

ИНТЕРПОЛЯЦИЯ

Методы
и компьютерные
технологии
их реализации

А. М. Половко
П. Н. Бутусов



Анатолий Половко

Павел Бутусов

Интерполяция

**Методы и компьютерные технологии
их реализации**

Санкт-Петербург

«БХВ-Петербург»

2004

УДК 618.3.06
ББК 32.973
П52

Половко А. М., Бутусов П. Н.

П52 Интерполяция. Методы и компьютерные технологии их реализации. — СПб.: БХВ-Петербург, 2004. — 320 с.: ил.

ISBN 5-94157-493-2

Книга содержит основы теории интерполяции: приведена классификация методов, рассматриваются точные и приближенные, однопараметрические и многопараметрические методы. Методика и компьютерные технологии иллюстрируются на примерах решения задач с помощью универсальных программных средств символьной математики Mathematica, Maple, Derive, Mathcad, MATLAB. Впервые описаны программные средства, позволяющие автоматизировать выбор вида функции интерполяции (CurveExpert, TableCurve, Simple Formula). Материал сопровождается большим количеством примеров и решением задач из различных областей знаний: химии, биологии, экономики, теории и практики надежности, эдукологии, теории массового обслуживания, таксации.

*Для преподавателей, студентов и специалистов,
занимающихся проблемами моделирования, планирования
и статистической обработки эксперимента*

УДК 618.3.06
ББК 32.973

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Римма Смоляк</i>
Компьютерная верстка	<i>Екатерины Трубниковой</i>
Корректор	<i>Евгений Камский</i>
Дизайн обложки	<i>Игоря Цырульникова</i>
Зав. производством	<i>Николай Тверских</i>

Рецензенты:

Доктор технических наук, профессор Санкт-Петербургского института государственной противопожарной службы МЧС России Щербаков О. В.

Доктор технических наук, профессор Санкт-Петербургской государственной лесотехнической академии им. С. М. Кирова Гуров С. В.

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 25.03.04.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 25,80.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02 от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-493-2

© Половко А. М., Бутусов П. Н., 2004
© Оформление, издательство "БХВ-Петербург", 2004

Содержание

Предисловие	7
Введение	9
1. Для кого эта книга	9
2. Научное и прикладное содержание книги	13
3. Понятие компьютерная технология интерполяции	14
4. Рациональная методика изучения компьютерных технологий интерполяции	15
Глава 1. Основы теории интерполяции	17
1.1. Что такое интерполяция	17
1.2. Интерполяция точная в узлах	19
1.2.1. Общий метод решения задачи	19
1.2.2. Интерполяционные полиномы	22
1.2.3. Интерполяционная формула Лагранжа	23
1.2.4. Табличные разности	24
Центральные разности	26
1.2.5. Интерполяционные формулы Ньютона	27
Интерполяционные формулы Ньютона при неравноотстоящих узлах	30
1.2.6. Интерполяционные формулы Гаусса	31
1.2.7. Интерполяционная формула Стирлинга	32
1.2.8. Интерполяционная формула Бесселя	33
1.2.9. Сплайн-интерполяция	33
1.3. Интерполяция приближенная в узлах	35
1.3.1. Метод выбранных точек	42
1.3.2. Метод средних	43
1.3.3. Метод наименьших квадратов	44
1.3.4. Аппроксимация Паде	48
Глава 2. Компьютерные технологии решения задач интерполяции в системе Derive 5	51
2.1. Интерполяция точная в узлах. Универсальный метод	51
2.2. Интерполяция по методу Лагранжа. Функция POLY_INTERPOLATE	63
2.3. Интерполяция точная в узлах нелинейными функциями	64
2.4. Интерполяция приближенная в узлах	67
2.5. Аппроксимация многими функциями	71

2.6. Аппроксимация Паде	72
2.7. Решение задач интерполяции путем прямых вычислений	74
Глава 3. Компьютерные технологии решения задач интерполяции в системе Mathematica	77
3.1. Интерполяция точная в узлах. Универсальный метод	77
3.1.1. Функция Solve.....	77
3.1.2. Функция NSolve.....	78
3.1.3. Функция FindRoot.....	80
3.1.4. Решение задачи интерполяции универсальным методом	81
3.2. Интерполяция точная в узлах. Функция InterpolatingPolynomial	85
3.3. Интерполяция нелинейными функциями.....	87
3.4. Функция Interpolation [data].....	91
3.5. Интерполяция приближенная в узлах.....	92
3.6. Аппроксимация Паде	96
3.7. Компьютерные технологии интерполяции и графическое представление функций.....	97
Глава 4. Компьютерные технологии решения задач интерполяции в среде Maple	101
4.1. Интерполяция точная в узлах. Универсальный метод.....	101
4.2. Интерполяция точная в узлах (функция interp)	111
4.3. Интерполяция приближенная в узлах (аппроксимация).....	112
4.4. Интерполяция нелинейными функциями.....	114
4.5. Аппроксимация Паде	120
4.6. Аппроксимация Паде при помощи полиномов Чебышева	121
4.7. Сплайн-интерполяция	122
4.8. Представление функции интерполяции полиномами	124
Глава 5. Компьютерные технологии решения задач интерполяции в системе Mathcad	127
5.1. Интерполяция точная в узлах. Универсальный метод.....	127
5.2. Кусочно-линейная интерполяция.....	141
5.3. Сплайн-интерполяция	142
5.4. Линейная аппроксимация	144
5.5. Полиномиальная аппроксимация.....	146
5.6. Аппроксимация линейной комбинацией функций.....	148
5.7. Аппроксимация нелинейной функцией.....	150
Глава 6. Компьютерные технологии решения задач интерполяции в системе Matlab	153
6.1. Начальные сведения о системе Matlab	153
6.2. Интерполяция точная в узлах. Универсальный метод.....	163
6.3. Интерполяция нелинейными функциями.....	172
6.4. Сплайн-интерполяция	173
6.5. Интерполяция точная в узлах (функция interp1)	174
6.6. Интерполяция приближенная в узлах (аппроксимация).....	176
6.7. Полиномиальная аппроксимация.....	177
6.8. Интерполяция кубическими полиномами.....	180

Глава 7. Компьютерные технологии многопараметрической интерполяции....	181
7.1. Введение.....	181
7.2. Интерполяция функций многих переменных в системе Derive	182
7.3. Интерполяция функций многих переменных в системе Maple	185
7.4. Интерполяция функций многих переменных в системе Mathematica	187
7.5. Интерполяция функций многих переменных в системе Mathcad	194
7.6. Интерполяция функций многих переменных в системе Matlab	195
7.7. Многопараметрическая интерполяция точная в узлах.....	196
Глава 8. Интерполяция в нашей профессии	205
8.1. Интерполяция – научная основа моделирования	205
8.2. Интерполяция и компьютерные технологии в задачах физики.....	206
8.3. Интерполяция в экономических задачах.....	218
8.4. Интерполяция в задачах таксации	229
8.5. Интерполяция в задачах массового обслуживания	234
8.6. Интерполяция в задачах надежности.....	242
8.6.1. Оценка надежности техники по опытным данным.....	243
8.6.2. Определение показателей надежности по данным эксплуатации восстанавливаемой техники	247
8.7. Интерполяция в обучении	251
Глава 9. Автоматизация решения задач интерполяции.....	263
9.1. Программа TableCurve 2D	264
9.2. Программа CurveExpert 1.3.....	273
9.3. Программа SIMPLE FORMULA.	281
Приложения	287
Приложение 1. Абстрактные математические задачи.....	291
Приложение 2. Задачи интерполяции в физике и химии	295
Приложение 3. Задачи интерполяции в экономике.....	303
Приложение 4. Задачи интерполяции в таксации.....	307
Приложение 5. Задачи интерполяции в надежности.....	309
Приложение 6. Задачи интерполяции в системах массового обслуживания	313
Список литературы.....	317
Предметный указатель.....	319

Предисловие

Законы природы это модели ее явлений. Этапами процесса открытия этих законов человеком являются: наблюдения, размышления, эксперимент, открытие. Результатами эксперимента могут быть: таблицы, диаграммы, графики, которые характеризуют изучаемый объект числом или совокупностью чисел и не могут быть физическими законами. Только математическая модель объекта в виде формулы может быть законом. Интерполяция, теория размерностей и теория подобия — это научные основы моделирования. При этом интерполяция является тем мостом, который объединяет опыт и знание, эксперимент и открытие.

Только через познание настоящего можно "путешествовать" в прошлое и будущее. Открыв закон можно прогнозировать. В вопросах прогнозирования интерполяция посредством полученных ею математических моделей служит превосходным гидом в научном предвидении.

Нельзя проводить эксперимент без плана, наугад. Нужно знать объект изучения и предполагаемую его математическую модель. Этим определяется объем эксперимента и план его проведения. Интерполяция требует от экспериментатора знания вида функции интерполяции. Здесь она строгий и требовательный путеводитель проведения эксперимента.

Области применения интерполяции — открытие и уточнение законов природы, прогнозирование, планирование и обработка данных эксперимента, моделирование, управление различными объектами и т. п. При этом интерполяция не имеет ограничений применения, ей подвластны любые области знаний, любые профессии. Широта использования, практическая направленность и значимость — вот основные черты интерполяции.

Интерполяция капризна. Для решения ее задач недостаточно знать математические методы и алгоритмы, уметь программировать, уметь работать на персональном компьютере, знать функции и команды интерполяции универсальных программных средств символьной математики. Необходимо еще знать компьютерные технологии интерполяции. Без этих знаний, решая задачу интерполяции, можно получить не математическую модель изучаемого

явления, а всего лишь математическое выражение, не имеющее никакого смысла.

Приобретя эту книгу, читатель будет научно обоснованно решать задачи интерполяции из любой области знаний. При этом он может воспользоваться любой из следующих математических систем: Mathenatica, Maple, Derive, Mathcad, Matlab.

Кроме того, он сможет существенно облегчить свой труд, применяя программные средства автоматизации TableCurve, CurveExpert, SIMPLE FORMULA, которые в книгах по компьютерным технологиям решения задач публикуются впервые.

Авторы благодарны профессорам С. В. Гурову, О. В. Щербакову за замечания, рекомендации и пожелания, высказанные ими в рецензиях на рукопись книги. Большинство из них учтены при окончательном редактировании книги. Большая благодарность Идалии Игнатьевне Кононец за редактирование рукописи в процессе ее написания. Спасибо всем сотрудникам редакции за содействие в издании книги.

Авторы будут очень признательны читателям, которые пришлют свои замечания и пожелания в адрес редакции.

Введение

1. Для кого эта книга

Ответить на этот вопрос трудно, потому что она нужна людям многих профессий и специальностей.

Она необходима студентам, изучающим такие предметы, как информатика, моделирование, компьютерные технологии, прикладная математика, планирование и обработка результатов эксперимента, а также и множество других специальных дисциплин.

Книга нужна студентам многих технических специальностей при проведении лабораторных работ по предметам естественно-научного и специального цикла дисциплин, когда требуется получать характеристики исследуемого объекта в виде таблиц и графиков. В физике — при изучении закономерностей явлений природы, в химии — при исследовании реакций, в электротехнике — при исследовании характеристик электрических машин, в таксации — при исследовании характеристик лесных ресурсов, в теории надежности — при обработке данных об отказах техники и т. д. и т. п. Компьютерные технологии интерполяции необходимы студентам экономических специальностей при изучении законов ценообразования, при оценке экономической эффективности управления предприятием.

Эта книга полезна также преподавателям дисциплин, в которых интерполяция используется как математический аппарат исследования. Они найдут варианты задач из различных областей знаний, необходимые при постановке и проведении лабораторных работ и упражнений.

Кроме того, книга может оказать помощь тем, кому приходится решать математические задачи на ПК. При решении задачи интерполяции попутно приходится иметь дело с векторами и матрицами, вычислять значения функций, решать системы линейных и нелинейных уравнений, строить графики и по их виду определять функции, вычислять погрешности расчетов.

Решая задачу интерполяции, пользователь ПК изучает одновременно соответствующее программное средство символьной математики, которое необходимо ему для решения и других математических задач. Таким образом, интерполяция может служить средством изучения универсальных математических систем, таких как Mathematica, Maple, Derive, Mathcad, Matlab. При этом процесс решения задач интерполяции исключительно увлекательное занятие, похожее на компьютерные игры. Действительно, вам предлагается таблица данных, а вы должны найти аналитическую функцию, которая может быть физическим законом изучаемого явления. Это даже интересней, чем игра в шахматы или преферанс.

Книга необходима научным работникам, исследования которых требуют применения интерполяции как математического аппарата моделирования при планировании, обработке данных эксперимента и иных математических расчетов.

Для физика-теоретика интерполяция — это один из способов отыскания законов физических явлений. Его сущность состоит в том, чтобы по данным эксперимента найти математическую модель объекта исследования, которая при определенных условиях является физическим законом. Методы интерполяции позволяют найти множество функций, которые будут с высокой точностью аппроксимировать данные таблицы. Однако среди этого множества есть только одна функция (если она есть вообще), которая является физическим законом и которую нужно распознать. Интерполяция подсказывает исследователю физический закон, который потом нужно доказать теоретически. Приведем один типичный пример.

