРТСМЕНТ. ДВЕ СЛУЧАЙНЫЕ «СЛУЧАЙНЫЕ» ФУНКЦИИ MATHCAD

На рисунке 22.2 было показано, как встроенная в Mathcad функция `runif` генерирует вектор, содержащий случайные числа в интервале от 150 до 200. Числа 150 и 200 — это условно минимальный и максимальный рост человека в санти-

[7] Программиста призвали в армию, поставили в начале шеренги и отдали команду: «По порядку рассчитайсь!». Наш программист сначала уточнил, по какой системе (двоичная, восьмеричная, десятичная, шестнадцатеричная или какая-то другая особая) нужно вести счет, а потом выкрикнул: «Нулевой!»

метрах. При этом было сделано допущение, что процент людей с определенным диапазоном роста одинаков. На рисунке 22.5 показана гистограмма (частота встречи) чисел, генерируемых функцией `runif`. При этом мы отказались от сантиметров и перешли к старым русским аршинам и вершкам и допустили, что рост человека может меняться от двух аршин до сажени (три аршина). Почему мы отказались от сантиметров? Об этом можно узнать в этюде 14 на стр. 324.

$$сажень := 7\ ft = 213.36\ cm \quad аршин := \frac{сажень}{3} = 71.12\ cm \quad вершок := \frac{аршин}{16} = 4.445\ cm$$

$n := 100000$

$V_1 := \mathrm{runif}(n, 2\ аршин, сажень) \qquad min(V_1) = 1.422 \qquad \max(V_1) = 2.134$

$H_1 := \mathrm{histogram}(16, V_1)$

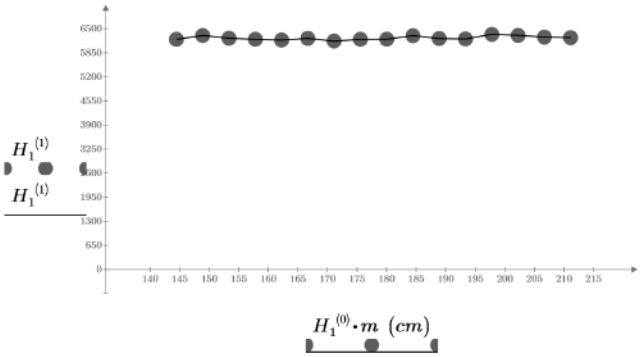


Рис. 22.5
Гистограмма равномерного распределения

Но в среде Mathcad есть функция с именем `rnorm`, позволяющая генерировать случайные числа с *нормальным* (см. рис. 22.6), а не с *равномерным* (см. рис. 22.5) распределением. Подбирая значения первого (математическое ожидание) и второго (среднеквадратическое отклонение) аргументов этой функции, можно добиться требуемого распределения студентов по росту. Нормальное распределение мы использовали в этюде 12, когда определяли параметры оптимального пожарного ведра — см. рис. 12.9 (стр. 283), рис. 12.10 (стр. 284), рис. 21.12 и рис. 21.13 (стр. 285). Но на сортировку характер распределения школьников, студентов или солдат по росту это не влияет.

Задание читателям

1. Реализуйте в среде Mathcad алгоритмы сортировки, перечисленные выше.
2. Возьмите большой массив данных по росту людей и сформируйте из него матрицу, где в столбцах были бы те значения, которые находятся в интервале один вершок. Должна получиться матрица, похожая на гистограмму в этюде 14 на рис. 14.20 (стр. 323). В столбцах этой матрицы разместите данные по росту с увеличением. Пустые места матрицы заполните строковой константой «–».
3. Рассчитайте массы грузиков (см. этюд 21), нанизанных на нить, чтобы она провисала так, как показано на рис. 22.6, естественно, в перевернутом виде.

$$V_2 := \mathrm{rnorm}\left(n, \frac{2\ \text{аршин} + \text{сажень}}{2}, 0.079\right) \qquad min\left(V_2\right) = 1.447 \qquad \max\left(V_2\right) = 2.155$$

$$H_2 := \mathrm{histogram}\left(16, V_2\right)$$

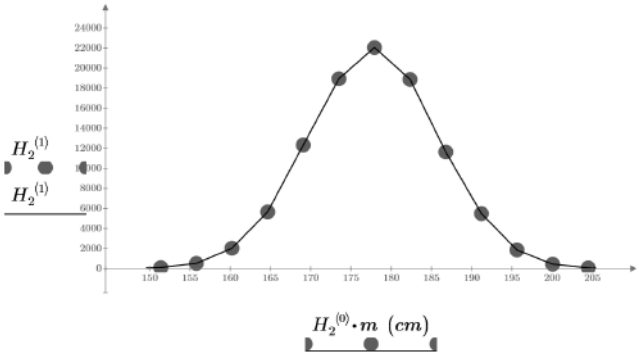


Рис. 22.6

Генерация вектора с нормальным распределением в среде Mathcad

Этюд 23

ПЛОТНОСТЬ СОЛЕНОЙ ВОДЫ, ИЛИ КАК КОРАБЛЬ РАССЧИТАЕТЕ И НАЗОВЕТЕ, ТАК ОН И ПОПЛЫВЕТ

В этюде обсуждаются вопросы расчета устойчивого плавания судов в пресной и морской воде (остойчивости) на основе рассмотрения сил, действующих на судно. Предложена лабораторная работа, охватывающая физику, математику и информатику.

Математика: решение алгебраических уравнений.

Физика: плотность, закон Архимеда, центр масс, центр тяжести, центр приложение сил, остойчивость судна, водоизмещение.

Информатика: облачные вычисления.

Сайт этюда: **https://www.ptcusercommunity.com/message/470168**

Первый автор этой книги работает в Международной ассоциации по свойствам воды и водяного пара **(www.iapws.org)**. Ее члены каждый год собираются в различных городах мира, тесно связанных с водой[1]. В 2014 г., например, такая встреча проходила в Москве[2], а заключительный банкет, без которого не обходится ни одно научное мероприятие, состоялся на борту теплохода, курсировавшего под стенами Кремля. В 2015 г. такая встреча была в Стокгольме, а финальный банкет прошел в музее легендарного шведского парусного корабля «Ваза», который был спущен на воду в 1628 г. и сразу... затонул. В 1961 г. корабль был поднят, затем законсервирован, подвергнут реставрации и в настоящее время экспонируется в специально построенном для него музее (рис. 23.1).

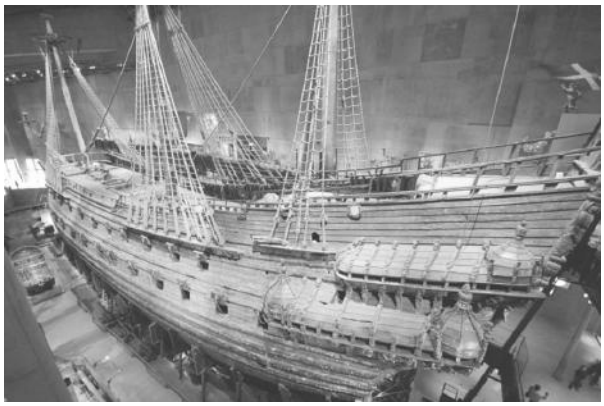


Рис. 23.1
Реставрированный корабль «Ваза»

Но встречи в рамках Международной ассоциации по свойствам воды и водяного пара, конечно, состоят не только из банкетов. Приходится и работать — рассматривать и утверждать документы, описывающие порядок расчета свойств этой важной субстанции [90]. Эти *формуляции* становятся стандартами для всех стран, и по ним выпускаются справочники, создаются программы для компьютеров. Российский подобный справочник [91] примечателен тем, что он дополнен сайтами Интернета с открытыми интерактивными сетевыми расчетами свойств воды и

[1] Почти все поселения мира возникали на берегах рек, озер, морей, океанов. Эти водоемы не разделяют людей, а наоборот, дают им удобные пути сообщения.

[2] В названии российской столицы, по одной из легенд, присутствует вода: МосКВА — аКВА.

водяного пара, сделанными по технологии Mathcad Application/Calculation Server (см. стр. 9) и Elsevier/Knowell Interactive Calculation (см. стр. 21). На рисунке 23.2 отображен один из таких интернет-расчетов, по которому можно определить плотность морской воды в зависимости от ее температуры, давления и солености.

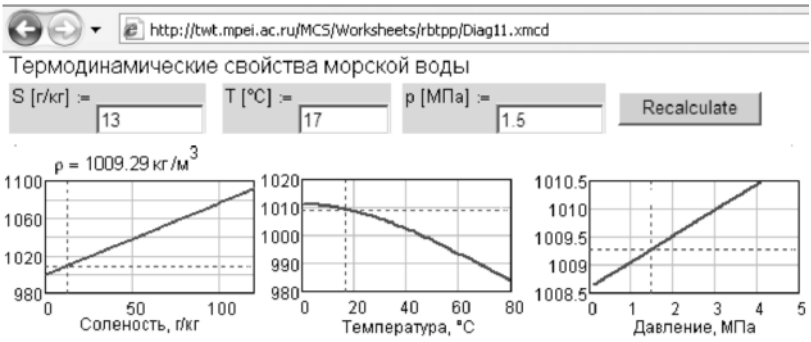


Рис. 23.2
Сайт, на котором можно оценить плотность морской воды

Если мы в Черном море, один килограмм воды которого содержит 13 г солей, погрузимся на глубину 150 м, где давление составит 1.5 МПа (примерно 15 атмосфер), а температура будет равна, допустим, 17 градусов по шкале Цельсия, то плотность воды там окажется равной 1009.29 килограмм на кубический метр. Но мы можем вполне спокойно сказать, что плотность воды там будет иметь порядок 1010 кг/м^3, учитывая погрешности измерения и методики расчета плотности[3]. Кроме того, соленость воды определить довольно сложно, если принять во внимание тот факт, что состав воды в разных морях, океанах и соленых озерах разный. Сайт на рисунке 23.2 не только выдает нужное значение плотности воды, но и рисует три изолинии, показывающие, как будет меняться плотность воды, если менять один параметр (температуру, например) и оставлять постоянными два других (давление и соленость воды). Для чистой (пресной) воды, термодинамические свойства которой определяются только двумя параметрами — давлением и температурой[4], можно построить поверхность или семейство кривых. Но морская вода требует три отдельных декартовых графика. Мы такой прием уже использовали, когда в этюде 14 раскладывали функцию четырех аргументов на четыре отдельных плоских графика.

Интернет-расчет, показанный на рисунке 23.2, вспомнился автору, когда он сидел на банкете в стокгольмском музее у борта этого незадачливого судна (рис. 23.1). И вот почему.

По одной из версий[5], опрокидывание корабля «Ваза» произошло из-за того, что не было принято во внимание изменение плотности воды при выходе корабля из пресноводного залива, где была расположена верфь, в соленое открытое море. Бо-

[3] В трамвае висит табличка: «Мест для стояния 229». Сразу возникает вопрос, как определили так точно число этих мест. Можно было спокойно округлить 229 до 230. Что вы скажите, если электронный термометр покажет вам вашу температуру: 36.589, а не 36.6?!

[4] Правильнее с точки зрения физики говорить «температура и давление», а не «давление и температура», так как температура в первую очередь определяет свойство воды, а не давление. Но тут главенствуют законы языка, а не теплофизики: «температура и давление» труднее выговаривать чем «давление и температура».

[5] Есть филологическая версия этого кораблекрушения. Для европейского уха «Ваза» — это в первую очередь не королевская династия и не корабль, названный в ее честь, а... сосуд для хранения чего-то. Одна из разновидностей вазы — это тазик, только сделанный не из металла, а из стекла или фарфора: ваза не для цветов, а для фруктов. Вспомним название этюда и английскую народную песенку в переводе С. Маршака: «Три мудреца в одном тазу пустились по морю в грозу...» Истинно сказано: как корабль назовешь, так он и поплывет.

лее плотная вода приподняла корабль, порыв ветра накренил его, и вода через открытые пушечные люки хлынула в трюмы...

Так это было или иначе, можно спорить, но то, что тут проявила себя некая инженерная ошибка, связанная с неправильными расчетами, — факт бесспорный.

Давайте посмотрим, как можно применить пакет Mathcad для оценки нашей гипотезы [85].

В качестве модели судна возьмем деревянный брусок в виде параллелепипеда со сторонами 10 мм (ширина «палубы» — переменная *ш*), 30 мм (высота судна — *в*) и 250 мм (длина — *д*: рис. 23.3, слева). Это упрощение не так уж далеко от жизни: реальные современные торговые суда имеют почти прямоугольные центральные поперечные сечения (см. рис. 23.3, справа), и только нос и корма сделаны заостренными для уменьшения сопротивления воды при движении. «Прямоугольный» корабль проще и дешевле строить, а потом заполнять его грузом — контейнерами, например, которые также имеют форму параллелепипеда.

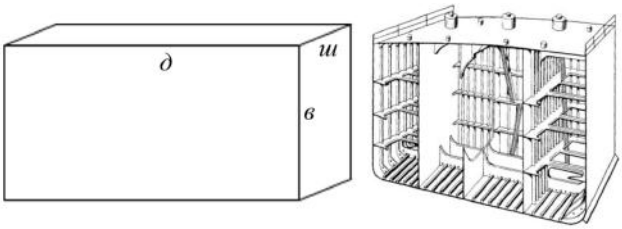


Рис. 23.3

Модель судна и центральная секция реального судна

Поместим нашу модель судна в воду таким образом, чтобы стороны 10 мм (*ш*) и 250 мм (*д*) образовывали палубу и днище нашего «судна», а сторона 30 мм (*в*) — высоту от днища до палубы. Жизненный опыт подсказывает, что такое положение деревянного бруска в воде не будет устойчивым (см. рис. 23.4, слева); сила $F_Т$ (сила тяжести, приложенная к центру масс[6]) и сила $F_А$ (выталкивающая сила Архимеда, приложенная к центру водоизмещения, к центру масс объема жидкости, вытесненной плавающим телом) развернут наше «судно», и оно ляжет набок.

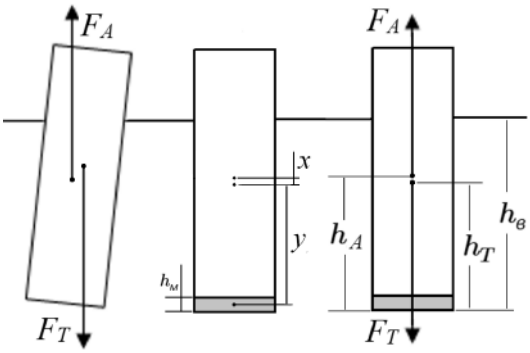


Рис. 23.4

Силы, действующие на «судно» (слева — без балласта, центр и справа — с балластом)

[6] Тут правильнее говорить не о центре масс, а о центре тяжести. Но в однородном гравитационном поле центр масс (геометрическая точка, характеризующая движение тела или системы частиц как целого) и центр тяжести (геометрическая точка, через которую проходит равнодействующая всех сил тяжести, действующих на частицы тела при любом положении последнего в пространстве) совпадают. В этюде 9 мы работали с неоднородным гравитационным полем, когда рассматривали движение спутников вокруг планет. Когда же мы приблизились к земле, то стали допускать, что это поле однородное.

Объясняется это тем, что центр масс оказался выше центра водоизмещения, и при любом малом отклонении такого «судна» от положения равновесия оно под действием пары сил перевернется. Но это условие необходимое, но не достаточное, и мы это отметим ниже.

Для снижения точки расположения центра масс давайте немного сострогаем рубанком днище нашего «судна» и прикрепим туда металлическую полоску толщиной $h_м$. Это будет тот самый *балласт*, который должен удержать «судно» от опрокидывания (см. рис. 23.4, центр и справа). Парусные военные корабли строились высокими, чтобы пушки стреляли дальше и сподручнее было брать на абордаж вражеские корабли. Без довольно массивного балласта такой корабль не удержался бы на плаву даже без поднятых парусов.

Проанализируем устойчивость (*остойчивость*, как говорят профессионалы [92]) нашего «судна» с помощью пакета Mathcad (рис. 23.5).

$$\left\| \begin{array}{l} \text{“Вес вытесненной воды”} \\ \left\| \begin{array}{l} \text{“Объем подводной части”} \\ h_в \cdot ш \cdot д \end{array} \right| \cdot \rho_в \cdot g \end{array} \right| = \left\| \begin{array}{l} \text{“Вес судна”} \\ m \cdot g \end{array} \right| \xrightarrow{solve,\, h_в} \frac{m}{ш \cdot д \cdot \rho_в}$$

$$\begin{bmatrix} x \cdot m_д = y \cdot m_м \\ h_T = \dfrac{h_м}{2} + y \\ h_м + \dfrac{в - h_м}{2} = \dfrac{h_м}{2} + x + y \end{bmatrix} \xrightarrow{solve,\, \begin{bmatrix} x \\ y \\ h_T \end{bmatrix}} \begin{bmatrix} \dfrac{в \cdot m_м}{2 \cdot m_д + 2 \cdot m_м} & \dfrac{в \cdot m_д}{2 \cdot m_д + 2 \cdot m_м} & \dfrac{в \cdot m_д + m_д \cdot h_м + h_м \cdot m_м}{2 \cdot m_д + 2 \cdot m_м} \end{bmatrix}$$

Рис. 23.5

Вывод формул для расчета высоты центра водоизмещения ($h_в$) и центра масс (h_T)

Высоту центра водоизмещения (центра объема жидкости, вытесненной плавающим телом) определить несложно через закон Архимеда — см. верхний оператор на рисунке 23.5, где уравнение (вес вытесненной воды равен весу судна) решается относительно переменной $h_в$. Для определения центра масс нужно решить уже не одно уравнение, а целых три — см. нижний оператор на рисунке 23.5. Вот эти уравнения:

- произведение массы деревянной части судна $m_д$ на расстояние от ее центра до центра масс всего судна (x) равно произведению массы металлической части судна $m_м$ на расстояние от ее центра до центра масс всего судна (y) — см. рис. 23.4, центр;
- высота центра масс судна $h_т$ равна половине высоты балласта $h_м$ плюс значение y;
- высота центра масс деревянной части судна $h_д$ вычисляется двумя способами: $h_м + (в - h_м) / 2$ и $h_м / 2 + x + y$.

Мы могли бы формулу для h_T вывести вручную, но, имея под рукой пакет Mathcad, можно автоматизировать эту работу, составив балансы крутящих моментов (первое уравнение) и расстояний (два вторых уравнения).

На рисунке 23.6 показан расчет остойчивости судна с балластом в пресной воде ($\rho = 1$ г/см^3) с использованием формулы для $h_т$, выведенной на рисунке 23.5. Высота $h_А$ (13.55 мм) оказалась больше высоты $h_т$ (13.313 мм). Наше «судно» с балластом будет держаться вертикально — пара сил будет возвращать его в исходное положение при небольших отклонениях от вертикали. Но это равновесие неустойчивое (локальная потенциальная яма) — при сильном отклонении судна от вертикали оно все-таки ляжет набок.

$ш := 10\ mm \qquad в := 30\ mm \qquad д := 250\ mm \qquad h_м := 0.5\ mm$

$\rho_в := 1.0\ \frac{gm}{cm^3}$ Плотность воды

$\rho_д := 0.8\ \frac{gm}{cm^3}$ Плотность дерева

$\rho_м := 7\ \frac{gm}{cm^3}$ Плотность металла

$m_м := ш \cdot д \cdot h_м \cdot \rho_м = 8.75\ gm \qquad m_д := ш \cdot д \cdot (в - h_м) \cdot \rho_д = 59\ gm$

$m := m_м + m_д = 67.75\ gm$

$h_в := \frac{m}{ш \cdot д \cdot \rho_в} = 27.1\ mm$

$h_T := \frac{в \cdot m_д + m_д \cdot h_м + h_м \cdot m_м}{2 \cdot m_д + 2 \cdot m_м} = 13.313\ mm \qquad h_A := \frac{h_в}{2} = 13.55\ mm$

Рис. 23.6

Расчет остойчивости судна с балластом в пресной воде

Давайте будем постепенно повышать плотность воды (выводить наше «судно» из пресноводной гавани в открытое море) и смотреть, что с ним произойдет. Для этого введем в расчет функцию с именем Δ (разность $h_А$ и $h_Т$) и с аргументом $\rho_в$, построим ее график и рассчитаем значение плотности воды, где эта функция поменяет знак — где силы $h_А$ и $h_Т$ начнут переворачивать наше «судно» (рис. 23.7).

$\Delta(\rho_в) := \frac{\frac{m}{ш \cdot д \cdot \rho_в}}{2} - h_T \qquad \rho_в := 1\ \frac{gm}{cm^3}, 1.0001\ \frac{gm}{cm^3} .. 1.03\ \frac{gm}{cm^3}$

$\rho_{в_кр} := \mathrm{root}\left(\Delta(\rho_в), \rho_в, 1\ \frac{gm}{cm^3}, 1.03\ \frac{gm}{cm^3}\right) = 1.018\ \frac{gm}{cm^3}$

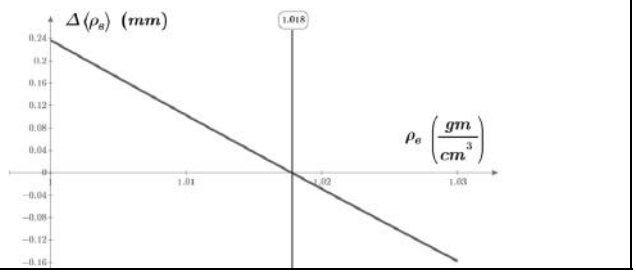


Рис. 23.7

График остойчивости судна в зависимости от плотности воды

Расчет на рисунке 23.7 показывает, что критическая плотность воды для нашего «судна» составляет 1.018 г на мм. По левому графику на рис. 23.2 можно оценить соленость воды при такой ее плотности.

Есть легенда о том, как древние русичи спаслись от набега викингов — предков строителей корабля «Ваза». Когда эти морские разбойники поднялись по реке на своих ладьях к поселению русичей и в полном вооружении приготовились к высадке на берег, русичи стали высыпать в воду из мешков самое дорогое, что у них было, — соль. Плотность воды в реке увеличилась, ладьи викингов потеряли устойчивость и перевернулись. Викинги частью потонули, а частью были взяты в плен.

Эту легенду авторы придумали при работе над книгой. Но она с точки зрения физики и математики ничем не хуже древнегреческой легенды о том, как был сожжен вражеский флот с помощью зеркал, сфокусировавших солнечные лучи на деревянных кораблях [82].

Но если говорить серьезно, то можно предложить такую **лабораторную работу**.

Изготавливается «судно» в виде линейки, сложенной из деревянной и металлической реек (см. рис. 23.4, центр и справа). Такой «линейный корабль» спуска-

ется на воду — опускается в емкость с водой. При этом «судно» спроектировано так (см. рис. 23.6), что оно устойчиво держится вертикально в пресной воде. Затем в воду начинают подсыпать поваренную соль, аккуратно перемешивают воду и следят, когда «судно» начнет... заваливаться набок.

Затем можно будет нагревать этот водный раствор и смотреть, как судно будет возвращаться к первоначальному вертикальному положению за счет того, что водный раствор с более высокой температурой имеет меньшую плотность (см. центральный график на рис. 23.2). Все это можно дополнить расчетами на компьютере остойчивости судна (решение алгебраических уравнений — см. рис. 23.6 и 23.7) и процессов тепло- и массопереноса (решение дифференциальных уравнений [60]). Форму судна можно делать более сложной, приближенной к реальной и т. д. Это будет хорошей учебной темой, связывающей на школьных занятиях математику, физику, химию и информатику. На уроке можно попытаться создать анимацию переворачивания судна [44] или его раскачивания с рисованием векторов действующих на судно сил [33]...

Конечно, расчет остойчивости реальных кораблей существенно сложнее. Это очень красивая инженерная задача, к решению которой приложили руку почти все великие математики. Нам сегодня остается только восхищаться, как они могли это делать без компьютеров. Кстати, на современных судах компьютеры управляют специальными «плавниками» и другими устройствами, снижающими качку даже при сильном волнении моря и предотвращающими опасный крен судна. Удивительно, но у реального корабля центр водоизмещения находится все-таки ниже центра масс, но его устойчивость создается в том числе и сложной формой подводной части корабля, компенсирующей отклонение центра масс смещением места расположения центра водоизмещения. Простейший пример. Если наше судно-брусок без балласта (рис. 23.4, слева) ляжет набок, то несложно подсчитать, что центр масс по-прежнему будет выше центра приложения архимедовых сил. Но при наклоне такого судна (вернее, плота) от бокового ветра, например, центр приложения архимедовых сил (центр водоизмещения) будет смещаться так, чтобы вернуть судно в горизонтальное положение — в глобальную потенциальную яму.

Кстати, наше судно-линейка («линейный корабль») — это своеобразный *ареометр*, прибор для измерения плотности жидкости, принцип работы которого основан на законе Архимеда. Ареометр представляет собой стеклянную трубку, нижняя часть которой при калибровке заполняется дробью или ртутью (балластом) для достижения необходимой массы. В верхней части трубки находится шкала, которая проградуирована в значениях плотности раствора или концентрации растворенного вещества[7]. Подобную «шкалу» можно видеть и по бортам судов для контроля его загрузки. Суда класса «река-море» имеют несколько таких шкал. Верхнюю риску на этой шкале называют *грузовой маркой* (load line), которая не должна погружаться в воду при загрузке судна в морском или речном порту. Иначе судно может перевернуться. И таких случаев, увы, было множество.

Перефразируя А. П. Чехова, можно сказать, что в корабле все должно быть прекрасно — и силуэт, и название и... расчеты, по которым он спроектирован. Такой небольшой, но поучительный расчет мы только что провели.

ДИВЕРТИСМЕНТ 1. ЦЕНТР МАСС, ЦЕНТР ТЯЖЕСТИ...

Как вычислить центр масс не прямоугольника (см. рис. 23.4), а более сложной плоской фигуры? Этот вопрос мы уже задавали в этюде 21. Если в Интернете сде-

[7] См. пример табличной и графической зависимости плотности и концентрации в этюде 2 на рис. 2.3 (стр. 43). В фильме «Гангстеры и филантропы» рассказано об одном химике, которого уволили с работы. Он зашел в кафе, заказал стакан водки и стал машинально помешивать ее... ареометром, который случайно оказался в нагрудном кармане его пиджака. Служащие кафе решили, что это ревизор, проверяющий крепость спиртного, и дали ему взятку. Этот химик прободрился и стал так «химичить» в других заведениях общепита...

лать соответствующий запрос, то будут выданы либо формулы для простых фигур (треугольник, полукруг, сектор или сегмент круга и т. д.), либо «страшная» формула с двойным интегралом, по которой практически невозможно сделать расчет по конкретному случаю. А вот он такой конкретный случай.

Фигура (допустим, что это поперечное сечение судна) имеет форму некой дуги — полуокружности с наружным радиусом 1 м и внутренним 0.9 м — см. рис. 23.8. Необходимо определить координаты центра масс такой фигуры. Можно поступить как в дивертисменте этюда 21 — превратить дугу в прямоугольную матрицу с нулями и единицами и далее обработать ее.

Но давайте поступим не как в этюде 21, а несколько иначе — равномерно заполним нашу фигуру множеством точек и определим ординату точки, суммарное расстояние от которой до остальных точек будем минимально. Будет ли эта точка центром масс — это отдельный вопрос. Ответить на него мы поручаем читателю. Что-то подобное мы проделывали в этюде 7, когда определяли крутящий момент сил у провисающей цепи (см. рис. 7.14 и 7.15 на стр. 151). Если этот параметр поделить на вес цепи, по мы получим координату центра тяжести.

На рисунке 23.8 сделан расчет центра масс методом, который можно назвать методом Монте-Карло. Метод Монте-Карло, кстати, часто применяется при интегрировании — при определении, например, площадей сложных фигур.

```
ORIGIN := 1   n := 10^5   X := runif(n, −1, 1) m   Y := runif(n, −1, 0) m

(X  Y) := | j ← 1
          | for  i ∈ 1 .. n
          |   if  (X_i)^2 + (Y_i)^2 ≤ (1m)^2 ∧ (X_i)^2 + (Y_i)^2 ≥ (90cm)^2
          |     | (XX_j ← X_i   YY_j ← Y_i)
          |     | j ← j + 1
          | (XX  YY)

n_1 := last(X) = 15101

M(y) := | M ← 0
        | for  i ∈ 1 .. n_1
        |   M ← M + √((X_i − 0m)^2 + (Y_i − y)^2)
        | M

y := −0.5     y := Minimize(M, y) = −750·mm
```

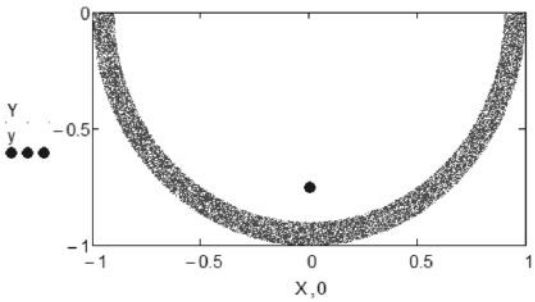


Рис. 23.8
Приближенное вычисление центра масс плоской фигуры

Мы берем прямоугольник размером 1 м на 2 м и наполняем его n точками со случайными координатами. Такую работу мы проделывали в этюде 16, когда вычисляли число π методом Монте-Карло — см. рис. 16.3 на стр. 360. Можно также

заполнить прямоугольник точками, расположенными в узлах равномерной сетки. Затем мы массив точек фильтруем — оставляем в векторах X и Y только те точки, которые попали в «днище судна» — в полуокружность толщиной 10 см. Эти точки и отображены на графике рис. 23.8, формируя заданную фигуру. В оператор if можно будет добавлять новые условия, формируя сложную фигуру и следя за правильностью выкладок, глядя на точки графика. Исходных точек в нашей конкретной задаче было 10 000 (n), а в «днище» попала 15 101 точка (n_1). Далее создается целевая функция M(y), возвращающая суммарное расстояние от точки с координатами (0, y) до остальных точек. Наша фигура на рис. 23.8 симметрична относительно центральной вертикальной оси, и ясно, что искомый центр масс должен находиться на этой оси. Но в общем случае функция с именем M должна иметь два аргумента x и y, а в ее формуле вместо нуля должна стоять переменная (независимый параметр оптимизации) x. Если же задача не плоская, а объемная, то функция M будет иметь три аргумента: x, y и z, а далее все делается так, как описано здесь — хватило бы только памяти и быстродействия компьютера. Далее у функции M(y) численно ищется минимум вблизи заданного значения (0, –0.5). Ответ: центр масс нашей утолщенной полуокружности отстоит от ее центра на 0.75 м — см. точку на графике рисунка 25.8. Как мы уже писали в этюде 19, вспоминая одного известного киноперсонажа: «Это не эстетично, зато дешево, надежно и практично». А точно ли мы сделали данное определение? Проверим это на фигуре, координаты центра масс которой известно.

На рисунке 23.9 показан расчет центра масс полусферы. Она получается, когда, например, нужно определить центр приложения архимедовых сил к нашему «кораблю» (см. рис. 23.8), максимально погрузив его в воду. Для полусферы известна точная формула расчета центра масс (см. выражение для переменной y_t). Точность нашего приближенного расчета оказалась удовлетворительной — в пределах долей процента (5 мм — см. также точку и кружок на рис. 23.9).

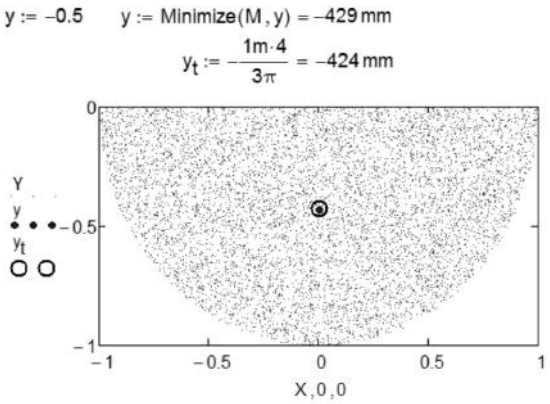


Рис. 23.9

Проверка приближенного вычисления центра масс плоской фигуры

А вот еще одна лабораторная работа на стыке физики и информатики. Берется плоская сплошная фигура сложной формы, вырезанная из картона. В ней сверлится дырочка, через которую она вешается на гвоздик. На этот же гвоздик вешается грузик на нитке (отвес). По нитке проводится прямая черта на фигуре (вариант: все это фотографируется, отправляется в компьютер и оцифровывается). Далее все действия повторяются для другой просверленной дырочки. Можно все это сделать и для третьей, четвертой дырочки, то это будет лишним. Точка пересечения этих двух прямых линий и будет центром тяжести (центром масс) нашей плоской фигуры. Анимация таких манипуляций расположена, например, в статье

«Центр масс» Википедии. Это свойство связано с тем, что потенциальная энергия свободно подвешенного тела будет минимизироваться — тело будет стараться расположиться как можно ниже. Это явление мы рассмотрели в этюде 21, когда рассчитывали форму провисающей нити с грузиками (см. рис. 21.1 на рис. 422). Далее реальная картонная фигура с двумя прямыми начерченными линиями фотографируется, оцифровывается с получением массива координат ее точек, по которым находится центр масс по вышеописанной компьютерной методике. Заканчивается лабораторная работа сравнением координат двух полученных точек из физического и компьютерного экспериментов. Облегченный вариант лабораторной работы. Рисуется на экране плоская фигура, у которой с помощью вышеописанной методики определяется центр масс. Далее эта фигура распечатывается на плотной бумаге и вырезается из бумаги ножницами. У вырезки определяется центр тяжести, двукратным подвешиванием ее с отвесом на гвоздике. Проверяется совпадение рассчитанной и определенной опытном путем точек.

ДИВЕРТИСМЕНТ 2.
РЫБА ИЩЕТ ГДЕ ГЛУБЖЕ, А ЧЕЛОВЕК... В ИНТЕРНЕТЕ

А почему рыба так поступает? Одна из причин такого поведения рыб в том, что в стоячем водоеме, покрытом льдом, самое «теплое» место — это дно. Это можно показать, построив график изменения плотности воды от температуры. Саму же зависимость можно найти в Интернете — в «облаке», непосредственно подсоединив ее к Mathcad-расчету. В этюде 1 мы уже делали это, рассчитывая ускорение свободного падения в зависимости от высоты над уровнем моря и географической широты — см. рис. 1.27 на стр. 37. Давайте теперь опустимся ниже «уровня моря» и рассчитаем там температуру.

Если сделать ссылку на «облачную» функцию из списка на рис. 1.29 (стр. 39), то несложно построить график изменения плотности воды в зависимости от температуры при атмосферном давлении (рис. 23.10) и еще раз убедится в том, что зимой на дне водоема, покрытого льдом, температура будет всегда равна плюс четырем градусам по шкале Цельсия. Если, конечно, водоем не промерзнет до дна.

Reference:http://twt.mpei.ac.ru/rbtpp/MC-WSP/M15/wspD1PT.xmcdz

$t := 0\,°C, 0.1\,°C .. 7\,°C$

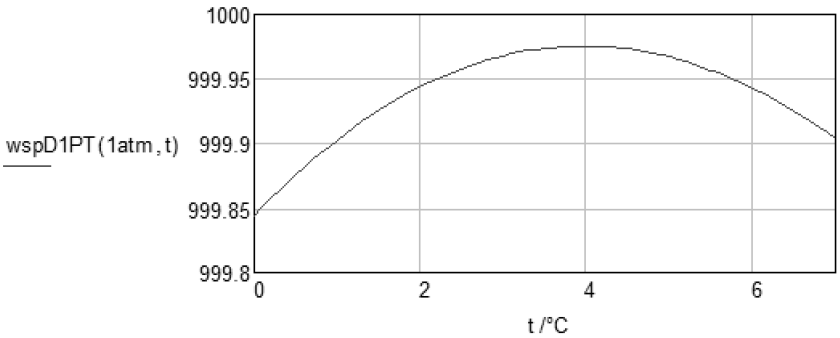


Рис. 23.10
Плотность воды и температура

Задание читателям

1. Написать программу определения центра масс тел сложной формы, состоящей из простых элементов с известной формулой.