Нами в *главе 8* решена задача интерполяции, в которой надо было найти зависимость температуры кипения воды от давления. Среди множества возможных функций найдена одна, которая соответствует минимальной погрешности. Эта функция имеет вид: $T = T_0 \sqrt[4]{P}$.

Здесь:

□ P — давление в атм.;

□ T_0 — температура кипения воды при $P = 1$ атм.

Полученная формула может быть физическим законом, если она будет доказана теоретически. Доказательство необходимо потому, что могут существовать и другие математические функции, описывающие данное явление с высокой точностью. Например, кубическая сплайн-интерполяция позволяет получить почти с нулевой погрешностью зависимость температуры кипения воды от давления, однако при этом математическое выражение не будет физическим законом.

Интерполяция может служить инструментом проверки истинности закона, полученного теоретически. Типичным примером является задача о расши-

рении Вселенной, решенная в *главе 8*. Интерполяция подтвердила сформулированный астрономом Хабблом в 1929 году линейный закон расширения Вселенной, а кроме того, позволила уточнить вид линейной функции. По Хабблу, скорость V удаления галактик от Земли пропорциональна расстоянию R , т. е. $V=KR$, где K — коэффициент Хаббла. Обработка экспериментальных данных путем решения задачи интерполяции дает следующий линейный закон: $V=KR+a$ (где a — *некая постоянная величина*). Доказательства или опровержения этого уточнения закона Хаббла пока астрономами не получено.

Книга будет полезна экономистам при решении практических задач. Экспериментальные данные при экономических анализах обычно представляются в форме таблиц, диаграмм, графиков. При этом эксперимент проводится пассивный (т. е. на объекте при обычном его функционировании). Активный эксперимент (т. е. на объекте при специальных условиях его работы) в экономике практически не проводится, т. к. он требует больших денежных затрат и в большинстве случаев бывает рискованным. Представление данных в виде таблиц, диаграмм или графиков позволяет лишь качественно судить об экономическом состоянии исследуемого объекта. Для оценок количественных необходимо иметь математическую модель в виде формул, которые возможно вывести путем решения задачи интерполяции. Полученная математическая модель может заменить активный эксперимент. При этом цена модели в сотни и тысячи раз дешевле активного эксперимента, да к тому же и никакого риска.

Круг экономических задач, требующих разработки математических моделей методами интерполяции, велик. Это задачи ценообразования, оптимальной платы за услуги, анализа влияния различных факторов на экономическую эффективность производства, транспортные задачи и многие другие.

В качестве примера в *главе 8* с помощью методов интерполяции решена задача оптимальной платы за посещение леса. Оказалось, что для получения максимальной прибыли цена разового посещения людьми леса в зимнее время должна быть 14.26 рублей.

Аналогом этой задачи может быть определение оптимальной стоимости за различные услуги (например, обучение, услуги парикмахерской, платной стоянки автомобилей, прачечной и т. п.).

Книга будет полезна инженерам различных специальностей. В качестве примеров в *главе 8* приводятся решения задач из теории массового обслуживания, таксации, теории и практики надежности, оценки знаний и установления рейтинга обучаемых. Рассмотрим вкратце результаты этих решений.

Отказы техники — явления случайные. Они характеризуются законами распределения времени до момента отказа невосстанавливаемой техники и между моментами отказа восстанавливаемой техники. Законы распределения получают по экспериментальным данным об отказах или данным эксплуа-

тации. При этом статистические данные являются функциями числа отказов от времени, представленными в виде таблиц. Интерполяция табличных данных позволяет получить функцию, которая является законом распределения времени до момента отказа техники. Анализ статистических данных об отказах методами интерполяции позволяет получить и другие характеристики надежности (например, интенсивность отказов, вероятность безотказной работы в течение времени t , параметр потока отказов).

Решенная в *главе 8* задача оценки надежности самолета ТУ-154М с применением метода интерполяции является одним из ярких примеров использования данных методов для анализа показателей надежности техники.

Интерполяция является наиболее важным математическим аппаратом для решения задач таксации. Данные о характеристиках лесов всегда представляются в табличной форме, где можно увидеть зависимость хода роста древостоев, плотности древесины, выхода фанерного и другого сырья и пр.

Получение математических моделей в задачах таксации возможно главным образом с помощью методов интерполяции. В *главе 8* приводится решение двух задач, одна из которых относится к многопараметрической интерполяции. Впервые получены линейная и квадратичная функции, выражающие зависимость среднего прироста древостоя ели от ее возраста, высоты, диаметра.

Особую роль играет интерполяция в тех случаях, когда некоторая сложная математическая функция анализируется в узком диапазоне аргумента. В таких случаях целесообразно (а часто необходимо) представить сложную функцию в узком диапазоне аргументов более простой. Это представление осуществляется путем решения задачи интерполяции.

В *главе 8* этим методом решены две задачи из области массового обслуживания, позволившие получить важные практические результаты. Получена математическая модель, выражающая зависимость необходимого числа обслуживающих органов от интенсивности потока заявок и интенсивности обслуживания, обеспечивающих заданную вероятность того, что заявка будет принята на обслуживание в произвольный момент времени.

Решена задача многопараметрической интерполяции. Получена функция регрессии, выражающая зависимость времени переходного процесса системы массового обслуживания от интенсивности потока заявок, интенсивности обслуживания и числа обслуживающих органов.

Результаты, полученные в этих двух задачах, трудно переоценить. Методики решения задач могут быть применены при изучении различных физических процессов. Это — исследование динамики систем управления, переходных процессов в электрических цепях, надежности сложных восстанавливаемых систем и многих других процессов, протекающих во времени.

Задачи, описанные и решенные в *главе 8*, лишь небольшая часть проблем, решаемых методами интерполяции. Область применения интерполяции как математического аппарата, значительно шире, чем приведенные примеры.

Наши примеры только иллюстрация компьютерных технологий решения задач интерполяции с применением универсальных математических программных средств символьной математики (хотя некоторые из примеров имеют самостоятельное значение).

2. Научное и прикладное содержание книги

Теория интерполяции совместно с теорией подобия и размерностей является научной основой моделирования, которое весьма полезно, а во многих случаях просто необходимо. Необходимость моделирования определяется следующими моментами:

- ☐ исследуемый объект слишком большой (корабль) или слишком мал (атом);
- ☐ объект удален от исследователя (галактика);
- ☐ объект недостижим во времени (был в прошлом или будет в будущем);
- ☐ объект опасен для исследователя (агрессивная среда);
- ☐ эксперимент слишком дорогой.

Во всех этих и многих других случаях, перечисление которых заняло бы много страниц, получение математической модели изучаемого объекта требует решения задач интерполяции.

Основное содержание книги — это компьютерные технологии решения прикладных задач из различных областей знаний, где требуется получение математических моделей объекта или явления.

Книга содержит теоретические основы интерполяции, практические задачи из физики, химии, экономики, теории и практики надежности, таксации, обучения.

Приведем некоторые практические вопросы, на которые может дать ответ книга, если воспользоваться ее методологией и компьютерной технологией решения задач интерполяции:

- ☐ Какова должна быть цена товара для получения максимальной прибыли?
- ☐ Каков курс валюты будет завтра, через неделю, через месяц и стоит ли освобождаться от доллара и покупать евро?
- ☐ Каков закон расширения Вселенной?
- ☐ Как оценить знания студента по стобальной системе оценок?
- ☐ Сколько должно быть обслуживающих органов (касс, больничных коек, самолетовылетов и т. д.), чтобы с заданной вероятностью обслужить потребителя?
- ☐ Какой ожидается прирост древесины за время t на участке в n га.

- ☐ Сколько необходимо иметь пушек, ракет, другой военной техники для поражения цели противника?

Вопросы, на которые интерполяция дает верные ответы, можно продолжать чуть ли не до бесконечности.

Имеется много хороших математических книг, излагающих теорию интерполяции. В этих книгах приводится классификация методов, методики интерполяции и их сравнительный анализ, рекомендации по применению. К сожалению, там не излагаются компьютерные технологии, без которых в большинстве случаев невозможно решить практическую задачу.

Большое число книг по компьютерной математике содержат описание функций интерполяции, реализованных в данной математической системе. Однако в них так же, как и в книгах по теоретическим основам интерполяции, отсутствуют компьютерные технологии решения задач, включающие в себя следующую последовательность действий:

1. Формулировка задачи.
2. Выбор метода интерполяции.
3. Выбор вида функции интерполяции.
4. Решение задачи с помощью универсального математического программного средства.
5. Оценка адекватности модели.

В литературе по компьютерной математике эта и другие технологии не излагаются. В них описываются лишь функции и команды, позволяющие решить задачу интерполяции. При этом во многих случаях исследователю не ясно, какой используется метод интерполяции, с какой погрешностью решена задача.

Предлагаемая читателю книга является прикладной, в ней излагаются компьютерные технологии, основанные на строгих математических методах решения задач интерполяции с большим числом примеров из различных областей знаний. В этом смысле ей нет аналогов.

Компьютерные технологии решения задач интерполяции можно излагать с применением только одной математической системы. Однако при таком изложении можно потерять читателя. Изложение в книге пяти основных универсальных математических программных средств символьной математики — еще одно ее достоинство.

3. Понятие компьютерная технология интерполяции

В теории и на практике стали модными понятия "информационные технологии" и "компьютерные технологии". Понятие "информационные техноло-

гии" четко не определено. Термин "компьютерные технологии" достаточно прост и очевиден. *Технология* — это последовательность действий для получения результата, в нашем случае для получения математической модели изучаемого объекта. Она реализуется программой на языке универсального программного средства символьной математики.

Компьютерная технология интерполяции есть последовательность выполнения функций и команд компьютера для решения задачи интерполяции.

Она состоит из следующих действий:

- ☐ выбор вида функции интерполяции с помощью компьютера;
- ☐ использование функций и команд универсального программного средства для получения математической модели;
- ☐ оценка адекватности модели.

Из перечисленного очевидно, что для решения задачи недостаточно знать математические методы, функции и команды универсального математического программного средства, владеть методами работы с клавиатурой и мышью ПК. Необходимо еще уметь выбрать вид функции интерполяции и оценить полученное решение. Для этого, кроме функций и команд интерполяции, необходимо знать:

- ☐ способы построения графиков функций, заданных в табличном и формульном видах;
- ☐ соответствие графика, построенного по данным таблицы аналитической функции;
- ☐ способы вычисления значений функции и ее табулирование;
- ☐ операции с векторами и матрицами;
- ☐ решение систем линейных и нелинейных уравнений;
- ☐ способы вычисления табличных разностей.

Решение задачи интерполяции на ПК позволит пользователю изучить и активно использовать для решения многих других задач универсальные программные средства символьной математики: Mathematica, Maple, Derive, Mathcad, Matlab.

4. Рациональная методика изучения компьютерных технологий интерполяции

Наиболее эффективным способом изучения компьютерных технологий интерполяции является решение задач. Для этого необходимо иметь на вашем компьютере любую из пяти математических систем, описанных в главах 2—6. Для начала рекомендуем поставить одну из следующих: Mathematica, Maple, Derive. Эти системы символьной математики являются наиболее интеллектуальными, способными давать решение в виде математических выражений.

Если такое программное средство имеется на вашем компьютере, то приступайте к формулировке задачи интерполяции. В постановку задачи входят следующие пункты:

- ☐ дано;
- ☐ определить;
- ☐ допустимая погрешность интерполяции.

Если у вас нет своей задачи, то рекомендуем взять любую задачу однопараметрической интерполяции из приложения.

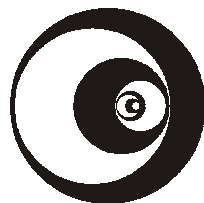
После формулировки задачи обратитесь к *главе 2* и выберите метод интерполяции. Если вы решаете задачу интерполяции впервые, то рекомендуем выбрать полиномиальную интерполяцию точную в узлах, а затем приближенную в узлах. Эти методы наиболее понятны и легко реализуемы на ПК.

Не спешите решать задачу после выбора метода. Изучите предварительно компьютерную технологию интерполяции, соответствующую математической системе, установленной на вашем ПК. Обратите особое внимание на примеры, повторите их решение.

Теперь приступайте к решению вашей задачи. Успех вам обеспечен.

Если читатель знает методы интерполяции, а тем более универсальные программные средства символьной математики, то ему необходимо изучить только компьютерные технологии, обратив особое внимание на способы выбора вида функции интерполяции и методы оценки адекватности математической модели. Если возникнут трудности в выборе вида функции интерполяции, обратитесь за помощью к одной из следующих программ: Formula, CurveExpert, TableCurve.