2. Рассчитать, насколько пушка корабля будет дальше стрелять, если поднять ее с нижней палубы на верхнюю. Подсказка в этюде 7.

3. Попытаться доказать обоснованность и погрешность метода расчета центра масс, показанного на рис. 23.8 и 23.9.

4. Рассчитать остойчивость судна, сечение которого показано на рис. 23.8, используя разные материалы корпуса (сталь, дерево, пенопласт и проч.) и разные жидкости (ртуть, глицерин, вода с разной соленостью и проч.). Показать, как такое судно будет вести себя, если его накренить.

5. Определить центр масс плоской «фигуры», с которой мы работали в этюде 5 — «фигуры» Че Гевары (см. рис. 5.1 на стр. 106). Сфотографировать своих друзей и знакомых и определить их центр «масс». Считают, что он должен находиться в районе пупка (см. пупок у покемона на рис. 19.7 (стр. 382), но так ли это?

6. Оцените плотность морской воды на дне Марианской впадины с учетом и без учета сжимаемости воды.

7. Вспомните о подводной лодке, например о «Наутилусе» (см. рис. 6.15 на стр. 135), и рассчитайте процесс ее погружения и всплытия за счет заполнения и продувки танков.

Этюд 24

РАСЧЕТЫ НА ~~САЛФ~~ТАБЛЕТКЕ, ИЛИ ...И НА МАРСЕ БУДУТ ЯБЛОНИ ЦВЕСТИ

В этюде рассмотрены вопросы движения ракеты-носителя: одно- и двухступенчатой. Рассказано о формуле Циолковского, о выборе места для космодрома.

Математика: решение алгебраического и дифференциального уравнения.

Физика: движение ракеты, формула Циолковского, теории относительности Эйнштейна, ускорение свободного падения, первая космическая скорость, вращение Земли вокруг своей оси.

Информатика: решение уравнения, оператор выбора.

Сайт этюда: **https://www.ptcusercommunity.com/thread/472716**

Есть такой советский культовый фильм «Девять дней одного года»[1]. В нем нет погонь и драк, на которые «подсели» современные юные и уже не совсем юные зрители, а есть прекрасные диалоги в исполнении прекрасных актеров: Татьяны Лавровой, Иннокентия Смоктуновского, Алексея Баталова и др.

Рис. 24.1
Кадр из фильма «Девять дней одного года»

Вот один такой диалог из фильма (рис. 24.1): актеры Евгений Евстигнеев (физик-теоретик Николай Иванович — слева) и Михаил Казаков (физик-романтик Валерий Иванович — справа):

— Скажите, Валерий Иванович, как глубоко вы собираетесь забраться в глубины нашей галактики?

— На 500 световых лет.

— С какой скоростью?

— Близкой к скорости света.

— Вес корабля?

— Сто тысяч тонн.

[1] Его можно посмотреть на сайте: **http://kino-ussr.ru/main/217-devyat-dney-odnogo-goda-1961.html**. Второе название этюда — это, кстати говоря, строка из знаменитой песни тех времен. Ее распевали на улицах, когда встречали космонавтов после очередной космической экспедиции: «Утверждают космонавты и мечтатели, что...» Прослушать песню можно здесь **http://pesnifilm.ru/load/mechte_navstrechu/i_na_marse_budut_jabloni_cvesti_quot_mechte_navstrechu_quot/833-1-0-1846**

— Горючее?

— Самое современное.

— Сейчас мы подсчитаем, сколько вам понадобится горючего. Будьте добры, салфеточку.

...

— Николай Иванович, вы закончили свои подсчеты?

— Да, пожалуйста. При весе космического корабля сто тысяч тонн при скорости, близкой к скорости света, для облета части галактики в разумный для человеческой жизни срок вам потребуется десять в двадцать второй степени тонн самого современного экстра-горючего. Справка. Наша планета весит несколько меньше, счастливого пути!

Диалог, конечно, наивный, но представим себе, что у Николая Ивановича в руках была не салфетка, а... «таблетка» (планшетный компьютер — см. первое название этюда) с программой Mathcad, и он на ней делает расчеты (рис. 24.2).

$$v_{end} = I \cdot \ln\left(\frac{M_1}{M_2}\right) \xrightarrow{solve, M_1} M_2 \cdot e^{\frac{v_{end}}{I}}$$

$$M_2 := 100 \cdot 1000 \text{ т}$$

$$v_{end} := 0.995\ c = 298\ 1000 \text{ км/с}$$

$$I := 777 \cdot 1000 \frac{\text{тс}}{\text{т/с}}$$

$$M := M_2 \cdot \exp\left(\frac{v_{end}}{I}\right) - M_2 = 1.003 \cdot 10^{22} \text{ т}$$

$$M_{Earth} := 5.97219 \cdot 10^{21} \text{ т}$$

Рис. 24.2

Предположение расчета на салфетке

В расчете на рисунке 24.2 использована *формула Циолковского* (кстати, сам Циолковский упоминается в фильме[2]): конечная скорость ракеты v_{end} пропорциональна натуральному логарифму отношения стартовой массы ракеты M_1 к ее конечной массе M_2. Разность же $M_1 - M_2$ — это масса топлива и окислителя (далее — просто топлива, как в фильме). Коэффициент пропорциональности в этом уравнении (величина I) — это отношение тяги двигателя F к массовому расходу топлива μ (величины F и μ мы будем использовать в дальнейших расчетах).

Мы приняли I равным 777 тысяч тонн силы/(тонн/с). Но в самом фильме эта цифра не озвучена, 777 тысяч — это наше предположение (три семерки на счастье!) Величина I — это также и скорость истечения газов из сопла ракеты (метры в секунду). Теоретическое значение этой скорости для ядерных ракетных двигателей может превышать 70 км/с. Скорость истечения для электрического двигателя может достигать 140 км/с. Поэтому вполне можно помечтать и о «трех семерках».

Уравнение Циолковского на рисунке 24.2 решается относительно переменной M_1. Если конечная скорость ракеты будет близка к скорости света (мы приняли, что эта скорость равна 0.995 скорости света c), то масса топлива и вправду превысит массу Земли (M_{Earth}). Но расчет наш, конечно, очень грубый («наивный»). Кроме того, он не учитывает то важное обстоятельство, что при скоростях, близких к

[2] В конце спора персонаж Казакова (Валерий Иванович) в конце диалога бросает своему оппоненту (персонажу Евгения Евстигнеева) такую фразу: «Когда Циолковский проектировал свою ракету, то в ресторане „Яр“ сидели ученые-скептики вроде вас и на салфеточках доказывали, что он сумасшедший».

скорости света, формула Циолковского не работает. Тут нужно будет отходить от законов классической механики и прибегать к теории относительности Эйнштейна.

Но давайте спустимся с галактических высот к околоземному пространству, немного (в сотню раз) притормозим ракету и подсчитаем, какие параметры могут быть у реальных, а не у полуфантастических космических аппаратов. Мы подобный расчет уже делали в этюде 9 — см. рис. 9.14 на стр. 213 и рис. 9.37 на стр. 236, но для одноступенчатых ракет.

На рисунке 24.3 сделана попытка расчета скорости ракеты, имеющей стартовую массу 250 т, из которых 10 т — это полезная нагрузка (P), 40 т — это масса пустой ракеты-носителя (m), а 200 т — масса топлива (M). Тяга двигателя ракеты (F) составляет 2000 тонн силы (тс) при расходе топлива 5 т в секунду (т/с — μ). При таких исходных данных двигатель будет работать 40 с (t_{end}).

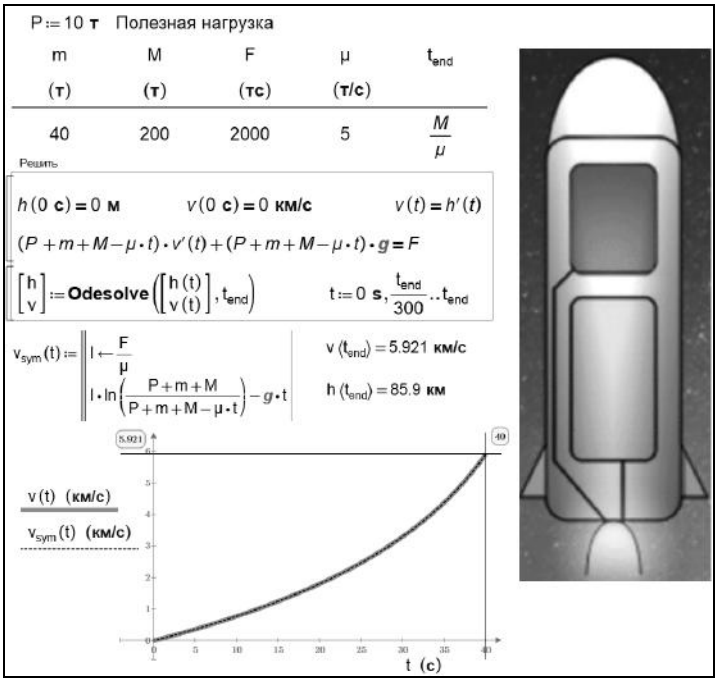


Рис. 24.3
Расчет одноступенчатой ракеты

На рисунке 24.3 показано численное решение дифференциального уравнения: двигатель толкает ракету вверх, а вниз ракету тянет ее вес. И все это уравновешивается произведением массы ракеты на ее ускорение — на первую производную ее скорости по времени.

Численное решение дифференциального уравнения движения ракеты — сгенерированная по точкам функция $v(t)$, сравнивается на графике с аналитическим решением (с функцией $v_{sym}(t)$), найденным с помощью пакета Maple (рис. 24.4). Этим действием мы фактически вывели формулу Циолковского, использованную в решении на рисунке 24.2, дополнительно вставив в нее учет ускорения свободного падения (веса ракеты). Но константа g — это не константа, а переменная величина, и об этом мы уже писали ранее (см. в этюде 1 рис. 1.27 на стр. 37) и еще поговорим ниже.

Примечание. На графике на рисунке 24.3 прорисована не одна, а две кривые: толстая сплошная бледная и тонкая пунктирная черная внутри толстой кривой. Так проверяет-

ся совпадение численного и аналитического (символьного) решений дифференциального уравнения.

Если в уравнении вес ракеты $(P + m + M - \mu \cdot t) \cdot g$ перенести в правую сторону со знаком «минус», то получится второй закон Ньютона в чистом виде. Математически, конечно же, это ничего не изменит, но будет нагляднее.

Диф_уравнение := $(P + m + M - \mu \cdot t) \cdot (v'(t) + g) = I \cdot \mu$

Начальные_условия := $v(0) = 0$

dsolve({Диф_уравнение, Начальные_условия})

$$v(t) = I \cdot \ln\left(\frac{M + P + m}{M + P + m - \mu\, t}\right) - g \cdot t$$

Рис. 24.4
Вывод уравнения Циолковского

Из расчета на рисунке 24.3 видно, что скорость ракеты в конце работы двигателя (5.921 км/с) не достигает первой космической скорости (7.9 км/с): ракета после ее разворота на 90° (это делают дополнительные двигатели или гироскопы) полетит не по околоземной, а по баллистической орбите и скоро упадет на землю.

Чтобы все-таки вывести наши 10 т полезной нагрузки на околоземную орбиту, можно увеличить запас топлива. Но лучше разделить ракету на две ступени (рис. 24.5).

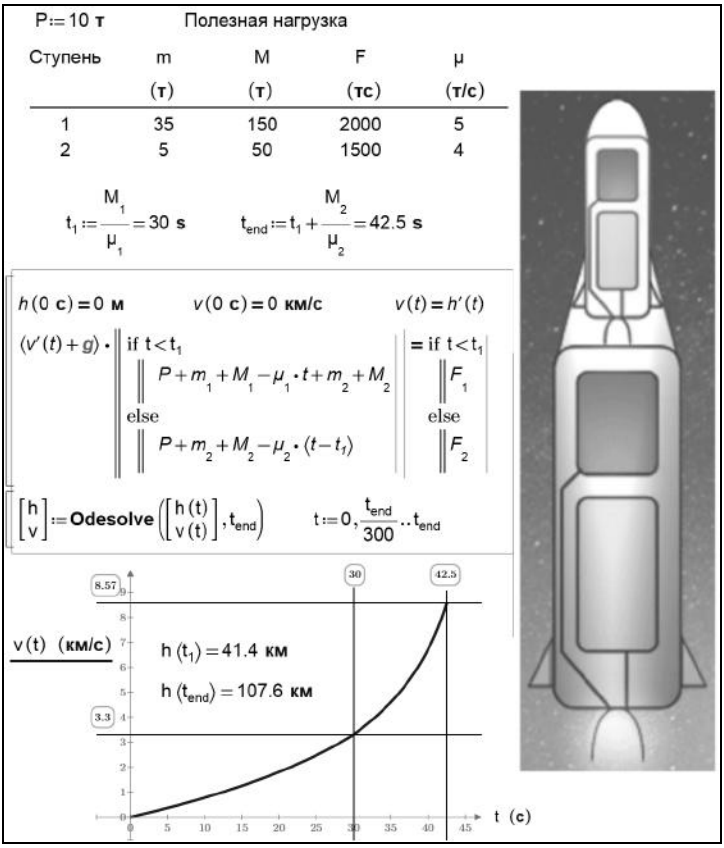


Рис. 24.5
Расчет полета двухступенчатой ракеты

Дифференциальное уравнение полета двухступенчатой ракеты будет содержать встроенные в Mathcad операторы `if`: если работает первая ступень с тягой F_1, то она несет полезную нагрузку P, сами две ступени (m_1 и m_2) — топливо второй ступени M_2 и расходуемое топливо первой ступени $M_1 - м_1 \cdot t$. Вторая же ступень (оператор `else`) с тягой F_2 будет нести только полезную нагрузку P, себя саму m_2 и расходуемое топливо $M_2 - м_2 \cdot (t - t_1)$, где t_1 — время работы первой ступени, а t — общее время полета ракеты. При таком раскладе скорость ракеты в конце последовательной работы двух ступеней (8.57 км/с) превысит первую космическую скорость и спутник будет выведен на орбиту. При этом ракета (вернее, ее вторая ступень) поднимается на 107,6 км, а не на 85,9 км (одноступенчатый вариант).

Кривые, показанные на рисунках 24.3 и 24.5, похожи на... верхнюю образующую и ребра монумента покорителям космоса, установленного в Москве (рис. 24.6).

Рис. 24.6
Монумент покорителям космоса

Интересно, заложена ли формула Циолковского в этот монумент или это случайное совпадение? Но сам Циолковский в виде памятника присутствует у подножия монумента.

В расчетах на рисунках 24.3–24.5 сделано, конечно, много допущений[3]: не учитывается действие на ракету потока воздуха, изменение величины ускорения свободного падения в зависимости от высоты и от географической широты. Кстати, о широте. Известно, что пуск ракет с Земли нужно осуществлять как можно ближе к экватору и разворачивать орбитальную станцию в восточном направлении. Во-первых, у экватора значение g меньше, чем на более высоких широтах, и, во-вторых и главных, ближе к экватору ракета получает бульший дополнительный разгон за счет вращения Земли.

На рисунке 24.7 показано, какое значение дополнительной горизонтальной скорости получит ракета при запуске ее в восточном направлении из разных точек Земли — с пяти различных космодромов, а также с полюса и экватора. Если посмотреть на карте места расположения этих космодромов, то окажется, что восточнее их находятся места для падения отработавших ступеней ракет — океан или малонаселенная территория.

[3] В принципе, можно было не учитывать и вес ракеты — произведение ее массы на ускорение свободного падения. Это допущение, как мы уже отметили, заложено в формулу Циолковского.

$R := 6370\ km \qquad v(\phi) := \dfrac{2\ \pi\ R \cdot \cos(\phi)}{day}$

$Полюс := 90\ ° \qquad v(Полюс) = 0\ \dfrac{km}{s}$

$Плесецк := 63\ ° \qquad v(Плесецк) = 0.21\ \dfrac{km}{s}$

$Байконур := 46\ ° \qquad v(Байконур) = 0.322\ \dfrac{km}{s}$

$Цзюцюань := 41\ ° \qquad v(Цзюцюань) = 0.35\ \dfrac{km}{s}$

$Мыс_Канаверал := 28.5\ ° \qquad v(Мыс_Канаверал) = 0.407\ \dfrac{km}{s}$

$Куру := 5\ ° \qquad v(Куру) = 0.461\ \dfrac{km}{s}$

$Экватор := 0\ ° \qquad v(Экватор) = 0.463\ \dfrac{km}{s}$

Рис. 24.7
Линейные скорости на поверхности Земли

Есть еще один, «астрономический» фактор: плоскости орбит, не совпадающие с экваториальными, неустойчивы — возмущающее действие Земли, Луны и Солнца может «уронить» такой спутник.

Ракеты-носители имеют несколько ступеней не только для экономии топлива и не только для того, чтобы вывести на орбиту бульшую полезную массу при заданном количестве топлива, но и для того, чтобы отработанные части ракеты не выходили на орбиту, а падали в заданном районе или сгорали в плотных слоях атмосферы, не засоряя околоземное пространство. Об этом мало задумывались в начале космической эры, но сейчас это вылилось в большую проблему. Часто можно слышать о том, что космическая станция изменила свою орбиту, чтобы не столкнуться с «космическим мусором».

С дискуссией по использованию Mathcad для расчета ракет-носителей можно ознакомиться по адресу: **https://www.ptcusercommunity.com/thread/128503**. Там же размещены файлы с приведенными выше расчетами и сделана попытка расчета реальной ракеты-носителя «Союз» с тремя (вернее, с двумя с половиной) ступенями.

ДИВЕРТИСМЕНТ. ЗАКОН РАДИОАКТИВНОГО РАСПАДА

В этюде был упомянут ядерный реактивный двигатель. Кроме того, этюд описывает решение дифференциального уравнения. Давайте еще раз вернемся к «диффурам» и решим типичную задачу ядерной физики. Тем более, что в этюде 7 на стр. 175 мы посетовали на то, что у нас нет отдельного этюда по этому разделу физики. И еще тем более то, пакет Mathcad создавался Алланом Раздоу (Allan Razdow) — физиком-ядерщиком по образованию, что мы уже отметили во введении.

Итак, радиоактивный распад.

В силу того, что спонтанный радиоактивный распад имеет вероятностную природу, выведем зависимость оставшегося количества не распавшихся частиц N от времени. Обозначим среднее время «жизни» радиоактивной частицы τ. Тогда

$$dN = -N \cdot \frac{dt}{\tau}.$$

Разделяя переменные, получаем уравнение радиоактивного распада:

$$\frac{dN}{N} = -\frac{dt}{\tau}.$$

И его решение:

$$N(t) = N_0 e^{-\frac{dt}{\tau}},$$

где N_0 — первоначальное количество частиц.

Часто для описания процессов радиоактивного распада применяют так называемый «период полураспада» — промежуток времени, в течение которого количество не распавшихся частиц уменьшается в два раза. Чтобы выделить этот параметр в нашем уравнении, выполним несложные преобразования:

$$N(t) = N_0 e^{-\frac{dt}{\tau}} = N_0 (2^{\log_2 e})^{-\frac{dt}{\tau}} = N_0 2^{-\frac{dt}{\tau/\log_2 e}} = N_0 2^{-\frac{dt}{T_{1/2}}},$$

где $T_{1/2} = \tau/\log_2 e$ — период полураспада.

Между периодом полураспада и временем жизни частицы устанавливается связь

$$T_{1/2} = \tau/\log_2 e$$

либо

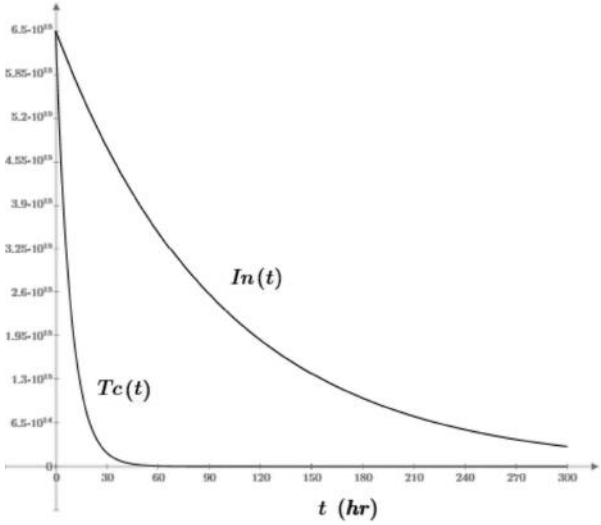
$$\tau = T_{1/2} \log_2 e = \frac{T_{1/2}}{\ln 2}.$$

Как видите, всё достаточно просто. Однако, при расчете реальных реакций выясняется, что радиоактивные элементы не «одиноки», порождаемые ими элементы, как правило, также являются радиоактивными, имеющими другой период полураспада, они порождают следующие элементы (а иногда и несколько видов) и расчет существенно затрудняется.

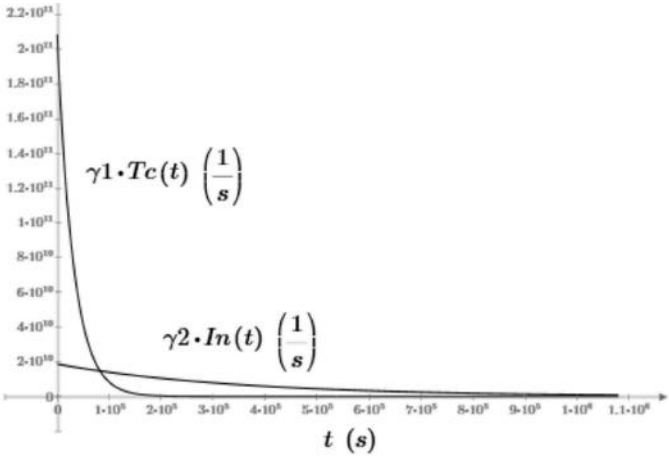
Для начала рассмотрим актуальную задачу создания полезного радиоактивного изотопа. Для примера выберем элемент технеций — легендарный элемент, первый из искусственно синтезированных химических элементов. Отдельный интерес вызывает изотоп технеция — технеций-99. Дело в том, что этот изотоп является очень удачным источником гамма излучения и поэтому очень важен в ядерной медицине. Излучение этого изотопа считается относительно «мягким», оно способно действовать только на очаг, слабо затрагивая соседние органы. Интересной особенностью технеция является его способность накапливаться в омертвевших участках сердечной мышцы, что можно увидеть при помощи радиометрической аппаратуры. Это позволяет проводить диагностику инфаркта миокарда в кратчайшие сроки — через 24 ч после его начала.

Обратим внимание на период полураспада технеция-99. Он составляет 6 ч. Здесь важно понять, что, с одной стороны, малые периоды полураспада означают то, что элемент является активным и испускает много радиоактивных частиц, но, с другой стороны, именно по этой причине он является короткоживущим и его излучение быстро прекращается. Проследим это на графиках. Рассмотрим распад технеция-99 с периодом полураспада 6 ч и индий-111 с периодом полураспада 2.8 сут. (последний препарат также широко применяется в ядерной медицине).

Количество не распавшихся атомов:



Активность препарата



Итак, период полураспада технеция-99 6 ч. Много это или мало? Для проведения качественных исследований достаточно малый период полураспада гарантирует высокую активность в сочетании с малым количеством препарата. Но... Этого очень мало для транспортировки препарата! Ведь буквально через 3 ч его станет в два раза меньше! Этот продукт сверхскоропортящийся. А его цена достигает десятков миллионов долларов за один грамм! Ученые нашли выход из положения, создав «генератор технеция». Как выяснилось, технеций-99 образуется в процессе спонтанного распада из другого радиоактивного препарата — молибдена-99. Период полураспада молибдена-99 составляет 66 ч. Этот элемент великолепно подходит для создания технеция-99. Общий ход реакции выглядит так:

$$^{99}Mo \rightarrow {}^{99}Tc^{m} + \beta^{-} \rightarrow {}^{99}Tc + \gamma \rightarrow {}^{99}Ru + \beta^{-}.$$

Гамма-кванты излучаются только при изомерном переходе с периодом полураспада 6 ч. Период полураспада технеция в последней реакции составляет $2{,}12 \cdot 10^{5}$ лет.

Составим уравнения реакций:

Ограничения

$$Mo(0\ day) = 6.5 \cdot 10^{15}$$

$$Tcm(0\ day) = 0 \quad Tc(0\ day) = 0$$

$$\frac{d}{dt} Mo(t) = -\beta 1 \cdot Mo(t)$$

$$\frac{d}{dt} Tcm(t) = \beta 1 \cdot Mo(t) - \gamma 2 \cdot Tcm(t)$$

$$\frac{d}{dt} Tc(t) = -\beta 3 \cdot Tc(t) + \gamma 2 \cdot Tcm(t)$$

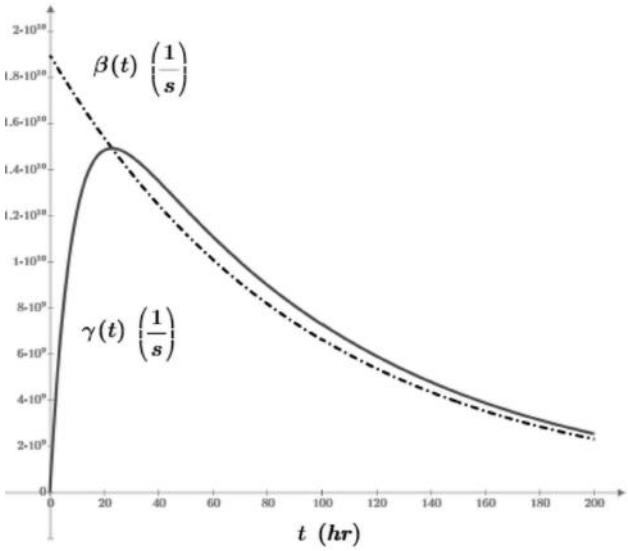
Решатель

$$\begin{bmatrix} Mo \\ Tcm \\ Tc \end{bmatrix} := \mathrm{odesolve}\left(\begin{bmatrix} Mo(t) \\ Tcm(t) \\ Tc(t) \end{bmatrix}, T\right)$$

Проанализируем активность гамма- и бета-излучения:

$$\gamma(t) := \gamma 2 \cdot Tcm(t)$$

$$\beta(t) := \beta 1 \cdot Mo(t) + \beta 3 \cdot Tc(t)$$



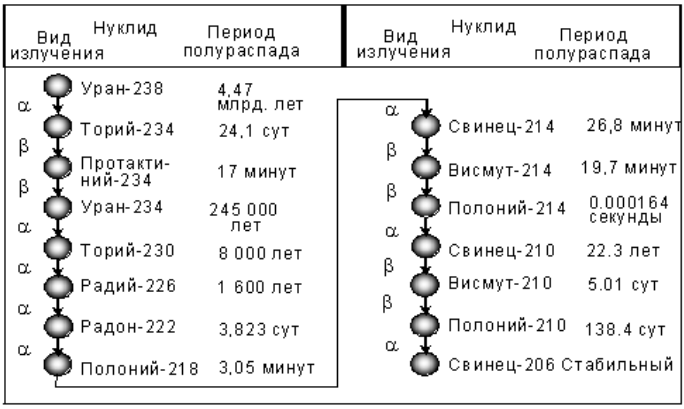
Видно, что примерно через 30 ч активность технеция достигает максимального значения.

Молибден-99, необходимый для осуществления реакции, вырабатывают в ядерном реакторе, затем в виде химического соединения молибдата натрия отправляется потребителю.

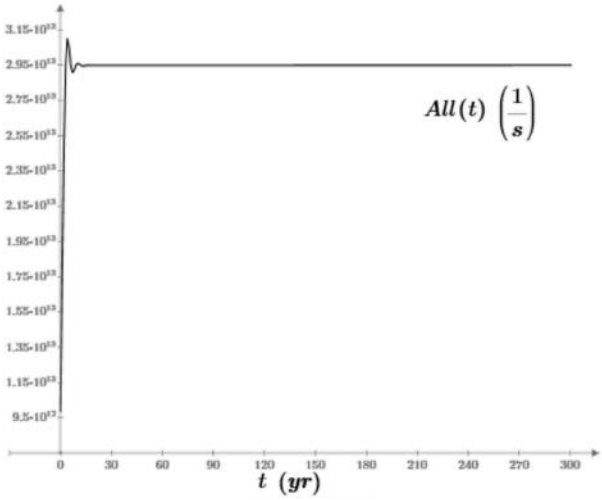
Вековое равновесие

Обратная ситуация складывается в ядерных реакциях, в которых участвуют долгоживущие изотопы (например, реакции в ряду урана-238). Период полураспада урана-238 составляет 4,47 млрд лет. Периоды полураспада всех образующихся элементов существенно меньше. Это приводит к тому, что главным «поставщи-

ком и регулятором» ядерных процессов является самый долгоживущий изотоп. По истечении времени сравнимого с периодом полураспада этого изотопа наступает «вековое равновесие» — процессы радиоактивного распада всех образующихся элементов следуют закону распада долгоживущего изотопа. В этом несложно убедиться на примере упомянутого урана-238. Его ряд выглядит следующим образом:



Составив уравнения радиоактивного распада, получим следующий график зависимости общей активности всех элементов ряда от времени (функция названа All(t)).



Видно, что начальные переходные процессы продолжались в течение крайне малого времени (менее 30 лет). Подробный анализ показывает, однако, что «вековое равновесие» наступает чуть позже, за период, составляющий несколько периодов полураспада самого долгоживущего дочернего элемента. Для уранового ряда это составляет около 830 000 лет.

Задание читателям

1. Рассчитать параметры полета реальной трехступенчатой ракеты «Союз», взяв ее стартовые и иные данные из Википедии, например.

Этюд 25

ГИБРИДНЫЕ РАСЧЕТЫ НА КОМПЬЮТЕРЕ, ИЛИ ЕЩЕ РАЗ ПРО КОНУС

В этюде сделан ретроспективный взгляд на процесс информатизации инженерного образования, в частности на способы решения задач на аналоговых, цифровых и гибридных компьютерах. Упомянуты двоичные, троичные, восьмеричные, десятеричные и шестнадцатеричные форматы вывода числовой информации «на печать», а также парадоксы, возникающие при этом. Рассмотрены преимущества и недостатки аналоговых, цифровых и гибридных компьютеров. Рассмотрены простейшие цифровые, аналоговые и гибридные компьютеры в связке с обыденными вещами (автомобили, часы и др.). Описано взаимодействие с аналоговыми вычислительными машинами как некий симбиоз науки и искусства. Дан анализ особенностей архитектуры и принципов работы легендарной советской АВМ МН-7. Проанализированы причины отказа от аналоговых компьютеров и перехода к цифровой технике. Даны примеры использования гибридных компьютеров в настоящее время, а также ренессанса аналоговых устройств и методов в наши дни. Введено понятие гибридного решения задачи на современном цифровом компьютере, включающего в себя элементы символьного (аналитического) и численного (приближенного) подходов. На примере решения задачи оптимизации (минимизация поверхности геометрических тел при фиксированном объеме) проанализированы инструменты систем компьютерной математики Mathcad (инженерный калькулятор) и Maple: решение алгебраических уравнений и систем, упрощение выражений, взятие производной и др. Сделан вывод о том, что современные математические программы дают нам новые возможности решения задач гибридным способом. Высказаны идеи оформления призовых кубков для победителей математических олимпиад.

Математика: числа с разным основанием, производная функции, поиск нулей функции.

Физика: гибридные автомобили.

Информатика: цифровые, аналоговые и гибридные компьютеры.

Сайт этюда: **https://www.ptcusercommunity.com/thread/139960**

Когда мы сейчас слышим слово «гибрид», то на ум, как правило, приходят гибридные автомобили[1]. Эти машины при движении попеременно или одновременно используют двигатель внутреннего сгорания и электродвигатель, питаемый электрогенератором и/или аккумулятором. За счет такой гибридности достигаются высокие показатели экономичности и решаются некоторые экологические проблемы: за городом такой автомобиль может «дымить», а в городе ехать как чистый электромобиль. Своеобразная гибридность, о которой не догадываются даже сами владельцы автомобилей, заложена и в некоторые паркетные внедорожники, передние колеса которых через механическую трансмиссию приводятся во вращение двигателем внутреннего сгорания, а задние при плохой дорожной обстановке, требующей полного привода, — электродвигателями, встроенными в ступицы колес. Некоторые троллейбусы также обладают гибридностью — при отсутствии кон-

[1] Последнее время слово «гибрид» стало принимать несколько негативный оттенок из-за... гибридных войн, когда военная сила дополняется другими мерами воздействия — социальными сетями Интернета, в частности. А вот бытовой пример своеобразной гибридности. Где лучше лечиться: в частной платной клинике или в бесплатной государственной больнице? Некоторые выбирают «гибрид» — ходят к «бесплатному» опытному врачу, но «благодарят» его подарками и «конвертами». Одни называют это бытовой коррупцией, а другие — своеобразным способом выживания российского здравоохранения через целевое неформальное налогообложение. Коррупцией же это явление обычно называют (обзывают) «государственные мужи и статс-дамы», посещающие закрытые бесплатные медицинские учреждения.

тактной сети (в историческом центре города, например, где провода портят вид улиц и площадей) они могут двигаться за счет накопленной в аккумуляторах энергии. Аккумуляторы при этом могут быть не только электрическими, но и в виде раскрученных маховиков, баллонов со сжатым воздухом и т. д.

Но гибридными бывают и другие «машины» — вычислительные, героини этой книги. Такие компьютеры состоят из *цифровых* и *аналоговых* блоков. Что это такое?

Как сложить 3 плюс 2, но не в уме, а на компьютере? Можно подсесть к компьютеру, пощелкать на клавиатуре и получить ответ 5. Компьютер переведет тройку и двойку из десятичного в двоичный формат (11 и 10 — см. вторую строку на рис. 25.1, где эти два числа выведены в двоичным виде, отмеченным постфиксом b — binary), проведет над нулями и единицами простейшие побитовые действия сложения, получит двоичный ответ (101 — не сто один, а один-ноль-один) и переведет все это обратно в привычный десятичный формат (5 — пятерка). Под хранение нулей и единиц в цифровых компьютерах зарезервированы элементы памяти, находящиеся в одном из двух возможных состояний: заряжено–разряжено, намагничено–размагничено и т. д. Из таких *битов* образуются *байты* и другие более объемные информационные конструкции. Так работают электронные цифровые вычислительные машины (ЭЦВМ) — компьютеры с двумя «цифрами в голове»: ноль и единица. Были, правда, попытки создания и троичных цифровых компьютеров (советская ЭЦВМ «Сетунь», например), но они вскоре прекратились в том числе и из-за массового выпуска двоичных электронных элементов.

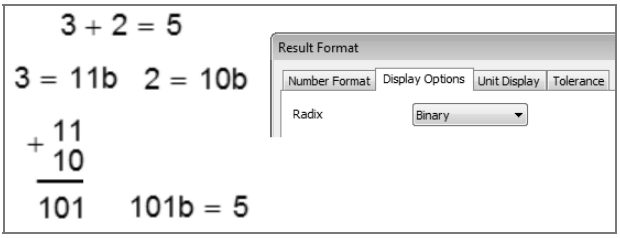


Рис. 25.1

Имитация сложения двух чисел на цифровом компьютере в среде пакета Mathcad 15

Но сложить тройку с двойкой можно и по-другому: взять два реостата, соединить их параллельно (см. рис. 25.2), выставить на первом реостате три ампера, а на втором два ампера и получить на выходе итоговый ответ: пять ампер.