Глава 1



Основы теории интерполяции

1.1. Что такое интерполяция

Любую функцию $f(x)$ можно представить в виде таблицы, графика или формулы. Если $f(x)$ задана в виде формулы, то представить ее в виде таблицы или графика нетрудно. Значительно труднее представить функцию в виде формулы, если она задана в виде таблицы или графика. Не менее трудно представить заданную функцию новой, более простой, существенно упрощающей дальнейшие расчеты. Интерполяция и является областью знаний, позволяющей решать такие задачи.

Пусть функция задана в виде таблицы (табл. 1.1).

Таблица 1.1

x	x_1	x_2	x_3		x_n
y	y_1	y_2	y_3		y_n

Здесь:

- x_i — аргумент;
- y_i — функция;
- $i = 1, 2, \dots, n$ — индексная переменная соответственно аргумента и функции $f(x)$.

Тогда одной из задач интерполяции является определение значений функции при значениях аргументов, отсутствующих в таблице. Очевидно, что если представить функцию $f(x)$ формулой, то ее значение можно вычислить при любых значениях x , в т. ч. и при значениях, отсутствующих в таблице.

Тогда задача интерполяции может быть сформулирована так: известны значения функции при некоторых значениях аргументов, необходимо представить функцию формулой для всего диапазона значений аргументов. Такая формулировка не является строгой, т. к. может существовать большое количество различных формул, которые удовлетворяют дискретным значениям исходных данных, представленных в таблице. Для корректной постановки задачи интерполяции дополнительно должен быть указан вид функции (например, линейная, параболическая, логарифмическая и др.)

Интерполяция в научных исследованиях и инженерной практике находит широкое применение. Областями ее использования являются:

- ☐ моделирование;
- ☐ планирование и статистическая обработка эксперимента;
- ☐ определение значений функции при аргументах, отсутствующих в таблице;
- ☐ табулирование функции;
- ☐ представление сложной функции более простой в определенных границах значений ее аргументов;
- ☐ во всех других случаях, где нужно выполнить приближение одних функций другими, более простыми, с допустимой для практики точностью.

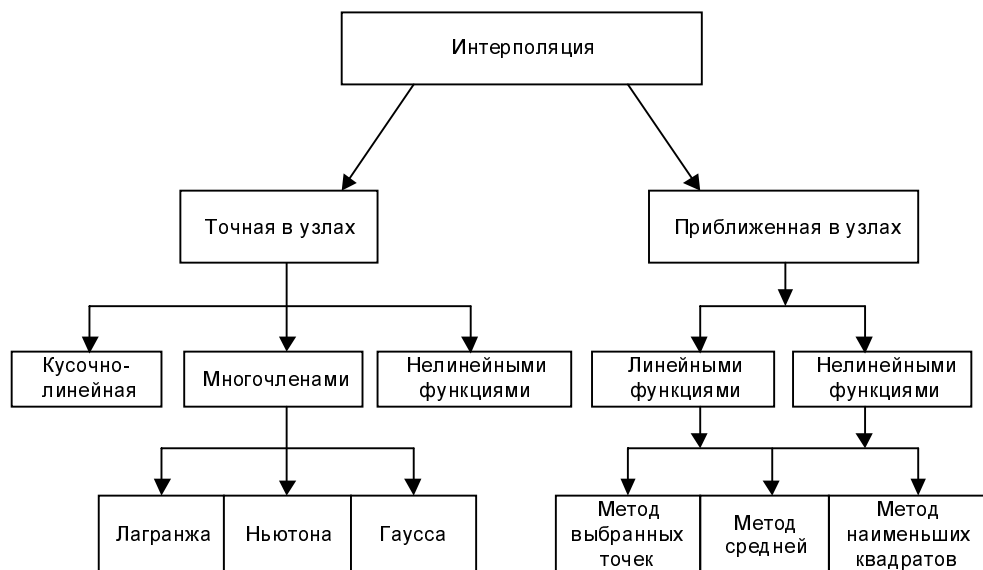


Рис. 1.1

Существует большое число методов интерполяции. Их можно классифицировать по множеству различных признаков: по точности в узлах интерполяции, виду функции интерполяции, критериям и применяемому математическому аппарату и др. На рис. 1.1 приведена подобная классификация.

В дальнейшем будут изложены все методы указанной классификации, их достоинства и недостатки, область применения, компьютерные технологии решения задач на ЭВМ. Следует при этом иметь в виду, что классификация, приведенная на рис. 1.1, далеко не полная. Она показана скорее для того, чтобы читатель представил себе объем излагаемого материала. Более полное и глубокое понятие о методах интерполяции можно найти в литературе [18-23]. В настоящей книге читатель найдет систематическое изложение компьютерных технологий решения задач интерполяции с помощью универсальных программных средств символьной математики, чего нет в упомянутой выше литературе.

1.2. Интерполяция точная в узлах

1.2.1. Общий метод решения задачи

Интерполяция точная в узлах — такая интерполяция, при которой значения функции интерполяции совпадают с ее действительными значениями во всех узлах (рис. 1.2).

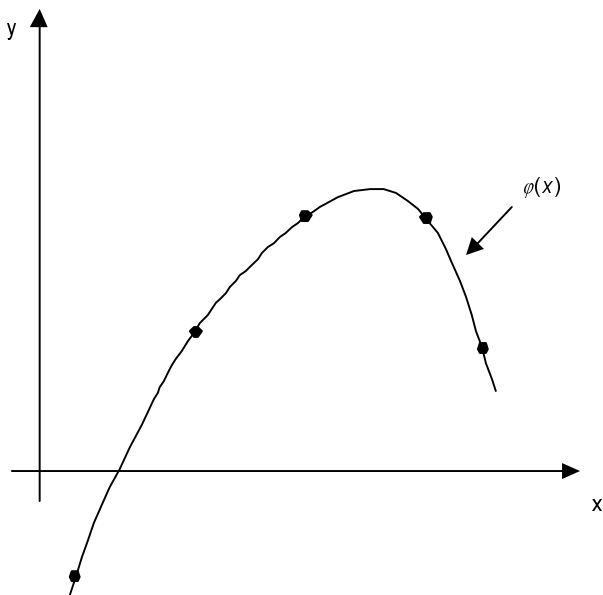


Рис. 1.2

На рисунке точками обозначены значения функции $f(x)$ при соответствующих значениях аргументов, сплошной линией обозначена функция $\varphi(x)$, полученная в результате интерполяции (функция интерполяции). Из рисунка видно, что значения функций в узлах x_i (где $i = 1, 2, \dots, n$) совпадают.

Такая интерполяция применяется в следующих случаях:

- при отыскании закона некоторого физического явления по экспериментальным данным, полученным с высокой точностью;
- при табулировании математических функций и определении их значений, когда значения аргументов отсутствуют в таблицах.

Решение задачи интерполяции в таких случаях сводится к решению систем линейных или нелинейных уравнений. Покажем это на примерах.

Пример 1.1

Пусть в результате эксперимента получены данные, приведенные в табл. 1.2.

Таблица 1.2

	Значения переменных				
x	1	2.4	4.0	6.8	9
y	2	4.1	6.5	10.7	14

Необходимо функцию $f(x)$ представить в виде формулы $y = \varphi(x)$ и найти ее значения при $x = 3$ и при $x = 8$. В постановке задачи не указан вид функции $\varphi(x)$. Способы ее отыскания будут описаны позже, а сейчас представим, что эта функция линейна $y = a + bx$. Для определения неизвестных a и b составим следующую систему уравнений:

$$\left. \begin{aligned} 2 &= a + b \\ 14 &= a + 9b \end{aligned} \right\} \quad (1.1)$$

В системе уравнений (1.1) значения y взяты в узлах $x = 1$ и $x = 9$, при которых y равен соответственно 2 и 14. Решая эту систему линейных уравнений, получим: $a = 0.5$; $b = 1.5$, т. е. функция, заданная таблицей 1.2, может быть представлена формулой $y = 0.5 + 1.5x$. Подставляя в эту формулу значения x из таблицы 1.2, убеждаемся, что при всех значениях аргумента, заданного таблицей, значения функции совпадают с табличными. Теперь вычислим значение y при $x = 3$ и при $x = 8$. Подставляя эти значения в уравнение $y = 0.5 + 1.5x$, получим: $y = 5$ и $y = 12.5$. Для определения неизвестных a и b при составлении системы уравнений (1.1) можно выбрать из таблицы любые два значения x и y . Решение будет прежним.

Пример 1.2

Пусть данные эксперимента сведены в таблицу (табл. 1.3).

Таблица 1.3

	Значения переменных					
x	1	2	3	4	5	6
y	1.2	0.21	0.077	0.038	0.021	0.014

Из табл. 1.3 видно, что с увеличением x значение y убывает, причем далеко не линейно. На рис. 1.3 представлен график функции $y = f(x)$.

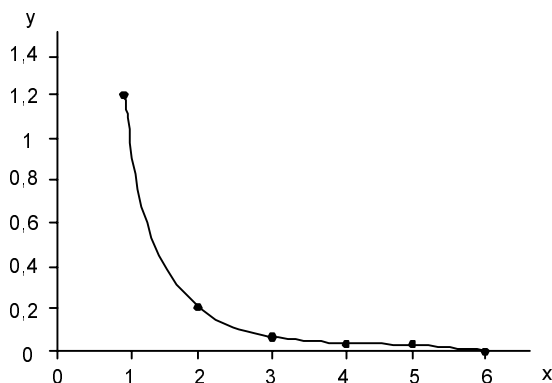


Рис. 1.3

Из рис.1.3 видно, что функция имеет вид гиперболы, поэтому представим

ее в виде: $y = \frac{a}{x^b}$.

Определим значения a и b , для чего, используя данные табл. 1.3, составим следующую систему уравнений:

$$\left. \begin{aligned} 1.2 &= \frac{a}{1^b} \\ 0.014 &= \frac{a}{6^b} \end{aligned} \right\} \quad (1.2)$$

Решая эту систему нелинейных уравнений, получим: $a = 1.2$; $b = 2.5$. Тогда функция интерполяции $y = \varphi(x)$ будет иметь вид:

$$\begin{array}{cccc}
 1 & x_0 & x_0^2 & \dots x_0^n \\
 1 & x_1 & x_1^2 & \dots x_1^n \\
 & \dots & & \\
 1 & x_n & x_n^2 & \dots x_n^n
 \end{array} \quad (1.5)$$

Главный определитель этой системы (1.5), называемый определителем Вандермонда, не равен нулю, если узлы интерполяции $x_0, x_1, x_2, \dots, x_n$ различны. Это является признаком того, что система (1.4) имеет решение и притом единственное. А это означает, что существует единственный полином степени n , являющийся интерполяционным, т. е. удовлетворяющий условию: $y(x_i) = y_i$ (где $i = 0, 1, 2, \dots, n$).

Геометрически это означает, что через $n + 1$ точку, расположенную на плоскости (x, y) , можно провести единственную кривую, проходящую через все точки и представляющую собой многочлен n -ой степени. Например, если число точек равно 2, то через эти точки можно провести единственную прямую $y = a_0 + a_1x$; если число точек равно 3, то через эти точки можно провести единственную параболу, представляющую собой многочлен второй степени: $y = a_0 + a_1x + a_2x^2$.

Интерполяционный многочлен может быть получен без решения системы уравнений (1.4). Существуют различные формулы, позволяющие получить интерполяционный многочлен, не требующие решения системы уравнений (например, формулы Лагранжа, Ньютона, Гаусса, Бесселя, Стирлинга). Наиболее популярными из них являются формулы Лагранжа и Ньютона.

1.2.3. Интерполяционная формула Лагранжа

Формула Лагранжа имеет вид:

$$\begin{aligned}
 y_n(x) = & y_0 \frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_n)} + y_1 \frac{(x-x_0)(x-x_2)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_n)} + \dots \\
 & + y_n \frac{(x-x_0)(x-x_1)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\dots(x_n-x_{n-1})}.
 \end{aligned} \quad (1.6)$$

Здесь:

- $x_0, x_1, \dots, x_n$ — узлы интерполяции;
- $y_0, y_1, \dots, y_n$ — значения функции в этих узлах.

Покажем, что формула (1.6) является интерполяционным полиномом.

Пусть $x = x_0$, тогда все члены, кроме первого, обращаются в ноль, а числитель и знаменатель в первом члене сокращаются, в результате чего $y_n(x_0) = y_0$. При $x = x_1$ второй член выражения (1.6) равен y_1 , а все остальные обращаются в ноль и т. д.

Таким образом, справедливыми являются следующие равенства: $y_n(x_0) = y_0$, $y_n(x_1) = y_1$, ..., $y_n(x_n) = y_n$.

Равенства означают, что формула (1.6) является интерполяционной. Из этой формулы также очевидно, что многочлен, полученный по формуле Лагранжа, будет степени не выше n .

Пример 1.3

Пусть функция задана в виде таблицы (табл. 1.4).

Таблица 1.4

	Значения переменных			
x	1	2	4	6
y	2	9	41	97

Необходимо представить функцию $y = f(x)$ в виде многочлена, используя интерполяционную формулу Лагранжа, и определить значения функции при $x = 3$ и при $x = 5$.