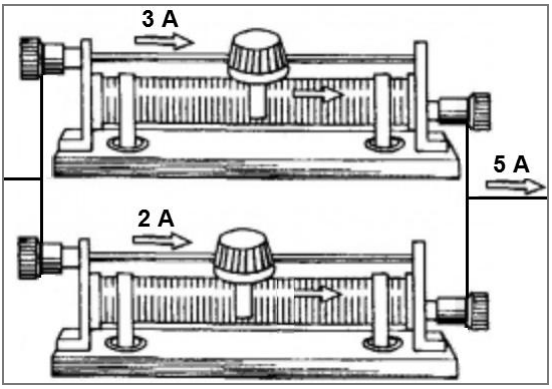


Рис. 25.2

Сложение двух чисел на аналоговом компьютере

Устройство, показанное на рис. 25.2, достаточно дооборудовать тремя амперметрами, чтобы получился «компьютер», складывающий два числа. Это и есть простейшая аналоговая вычислительная машина (АВМ). Математическая операция «сложение» тут имеет прямую электрическую *аналогию* — параллельное соединение двух проводников с током. К схеме, показанной на рис. 25.2, можно подключить дополнительные проводники с реостатами и получить своеобразный сумматор, который мы часто используем в электронных таблицах. Достаточно, например, выставить на таких реостатах зарплату отдельных сотрудников и получить итоговую сумму на «итоговом» амперметре. Аналог более сложной операции — операции дифференцирования — содержится в электрической катушке, а интегрирования — в конденсаторе с обратными связями. Аналоги математических действий можно найти и в пневматических[2], гидравлических, механических и прочих неэлектронных устройствах. На этих принципах и построены аналоговые вычислительные машины. Но в аналоговом сложении есть и серьезные недостатки. Максимальное значение используемых в аналоговых вычислительных машинах переменных ограничено. Если в примере с расчетом зарплаты на АВМ мы положим, что 1 рубль равен 1 амперу, то суммарный ток (который показывает, сколько денег должен получить некий коллектив работников) может быть весьма значительным и превысить возможности нашей электрической схемы. А если мы положим, что 1 рубль равен 0.01 амперу, например, то как измерять копейки? Но безусловным преимуществом АВМ является то, что они могут легко оперировать с функциями, а не только с переменными. Для сложения двух функций с аргументом время на цифровом компьютере нужно выполнить сложную внутреннюю процедуру. На аналоговом же компьютере для этого достаточно соединить два провода.

Московский энергетический институт (alma mater авторов этой книги) в свое время был пионером в использовании вычислительной техники не только для научных исследований и учебного процесса (см. ниже), но и в сфере администрирования. В частности, Информационно-вычислительный центр МЭИ (ИВЦ) разработал одну из первых в СССР программ для ЭЦВМ расчета зарплат сотрудников и стипендий студентов. Компьютеры печатали на принтерах (их тогда называли АЦПУ — алфавитно-цифровое печатающее устройство) ведомости на зарплату — колонки фамилий и цифр с итоговой суммой внизу. Главный бухгалтер МЭИ эти ведомости должен был подписывать. Но прежде чем это сделать, он проверял на *счетах* правильность числа, стоящего внизу ведомости. Когда этому человеку говорили, что в итоговой сумме никак не может быть ошибки, что ошибка возможна только при начислении той или иной зарплаты тому или иному работнику, и это простым суммированием не зафиксируешь, то ответ был примерно таков — компьютер в случае чего в тюрьму не посадят, а меня посадят, если я подпишу неправильную ведомость... Так до самой пенсии этот главбух щелкал костяшками счет и суммировал числа в ведомостях на зарплату перед их утверждением... Даже тогда, когда появились доступные настольные электронные калькуляторы, этот главбух не изменил своей деревянной «АВМ». К этим счетам потом прибили памятную табличку и подарили этому главбуху при торжественном проводе его на пенсию... Это было довольно давно. Но вот совсем недавно один из авторов книги сдавал белье в прачечную, и приёмщица, выписывая квитанцию, рассчитала сумму к оплате (произведение массы белья на стоимость стирки килограмма) на тех же деревянных счетах. Автор не удержался, сфотографировал этот «компьютер» и поместил фото на сайте **https://www.ptcusercommunity.com/message/439149**, где участники данного форума делятся своими воспоминаниями о старинных счетных

[2] Пневматические счетно-управляющие устройства использовались, а где-то используются и до сих пор на взрывоопасных производствах, например.

устройствах: счетах, арифмометрах, логарифмических линейках[3] и прочей вычислительной старине и экзотике.

Но если говорить о серьезных вещах, то следует отметить, что начиная с 50-х годов прошлого столетия МЭИ был в числе немногих отечественных вузов, уделявших особое внимание использованию в учебном процессе и научных исследованиях средств вычислительной техники.

Слыша об аналоговых вычислительных машинах, люди старшего поколения с образованием по электронике или теории управления сразу вспомнят легендарную машину МН-7 (рис. 25.3), на верхней панели которой собирались сложные электрические схемы, на которых довольно быстро и точно моделировались дифференциальные, интегродифференциальные и прочие довольно сложные уравнения и их системы до 6-го порядка включительно. Подобные задачи на цифровых машинах тех времен (Урал, МИР-1, БЭСМ-6, СМ-4 и т. д.) решались сутками, если вообще решались[4]. В МН-7 значения моделировались напряжением. Как уже упоминалось, у АВМ есть ограничения по максимальному значению переменной. Для МН-7 максимальное значение напряжения составляло 100 вольт. Это означало, что перед тем как собрать схему, необходимо было произвести расчеты масштабирующих коэффициентов, чтобы результат не превышал номинальное напряжение в 100 вольт. При этом возникали сложности с представлением малых величин. Как быть с величиной, которая составляет одну десятитысячную от максимального значения? А если разница между максимальным и минимальным значениями составляет 8–10 порядков?

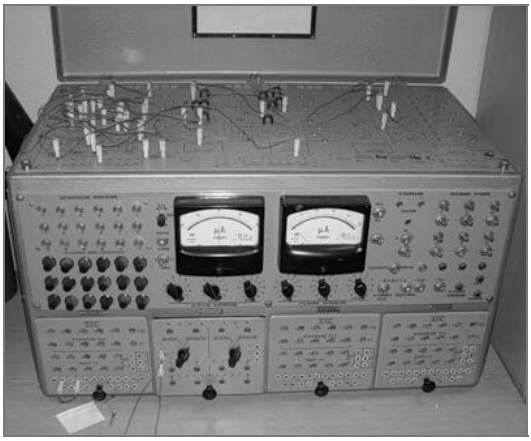


Рис. 25.3
Аналоговая вычислительная машина МН-7

[3] Логарифмическая линейка, если кто забыл, сводит умножение к сложению, а деление к вычитанию, что вытекает из свойства логарифмов. Эти арифметические действия осуществляются аналогово — через перемещение линеек с логарифмической шкалой и бегунка. Некоторые преподаватели на заре ЭВМ проверяли компьютерные распечатки курсовых и дипломных работ студентов (а они часто обгоняют своих наставников в плане использования компьютеров) с помощью... логарифмической линейки.

[4] Исходные данные и программы для их обработки в те времена хранились в основном на перфокартах. В позднесоветское время можно было купить дефицитную книгу, сдав определенное количество макулатуры. Некоторые предприимчивые сотрудники НИИ и вузов стали сдавать мешками в качестве такой макулатуры... перфокарты. Так, говорят, были потеряны многие уникальные программы решения задач на ЭЦВМ. Потом, правда, спохватились и запретили принимать в отделениях утильсырья эти бумажные носители информации. В наши дни в лом (на выплавку драгметаллов) пустили многие уникальные ЭЦВМ, вместо того, чтобы разместить их в музеях.

Решение систем дифференциальных уравнений зависит от начальных условий. Малейшее отклонение начальных условий от заданных значений могло привести к другому решению. Поэтому процессу задания начальных условий и выставления нулей усилителей при моделировании на аналоговых вычислительных машинах уделялось большое внимание. Перед тем как запустить собранную модель, необходимо было убедиться, что на входах усилителей именно те значения, которые нужны (0 вольт). Отклонение в сотые доли при интегрировании могло дать весьма значительную ошибку.

МН-7 работала на ламповых операционных усилителях, характеризующихся малыми значениями паразитных токов и напряжений и практически полным отсутствием зависимости характеристик усилителя от температуры окружающей среды. При появлении АВМ, использующих транзисторы, процедура выставления нулей усилителей превратилась в целое приключение. Порой казалось, что характеристики усилителей зависят не только от окружающей среды, но и от фаз луны, стройности проходившей мимо лаборантки и качества обеда в студенческой столовой. А если таких усилителей в комплексе 128 штук? На МН-7 всего было 18 усилителей.

Недаром профессор МЭИ Илья Маркович Тетельбаум писал [93], что «...аналоговое моделирование — это не формализованная процедура, а экспериментальный поиск. Поэтому можно говорить об искусстве моделирования так же, как и об искусстве эксперимента».

Но скорость цифровых компьютеров росла, наращивалась их память, появились изощренные быстрые численные алгоритмы для тех же дифференциальных уравнений, что позволило решать их за приемлемое время и на цифровых ЭВМ. Из-за этого и по другим причинам аналоговые вычислительные машины как бы ушли в тень. Вернее, так. Они выжили, но в связке с цифровыми устройствами в виде *гибридных* машин — в качестве аналоговых блоков к уже привычным цифровым компьютерам. Такие блоки применяются, например, в специализированных компьютерах, обслуживающих тренажеры, — там, где счет должен вестись в режиме реального времени. В самолетных тренажерах, например. Без аналоговых блоков такие задачи под силу только суперкомпьютерам, время работы которых слишком дорого. Кстати, если аналоговый сумматор, показанный на рис. 25.3, дооборудовать не аналоговыми (стрелочными — см. переднюю панель прибора на рис. 25.3), а современными цифровыми амперметрами, то и получится примитивная гибридная вычислительная машина.

Слово «гибрид» можно приложить и к современным методикам решения задач в средах математических пакетов, установленных на цифровых компьютерах.

Давайте поговорим об этом.

Математическую задачу, как известно, можно решить двумя способами: *численно* или *аналитически* (*символьно*[5]). В современных компьютерных математических программах (Maple, Mathcad, MATLAB, Mathematica и др.) для этого предусмотрены специальные инструментальные средства. Символьные инструменты работают с самими символами переменных[6], а численные — с числами, хранящимися в переменных. В умелом сочетании использования этих инструментов и заключена сущность слова «гибрид», стоящего в названии этюда.

Давайте рассмотрим конкретный несложный пример такого гибридного решения математической задачи на компьютере.

---

[5] Более правильное, но более длинное название — компьютерные инструменты аналитических преобразований. Но у нас, как обычно, прижилась калька с английского — symbolic. Другой пример, приведенный ранее, — АЦПУ и принтер.

[6] Вернее, так. Если переменная хранит число, то символьная математика работает с ним, а если переменная не хранит числа, то с самим символом. Но есть специальные операторы, способные глушить число в переменной для символьной работы с ней.

Вернемся к автомобилям, простым и гибридным. Они, в отличие от чистых электромобилей, должны заправляться бензином или дизельным топливом. На крупных АЗС можно видеть цистерны для хранения топлива в виде прямых круговых цилиндров. А какие пропорции у этих емкостей, если нет каких-либо особых ограничений? Несложно доказать, что если диаметр основания цилиндра d равен его высоте h, то такое геометрическое тело будет иметь наименьшую поверхность S[7] при заданном объеме V [94]. На сооружение такой емкости пойдет меньше металла, краски, теплоизоляции, если, к примеру, хранится горячая вода или охлажденный хладагент[8]. А если цистерна будет открытой и у ней нет верхней крышки!? Тут также несложно подсчитать, что в такой емкости соотношение d / h будет равно уже не единице, а двум (высота цилиндра равна радиусу его основания). А если емкость имеет форму прямого кругового конуса и предназначена, например, для хранения сыпучих материалов! Какое тут должно быть оптимальное соотношение d к h? Здесь уже сходу в уме такую задачу не решить — нужно будет привлекать математиков или... компьютер. Мы такую задачу о конусе решали в этюде 12 — см. 12.4–12.6 на стр. 277 и 278.

Усложним эту задачу. На рисунке 25.4 показано решение в среде Maple задачи об оптимальном соотношении диаметра и высоты емкости в виде некоего «бокала»: в виде прямого кругового конуса с диаметром основания d и высотой h, с круглой крышкой (ее на рис. 25.1 не видно) и круглой подставкой. У крышки и подставки такой же диаметр, как и у основания конуса. Спрашивается, чему равно оптимальное отношение d к h для такой емкости?

На рисунке 25.4 показано решение задачи о «бокале с крышкой» в среде Maple. Этот пакет тут выбран неслучайно, так как у него (наряду с пакетом Mathematica[9]) основной инструмент — это символьная, а не численная математика, и мы это подчеркивали во введении. На рисунке 25.4 аналитически решается уравнение объема конуса по переменной h — выводится формула, связывающая высоту конуса с его заданным объемом V и диаметром основания d (пункт 1 на рис. 25.4)[10].

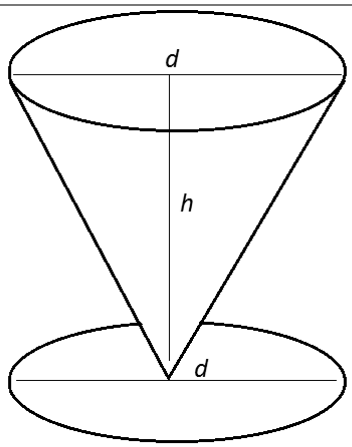
---

[7] Как известно, из всех геометрических тел наименьшую поверхность при заданном объеме имеет шар. Но шарообразный резервуар изготавливать довольно сложно. Их делают только для хранения газов под давлением (газгольдеры, танки для перевозки сжиженного газа на специализированных судах и автомобилях).

[8] Это так называемые аккумуляторы тепла или холода, применяемые для повышения энергоэффективности и энергосбережения [2]. Компрессорная часть такого кондиционера работает только ночью, когда есть избыток электроэнергии и она дешевле чем днем, отправляя холод в виде остуженного хладагента не в жилое помещение, а в специальный бак, из которого позже жарким днем будет забираться холод. Таким же образом зимой можно аккумулировать тепло, сжигая природный газ только ночью, когда его избыток. В энергоснабжении зданий в настоящее время также вклиниваются элементы гибридности, связанные, например, с тем, что для производства тепла, холода и электроэнергии (тригенерация) используются устройства с различными принципами действия. В традиционный плакат, который висит во многих учебных заведениях энергетического профиля «Давать тепло и свет — наша профессия!» нужно вписать еще одно слово «холод». Но «давать холод в дом» — это как-то странно звучит.

[9] Ниже мы будем работать с пакетом Mathcad, «так как у него (наряду с пакетом Matlab) основной инструмент — это численная, а не символьная математика». Использование при решении задач разных пакетов можно также считать еще одним проявлением гибридности. Формулы для объема и поверхности тел можно искать либо в уме, либо в справочниках — бумажных или интернетовских. Но лучше тут поступить опять же «гибридно»: вспомнить формулу, а потом проверить ее правильность. А еще лучше постараться вывести нужную формулу самому. Так можно тренировать свой мозг. Некоторые пожилые люди, звоня своим близким, стараются выуживать номер телефона именно из своей памяти, а не из памяти телефона.

[10] Это несложное преобразование можно сделать и без компьютера и даже без листка бумаги — в уме и сразу вписать ответ. Сочетание ручных («головных») и компьютерных символьных преобразований — это еще один пример гибридности.



$$> h_1 := solve\left(V = \frac{1}{3}\pi\cdot\frac{d^2}{4}\cdot h, h\right)$$

$$h_1 := \frac{12\,V}{\pi\,d^2} \qquad \textbf{(1)}$$

$$> S := subs\left(h = h_1, \pi\cdot\frac{d}{2}\cdot\sqrt{\left(\frac{d}{2}\right)^2 + h^2} + \pi\cdot\frac{d^2}{4} + \pi\cdot\frac{d^2}{4}\right)$$

$$S := \frac{1}{4}\pi\,d\sqrt{d^2 + \frac{576\,V^2}{\pi^2\,d^4}} + \frac{1}{2}\pi\,d^2 \qquad \textbf{(2)}$$

$$> d_{optimal} := solve\left(\frac{\mathrm{d}}{\mathrm{d}\,d}S, d\right)$$

$$d_{optimal} := \frac{96^{1/6}\left(\pi^4\,V^2\left(2\sqrt{7} - 5\right)\right)^{1/6}}{\pi}, \qquad \textbf{(3)}$$

$$> A := simplify\left(\frac{d_{optimal}}{h_1\Big|_{d = d_{optimal}}}\right)$$

$$A := \frac{1}{3}\sqrt{6}\sqrt{2\sqrt{7} - 5} \qquad \textbf{(4)}$$

$$> evalf(A)$$

$$0.4408345283 \qquad \textbf{(5)}$$

Рис. 25.4

Символьный расчет оптимального «бокала» с плоской крышкой

Далее полученная формула для h_1 подставляется в формулу площади поверхности S «бокала с крышкой» (пункт 2). Затем у выражения для S берется первая производная по переменной d, у которой ищутся нули[11] (пункт 3). Известно, что у гладкой непрерывной функции (а мы с ней тут имеем дело) в точке минимума производная равна нулю. Это положение и позволило нам вывести некую новую

[11] Таких решений оказалось несколько, но мы оставили для дальнейших преобразований одно из них. Нужный ли мы взяли нуль функции, это отдельный вопрос для отдельного анализа. Может быть, у нашей задачи есть несколько решений?

математическую константу[12] — см. пункт 4 (точное воспроизведение константы в радикалах[13]) и пункт 5 (расчет приближенного значение этой константы в виде десятичного числа с плавающей точкой[14]).

Но если у нашего «бокала» плоскую крышку заменить на полусферу, то ответа уже так просто получить не удастся. Вернее, он будет получен, но в виде... новой задачи — задачи о поиске нулей полинома высокой степени. Можно, конечно, попытаться решить эту задачу аналитически, но лучше вспомнить, что есть и... численные методы решения подобных задач.

На рисунке 25.5 показано численное решение еще более сложной задачи — задачи об оптимальных пропорциях тела, составленного из полусферы, цилиндра и конуса, стоящего на круглой подставке. Полный объем такого тела (теперь это уже не бокал, а некий кубок, предназначение которого будет обрисовано ниже) равен 1 литру. Спрашивается, чему должны быть равны значения переменные d, L и h, при которых наружная поверхность тела (кубка) будет минимальна.