Подставляя данные табл. 1.4 в формулу (1.6), получим:

$$y(x) = 2 \frac{(x-2)(x-4)(x-6)}{(1-2)(1-4)(1-6)} + 9 \frac{(x-1)(x-4)(x-6)}{(2-1)(2-4)(2-6)} +$$

$$+ 41 \frac{(x-1)(x-2)(x-6)}{(4-1)(4-2)(4-6)} + 97 \frac{(x-1)(x-2)(x-4)}{(6-1)(6-2)(6-4)}.$$

В результате очевидных преобразований получим: $y(x) = 3x^2 - 2x + 1$. Подставляя в эту формулу $x = 3$, а затем $x = 5$, получим $y(3) = 22$, $y(5) = 66$. Существенным недостатком формулы Лагранжа является то, что результаты предыдущих вычислений теряются, если добавляется или убирается хотя бы одно значение $y(x)$. Достоинство формулы состоит в том, что она пригодна для случая постоянного и переменного шага изменения аргумента x .

1.2.4. Табличные разности

Пусть значения функции в узлах интерполяции равны $y_0, y_1, \dots, y_n$. Тогда конечными разностями Δy_i (где $i = 0, 1, 2, \dots, (n-1)$) первого порядка называются выражения:

$$\begin{aligned} \Delta y_0 &= y_1 - y_0 \\ \Delta y_1 &= y_2 - y_1 \\ \Delta y_2 &= y_3 - y_2 \\ &\dots\dots\dots \\ \Delta y_{n-1} &= y_n - y_{n-1} \end{aligned} \tag{1.7}$$

Разности $\Delta^2 y_i$ (где $i = 0, 1, 2, \dots, (n - 2)$) разностей первого порядка называются разностями второго порядка:

$$\begin{aligned}\Delta^2 y_0 &= \Delta y_1 - \Delta y_0 \\ \Delta^2 y_1 &= \Delta y_2 - \Delta y_1 \\ &\dots\dots\dots \\ \Delta^2 y_{n-2} &= \Delta y_{n-1} - \Delta y_{n-2}\end{aligned}\tag{1.8}$$

Можно получать разности третьего, четвертого порядка и т. д. В общем случае конечные разности k -го порядка образуются как разности $(k - 1)$ -го порядка. Заметим, что число конечных разностей убывает с ростом их порядка: число табличных разностей первого порядка на единицу меньше, чем значений y , т. е. равно n ; число табличных разностей второго порядка будет $n - 1$, третьего — $n - 2$ и т. д. Наиболее удобно представлять конечные разности в виде таблицы (табл. 1.5).

Таблица 1.5

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$	$\Delta^6 y$
x_0	y_0						
$x_1 = x_0 + h$	y_1	Δy_0					
$x_2 = x_1 + h$	y_2	Δy_1	$\Delta^2 y_0$				
$x_3 = x_2 + h$	y_3	Δy_2	$\Delta^2 y_1$	$\Delta^3 y_0$			
$x_4 = x_3 + h$	y_4	Δy_3	$\Delta^2 y_2$	$\Delta^3 y_1$	$\Delta^4 y_0$		
$x_5 = x_4 + h$	y_5	Δy_4	$\Delta^2 y_3$	$\Delta^3 y_2$	$\Delta^4 y_1$	$\Delta^5 y_0$	
$x_6 = x_5 + h$	y_6	Δy_5	$\Delta^2 y_4$	$\Delta^3 y_3$	$\Delta^4 y_2$	$\Delta^5 y_1$	$\Delta^6 y_0$

По конечным разностям можно определить, является ли данная функция многочленом, найти степень многочлена, построить интерполяционный полином. Функция является многочленом степени n только в том случае, если конечные разности n -го порядка при постоянном шаге таблицы являются величинами постоянными.

В табл. 1.6 приведены табличные разности в случае примера 1.3.

Из таблицы видно, что конечные разности второго порядка одинаковы. Это означает, что функция $y(x)$ является многочленом второй степени, что и было получено при решении задачи интерполяции по формуле Лагранжа.

Таблица 1.6

	Значения переменных					
x	1	2	3	4	5	6
y	2	9	22	41	66	97
Δy		7	13	19	25	31
$\Delta^2 y$			6	6	6	6
$\Delta^3 y$				0	0	0

Таблица 1.7

X	Y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$	$\Delta^6 y$
x_{-4}	y_{-4}						
		Δy_{-4}					
x_{-3}	y_{-3}		$\Delta^2 y_{-4}$				
		Δy_{-3}		$\Delta^3 y_{-4}$			
x_{-2}	y_{-2}		$\Delta^2 y_{-3}$		$\Delta^4 y_{-4}$		
		Δy_{-2}		$\Delta^3 y_{-3}$		$\Delta^5 y_{-4}$	
x_{-1}	y_{-1}		$\Delta^2 y_{-2}$		$\Delta^4 y_{-3}$		$\Delta^6 y_{-4}$
		Δy_{-1}		$\Delta^3 y_{-2}$		$\Delta^5 y_{-3}$	
x_0	y_0		$\Delta^2 y_{-1}$		$\Delta^4 y_{-2}$		$\Delta^6 y_{-3}$
		Δy_0		$\Delta^3 y_{-1}$		$\Delta^5 y_{-2}$	
x_1	y_1		$\Delta^2 y_0$		$\Delta^4 y_{-1}$		$\Delta^6 y_{-2}$
		Δy_1		$\Delta^3 y_0$		$\Delta^5 y_{-1}$	
x_2	y_2		$\Delta^2 y_1$		$\Delta^4 y_0$		
		Δy_2		$\Delta^3 y_1$			
x_3	y_3		$\Delta^2 y_2$				
		Δy_3					
x_4	y_4						

Центральные разности

Если за начальные значения x_0, y_0 взять их значения в середине таблицы, то можно получить табличные разности по обе стороны начальных значений.

Тогда табличные разности, находящиеся в строке начальных значений x_0, y_0 и в строках, примыкающих к ней, называют центральными табличными разностями. Такие разности приведены в табл. 1.7 (помечены стрелками). К ним относятся разности:

$$\Delta y_{-1}, \Delta y_0, \Delta^2 y_{-1}, \Delta^3 y_{-1}, \Delta^3 y_{-2}, \Delta^4 y_{-2}, \Delta^5 y_{-2}, \Delta^5 y_{-3}, \Delta^6 y_{-3}.$$

В общем случае $x_i = x_0 + i h$, где:

$$\square i = 0, \pm 1, \pm 2, ;$$

$$\square \Delta y_i = y_{i+1} - y_i;$$

$$\square \Delta^2 y_i = \Delta y_{i+1} - \Delta y_i \text{ и т. д.}$$

Интерполяционные формулы, использующие центральные разности, называются интерполяционными формулами с центральными разностями. К ним относятся формулы Гаусса, Стирлинга, Бесселя.

1.2.5. Интерполяционные формулы Ньютона

Интерполяционная формула Ньютона при равноотстоящих узлах имеет вид:

$$y_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots + c_n(x - x_0)(x - x_1)(x - x_2) \dots (x - x_{n-1}). \quad (1.9)$$

Здесь:

$$\square x_i - \text{узлы интерполяции с постоянным шагом (где } i = 0, 1, 2, \dots, (n - 1)), \text{ т. е. } x_1 = x_0 + h, x_2 = x_1 + h, \dots, x_{n-1} = x_{n-2} + h;$$

$$\square c_i - \text{коэффициенты интерполяционной формулы (где } i = 0, 1, 2, \dots, n)$$

Из формулы (1.9) видно, что функция $y(x)$, как и в случае формулы Лагранжа, является многочленом степени n , но записанной в иной форме.

Определим коэффициенты c_i в формуле (1.9).

При $x = x_0$ $y(x_0) = c_0$. Тогда $c_0 = y_0$.

$$\text{При } x = x_1 \quad y(x_1) = y_1 = c_0 + c_1(x_1 - x_0) \quad \text{или} \quad c_1 = \frac{y_1 - c_0}{x_1 - x_0} = \frac{y_1 - y_0}{x_1 - x_0} = \frac{\Delta y_0}{h}.$$

$$\text{При } x = x_2 \quad y_n(x_2) = y_2 = c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1).$$

Подставляя вместо c_0 и c_1 их значения, получим:

$$y_2 = y_0 + \frac{\Delta y_0 (x_2 - x_0)}{h} + c_2 (x_2 - x_0)(x_2 - x_1) = y_0 + \frac{\Delta y_0 \cdot 2h}{h} + c_2 \cdot 2h \cdot h.$$

Найдем c_2 :

$$c_2 = \frac{y_2 - y_0 - \frac{y_1 - y_0}{h} 2h}{2h^2} = \frac{(y_2 - y_1) - (y_1 - y_0)}{2h^2} = \frac{\Delta y_1 - \Delta y_0}{2h^2} = \frac{\Delta^2 y_0}{2h^2}$$

Вычислительные процедуры очевидны. Продолжая их, получим следующую общую формулу для коэффициентов многочлена (1.9):

$$c_k = \frac{\Delta^k y_0}{k! h^k}, \quad k = 1, 2, \dots, n \quad (1.10)$$

Подставляя найденные выражения коэффициентов в (1.9), получим:

$$\begin{aligned} y_n(x) = & y_0 + \frac{\Delta y_0}{1! h} (x - x_0) + \frac{\Delta^2 y_0}{2! h^2} (x - x_0)(x - x_1) + \\ & + \frac{\Delta^3 y_0}{3! h^3} (x - x_0)(x - x_1)(x - x_2) + \dots + \frac{\Delta^n y_0}{n! h^n} (x - x_0)(x - x_1) \dots (x - x_{n-1}) \end{aligned} \quad (1.11)$$

Заметим, что в этой формуле конечные разности находятся в верхней кривой строке таблицы конечных разностей. Формулу (1.11) можно записать в более простом виде. Обозначим $\frac{(x - x_0)}{h} = t$. Тогда

$$\begin{aligned} \frac{(x - x_1)}{h} &= \frac{(x - (x_0 + h))}{h} = t - 1 \\ \frac{(x - x_2)}{h} &= \frac{(x - (x_0 + 2h))}{h} = t - 2, \dots \\ \frac{(x - x_{n-1})}{h} &= \frac{(x - (x_0 + (n-1)h))}{h} = t - n + 1. \end{aligned}$$

Подставляя полученные выражения в (1.11), получим:

$$\begin{aligned} y(x_0 + t h) = & y_0 + \Delta y_0 \frac{t}{1!} + \Delta^2 y_0 \frac{t(t-1)}{2!} + \\ & + \Delta^3 y_0 \frac{t(t-1)(t-2)}{3!} + \dots + \Delta^n y_0 \frac{t(t-1)(t-2) \dots (t-n+1)}{n!} \end{aligned} \quad (1.12)$$

Формула (1.12) может быть записана в следующем символическом виде:

$$y_n(x_0 + t h) = (1 + \Delta)^t y_0. \quad (1.13)$$

Интерполяционная формула (1.12) дает хорошие результаты при $t < 1$, т. е. при значениях x в начале таблицы для функции $y(x)$. Эта формула часто называется интерполяционной формулой Ньютона для интерполирования вперед.

Формула Ньютона (1.12) существенно отличается от формулы Лагранжа (1.6). В формуле Лагранжа каждый член является многочленом степени n , т. е. все члены формулы Лагранжа равноценны. В формулу же Ньютона входят алгебраические члены повышающихся степеней. Кроме того, коэффициентами являются конечные разности, деленные на факториалы n . Это позволяет (особенно при малых t) ограничиться числом членов, что может

существенно упростить вычисления. Погрешности формулы (1.12) возрастают при интерполировании в конце таблицы. Для интерполирования в конце таблицы Ньютоном была предложена следующая формула:

$$y_n(x) = c_0 + c_1(x - x_n) + c_2(x - x_n)(x - x_{n-1}) + \dots + c_n(x - x_n)(x - x_{n-1}) \dots (x - x_1). \quad (1.14)$$

По аналогии с предыдущими доказательствами, получим следующие значения коэффициентов $c_0, c_1, \dots, c_n$:

$$c_0 = y_n, c_1 = \frac{\Delta y_{n-1}}{h}, c_2 = \frac{\Delta^2 y_{n-2}}{2!h^2}, \dots, c_k = \frac{\Delta^k y_{n-k}}{k!h^k}, k = 1, 2, \dots, n.$$

Подставляя значения коэффициентов в формулу (1.14), получим:

$$y_n(x) = y_n + \frac{\Delta y_{n-1}}{h}(x - x_n) + \frac{\Delta^2 y_{n-2}}{2!h^2}(x - x_n)(x - x_{n-1}) + \dots + \frac{\Delta^n y_0}{n!h^n}(x - x_n)(x - x_{n-1}) \dots (x - x_1) \quad (1.15)$$

Обозначая $\frac{(x - x_n)}{h} = t$ и подставляя в (1.15), получим:

$$y_n(x_n + th) = y_n + \Delta y_{n-1} \frac{t}{1!} + \Delta^2 y_{n-2} \frac{t(t+1)}{2!} + \dots + \Delta^n y_0 \frac{t(t-1)(t-2) \dots (t+n-1)}{n!} \quad (1.16)$$

Эта формула называется второй интерполяционной формулой Ньютона для интерполирования назад. Она дает хорошие результаты при интерполировании в конце таблицы. При этом следует иметь в виду, что шаг h имеет отрицательное значение $x_{k-1} - x_k = -h$ (где $k = 1, 2, \dots, n$), а конечные разности находятся в нижней косой строке таблицы разностей. Формулы Ньютона могут давать значительные погрешности при интерполировании в середине таблицы. В таких случаях следует применять интерполяционные формулы Гаусса, Стирлинга, Бесселя.

Пример 1.4

Пусть функция $y(x)$ представлена в виде табл. 1.6. Необходимо найти интерполяционный полином по формуле Ньютона для интерполирования вперед. В нашем случае $y_0 = 2$, $h = 1$, табличные разности имеют значения: $\Delta y_0 = 7$, $\Delta^2 y_0 = 6$, $\Delta^3 y_0 = 0$. Воспользуемся формулой (1.11):

$$y(x) = y_0 + \frac{\Delta y_0}{1!h}(x - x_0) + \frac{\Delta^2 y_0}{2!h^2}(x - x_0)(x - x_1).$$

Подставляя в эту формулу значения узлов интерполяции x_0, x_1 , значение функции y_0 , табличные разности $\Delta y_0, \Delta^2 y_0$, получим:

$$y(x) = 2 + 7(x - 1) + 3(x - 1)(x - 2).$$

После очевидных преобразований интерполяционный многочлен будет иметь вид:

$$y(x) = 3x^2 - 2x + 1.$$

Решение совпадает с решением, полученным по формуле Лагранжа.

Интерполяционные формулы Ньютона при неравноотстоящих узлах

В случае, когда шаг $h \neq \text{const}$ (неравноотстоящие узлы), интерполяционная формула Ньютона имеет вид:

$$\begin{aligned} y_n(x) = & y_0 + \sigma y_0(x - x_0) + \sigma^2 y_0(x - x_0)(x - x_1) + \dots \\ & + \sigma^n y_0(x - x_0)(x - x_1) \dots (x - x_{n-1}). \end{aligned} \quad (1.17)$$

Внешне формула похожа на формулы Ньютона для равноотстоящих узлов, когда $h = \text{const}$. Ее отличие состоит в способе вычисления табличных разностей $\sigma^k y_0$ (где $k = 1, 2, \dots, n$). В формуле (1.17) $\sigma^k y_0$ являются разделенными разностями или разностными отношениями, представляющими собой отношения разности значений функции к разности значений соответствующих аргументов. Разностные отношения имеют вид:

□ разностные отношения первого порядка

$$\delta y_0 = \frac{y_1 - y_0}{x_1 - x_0}, \delta y_1 = \frac{y_2 - y_1}{x_2 - x_1}, \dots, \delta y_{n-1} = \frac{y_n - y_{n-1}}{x_n - x_{n-1}};$$

□ разностные отношения второго порядка

$$\delta^2 y_0 = \frac{\delta y_1 - \delta y_0}{x_2 - x_0}, \delta^2 y_1 = \frac{\delta y_2 - \delta y_1}{x_3 - x_1}, \dots, \delta^2 y_{n-2} = \frac{\delta y_{n-1} - \delta y_{n-2}}{x_n - x_{n-2}}.$$

В общем виде разностные отношения определяются выражением:

$$\delta^k y_i = \frac{\delta^{k-1} y_{i+1} - \delta^{k-1} y_i}{x_{i+k} - x_i}, \quad i = 0, 1, 2, \dots, n-1, \quad k = 1, 2, \dots, n. \quad (1.18)$$

Как и в случае равноотстоящих узлов, степень многочлена определяется по значениям разностных отношений. Если разностные отношения n -го порядка постоянны, то функция представляет собой многочлен n -ой степени. Указанное свойство разностных отношений позволяет в случае интерполяции многочленами существенно упростить вычисления. В качестве примера проинтерполируем функцию $y(x)$, заданную в виде табл. 1.4. В данном случае шаг таблицы переменный. Определим разностные отношения.

□ Разностные отношения первого порядка:

$$\delta y_0 = \frac{9-2}{2-1} = 7, \quad \delta y_1 = \frac{41-9}{4-2} = 16, \quad \delta y_2 = \frac{97-41}{6-4} = 28.$$

□ Разностные отношения второго порядка:

$$\delta^2 y_0 = \frac{16-7}{4-1} = 3, \quad \delta^2 y_1 = \frac{28-16}{6-2} = 3.$$

Поскольку разностные отношения второго порядка одинаковы, то функция представляет собой многочлен второй степени. Подставляя в формулу (1.17) данные таблицы и разностные отношения, получим:

$$y(x) = 2 + 7(x-1) + 3(x-1)(x-2) = 3x^2 - 2x + 1$$

1.2.6. Интерполяционные формулы Гаусса

Пусть функция $y(x)$, представленная в виде таблицы, имеет $2n + 1$ узлов интерполирования:

$x_{-n}, x_{-(n-1)}, \dots, x_{-1}, x_0, x_1, x_2, \dots, x_n$ и постоянный шаг $h = x_i - x_{i-1} = \text{const}$.

Представим функцию в виде многочлена степени не выше $2n$. Интерполяционная формула Гаусса имеет вид:

$$\begin{aligned} y(x) = & c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) + \\ & + c_3(x-x_{-1})(x-x_0)(x-x_1) + c_4(x-x_{-1})(x-x_0)(x-x_1)(x-x_2) + \\ & + c_5(x-x_{-2})(x-x_{-1})(x-x_0)(x-x_1)(x-x_2) + \dots \\ & + c_{2n-1}(x-x_{-(n-1)}) \dots (x-x_{-1})(x-x_0)(x-x_1) \dots (x-x_{n-1}) + \\ & + c_{2n}(x-x_{-(n-1)}) \dots (x-x_{-1})(x-x_0)(x-x_1) \dots (x-x_{n-1})(x-x_n). \end{aligned} \quad (1.19)$$

Определим коэффициенты c_i в формуле (1.19).

При $x = x_0$ $y(x_0) = y_0 = c_0$.

При $x = x_1$ $y(x_1) = y_1 = c_0 + c_1(x_1 - x_0)$ или $c_1 = \frac{y_1 - y_0}{x_1 - x_0} = \frac{\Delta y_0}{h}$.

При $x = x_{-1}$ $y(x_{-1}) = y_{-1} = c_0 + c_1(x_{-1} - x_0) + c_2(x_{-1} - x_0)(x_{-1} - x_1)$ или

$$c_2 = \frac{y_{-1} - c_0 - c_1(x_{-1} - x_0)}{(x_{-1} - x_0)(x_{-1} - x_1)} = \frac{y_{-1} - y_0 - \frac{\Delta y_0}{h}(x_{-1} - x_0)}{(x_{-1} - x_0)(x_{-1} - x_1)} = \frac{\Delta^2 y_{-1}}{2! h^2}.$$

Вычислительные процедуры очевидны. Продолжая их, получим:

$$c_3 = \frac{\Delta^3 y_{-1}}{3! h^3}, \quad c_4 = \frac{\Delta^4 y_{-2}}{4! h^4}, \quad \dots, \quad c_{2n} = \frac{\Delta^{2n} y_{-n}}{(2n)! h^{2n}}.$$

Введем новую переменную $\frac{x - x_0}{h} = t$.

Подставим в (1.19) новую переменную t и значения коэффициентов c_i .

После очевидных преобразований получим:

$$\begin{aligned} y(x) = & y_0 + t \Delta y_0 + \frac{t(t-1)}{2!} \Delta^2 y_{-1} + \frac{(t+1)t(t-1)}{3!} \Delta^3 y_{-1} + \\ & + \frac{(t+1)t(t-1)(t-2)}{4!} \Delta^4 y_{-2} + \frac{(t+2)(t+1)t(t-1)(t-2)}{5!} \Delta^5 y_{-2} + \dots \\ & + \frac{(t+n-1)\dots(t-n)}{(2n)!} \Delta^{2n} y_{-n} . \end{aligned} \quad (1.20)$$

Формула (1.20) называется первой интерполяционной формулой Гаусса. В нее входят центральные разности, расположенные в строке начальных значений x_0, y_0 и в смежной нижней строке таблицы 1.7.

Вторая интерполяционная формула Гаусса, получаемая аналогично первой, имеет вид:

$$\begin{aligned} y(x) = & y_0 + t \Delta y_0 + \frac{(t+1)t}{2!} \Delta^2 y_{-1} + \frac{(t+1)t(t-1)}{3!} \Delta^3 y_{-2} + \\ & + \frac{(t+2)(t+1)t(t-1)}{4!} \Delta^4 y_{-2} + \dots + \frac{(t+n)(t+n-1)\dots(t-n+1)}{(2n)!} \Delta^{2n} y_{-n} . \end{aligned} \quad (1.21)$$

Формула содержит центральные разности, находящиеся в строке начальных значений x_0, y_0 и в смежной верхней строке таблицы 1.7.

Интерполяционные формулы Гаусса целесообразно применять при интерполировании в средней части таблицы функции $f(x)$. Недостатком формул Гаусса является ограничение возможности их использования только для случая равноотстоящих узлов, когда шаг $h = \text{const}$.

1.2.7. Интерполяционная формула Стирлинга

Формула Стирлинга представляет собой среднее арифметическое первой и второй формул Гаусса. Она имеет вид:

$$\begin{aligned} y(x) = & y_0 + t \frac{\Delta y_{-1} + \Delta y_0}{2} + \frac{t^2}{2!} \Delta^2 y_{-1} + \frac{t(t^2 - 1^2)}{3!} \cdot \frac{\Delta^3 y_{-2} + \Delta^3 y_{-1}}{2} + \\ & + \frac{t^2(t^2 - 1^2)}{4!} \Delta^4 y_{-2} + \frac{t(t^2 - 1^2)(t^2 - 2^2)}{5!} \cdot \frac{\Delta^5 y_{-3} + \Delta^5 y_{-2}}{2} + \\ & + \frac{t^2(t^2 - 1^2)(t^2 - 2^2)}{6!} \Delta^6 y_{-3} + \dots + \frac{t^2(t^2 - 1^2)(t^2 - 2^2) \dots (t^2 - (n-1)^2)}{(2n)!} \Delta^{2n} y_{-n} . \end{aligned} \quad (1.22)$$

В формуле $t = \frac{x - x_0}{h}$.

Формулу Стирлинга целесообразно применять при $t \leq 0.25$. Она может использоваться лишь при постоянном шаге изменения аргумента. В этом ее существенный недостаток.

1.2.8. Интерполяционная формула Бесселя

Пусть функция $y(x)$, представленная в виде таблицы, имеет $2n + 2$ равноотстоящих узла интерполирования $x_{-n}, x_{-(n-1)}, \dots, x_0, x_1, \dots, x_{n-1}, x_n, x_{n+1}$ с постоянным шагом $h = x_i - x_{i-1}$ ($i = -n, \dots, n + 1$).

Формула Бесселя представляет собой многочлен степени $2n + 2$, значения которого в узлах интерполирования совпадают с функцией $y(x)$.

Формула Бесселя имеет вид:

$$\begin{aligned}
 y(x) = & \frac{y_0 + y_1}{2} + (t - \frac{1}{2}) \Delta y_0 + \frac{t(t-1)}{2!} \cdot \frac{\Delta^2 y_{-1} + \Delta^2 y_0}{2} + \frac{(t - \frac{1}{2})t(t-1)}{3!} \Delta^3 y_{-1} + \\
 & + \frac{t(t-1)(t+1)(t-2)}{4!} \cdot \frac{\Delta^4 y_{-2} + \Delta^4 y_{-1}}{2} + \frac{(t - \frac{1}{2})t(t-1)(t+2)(t-2)}{5!} \Delta^5 y_{-2} + \\
 & + \frac{t(t-1)(t+1)(t+2)(t-2)(t-3)}{6!} \cdot \frac{\Delta^6 y_{-3} + \Delta^6 y_{-2}}{2} + \dots \\
 & \dots + \frac{t(t-1)(t+1)(t-2)(t+2)\dots(t-n)(t+n-1)}{(2n)!} \cdot \frac{\Delta^{2n} y_{-n} + \Delta^{2n} y_{-n+1}}{2} + \\
 & + \frac{(t - \frac{1}{2})t(t-1)(t+1)(t-2)(t+2)\dots(t-n)(t+n-1)}{(2n+1)!} \cdot \Delta^{2n+1} y_{-n}
 \end{aligned} \tag{1.23}$$

Здесь: $t = \frac{x + x_0}{h}$.

Интерполяционную формулу Бесселя целесообразно применять при $0.25 \leq t \leq 0.75$. Пользоваться формулой можно лишь при условии постоянного шага изменения аргумента x .