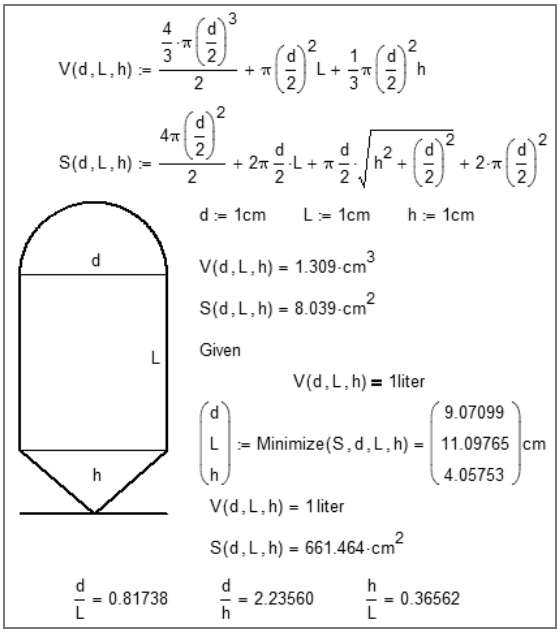


Рис. 25.5

Численный расчет оптимального кубка (Mathcad 15)

На рисунке 25.4 мы продемонстрировали *математический* подход к решению задачи: нас интересовали не размеры конкретного оптимального бокала в сантиметрах, а математическая формула, по которой можно рассчитать пропорции такого тела произвольного объема. Решение же, показанное на рис. 25.5, содержит некий *инженерный* подход. Человеку (инженеру), проектирующему конкретную

[12] Есть сайт в Интернете (**http://www.people.fas.harvard.edu/~sfinch**), на котором собраны всевозможные математические константы, начиная от π и e и кончая совершенной экзотикой, мало понятной многим даже хорошо знающим математику.

[13] У простого конуса без основания соотношение d к h будет равно корню из двух, а у конуса с крышкой — в два раза меньше. Можно запомнить эту закономерность для проектирования конических емкостей (математический подход), а можно делать расчеты для каждого конкретного случая (инженерный подход).

[14] Тут можно начать вычислительную гонку — объявить конкурс на то, кто вычислит наиболее длинное десятичное значение этой константы. Сейчас такая гонка ведется в отношении главной математической константы — числа π, которое фигурирует в наших формулах вычисления объема и площади.

емкость, недосуг искать в справочниках или выводить самому общие закономерности, а потом обдумывать, как их приложить к своей практической задаче[15], — ему нужно строить конкретный объект. Пакет Mathcad тут выбран неслучайно. Его еще называют инженерным суперкалькулятором. В среде Mathcad, например, можно пользоваться единицами измерения в полном объеме. Чего не скажешь о пакете Maple [4]. И это понятно. Для математика, решающего абстрактные задачи, метры, килограммы и секунды не нужны. Инженеру же без них никак не обойтись.

В решении на рис. 25.5 вводятся две функции пользователя с именами `V` (объем кубка) и `S` (площадь наружной поверхности кубка), у которых по тройке (заметим!) аргументов `d`, `L` и `h`. Им придается первое приближение (предположение), равное одному сантиметру. Mathcad-функция `Minimize` будет по особому численному алгоритму менять значения, хранящиеся в переменных `d`, `L` и `h` так, чтобы функция `S` (целевая функция поиска) приняла минимальное значение, а объем тела `V` оставался бы равен одному литру (ограничение при поиске минимума целевой функции).

Примечание. Если читатель попробует, как говорят студенты, вручную «набить» задачу на рис. 25.5 на компьютере с Mathcad 15, то он получит... неверный ответ. Такие ситуации приводят в ступор многих пользователей Mathcad. Они хватаются за голову и делают удивленные лица. Именно такой человек изображен на заставке форума «Сумасшедший дом Mathcad», о котором мы уже упоминали в книге (см. стр. 52). Дело в том, что установленная по умолчанию точность численного решения задачи об оптимальном кубке — это 0.001 м или 1 мм (0.1 см), если решается размерная задача с опорой на международную систему измерений СИ, где метр — это базовая единица длины. А этого недостаточно для нашего не «метрового», а «сантиметрового» кубка. Чтобы правильно решить задачу на рис. 25.5, нужно изменить значение встроенной переменной `CTOL` с 10^{-3} на 10^{-7}, например. В среде Mathcad Prime, кстати, эта недоработка исправлена — см. обсуждения этого вопроса на сайте **https://www.ptcusercommunity.com/thread/135400**. Из-за этой особенности пользователи Mathcad иногда отключают механизм размерностей при решении некоторых задач. И второе. Численные методы расчетов часто называют *приближенными* методами. И в этом мы только что убедились, решив задачу об оптимальном кубке приближенно — с точностью 100 нанометров. Но и символьные методы решения задач могут быть приближенными. Это случается тогда, например, когда сложную функцию вблизи рассматриваемой точки заменяют на ряд более простых функций — пример на рис. 4.11 и 4.12 в этюде 4 на стр. 70 и 71. Это уточнение может касаться аналоговых и цифровых вычислительных машин. Считается, что первые решают задачу приближенно и на это решение оказывают влияние разные внутренние и внешние факторы (см. выше). Цифровые же машины более детерминированы и решают задачи с предсказуемой точностью при каждом прогоне решения.

Наша задача о кубке решена (рис. 25.5), но остались сомнения, будет ли влиять объем тела `V` на оптимальные соотношения `d / L`, `d / h` и `h / L`, показанные в конце рисунка 25.5. По задаче на рис. 25.4 таких сомнений не было: `d / h` оставалось константой при любом объеме «бокала». На то оно и математическое, а не инженерное решение!

В решении на рис. 25.5 был использован KISS-принцип. KISS — это аббревиатура английской фразы «Keep it simple, stupid» — «Делай это проще, дурачок!» Она призывает решать поставленные задачи простыми методами и прибегать к изощренным алгоритмам только тогда, когда простые методы дают сбой или... сложны в реализации. А простота задачи на рис. 25.5 состояла в том, что мы не стали ее анализировать глубже и не сократили число аргументов функции `S` с трех до двух. Сделаем это!

На рисунке 25.6 показано *гибридное* решение задачи о нашем кубке: сначала используются символьные инструменты Mathcad[16] для получения функции `S` уже

[15] Но всегда нужно помнить, что «нет ничего практичнее хорошей теории».
[16] Они, кстати, в старых версиях Mathcad были взяты из пакета Maple.

не с тремя (рис. 25.5), а с двумя (`d` и `L`) аргументами и одним параметром `V`. Для этого были задействованы символьные операторы `solve` (решить) и `substitute` (подставить; в среде Maple он называется `subs`). Далее можно было бы пойти по пути, намеченному на рис. 25.4, — попытаться взять частные производные функции `S` по переменным `d` и `L` и решить систему уже двух уравнений. Но можно перейти на «численные» рельсы и искать у функции `S` минимум с помощью той же встроенной функции `Minimize`.

$$V=\frac{\frac{4}{3}\cdot\pi\cdot\left(\frac{d}{2}\right)^3}{2}+\pi\cdot\left(\frac{d}{2}\right)^2\cdot L+\frac{1}{3}\cdot\pi\cdot\left(\frac{d}{2}\right)^2\cdot h \xrightarrow{solve,h} -\frac{\pi\cdot d^3+3\cdot\pi\cdot L\cdot d^2-12\cdot V}{\pi\cdot d^2}$$

$$\frac{4\cdot\pi\cdot\left(\frac{d}{2}\right)^2}{2}+2\cdot\pi\cdot\frac{d}{2}\cdot L+\pi\cdot\frac{d}{2}\cdot\sqrt{h^2+\left(\frac{d}{2}\right)^2}+2\cdot\pi\cdot\left(\frac{d}{2}\right)^2 \xrightarrow{substitute,\,h=-\frac{\pi\cdot d^3+3\cdot\pi\cdot L\cdot d^2-12\cdot V}{\pi\cdot d^2}} \cdots$$

$$\cdots\rightarrow \frac{\pi\cdot d\cdot\left(2\cdot L+2\cdot d+\sqrt{\frac{d^2}{4}+\frac{\left(\pi\cdot d^3+3\cdot\pi\cdot L\cdot d^2-12\cdot V\right)^2}{\pi^2\cdot d^4}}\right)}{2}$$

$$V:=1\ \boldsymbol{L} \qquad S(d,L):=\frac{\pi\cdot d\cdot\left(2\cdot L+2\cdot d+\sqrt{\frac{d^2}{4}+\frac{\left(\pi\cdot d^3+3\cdot\pi\cdot L\cdot d^2-12\cdot V\right)^2}{\pi^2\cdot d^4}}\right)}{2}$$

$$d:=1\ \boldsymbol{cm} \qquad L:=1\ \boldsymbol{cm} \qquad \mathbf{Minimize}(S,d,L)=\begin{bmatrix}9.070739\\11.099027\end{bmatrix}\boldsymbol{cm}$$

Рис. 25.6
Гибридный расчет оптимального кубка: Mathcad Prime

На сайте книги можно скачать файлы Maple и Mathcad. Там же расположена анимация процесса поиска оптимального размера конического бокала со сферической, а не плоской крышкой, но без «ножки» (кубок «Пей до дна!» или рожок с мороженым). На рисунке 25.7 показаны три кадра этой анимации с графическим отображением целевой функции.

На другом сайте **(https://www.ptcusercommunity.com/groups/optimisation with-mathcad)** собраны анимации, иллюстрирующие разные математические задачи с оптимизацией. Одна из анимаций в этой группе — это анимация проектирования кубка оптимального размера без ножки, составленного из полусферы, цилиндра и конуса: см. **https://www.ptcusercommunity.com/videos/2239**. Обсуждение этой математической задачи можно найти на сайте **https://www.ptcusercommunity.com/message/196864.**

Спроектированный нами кубок, показанный на рис. 25.5, имеет пропорции, найденные там же и на рис. 25.6. Но в кубке главное не минимум поверхности[17], а некая красота пропорций. Есть ли такая «красота» на рис. 25.5? Об эстетической стороне кубка разговор должен быть особый. Сейчас же мы можем поступить так: выгравировать на цилиндрической части нашего оптимального кубка математические выкладки, показанные выше, и вручить такой кубок... победителю математической олимпиады. Но не традиционной, а гибридной: ее участники для решения задач могут использовать не только свои знания и смекалку, но и возможности современных математических пакетов и даже помощь Интернета.

Гибридность также можно отметить в сочетании компьютерных символьных преобразований с аналогичными преобразованиями, делаемыми самим пользова-

[17] А почему бы и нет? Кубок можно сделать из дешевого металла, а потом покрыть слоем золота или серебра. Наш расчет минимизирует затраты на эту операцию.

телем компьютера — человеком. Это мы делали, например, решая задачу о линзе — см. рис. 20.4 на стр. 402.

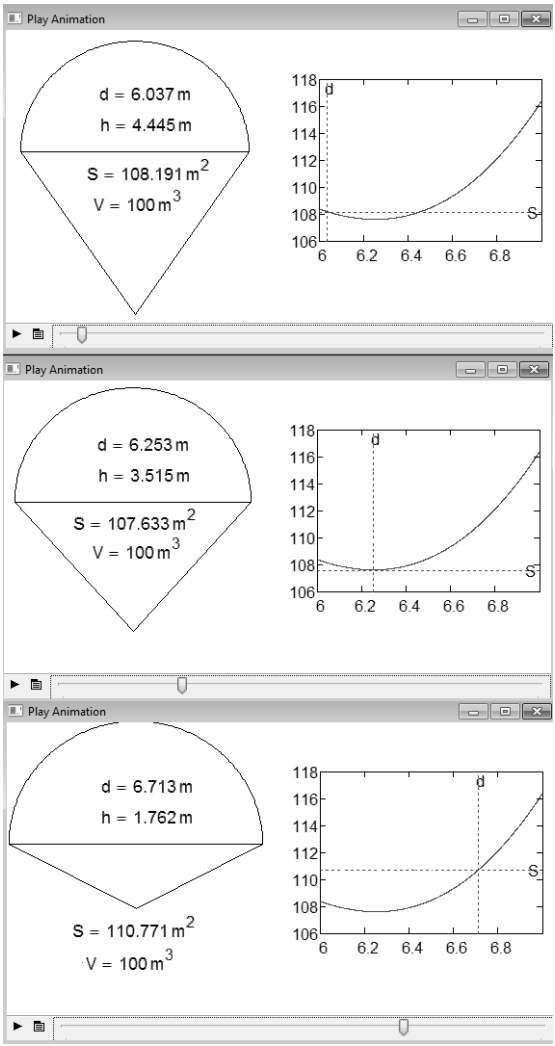


Рис. 25.7

Три кадра анимации задачи оптимизации размера тела, составленного из конуса и полусферы

Вывод.

Умелое использования аналитических и численных методов, различных математических пакетов и Интернета повышает эффективность и точность решения задач, делает сам процесс приятным за счет перекладывания рутинных действий на плечи компьютера и оставления для человека только интеллектуальной, творческой работы. Такие решения можно дополнять плоской и трехмерной графикой и анимацией.

Авторы благодарны Сергею Хорькову, принявшему участие в написании этюда.

Задание читателям

1. Придумать и переслать авторам новые задачи для этой книги.

ЗАДАЧИ, РЕШЕННЫЕ В КНИГЕ И НА САЙТЕ КНИГИ

1. Колебание математического маятника — рис. 1.1, рис. 9.23 и 9.24*a*.
2. Колебание физического маятника — рис. 9.24*b* и 9.25.
3. Рост Эркюля Пуаро — рис. 1.2.
4. Рост тургеневского Герасима — рис. 1.2.
5. Пряничный человечек — рис. 1.3.
6. Удельный расход условного топлива — рис. 1.4.
7. Пересчет видов концентрации — рис. 1.6–1.9.
8. Моляльность водного раствора — рис. 1.6 и 1.9.
9. КПД термодинамического цикла Отто — рис. 1.10–1.12 и 1.14.
10. Энтропия идеального газа — рис. 1.12.
11. Внутренняя энергия идеального газа — рис. 1.13.
12. Движение тела в пространстве — рис. 2.1 и 2.2.
13. Плотность и концентрация водного раствора соли — рис. 2.3.
14. Уравнение прямой линии — рис. 2.4.
15. Длина вектора — рис. 2.2 и 2.6.
16. Старт автомобиля — рис. 2.7.
17. Перевод чисел в двоичный, восьмеричный и шестнадцатеричный формат — рис. 3.1.
18. Перевод чисел в произвольный формат — рис. 3.2.
19. Формат суммы денег — рис. 3.3.
20. Перевод арабского числа в римское — рис. 3.4.
21. Перевод римского числа в арабское — рис. 3.5.
22. Частотный словарь текста — рис. 3.6.
23. Суммирование простых дробей — рис. 3.7.
24. Площадь прямоугольника — рис. 3.8.
25. Задача о движении моторной лодки по реке — рис. 4.1, 4.2, 4.10–4.12.
26. Задача о движении двух моторных лодок по реке — рис. 4.15 и 4.16.
27. Золотое сечение — рис. 4.14.
28. Крейсерская скорость судна — рис. 4.17, 4.18 и 4.20.
29. Остановка моторной лодки — рис. 4.21 и 4.22.
30. Кафель для ванной комнаты математика — рис. 4.28 и 4.29.
31. Диаметр шара — рис. 4.30.
32. Копирование картины Ван Гога — рис. 4.36, 4.38, 4.40–4.42.
33. Кружевная салфетка — рис. 4.43–4.46.
34. Вышивка Ришелье — рис. 4.47 и 4.48.
35. Портреты Че Гевары и Элизабет Тейлор — рис. 5.6 и 5.7.
36. Пасьянс — рис. 5.14.
37. Треугольник и точка — рис. 5.16.
38. Стороны треугольника — рис. 5.17–5.20.
39. Задача о собаке, корове и лошади — рис. 6.1.
40. Задача о деньгах двух братьев — рис. 6.2.
41. Задача о дороге к храму — рис. 6.3.
42. Разложение 1000 вершков на сажени, аршины и вершки — рис. 6.4.
43. Задача о медных чайниках — рис. 6.5.
44. Задача о миллиарде секунд — рис. 6.6.
45. Задача о фунте и золотнике — рис. 6.7.
46. Задача о дороге, вымощенной досками — рис. 6.8.
47. Поиск кубического корня — рис. 6.9.
48. Задача о купце и сукне — рис. 6.10.
49. Задача о курсе рубля — рис. 6.11.
50. Задача о французских деньгах — рис. 6.12 и 6.13.

51. Размеры подводной лодки «Наутилус» — рис. 6.15 и 6.16.
52. Расчет геометрии арки — рис. 7.1 и 7.27.
53. Расчет провисания цепи — рис. 7.3 и 21.1.
54. Задача о висячем мосте — рис. 7.5.
55. Задача о канатной дороге — рис. 4.49–4.51, 7.9, 7.14 и 7.15.
56. Задача о трех скоростях велосипеда — рис. 7.20.
57. Задача об электрической цепи — рис. 7.21–7.24, 7.28 и 7.29.
58. Параболическое зеркало — рис. 7.35, 7.36 и 20.4.
59. Оптическая линза — рис. 7.38 и 20.6.
60. Кинематика паровой машины — рис. 8.1–8.4.
61. Задача о дворниках автомобиля — рис. 8.6.
62. Стопоходящая машина Чебышёва — рис. 8.7.
63. Стопоходящая машина Янсена — рис. 8.8.
64. Прави́ло Чебышёва — рис. 8.9 и 8.10.
65. Движение трех планет — рис. 9.1 и 9.3.
66. Движение пяти планет — рис. 9.5.
67. Движение трех планет в 3D — рис. 9.6.
68. Перехват спутника — рис. 9.7 и 9.8.
69. Обмен спутниками — рис. 9.9.
70. Движение четырех планет и четырех спутников — рис. 9.10.
71. Часы Кеплера — рис. 9.11.
72. Прыжок парашютиста — рис. 9.12.
73. Прыжок в воду — рис. 9.13.
74. Пуск ракеты из подводной лодки — рис. 9.14.
75. Движение астероида у Земли — рис. 9.15–9.18.
76. Торможение спутника в плотных слоях атмосферы Земли — рис. 9.20–9.22.
77. Гравитационный поезд — рис. 9.30–9.34.
78. Поворот трамвая — рис. 9.35.
79. Американские горки — рис. 9.36.
80. Старт ракеты — рис. 9.37 и 24.3.
81. Ускорение свободного падения — рис.9.38 и 9.39.
82. Рисование эллипса — рис. 9.40.
83. Факториал — рис. 10.1.
84. Числа Фибоначчи — рис. 10.2.
85. Прекрасные числа Фибоначчи — рис. 10.3.
86. Ханойские башни — рис. 10.4.
87. Фрактал Бодибилдинг — рис. 10.5.
88. Конь на шахматной доске — рис. 10.6.
89. Выключатель для дачи — рис. 11.6 и 11.8.
90. Машина для голосования — рис. 11.18 и 11.19.
91. Объем пирамиды — рис. 12.2 и 12.3.
82. Конус с минимальной площадью поверхности — рис. 12.4–12.6.
93. Оптимальный фильтр — рис. 12.7.
94. Максимальный объем пожарного ведра — рис. 12.8.
95. Оптимальное пожарное ведро — рис. 12.9–12.16.
96. Окружность, вписанная в треугольник, — рис. 13.1–13.4.
97. Движение окружности в треугольник — рис. 13.6–13.12.
98. Лемниската и эллипс — рис. 13.14.
99. Треугольник в ложбине — рис. 13.15.
100. Линейное сглаживание — рис. 14.5–14.7 и 14.9.
101. Сглаживание полиномом — рис. 14.10 и 14.11.
102. Линейно-кусочная интерполяция — рис. 14.12.
103. Интерполяция сплайнами — рис. 14.13.
104. Предсказание — рис. 14.14 и 14.15.
105. Сглаживание табличной зависимости — рис. 14.16.