1.2.9. Сплайн-интерполяция

Существенно повысить точность аппроксимации можно путем интерполяции функции $f(x)$ множеством полиномов невысокого порядка на узких интервалах всего интервала интерполяции (a, b) . Такие полиномы называются сплайнами. Сплайны, реализуемые в универсальных математических программных средствах, бывают второй, третьей и, редко, четвертой степени. Наиболее часто применяются кубические сплайны.

Показано, что кубический сплайн есть самая гладкая из функций, интерполирующих заданные точки. Параметры сплайна выбираются такими, чтобы сплайн проходил через ближайшие узловые точки, при этом в граничных точках значение сплайна должно быть равно значениям функции $f(x)$. Кроме того, в этих точках должны совпадать первая и вторая производные.

Таким образом, сплайн имеет следующие особенности:

- график функции интерполяции проходит точно через узловые точки каждого сплайна;
- в узловых точках разрывы функции и ее производных отсутствуют;
- сплайны обеспечивают высокую точность интерполяции во всем диапазоне интерполяции.

Пример 1.5

Допустим, функция $y = f(x)$ задана в виде таблицы (табл. 1.8).

Таблица 1.8

	Значения переменных									
x	1	2	3	4	5	6	7	8	9	10
y	0.5	-1	-0.5	5	18.5	43	81.5	137	212.5	311

```

> readlib(spline):
_x:=[1,2,3,4,5,6,7,8,9,10]:
_y:=[0.5,-1,-0.5,5,18.5,43,81.5,137,212.5,311]:
res:=spline(_x,_y,x,cubic):
evalf(res,4);

```

$$\begin{cases}
 2.000 - .9229x - .8657x^2 + .2886x^3 & x < 2. \\
 -1.487 + 2.300x - 2.477x^2 + .5572x^3 & x < 3. \\
 1.859 + .2925x - 1.808x^2 + .4828x^3 & x < 4. \\
 .01221 + 1.678x - 2.154x^2 + .5117x^3 & x < 5. \\
 5.145 - 1.402x - 1.538x^2 + .4706x^3 & x < 6. \\
 -24.10 + 13.22x - 3.976x^2 + .6060x^3 & x < 7. \\
 147.6 - 60.36x + 6.536x^2 + .1054x^3 & x < 8. \\
 -808.2 + 298.1x - 38.27x^2 + 1.972x^3 & x < 9. \\
 4271. - 1395. x + 149.8 x^2 - 4.994 x^3 & \text{otherwise}
 \end{cases}$$

[>

Time: 3.1s | Bytes: 3.06M | Available: 963M

Рис. 1.4

Необходимо выполнить интерполяцию сплайнами третьей степени. Сплайн-интерполяция реализована во всех универсальных математических программных средствах. Процедуры решения задачи приводятся в последующих главах. Решение нами выполнено в среде Maple и имеет вид, представленный на рис. 1.4.

Ответ приведен с округлением до трех значащих цифр после запятой.

Из ответа видно, что Maple выдала решение в виде условных выражений для $x < k$. Легко убедиться, что во всех узловых точках значение функции совпадает с табличным.

Универсальные программные средства символьной математики позволяют также получить решение в виде процедур.

1.3. Интерполяция приближенная в узлах

Интерполяцию приближенную в узлах, в математике называют аппроксимацией. Пусть результатом эксперимента по установлению некоторой физической закономерности является таблица. Необходимо найти функцию в виде формулы. Эксперимент без погрешностей провести практически невозможно. Поэтому в данном случае решение задачи интерполяции точной в узлах сомнительно, т. к. нет уверенности в высокой точности эксперимента. Естественным является желание найти такую функцию, которая устанавливала бы зависимость между x и y наиболее точно. Такая функция, называемая эмпирической, устанавливает закономерности изучаемого явления и нередко является физическим законом. Геометрический смысл задачи аппроксимации виден из рис. 1.5.

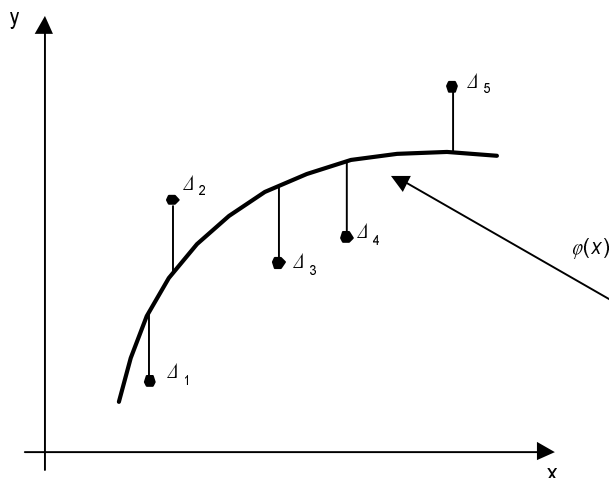


Рис. 1. 5

На рис. 1.5 приняты следующие обозначения:

- $M_i(x_i, y_i)$ — координаты точек, полученные экспериментально (помечены точками) ($i = 1, 2, \dots, n$);
- $\varphi(x)$ — эмпирическая функция;
- Δ_i — разность в i -той точке между эмпирической функцией и значением y_i , полученным из опыта, т. е. $\Delta_i = \varphi(x_i) - y_i$.

Разности Δ_i называются отклонениями или невязками. Они могут быть как положительными, так и отрицательными. Задачей аппроксимации является отыскание такой эмпирической функции $\varphi(x)$, которая в узлах интерполяции $x_1, x_2, \dots, x_n$ мало отличалась бы от экспериментальных данных.

При сформулированной выше постановке задачи у исследователя возникают следующие три вопроса.

- Какими критериями следует оценивать близость эмпирической функции $\varphi(x)$ и данных эксперимента?
- Как выбрать вид эмпирической функции?
- Как получить эмпирическую функцию $\varphi(x)$, удовлетворяющую условиям близости к экспериментальным данным?

Ответим на эти вопросы.

Критериями близости могут быть следующие условия:

- алгебраическая сумма всех отклонений равна нулю, т. е.

$$\delta = \sum_{i=1}^n \Delta_i = 0; \quad (1.24)$$

- сумма квадратов отклонений является минимальной, т. е.

$$\delta = \sum_{i=1}^n \Delta_i^2 = \sum_{i=1}^n (\varphi(x_i) - y_i)^2 = \min; \quad (1.25)$$

- среднее значение всех отклонений является минимальным, т. е.

$$\delta = \frac{\sum_{i=1}^n \Delta_i}{n} = \min. \quad (1.26)$$

На практике наиболее часто для решения задач интерполяции применяются следующие функции:

- степенного вида

$$y = ax^b \quad \text{или} \quad y = ax^b + c; \quad (1.27)$$

- логарифмическая

$$y = a + b \ln(x); \quad (1.28)$$

□ дробно-линейные

$$y = a + \frac{b}{x}, \quad y = \frac{1}{a + b x}, \quad y = \frac{x}{a + b x}; \quad (1.29)$$

□ показательная

$$y = a b^x + c; \quad (1.30)$$

□ дробно-рациональная

$$y = \frac{b_0 + b_1 x + b_m x^m}{a_0 + a_1 x + \dots + a_n x^n}; \quad (1.31)$$

□ многочлен степени n

$$y = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n. \quad (1.32)$$

Наиболее простым является следующий способ выбора вида эмпирической функции. На плоскости x, y изображаются все точки $M(x, y)$ таблицы, полученной экспериментально. Проводится кривая, проходящая через все точки или вблизи к ним. Экспериментальная кривая сравнивается с графиками типичных функций, на основании этого сравнения выбирается эмпирическая функция. Может оказаться, что графики нескольких типичных функций похожи на экспериментальную кривую. В таком случае возможны следующие два способа выбора наиболее подходящей функции:

□ способ 1. Находится несколько эмпирических функций, вычисляются отклонения и в соответствии с выбранным критерием выбирается оптимальная;

□ способ 2. Осуществляется выравнивание функции $y = \varphi(x)$ путем ее преобразования в линейную функцию методом замены переменных x и y .

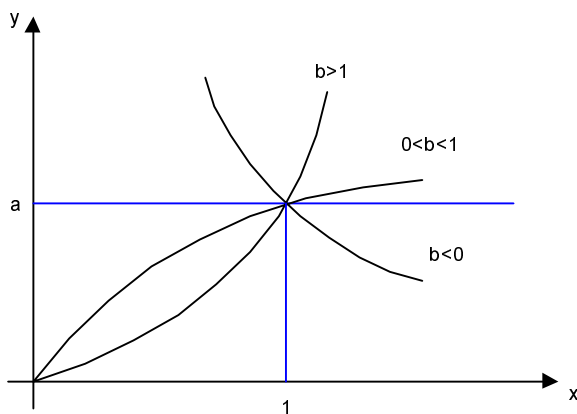


Рис. 1.6

Покажем графики типичных функций и способы их выравнивания.

□ Степенная функция $y = a x^b$. График функции приведен на рис. 1.6.

Выравнивание степенной функции можно выполнить логарифмированием: $\ln(y) = \ln(a) + b \ln(x)$. Тогда в новых переменных $Y = \ln(y)$, $X = \ln(x)$ имеем уравнение прямой $Y = \ln(a) + bX$. Таким образом, чтобы убедиться, что степенная функция может быть выбрана в качестве эмпирической, достаточно преобразовать экспериментальные данные x и y в натуральные логарифмы и убедиться, что точки в координатах X , Y лежат на прямой или расположены достаточно близко от нее.

□ Логарифмическая функция $y = a + b \ln(x)$. График функции представлен на рис. 1.7.

Выравнивание логарифмической функции осуществляется заменой переменных $\ln(x) = X$, $y = Y$. При такой замене переменных получаем линейную функцию $Y = a + bX$. Таким образом, если точки с координатами $(\ln(x_i), y_i)$ находятся на одной прямой, то эмпирической зависимостью является логарифмическая функция.

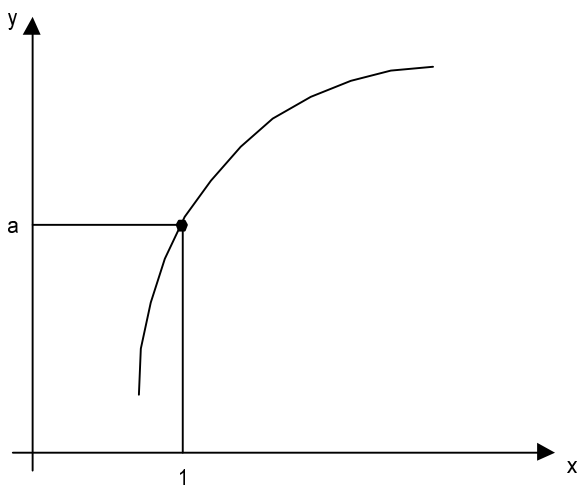


Рис. 1.7

□ Дробно-линейная функция $y = \frac{x}{a + b x}$. График функции показан на рис. 1.8.

Выравнивание дробно-линейной функции осуществляется заменой переменных $x = X$, $\frac{x}{y} = Y$. Тогда дробно-линейная функция приобретает вид:

$Y = a + bX$. Следовательно, дробно-линейная функция может быть эмпирической, если точки с координатами $(x_i, \frac{x_i}{y_i})$ будут располагаться на одной прямой.

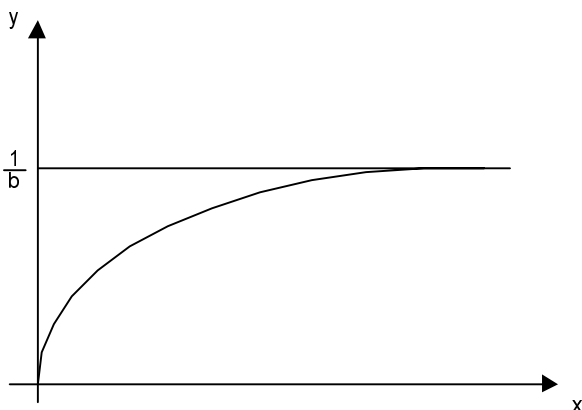


Рис. 1.8

□ Показательная функция $y = ab^x$. График показательной функции приведен на рис. 1.9.

Выравнивание показательной функции осуществляется логарифмированием $\ln(y) = \ln(a) + x \ln(b)$. Заменой переменных $\ln(y) = Y$, $x = X$ получаем линейную функцию вида $Y = \ln(a) + X \ln(b)$. Таким образом, показательная функция может быть эмпирической, если точки с координатами $(x_i, \ln(y_i))$ будут находиться на одной прямой.

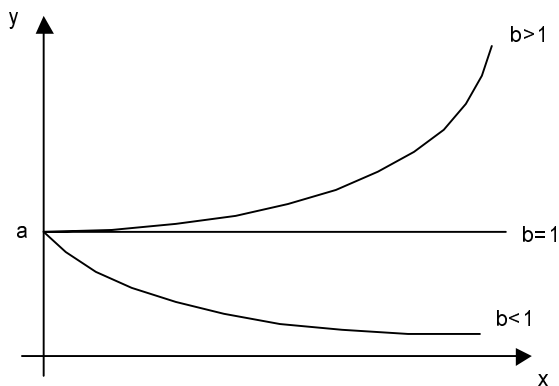


Рис. 1.9

Замечание

Для проверки линейности функции после ее выравнивания нет необходимости наносить точки таблицы на плоскость (x, y) . Достаточно убедиться, что первые табличные разности в новых координатах имеют одинаковые значения.

Аппроксимация многочленами степени n выравнивания не требует, даже если это возможно, т. к. установить принадлежность этой функции к данному классу и определить степень многочлена можно по табличным разностям.

Пример 1.6

Пусть в результате эксперимента получены данные, приведенные в табл. 1.9.

Таблица 1.9

	Значения переменных					
x	1	2	3	4	5	6
y	0.37	0.69	0.97	1.21	1.43	1.62

Необходимо определить эмпирическую функцию $\varphi(x)$. График функции имеет вид, показанный на рис. 1.10.

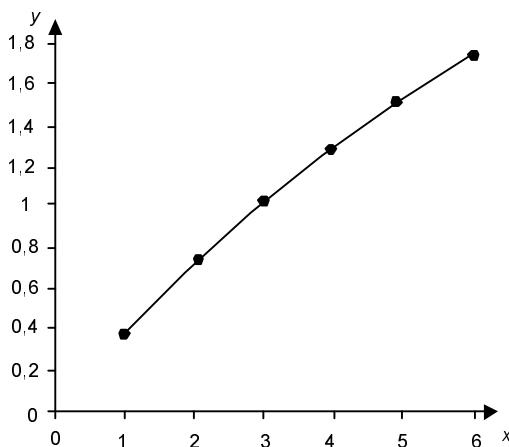


Рис. 1.10

Сравнивая график функции с графиками, приведенными выше, видим, что функция похожа на степенную $y = ax^b$, дробно-линейную $y = \frac{x}{a + bx}$, логарифмическую $y = a + b \ln(x)$.

Какую же из этих функций выбрать?

Выполним выравнивание этих функций. Для этого представим данные таблицы в новых координатах (X , Y), где:

□ $X = \ln(x)$, $Y = \ln(y)$ (для степенной функции);

□ $X = x$, $Y = \frac{x}{y}$ (для дробно-линейной);

□ $X = \ln(x)$, $Y = y$ (для логарифмической).

Найдем для всех этих функций первые табличные разности. Результаты вычислений приведены в таблицах (табл. 1.10, 1.11, 1.12.)

Таблица 1.10. Степенная функция

	Значения переменных					
x	1	2	3	4	5	6
y	0.37	0.69	0.97	1.21	1.43	1.62
$X = \ln(x)$	0	0.69	1.1	1.39	1.61	1.79
$Y = \ln(y)$	-0.99	-0.37	-0.03	0.19	0.36	0.48
Δy_0	0.9	0.83	0.76	0.77	0.67	

Таблица 1.11. Дробно-линейная функция

	Значения переменных					
x	1	2	3	4	5	6
y	0.37	0.69	0.97	1.21	1.43	1.62
$X = x$	1	2	3	4	5	6
$Y = x / y$	2.7	2.9	3.1	3.3	3.5	3.7
Δy_0		0.2	0.2	0.2	0.2	0.2

Таблица 1.12. Логарифмическая функция

	Значения переменных					
x	1	2	3	4	5	6
y	0.37	0.69	0.97	1.21	1.43	1.62
$X = \ln(x)$	0	0.69	1.1	1.39	1.61	1.79
$Y = y$	0.37	0.69	0.97	1.21	1.43	1.62
Δy_0	0.46	0.68	0.83	1	1.06	

Замечание

При определении табличных разностей в случае степенной и логарифмической функций следует вычислять разностные отношения, т. к. узлы этих функций в новых координатах неравноотстоящие.

Из выполненных расчетов видно, что только дробно-линейная функция может быть эмпирической аппроксимирующей функцией. Определим значение коэффициентов a и b дробно-линейной функции $y = \frac{x}{a+bx}$. Выберем следующие две точки из табл. 1.11: (1, 0.37) и (6, 1.62). Тогда система уравнений для определения неизвестных a и b будет иметь вид:

$$0.37 = \frac{1}{a+bx}$$

$$1.62 = \frac{6}{a+6x}$$

Решение этой системы уравнений дает следующие значения коэффициентов:

□ $a = 2.5;$

□ $b = 0.2.$

Подставляя эти значения в выражение для дробно-линейной функции, получим следующую эмпирическую аппроксимирующую функцию:

$$y = \frac{x}{2.5 + 0.2x}.$$

Методы аппроксимации можно классифицировать по критериям близости эмпирической функции и данных эксперимента. Согласно рис. 1.1, методам интерполяции приближенной в узлах (аппроксимации) являются:

- метод выбранных точек (основан на предположении, что сумма уклонений минимальна, при этом функция подбирается "на глаз", приближенно);
- метод средних (предполагает, что сумма всех уклонений равна нулю);
- метод наименьших квадратов (МНК) (основан на критерии оптимальности, предполагающем, что сумма квадратов всех уклонений минимальна).

Рассмотрим эти методы более подробно.

1.3.1. Метод выбранных точек

Сущность метода.

На координатную плоскость x, y наносятся экспериментально полученные точки $M_i(x_i, y_i)$. Выбираются m точек, через которые пройдет эмпирическая

1.3.2. Метод средних

[illegible]

Пусть, например, в качестве эмпирической функции выбран многочлен второй степени $\varphi(x) = a_0 + a_1x + a_2x^2$. Тогда $\Delta_i = (a_0 + a_1x_i + a_2x_i^2) - y_i$. Поскольку $\varphi(x)$ содержит три неизвестных, то разобьем диапазон изменения аргумента на три участка с числом точек, равным k в первом участке, m точек — во втором и $(n - m - k)$ точек в третьем участке. Подставляя значения отклонений в (1.33), получим следующую систему уравнений:

$$\begin{aligned}
a_0 k + a_1 \sum_{i=1}^k x_i + a_2 \sum_{i=1}^k x_i^2 &= \sum_{i=1}^k y_i \\
a_0 m + a_1 \sum_{i=k+1}^{k+m} x_i + a_2 \sum_{i=k+1}^{k+m} x_i^2 &= \sum_{i=k+1}^{k+m} y_i \\
a_0 (n - k - m) + a_1 \sum_{i=k+m+1}^n x_i + a_2 \sum_{i=k+m+1}^n x_i^2 &= \sum_{i=k+m+1}^n y_i
\end{aligned} \tag{1.34}$$

В результате решения этой системы уравнений получим коэффициенты эмпирической функции, представляющей собой параболу $\varphi(x) = a_0 + a_1 x + a_2 x^2$. Таким образом, задача аппроксимации свелась, как и в случае выбранных точек, к решению системы алгебраических уравнений.

Метод средних на практике дает хорошие результаты. Однако он требует большого объема экспериментальных данных, а также выполнения условия, чтобы сумма отклонений на каждом участке была равна нулю. Это может вызвать затруднения в случае, если эмпирическая формула содержит много неизвестных коэффициентов (например, многочлен высокой степени).

1.3.3. Метод наименьших квадратов

Сущность метода.

Метод наименьших квадратов (МНК) основан на предположении, что сумма квадратов отклонений должна достигать минимального значения.

Пусть функция задана в виде таблицы, значения которой $y_1, y_2, \dots, y_n$ получены экспериментально при значениях аргументов $x_1, x_2, \dots, x_n$ соответственно. Эмпирическую функцию аппроксимации будем искать в виде многочлена степени m :

$$\varphi(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^m.$$

За критерий оптимальности примем минимум суммы квадратов отклонений:

$$\xi = \sum_{i=1}^n (\varphi(x_i) - y_i)^2 \rightarrow \min$$

или

$$\xi = \sum_{i=1}^n (a_0 + a_1 x_i + a_2 x_i^2 + \dots + a_m x_i^m - y_i)^2 \rightarrow \min. \tag{1.35}$$

Следует иметь в виду, что решение задачи аппроксимации возможно, если $m < n - 1$. Если $m = n - 1$, то задача аппроксимации сводится к задаче интерполяции точной в узлах. При $m > n - 1$ решения не существует из-за недостаточного объема данных.

$$\begin{aligned} \frac{\partial \xi}{\partial a_0} &= 2 \sum_{i=1}^n (a_0 + a_1 x_i + a_2 x_i^2 + \dots + a_m x_i^m - y_i)^2 = 0 \\ \frac{\partial \xi}{\partial a_1} &= 2 \sum_{i=1}^n (a_0 + a_1 x_i + a_2 x_i^2 + \dots + a_m x_i^m - y_i)^2 x_i = 0 \\ \frac{\partial \xi}{\partial a_2} &= 2 \sum_{i=1}^n (a_0 + a_1 x_i + a_2 x_i^2 + \dots + a_m x_i^m - y_i)^2 x_i^2 = 0 \\ &\dots\dots\dots \\ \frac{\partial \xi}{\partial a_m} &= 2 \sum_{i=1}^n (a_0 + a_1 x_i + a_2 x_i^2 + \dots + a_m x_i^m - y_i)^2 x_i^m = 0. \end{aligned} \quad (1.36)$$
$$\begin{aligned}
a_0 \sum_{i=1}^n x_i + a_1 \sum_{i=1}^n x_i^2 + \dots + a_m \sum_{i=1}^n x_i^m &= \sum_{i=1}^n y_i \\
a_0 \sum_{i=1}^n x_i + a_1 \sum_{i=1}^n x_i^2 + a_2 \sum_{i=1}^n x_i^3 + \dots + a_m \sum_{i=1}^n x_i^{m+1} &= \sum_{i=1}^n x_i y_i \\
a_0 \sum_{i=1}^n x_i^2 + a_1 \sum_{i=1}^n x_i^3 + a_2 \sum_{i=1}^n x_i^4 + \dots + a_m \sum_{i=1}^n x_i^{m+2} &= \sum_{i=1}^n x_i^2 y_i \quad (1.37)
\end{aligned}$$

Решение задачи.

Таблица 1.13

	Значения переменных					
x	1	2	3	4	5	6
y	5.1	15.6	32.3	55.2	84.3	119.6

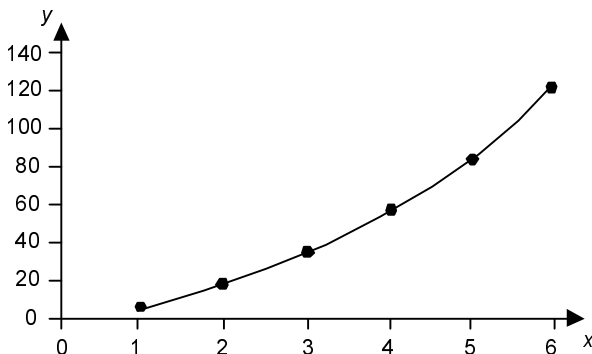


Рис. 1.11

Сравнивая график рис. 1.11 с графиками типичных функций, видим, что похожими являются степенная функция $y = ax^b$ и показательная $y = ab^x$. Однако ни одна из них не может быть аппроксимирующей, в чем можно убедиться, выполнив выравнивание этих функций и определив первые табличные разности, которые не являются величинами постоянными. Выберем в качестве аппроксимирующей функции многочлен степени n . Вначале примем $n = 2$, т. е. $\varphi(x) = a_0 + a_1 x + a_2 x^2$.

Определим коэффициенты a_0 , a_1 , a_2 , пользуясь методами выбранных точек, средних и наименьших квадратов.

□ Метод выбранных точек.

В соответствии с графиком (рис. 1.11) и данными таблицы 1.13 выберем следующие три точки $\{1, 5.1\}$, $\{3, 32.3\}$, $\{6, 119.6\}$ и составим систему уравнений:

$$a_0 + a_1 \cdot 1 + a_2 \cdot 1^2 = 5.1$$

$$a_0 + a_1 \cdot 3 + a_2 \cdot 3^2 = 32.3$$

$$a_0 + a_1 \cdot 6 + a_2 \cdot 6^2 = 119.6$$

Решая эту систему, получим:

- $a_0 = 0.8$;

- $a_1 = 1.2$;
- $a_2 = 3.1$.

Тогда эмпирическая функция будет иметь вид:

$$\varphi(x) = 0.8 + 1.2x + 3.1x^2.$$

□ Метод средних.

Разобьем функцию на три участка. Предположим, что на каждом из участков алгебраическая сумма погрешностей равна нулю. Тогда:

- $\Delta_1 + \Delta_2 = 0$;
- $\Delta_3 + \Delta_4 = 0$;
- $\Delta_5 + \Delta_6 = 0$.