106. Медиан-медианная регрессия — рис. 14.17.
107. Цена подержанного автомобиля — рис. 14.22–14.32 и 14.34.
108. Разложение числа на простые множители — рис. 15.2.
109. Задача коммивояжера — рис. 15.4–15.10.
110. Обратная задача коммивояжера — рис. 15.11.
111. Бег в укрытие — рис. 15.13.
112. Задача Штейнера — рис. 15.14 и 15.15.
113. Транспортная задача — рис. 15.16 и 15.17.
114. Задача о финансовой пирамиде — рис. 15.18 и 15.9.
115. Орел — решка — рис. 16.1.
116. Игра в кости — рис. 16.2.
117. Расчет числа π — рис. 16.3.
118. Трехсторонняя дуэль — рис. 16.5–16.8.
119. Задача о гусях — рис. 17.1.
120. Задача о рыбаках и рыбке — рис. 17.2 и 17.3.
121. Старая буквенная головоломка — рис. 18.2.
122. Новая буквенная головоломка — рис. 18.4.
123. Нахождение центра масс плоской фигуры — рис. 21.6 и 23.8.
124. Сортировка вектора — рис. 22.2 и 22.4.
125. Остойчивость судна — рис. 23.7.
126. Многоступенчатая ракета — рис. 24.5.

ПРОИЗВЕДЕНИЯ ЛИТЕРАТУРЫ И ИЗОБРАЗИТЕЛЬНОГО ИСКУССТВА, УПОМИНАЕМЫЕ В КНИГЕ

И. Ильф и Е. Петров. Золотой теленок.
Н. В. Гоголь. Мертвые души.
Марк Твен. Странствования за границей.
Л. Н. Толстой. Анна Каренина.
Л. Н. Толстой. Воскресенье.
Л. Н. Толстой. Война и мир.
С. А. Рачинский. 1001 задача для умственного счета.
Н. П. Богданов-Бельский. Картина «Устный счет. В народной школе С. А. Рачинского».
Ван Гог. Пейзаж в Арле после дождя.
Даниил Хармс. Старуха.
Даниил Хармс. Исторический эпизод.
Даниил Хармс. Праздник.
Козьма Прутков. Плоды раздумья.
Фильм «Девять дней одного года».
Сергей Довлатов. Соло на ундервуде.
Эдгар По. Золотой жук.
Артур Конан Дойль. Пляшущие человечки.
Энди Уорхол. Лиз.
Леонардо да Винчи. Мона Лиза.
А. С. Пушкин. Портрет.
А. С. Пушкин. Каменный гость.
А. С. Пушкин. Пиковая дама.
И. А. Крылов. Лебедь, Рак и Щука.
И. С. Тургенев. Муму.
А. П. Чехов. Репетитор.
А. П. Чехов. Грач.
Ф. М. Достоевский. Игрок.
Жан-Батист Мольер. Мещанин во дворянстве.
Жуль Верн. Двадцать тысяч лье под водой.
Жуль Верн. Таинственный остров.
Акира Куросава. Семь самураев.
Фильм «Великолепная семерка».
Фильм «Армагеддон».
Альберто Корда. Фотография Че Гевары.
М. А. Булгаков. Мастер и Маргарита.
Альф Прейсен. Козленок, который умел считать до десяти.
А. Н. Толстой. Приключения Буратино.
М. Жванецкий. Поздравим себя.
А. Н. Толстой. Гиперболоид инженера Гарина.
Джон Голсуорси. Сага о Форсайтах.
Агата Кристи. Убийство на поле для гольфа.
В. Ю. Драгунский. Главные реки Америки.
Н. А. Добролюбов. Луч света в темном царстве.

ДАЧНЫЕ МОТИВЫ В ФИЗИКО-МАТЕМАТИЧЕСКИХ ЭТЮДАХ

Римские числа на бревнах строящейся бани — этюд 3.
Прослушанный на веранде дачи рассказ «Золотой жук» — этюд 3.
Кирпич, подвешенный на электрический провод, — этюд 7.
Взгляд на кирпичную кладку стены — этюд 7.
Книжка-игрушка, найденная на чердаке дачи, быстро перелистывая которую можно видеть анимацию, — этюды 8 и 13.
Портрет пожарного ведра на пожарном щите дачного участка — этюд 12.
Тропинка от дачного домика к хозблоку — этюд 15.
Пролет журавлиного клина над дачей[1] — этюд 17.
Ну и, конечно, все этюды писались в основном на дачах и других местах отдыха авторов книги. Кроме того, Пафнутий Львович Чебышёв, чья стопоходящая машина рассматривалась в этюде 8, родился в 10 километрах от дачи одного из авторов книги — в деревне Окатово Боровского уезда Калужской губернии.

[1] Журавли проплывали в облаках. В «облаках» же хранились тексты и картинки книги.

ИДЕИ АРТОБЪЕКТОВ С МАТЕМАТИЧЕСКИМ СМЫСЛОМ

1. Пряничный человечек — этюд 3, рис. 1.3, стр. 17
2. Пол в ванной комнате — этюд 4, рис. 4.28, стр. 88
3. Компьютерная картина Ван Гога — этюд 4, рис. 4.36, стр. 92
4. Кружево — этюд 4, рис. 4.44, стр. 97
5. Вышивка «Ришелье» — этюд 4, рис. 4.48, стр. 99
6. Ваяние скульптуры из отдельных точек — этюд 5, рис. 5.13, стр. 115
7. Реальная и виртуальная цепь на картине — этюд 7, рис. 7.4, стр. 143
8. Помещение для велосипедов с оптимизированной «цепной» крышей — этюд 7, стр. 156
9. Дед Мороз на лыжах — этюд 8, рис. 8.8, стр. 187
10. Часы Кеплера — этюд 9, рис. 9.11, стр. 209
11. Шкатулка с оптимальными размерами — этюд. 10, рис. 10.7, стр. 247
12. Кубики оригами — этюд 11, рис. 11.13-11.17
13. Mathcad-часы — реализованная идея; этюд 15, рис. 15.1, стр. 337
14. Часы-глобус — этюд 15, рис. 15.3, стр. 339
15. Часы с задачей коммивояжера — этюд 15, рис. 15.4 и 15.5, стр. 341
16. Монна Лиза с задачей коммивояжера — реализованная чужая идея; этюд 15, рис. 15.7, стр. 343
17. Велосипед Аполлония — этюд 19, рис. 19.8, стр. 383
18. Гитара Аполлона — этюд 19, рис. 19.10, стр. 385
19. Фигура с раскрашенным центром масс — этюд 21, рис. 21.7, стр. ???
20. Ваза-параболоид — этюд 20, стр. ???
21. Кубок победителю математической олимпиады — этюд 24, рис. 24.5, стр. 451

МАТЕМАТИЧЕСКИЕ АРТОБЪЕКТЫ, ОПИСАННЫЕ В КНИГЕ

1. Новогодняя открытка со стопоходящей машиной (этюд 8)
2. Часы Кеплера (этюд 9)
3. Часы с задачей коммивояжера (этюд 15)
4. Mathcad-часы (этюд 15)
5. Портрет Моны Лизы (этюд 15)
6. Ваза в виде параболоида (этюд 20)
7. Кубок победителю математической олимпиады (этюд 25)

ЛАБОРАТОРНЫЕ РАБОТЫ НА СТЫКЕ МАТЕМАТИКИ, ФИЗИКИ И ИНФОРМАТИКИ, ОПИСАННЫЕ В КНИГЕ

1. Создание шифра для передачи текстовых сообщений (этюд 3, рис. 3.6, стр. 56).
2. Определение оптимального размера реальной и виртуальной провисающей цепи (этюд 4, рис. 4.51, стр. 103).
3. Угадывание образа по растровому рисунку (этюд 5, рис. 5.6, стр.110).
4. Ваяние скульптуры из отдельных точек (этюд 5, рис. 5.13, стр. 115).
5. Создание дорожки для велосипеда с квадратными колесами (этюд 7, рис. 7.19, стр. 157).
6. Создание стопоходящей машины (этюд 8, рис. 8.8, стр. 187).
7. Реальная и виртуальная машина для голосования (этюд 11, рис. 11.18 и 11.19, стр. 271–272).
8. Построение плоских кривых второго порядка на бумаге и на экране дисплея (этюд 19).
9. Статистические испытания игры в кости (этюд 16, рис. 162, стр. 358).
10. Провисание реальной и компьютерной цепи с грузиками (этюд 21, рис. 21.1, стр. ???).
11. Определение остойчивости судна (этюд 23, рис. 23.4, стр. ???).
12. Градуировка ареометра (этюд 23, стр. ???).
13. Определение центра масс плоской фигуры сложной формы (два варианта — этюд 21 и этюд 23).
14. Суммирование на реостатах (этюд 24, рис. 24.2, стр. ???).

ЛИТЕРАТУРА И ИНТЕРНЕТ-ИСТОЧНИКИ

1. *Очков*, *В. Ф.* 24 этюда на Бейсике / В. Ф. Очков, Ю. В. Пухначев. — М. : Финансы и статистика, 1988. URL: http://twt.mpei.ac.ru/ochkov/24/index.htm.
2. Теплотехнические этюды с Excel, Mathcad и Интернет / под общ. ред. В. Ф. Очкова. — СПб. : БХВ-Петербург, 2015. URL: http://twt.mpei.ac.ru/ochkov/TTMI.
3. *Очков*, *В. Ф.* Формулы в научных публикациях: проблемы и решения // Cloud of Science. — 2014. — Т. 1. — № 3. — С. 421–456. URL: http://twt.mpei.ac.ru/ochkov/formula.
4. *Очков*, *В. Ф.* Физические и экономические величины в Mathcad и Maple. — М. : Финансы и статистика, 2002. — 192 с. — (Диалог с компьютером). URL: http://twt.mpei.ac.ru/ochkov/Units/Forword_book.htm.
5. *Коробов*, *В. И.* Химические расчеты в среде Mathcad / В. И. Коробов, В. Ф. Очков. — Днепропетровск : Изд-во Днепропетровского национального университета, 2012. — 216 с. URL: http://www.exponenta.ru/educat/news/ ochkov/book_ochkov11.asp.
6. *Очков*, *В. Ф.* Mathcad 14 для студентов и инженеров: русская версия. — СПб. : БХВ-Петербург, 2009. — 752 с. URL: http://twt.mpei.ac.ru/ochkov/Mathcad_14/RusIndex.html.
7. *Коробов*, *В. И.* Химическая кинетика: введение с Mathcad/Maple/MCS / В. И. Коробов, В. Ф. Очков. — М. : Горячая линия — Телеком, 2009. — 384 с. URL: http://twt.mpei.ac.ru/TTHB/New-Chem-Kin/En-Ru-book.html.
8. *Korobov*, *V.* Chemical Kinetics with Mathcad and Maple / V. Korobov, V. Ochkov. — Springer, 2011. — 344 p. URL: http://twt.mpei.ac.ru/TTHB/New-Chem-Kin/En-ChemKin.html.
9. *Очков*, *В. Ф.* Mathcad 14 для студентов, инженеров и конструкторов. — СПб. : БХВ-Петербург, 2007. — 368 с. URL: http://twt.mpei.ac.ru/ochkov/Mathcad_14/index.html.
10. *Очков*, *В. Ф.* Mathcad 12 для студентов и инженеров. — СПб. : БХВ-Петербург, 2005. — 450 с. URL: http://twt.mpei.ac.ru/ochkov/Mathcad_12
11. *Solodov*, *A.* Differential Models. An Introduction with Mathcad / A. Solodov, V. Ochkov. — Springer, 2004. — 232 p. URL: http://twt.mpei.ac.ru/ochkov/Diff_MC/index_Eng.htm.
12. *Воскобойников*, *Ю. Е.* Программирование и решение задач в пакете Mathcad / Ю. Е. Воскобойников, В. Ф. Очков. — Новосибирск : Изд-во НГАСУ, 2002. — 138 с. URL: http://twt.mpei.ac.ru/ochkov/Prog_Mathcad/index.html.
13. *Солодов*, *А. П.* Mathcad/Дифференциальные модели / А. П. Солодов, В. Ф. Очков. — М. : Изд-во МЭИ, 2002. — 239 с. URL: http://twt.mpei.ac.ru/ochkov/ Diff_MC.
14. *Очков*, *В. Ф.* Советы пользователям Mathcad. — М. : Изд-во МЭИ, 2001. — 196 с. URL: http://twt.mpei.ac.ru/ochkov/Sovet_MC.
15. *Очков*, *В. Ф.* Решение инженерно-технических задач в среде Mathcad // Справочник «Теплоэнергетика и теплотехника». — М. : Изд-во МЭИ, 1999. — Кн. 1. — Разд. 5.5.2. — 528 с.
16. *Очков*, *В. Ф.* Mathcad 8 Pro для студентов и инженеров. — М. : КомпьютерПресс, 1999. — 523 с. URL: http://twt.mpei.ac.ru/ochkov/mc8Pro.book.
17. *Очков*, *В. Ф.* Mathcad 7 Pro для студентов и инженеров. — М. : КомпьютерПресс, 1998. — 384 с. URL: http://twt.mpei.ac.ru/ochkov/mc7Pro.book.
18. *Очков*, *В. Ф.* Mathcad PLUS 6.0 для студентов и инженеров. — М. : КомпьютерПресс, 1996. — 230 с. URL: http://twt.mpei.ac.ru/ochkov/mc6Pro.book.
19. *Очков*, *В. Ф.* Расчетный документ инженерного калькулятора как структурная схема // Вестник МЭИ. — 2010. — № 5. — С. 118–125. URL: http://twt.mpei.ac.ru/ochkov/Mathcad_14/MathcadWorkSheetAsFlowChart.pdf.
20. *Очков*, *В. Ф.* Развитие интерфейса Mathcad: от DOS к Internet // Exponenta Pro. — 2004. — № 7–8. URL: http://twt.mpei.ac.ru/ochkov/Interface_Mathcad.
21. *Очков*, *В. Ф.* Mathcad Application Server: опыт эксплуатации в России // Каталог SoftLine Direct. — 2006. — № 11. URL: http://twt.mpei.ac.ru/ochkov/3MAS.

22. http://twt.mpei.ac.ru/ochkov/VPU_Book_New/mas/From_WorkSheet_to_WebSheet.html.
23. http://twt.mpei.ac.ru/MCS/Worksheets/rbtpp/tcon.sol.sod.hl.lf.xmcd.
24. *Кондакова*, *Г. Ю.* Справочное издание «Интернет-версия справочника Теплоэнергетика и теплотехника. Инструментальные средства создания и развития» / Г. Ю. Кондакова, А. С. Копылов, К. А. Орлов [и др.]. — М. : Издательский дом МЭИ, 2007. URL: http://twt.mpei.ac.ru/TTHB/5/index.html.
25. http://twt.mpei.ac.ru/TTHB/tthb.html.
26. www.smath.info.
27. http://mathworld.wolfram.com/GingerbreadmanMap.html.
28. *Тихонов*, *А. Н.* Уравнения математической физики / А. Н. Тихонов, А. А. Самарский. — М. : Наука, 1977. — 735 с.
29. http://twt.mpei.ac.ru/MAS/Worksheets/Chem/solutions.html.
30. http://twt.mpei.ac.ru/MCS/Worksheets/Thermal/T90-T68.xmcd.
31. *Очков*, *В. Ф.* Задачи по физике: новый подход к решению // Открытое образование. — 2012. — № 6. URL: http://twt.mpei.ac.ru/ochkov/Mathcad-15/Physic.pdf.
32. *Очков*, *В. Ф.* Преподавание математики и математические пакеты // Открытое образование. — 2013. — № 2. URL: http://twt.mpei.ac.ru/ochkov/Mathcad-15/OchkovMath.pdf.
33. *Очков*, *В. Ф.* Скаляр и вектор в компьютерных вычислениях // Открытое образование. — 2013. — № 4. URL: http://twt.mpei.ac.ru/ochkov/Mathcad-15/ScalarVector.pdf.
34. *Очков*, *В. Ф.* Римские и арабские // Наука и жизнь. — М., 1986. — № 4.
35. *Очков*, *В. Ф.* Римские-арабские / В. Ф. Очков, Е. П. Богомолова // Информатика в школе. — 2014. — № 8. URL: http://twt.mpei.ac.ru/ochkov/Arab-Roman.pdf.
36. *Очков*, *В. Ф.* Mathcad и криптография // Информатика в школе. — 2013. — № 10. URL: http://twt.mpei.ac.ru/ochkov/Mathcad-15/MATHCAD-CRYPTOGRAPHY.pdf.
37. http://www.ph4s.ru/book_kripto.html.
38. http://lib.ru/INOFANT/POE/Goldbug.txt.
39. http://lib.ru/AKONANDOJL/sh_dancm.txt.
40. О математике: проблемы преподавания / сост. А. Д. Ярцева, А. В. Чернавский. — М. : Знак, 2012. URL: ozon.ru/context/detail/id/19725947.
41. *Богомолова*, *Е. П.* Решатели, или Великолепная семерка Mathcad / Е. П. Богомолова, В. Ф. Очков, Х. Мати // Открытое образование. — 2015. — № 3. URL: http://twt.mpei.ac.ru/ochkov/Solvers-OE.pdf.
42. *Очков*, *В. Ф.* Решение алгебраических уравнений и систем или Ван Гог в среде Mathcad. — М. : КомпьютерПресс, 2001. — № 9. URL: http://twt.mpei.ac.ru/ochkov/Carpet/index.htm.
43. *Очков*, *В. Ф.* История одного шедевра // Компьютерные инструменты в образовании. — 2000. — № 3, 4. URL: http://twt.mpei.ac.ru/ochkov/Lace/Lace.htm.
44. *Очков*, *В. Ф.* Живые кинематические схемы в Mathcad // Открытое образование. — 2013. — № 3. URL: http://twt.mpei.ac.ru/ochkov/Mathcad-15/kinematic.html.
45. *Очков*, *В. Ф.* Угадай образ // Информатика в школе. — 2011. — № 9 (73). URL: http://twt.mpei.ac.ru/ochkov/Mathcad-15/SolveImage.pdf.
46. *Доценко*, *В.* Пятое правило арифметики // Наука и жизнь. — М., 2004. — № 12. URL: http://www.nkj.ru/archive/articles/457.
47. *Очков*, *В. Ф.* На компьютер надейся, а сам не плошай! / В. Ф. Очков, В. С. Дробилко // Информатика в школе. — 2012. — № 3 (76). URL: http://twt.mpei.ac.ru/ochkov/Mathcad-15/USA.pdf.
48. *Карпушина*, *Н.* Проценты простые и сложные. Уроки арифметики в классической литературе // Наука и жизнь. — М., 2009. — № 9. URL: http://www.nkj.ru/archive/articles/16385.
49. *Меркин*, *Д. Р.* Введение в механику гибкой нити. — М. : Наука, 1980. — 240 с. URL: http://www.twirpx.com/file/357235.
50. *Очков*, *В. Ф.* Что нам стоит... мост построить. — М. : КомпьютерПресс, 2000. — № 3. URL: http://twt.mpei.ac.ru/ochkov/Bridge/Bridge.htm.

51. *Очков, В. Ф.* Активность на форумах — важная часть учебы и последующей инженерной деятельности студента / В. Ф. Очков, С. Герк // Открытое образование. — 2014. — № 5. URL: http://twt.mpei.ac.ru/ochkov/Ochkov-Gurke-OE-5-2014.pdf.
52. *Рабинович, И. М.* Курс строительной механики стержневых систем. — 1950. URL: http://dwg.ru/dnl/8150.
53. http://www.youtube.com/watch?feature=player_detailpage&v=yFSMLHovlak.
54. *Очков, В. Ф.* MCS на занятиях по математике, физике, информатике... // Компьютерные учебные программы и инновации. — 2008. — № 3. URL: http://twt.mpei.ac.ru/ochkov/Pendulum/index.html.
55. *Очков, В. Ф.* MAS на занятиях по математике, физике, информатике... // Компьютерные учебные программы и инновации. — 2006. — № 2. URL: http://twt.mpei.ac.ru/ochkov/Mathcad_12/Planet/index.html.
56. *Бутиков, Е. И.* Пакет компьютерных программ «Движение космических тел». URL: http://butikov.faculty.ifmo.ru/Planets/index.html.
57. *Chenciner, A.* A remarkable periodic solution of the three-body problem in the case of equal masses / A. Chenciner, R. Montgomery // Annals of Mathematics. — 152 (2000). — P. 881–901. URL: http://arxiv.org/abs/math/0011268.
58. *Очков, В. Ф.* Обратная задача коммивояжера или Необычные математические часы / В. Ф. Очков, Е. П. Богомолова // Открытое образование. — 2014. — № 2. — С. 22–28. URL: http://twt.mpei.ac.ru/ochkov/SalesMan.
59. *Очков, В. Ф.* Интерполяция, экстраполяция, аппроксимация, или «Ложь, наглая ложь и статистика» / В. Ф. Очков, Е. П. Богомолова // Cloud of Science. — 2015. — Т. 2. — № 1. — С. 61–88. URL: http://twt.mpei.ac.ru/ochkov/CoS_2_1.pdf.
60. *Очков, В. Ф.* Это страшное слово дифуры... / В. Ф. Очков, Е. П. Богомолова // Информатика в школе. — 2015. — № 1. — С. 55–58. URL: http://twt.mpei.ac.ru/ ochkov/ODE.pdf.
61. *Ochkov, V.* Journey from St. Petersburg to Moscow or Model of Gravity Train in Mathcad / V. Ochkov, K. Pisačić // Technical Journal University North Croatia. — Vol. 9. — № 1. — P. 1–5. URL: http://twt.mpei.ac.ru/ochkov/Tunnel.pdf.
62. *Очков, В.* Путешествие из Петербурга в Москву, или Свет в конце туннеля / В. Очков, К. Писачич // Информатика в школе. — 2015. — № 4. URL: http://twt.mpei.ac.ru/ochkov/Tunnel.pdf.
63. *Очков, В. Ф.* Принцип неопределенности программирования. — М. : КомпьютерПресс, 1996. — № 7. URL: http://twt.mpei.ac.ru/ochkov/IZBYT.htm.
64. *Очков, В. Ф.* Mathcad и нечеткие множества // КомпьютерПресс. — 1998. — № 1. URL: http://twt.mpei.ac.ru/ochkov/F_sets.htm.
65. *Очков, В. Ф.* Mathcad и нечеткая логика // КомпьютерПресс. — 1998. — № 8. URL: http://twt.mpei.ac.ru/ochkov/F_log.htm.
66. *Кузнецов, А. А.* Об информатике, ее подходах и предмете (философия информатики) / А. А. Кузнецов, С. А. Бешенков, Т. Б. Захарова [и др.] // Вестник Тамбовского университета. — 2005. — Т. 10. — № 3. — С. 236–249. — (Естественные и технические науки).
67. *Кузнецов, А. А.* Содержание обучения информатике в основной школе: на пути к фундаментализации / А. А. Кузнецов, И. В. Левченко, О. Ю. Заславская [и др.] // Вестник Московского городского педагогического университета. — 2010. — № 20. — С. 6–18. — (Информатика и информатизация образования).
68. *Коротенков, Ю. Г.* Обучение информатике и математике в рамках единой образовательной области «Математика и информатика» // Информатика и образование. — 2014. — № 8 (257). — С. 36–39.
69. *Очков, В. Ф.* Цена подержанного автомобиля, или Путь от корреляции к регрессии в среде Mathcad. — М. : КомпьютерПресс, 2001. — № 9. URL: http://twt.mpei.ac.ru/ochkov/car/car.htm.
70. *Богомолова, Е. П.* Проблемы оценивания результатов ЕГЭ по математике / Е. П. Богомолова, О. В. Максимова // Alma Mater. Вестник высшей школы. — 2014. — № 9. — С. 56–60.

71. *Богомолова*, *Е. П.* Влияние компьютерной поддержки математики на успеваемость студентов технических вузов / Е. П. Богомолова, О. В. Максимова // Открытое образование. — 2014. — № 6. — С. 65–71.
72. http://lib.ru/HARMS/staruha.txt.
73. http://az.lib.ru/t/tolstoj_lew_nikolaewich/text_0090.shtml.
74. http://www.tsp.gatech.edu.
75. *Мостеллер*, *Ф.* Пятьдесят занимательных вероятностных задач с решениями : пер. с англ. — М. : Наука, 1975.
76. *Гудман*, *С.* Введение в разработку и анализ алгоритмов : пер. с англ. / С. Гудман, С. Хидетниеми. — М. : Мир, 1975.
77. *Очков*, *В. Ф.* Две задачи, в решение которых внес коррективы компьютер / В. Ф. Очков, Ю. В. Пухначев // Программные продукты и системы. — 1989. — № 2.
78. *Очков*, *В. Ф.* Задача о рыбаках и рыбке / В. Ф. Очков, Е. А. Селиванов // Информатика в школе. — 2012. — № 10. URL: http://twt.mpei.ac.ru/ochkov/Mathcad-15/Fish.pdf.
79. *Очков*, *В. Ф.* На компьютер надейся, а сам не плошай! / В. Ф. Очков, В. С. Дробилко // Информатика в школе. — 2012. — № 3 (76). — С. 62–63. URL: http://twt.mpei.ac.ru/ochkov/Mathcad-15/USA.pdf.
80. http://www.etudes.ru/ru/imath.
81. *Очков*, *В. Ф.* Оптимизированный фрактал или ФМИ / В. Ф. Очков, Я. Калова, Е. В. Никульчев // Cloud of Science. — 2015. — Т. 2. — № 4. — С. 544–561. URL: http://twt.mpei.ac.ru/ochkov/Opt_Fractal.pdf.
82. *Очков*, *В. Ф.* Литературно-физическая композиция «Истории о зеркале и линзе» / В. Ф. Очков, Я. Калова, А. В. Соколов, Ю. В. Чудова // Открытое образование. — 2016. — № 1. — С. 34–42. URL: http://twt.mpei.ac.ru/ochkov/Mirror-Lens.pdf.
83. *Очков*, *В. Ф.* ...и на Марсе будут яблони цвести / В. Ф. Очков, Ю. В. Чудова, А. В. Соколов // Информатика в школе. — 2016. — № 2. — С. 24–28. URL: http://twt.mpei.ac.ru/ochkov/Rocket.pdf.
84. *Ochkov*, *V.* Thermal Engineering Studies with Excel, Mathcad and Internet / V. Ochkov, K. Orlov, V. Voloshchuk, N. Rogalev [Ed.] // Springer. — 2016. URL: https://www.ptcusercommunity.com/groups/thermal.
85. *Очков*, *В. Ф.* Как корабль рассчитаете и назовете, так он и поплывет / В. Ф. Очков, А. В. Соколов // Информатика в школе. — 2016. — № 4. — С. 56–59. URL: http://twt.mpei.ac.ru/ochkov/Ship.pdf.
86. J. Dennis Lawrence. Catalog of special plane curves. Dover Publications. — 1972.
87. *Lockwood*, *E. H.* Book of Curves. — Cambridge University Press, 2007.
88. *Rutter*, *J. W.* Geometry of Curves. — CRC Press, 2000.
89. *Чертов*, *А. Г.* Физические величины. — М. : Высшая школа, 1990.
90. *Очков*, *В. Ф.* Вода в научном, культурологическом и религиозном аспектах // Водоочистка. Водоподготовка. Водоснабжение. — 2009. — № 7. URL: http://twt.mpei.ac.ru/ochkov/WSCR/WaterSCR.pdf.
91. *Александров*, *А. А.* Теплофизические свойства рабочих веществ теплоэнергетики: интернет-справочник / А. А. Александров, К. А. Орлов, В. Ф. Очков. — М. : Издательский дом МЭИ, 2009. URL: http://twt.mpei.ac.ru/rbtpp.
92. Основы остойчивости морского судна: учебное пособие / Ф. М. Кацман, Б. П. Коваленко. — СПб. : ГМА им. адм. С. О. Макарова, 2003.
93. *Тетельбаум*, *И. М.* 400 схем для АВМ / И. М. Тетельбаум, Ю. Р. Шнейдер. — М. : Энергия, 1978.
94. *Ахиезер*, *Н. И.* Лекции по вариационному исчислению. — М. : Гостехиздат, 1955.

СОДЕРЖАНИЕ

Предисловие 3

Введение
Преподавание информатики, математики, физики, химии
и математические пакеты, или Между Сциллой и Харибдой 6

Этюд 1
Вычислительный ренессанс, или Ружье на сцене 12
Дивертисмент. Как высоко мы поднялись в свободном падении? 36

Этюд 2
Лебедь, Рак и Щука,
или Скаляр и вектор в компьютерных вычислениях 40

Этюд 3
Золотой жук, или Римско-арабская криптография 50
Дивертисмент. О математике и математиках 59

Этюд 4
Решатели, или Великолепная семерка Mathcad 63
Дивертисмент 1. Ван Гог в среде Mathcad 84
Дивертисмент 2. О бедном слове Given замолвите слово 100

Этюд 5
Угадай образ, или Игра в матрицу 105
Дивертисмент 1. Колумбово яйцо, или точечная скульптура 112
Дивертисмент 2. Пасьянс 115
Дивертисмент 3. И еще об угадывании мелодии 122

Этюд 6
Что почем, или Старая задача на новый лад 123

Этюд 7
Цепная функция, или Пятый элемент 137
Дивертисмент 1. Квадратное колесо 156
Дивертисмент 2. Реабилитация параболы 171

Этюд 8
Живые кинематические схемы, или Рука Геллы 178
Дивертисмент 1. Рука Геллы 188
Дивертисмент 2. Дверь в мир математики и физики 190

Этюд 9
Небесная механика, или Часы Кеплера 197
Дивертисмент 1. Беззаконная комета 215
Дивертисмент 2. Путешествие из Петербурга в Москву,
или Свет в конце туннеля 227
Дивертисмент 3. Сила, масса, импульс, ускорение... 235

Этюд 10
Рекурсия, или Ход конем 240
Дивертисмент. Оптимизированный фрактал 247

Этюд 11
Да — Нет в среде Mathcad, или Оригами Буля 256

Этюд 12
Нечеткое множество, или Оптимальное пожарное ведро 273
Дивертисмент. Полностью размерная функция 290

Этюд 13
Путешествие окружности в треугольнике, а треугольника
в ложбине, или Сам себе компьютерный режиссер 293

Этюд 14
Интерполяция, экстраполяция и сглаживание,
или Ложь, наглая ложь и статистика... 308

Этюд 15
Пойти туда, зная куда, или Новые швейцарские часы 336

Этюд 16
Орел — решка, или Трехсторонняя дуэль в Монте-Карло 356

Этюд 17
Летят гуси, или Задача о рыбаках и рыбке 366

Этюд 18
Буквенная головоломка,
или На компьютер надейся, а сам не плошай! 371

Этюд 19
Семь вычислительных кривых,
или Велосипед Аполлония и гитара Аполлона 376
Дивертисмент. Жук второго порядка 394

Этюд 20
Путешествие от дома к школе по маршруту ферма,
или Второе оптическое свойство гиперболы 398
Дивертисмент 1. Лупа для чтения физико-математических
справочников 408
Дивертисмент 2. Теорема Ферма и Mathcad 411
Дивертисмент 3. Матричная оптика 413

Этюд 21
Оптимизация потенциальной энергии, или Еще раз про цепь 421
Дивертисмент. Центр тяжести... 426

Этюд 22
Параллельные вычисления, или По порядку становись! 430
Дивертсмент. Две случайные «случайные» функции Mathcad 435

Этюд 23
Плотность соленой воды,
или Как корабль рассчитаете и назовете, так он и поплывет 438
Дивертисмент 1. Центр масс, центр тяжести... 443
Дивертисмент 2. Рыба ищет где глубже, а человек... в Интернете 446

Этюд 24
Расчеты на ~~салф~~таблетке, или ...И на Марсе будут яблони цвести 448
Дивертисмент. Закон радиоактивного распада 453

Этюд 25
Гибридные расчеты на компьютере, или Еще раз про конус 458

Задачи, решенные в книге и на сайте книги 469

Произведения литературы и изобразительного искусства, упоминаемые в книге 472

Дачные мотивы в физико-математических этюдах 473

Идеи артобъектов с математическим смыслом 474

Математические артобъекты, описанные в книге 475

Лабораторные работы на стыке математики, физики и информатики, описанные в книге 476

Литература и интернет-источники 477

Валерий Федорович ОЧКОВ,
Елена Петровна БОГОМОЛОВА,
Дмитрий Александрович ИВАНОВ

ФИЗИКО-МАТЕМАТИЧЕСКИЕ ЭТЮДЫ С MATHCAD И ИНТЕРНЕТ

Учебное пособие

Издание второе, исправленное и дополненное

Зав. редакцией естественнонаучной литературы *М. В. Рудкевич*
Выпускающие *С. Ю. Гаганов,* _____

ЛР № 065466 от 21.10.97
Гигиенический сертификат 78.01.10.953.П.1028 от 14.04.2016 г., выдан ЦГСЭН в СПб

Издательство «ЛАНЬ»
lan@lanbook.ru; www.lanbook.com
196105, Санкт-Петербург, пр. Ю. Гагарина, д. 1, лит. А.
Тел./факс: (812) 336-25-09, 412-92-72.
Бесплатный звонок по России: 8-800-700-40-71

ГДЕ КУПИТЬ

ДЛЯ ОРГАНИЗАЦИЙ:

Для того, чтобы заказать необходимые Вам книги, достаточно обратиться в любую из торговых компаний Издательского Дома «ЛАНЬ»:

по России и зарубежью
«ЛАНЬ-ТРЕЙД». 196105, Санкт-Петербург, пр. Ю. Гагарина, д. 1, лит. А.
тел.: (812) 412-85-78, 412-14-45, 412-85-82; тел./факс: (812) 412-54-93
e-mail: trade@lanbook.ru; ICQ: 446-869-967

www.lanbook.com
пункт меню **«Где купить»**
раздел **«Прайс-листы, каталоги»**

в Москве и в Московской области
«ЛАНЬ-ПРЕСС». 109263, Москва, 7-я ул. Текстильщиков, д. 6/19
тел.: (499) 178-65-85; e-mail: lanpress@lanbook.ru

в Краснодаре и в Краснодарском крае
«ЛАНЬ-ЮГ». 350901, Краснодар, ул. Жлобы, д. 1/1
тел.: (861) 274-10-35; e-mail: lankrd98@mail.ru

ДЛЯ РОЗНИЧНЫХ ПОКУПАТЕЛЕЙ:

интернет-магазин
Издательство «Лань»: http://www.lanbook.com

магазин электронных книг
Global F5: http://globalf5.com/

Подписано в печать __.__.17.
Бумага офсетная. Гарнитура Школьная. Формат 70×100 $^{1}/_{16}$.
Печать офсетная. Усл. п. л. 39,33. Тираж ___ экз.

Заказ № ___.

Отпечатано в полном соответствии
с качеством предоставленного оригинал-макета
в ПАО «Т8 Издательские Технологии».
109316, г. Москва, Волгоградский пр., д. 42, к. 5.