Поскольку $\Delta_i = a_0 + a_1x_i + a_2x_i^2 - y_i$, то после подстановки Δ_i в предыдущую систему уравнений получим:

$$2a_0 + a_1(x_1 + x_2) + a_2(x_1^2 + x_2^2) = y_1 + y_2$$

$$2a_0 + a_1(x_3 + x_4) + a_2(x_3^2 + x_4^2) = y_3 + y_4$$

$$2a_0 + a_1(x_5 + x_6) + a_2(x_5^2 + x_6^2) = y_5 + y_6$$

Подставляя в эти уравнения значения x_i и y_i из табл. 1.13, получим:

$$2a_0 + 3a_1 + 5a_2 = 20.7$$

$$2a_0 + 7a_1 + 25a_2 = 87.5$$

$$2a_0 + 11a_1 + 61a_2 = 203.9$$

В результате решения этой системы уравнений получим:

- $a_0 = 0.8$;
- $a_1 = 1.2$;
- $a_2 = 3.1$.

Тогда эмпирическая функция будет иметь вид:

$$\varphi(x) = 0.8 + 1.2x + 3.1x^2.$$

□ Метод наименьших квадратов.

В нашем примере функция интерполяции представляет собой многочлен второй степени. Тогда на основании (1.37) имеем:

$$a_0n + a_1 \sum_{i=1}^n x_i + a_2 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n y_i$$

$$a_0 \sum_{i=1}^n x_i + a_1 \sum_{i=1}^n x_i^2 + a_2 \sum_{i=1}^n x_i^3 = \sum_{i=1}^n x_i y_i$$

$$a_0 \sum_{i=1}^n x_i^2 + a_1 \sum_{i=1}^n x_i i + a_2 \sum_{i=1}^n x_i = \sum_{i=1}^n x_i^2 y_i.$$

В нашем случае

$$n = 6, \sum_{i=1}^n x_i = 21, \sum_{i=1}^n x_i^2 = 91, \sum_{i=1}^n x_i i = 441, \sum_{i=1}^n x_i^4 = 2275, \\ \sum_{i=1}^n y_i = 312.1, \sum_{i=1}^n x_i y_i = 1493.1, \sum_{i=1}^n x_i^2 y_i = 7654.5.$$

Подставляя эти значения в систему уравнений, получим:

$$6a_0 + 21a_1 + 91a_2 = 312.1$$

$$21a_0 + 91a_1 + 441a_2 = 1493.1$$

$$91a_0 + 441a_1 + 2275a_2 = 7654.5.$$

Результатами решения этой системы уравнений являются:

- $a_0 = 0.8;$
- $a_1 = 1.2;$
- $a_2 = 3.1.$

Тогда функция интерполяции будет иметь вид:

$$\varphi(x) = 0.8 + 1.2x + 3.1x^2.$$

В данном случае решение совпадает с полученным по методам средних и выбранных точек.

Табулируя полученную нами функцию, убеждаемся, что погрешность формулы равна нулю.

1.3.4. Аппроксимация Паде

Полиномиальная интерполяция может приводить к большим ошибкам в не узловых точках функции интерполяции. Более того, ошибки могут увеличиваться с повышением степени полинома.

Уменьшить погрешность интерполяции можно подбором функции интерполяции с различным базисом. Одним из таких способов является интерполяция дробно-рациональными функциями, при которой функция интерполяции представляется в виде отношения двух полиномов:

$$\varphi(x) = \frac{a_0 + a_1x + a_2x^2 + \dots + a_nx^n}{b_0 + b_1x + b_2x^2 + \dots + b_mx^m}.$$

Такая интерполяция называется аппроксимацией Паде.

Высокая точность этой интерполяции достигается полиномами Чебышева.

Полиномы Чебышева первого рода $T_n(x)$ и второго рода $U_n(x)$ являются решениями дифференциального уравнения

$$(1 - x^2) \frac{d^2 y(x)}{dx^2} - x \frac{dy(x)}{dx} + n^2 y(x) = 0.$$

Полиномы имеют вид:

$$T_n(x) = \cos(n \arccos x) = x^n - c_2^n x^{n-2} (1-x^2) + c_4^n x^{n-4} (1-x^2)^2 - c_6^n x^{n-6} (1-x^2)^3 + \dots$$

$$U_n(x) = \frac{\sin((n+1) \arccos x)}{\sin x} = c_1^{n+1} x^n - c_3^{n+1} x^{n-2} (1-x^2) + c_5^{n+1} x^{n-4} (1-x^2)^2 + \dots$$

Для малых n полиномы имеют вид:

$$\begin{array}{ll} T_0(x) = 1 & U_0(x) = 1 \\ T_1(x) = x & U_1(x) = 2x \\ T_2(x) = 2x^2 - 1 & U_2(x) = 4x^2 - 1 \\ T_3(x) = 4x^3 - 3x & U_3(x) = 8x^3 - 4x \\ T_4(x) = 8x^4 - 8x^2 + 1 & U_4(x) = 16x^4 - 12x^2 + 1 \end{array}$$

К сожалению, аппроксимация Паде требует аналитического задания функции. Она не позволяет решать задачи интерполяции при табличном задании функции.

Приведем результаты аппроксимации Паде функции $y = e^{-x} \sin(x)$ дробно-рациональной функцией и с помощью полиномов Чебышева.

Аппроксимация выполнена для случая полиномов третьей степени.

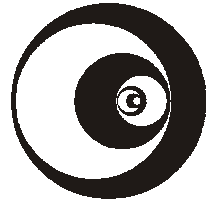
Аппроксимация Паде:

$$y = \frac{\frac{19}{120}x^3 - \frac{3}{4}x^2 + x}{1 + \frac{1}{4}x + \frac{3}{40}x^2 - \frac{1}{120}x^3} = \frac{0.158x^3 - 0.75x^2 + x}{1 + 0.25x + 0.075x^2 - 0.0083x^3}.$$

Аппроксимация Паде с помощью полиномов Чебышева:

$$\begin{aligned} y &= \frac{-0.374 T(0, x) + 1.091 T(1, x) - 0.374 T(2, x) + 0.04 T(3, x)}{T(0, x) + 0.212 T(1, x) + 0.029 T(2, x) - 0.00382 T(3, x)} \\ &= \frac{-0.912 \cdot 10^{-5} + 0.971x - 0.748x^2 + 0.16x^3}{0.971 + 0.223x + 0.0579x^2 - 0.0153x^3}. \end{aligned}$$

Из ответов видно, что результаты аппроксимации Паде алгебраическими полиномами и полиномами Чебышева различны.



Глава 2

Компьютерные технологии решения задач интерполяции в системе Derive 5

2.1. Интерполяция точная в узлах. Универсальный метод

Решение задачи интерполяции универсальным методом сводится к следующей последовательности действий:

1. Выбор вида функции интерполяции.
2. Образование системы уравнений в виде матрицы.
3. Решение системы уравнений.
4. Проверка правильности решения.

Опишем процесс реализации этих действия в системе Derive.

Исходные данные вводятся в виде матрицы размером $n \times 2$, где n — число узлов функции $y = f(x)$, заданной в виде таблицы. Функция $y = f(x)$ представляется графически, аппроксимирующая функция $\varphi(x)$ подбирается визуально путем сравнения графика, представленного в виде точек, с графиками типичных функций.

Вводится (n раз) в память ПК уравнение функции $\varphi(x)$. При этом каждое последующее уравнение образуется из предыдущего путем подстановки в предыдущее уравнение новых переменных x и y . Далее образуется система уравнений в виде матрицы, элементами которой при вводе являются метки $\#K$, присваиваемые программой каждому из уравнений.

Решение системы уравнений осуществляется с помощью функции Solve, при этом Derive выдает ответ в виде численных значений коэффициентов функции интерполяции. Правильность решения задачи проверяется путем табулирования функции $\varphi(x)$ и сравнения с таблицей исходных данных.

Процедуры интерполяции с помощью Derive подробно покажем на примере.

Пример 2.1

Пусть функция $y = f(x)$ задана в виде таблицы (табл. 2.1).

Таблица 2.1

	Значения переменных			
x	1	2	3	4
y	6.2	4.1	1.9	0.6

Необходимо найти $\varphi(x)$ — функцию интерполяции точную в узлах, если $\varphi(x)$ является многочленом n -ой степени: $y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$. В данном случае число узлов $n = 4$, тогда многочлен должен быть степени $n - 1$, т. е. $y = a_0 + a_1x + a_2x^2 + a_3x^3$.

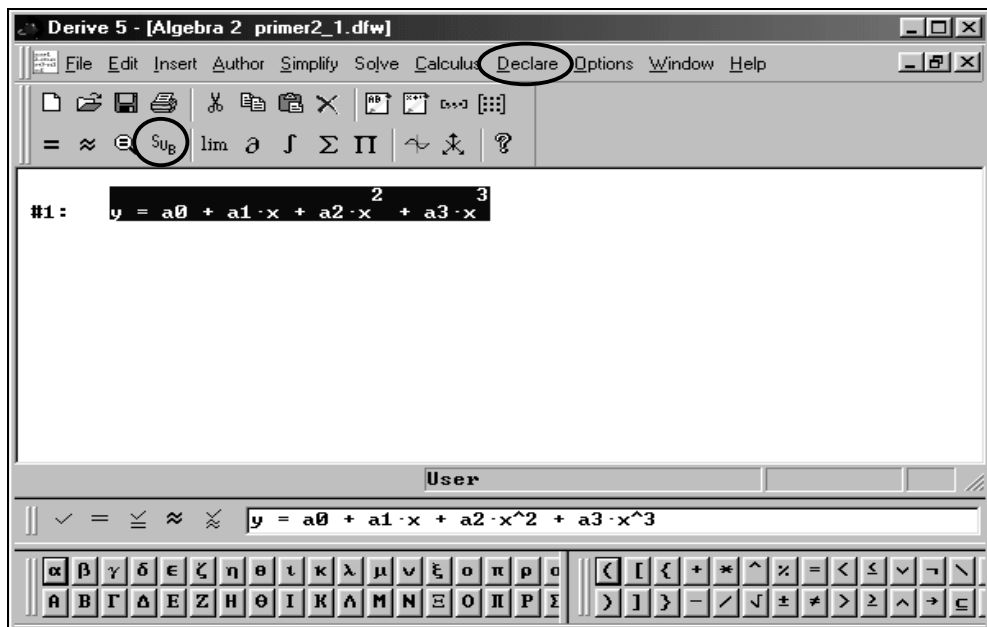


Рис. 2.1

Рассмотрим технологию решения задачи.

1. Настройка Derive на ввод переменных с индексами.

В главном меню (рис. 2.1) надо нажать на пункт меню **Declare**, откроется диалоговое окно **Input Settings** (рис. 2.2). В группе **Input Mode** надо активизировать переключатель **Word**. Одновременно в группе **Case Sensitivity** активизируется переключатель **Insensitive** и в поле **Radix** устанавливается десятичная система счисления (**Decimal**). Надо нажать на клавишу <Enter> или щелкнуть по кнопке **OK**.

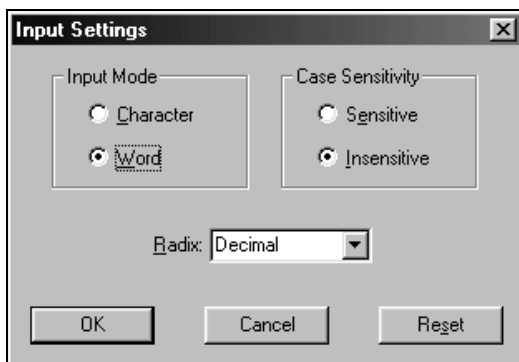


Рис. 2.2

2. Ввод уравнений.

Активизируется строка пользователя, где надо ввести уравнение:

$y = a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3$, которое записывается в память нажатием клавиши <Enter> или щелчком мыши по кнопке  (слева от строки пользователя). На экране отобразится уравнение с меткой #1 (см. рис. 2.1).

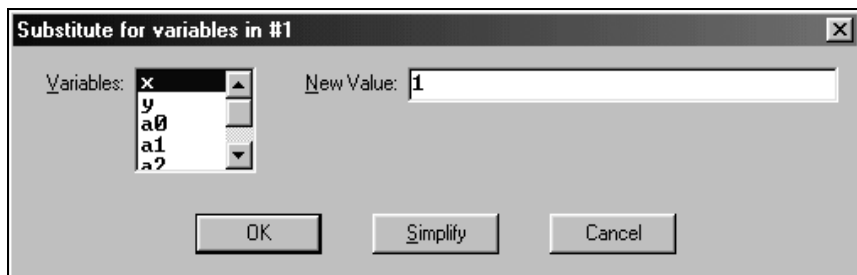


Рис. 2.3

Затем нужно щелкнуть мышью по кнопке **Sub** панели инструментов. На экране откроется диалоговое окно **Substitute for variables in #1** (рис. 2.3).

В поле **Variables** перечислены все переменные уравнения #1. Надо щелкнуть кнопкой мыши по переменной **x** и ввести ее значение в поле **New Value** (в нашем случае 1), затем щелкнуть по переменной **y** и ввести значение **y** (которое равно 6,2). После щелчка мыши по кнопке **OK** или нажатия клавиши <Enter> на экране отобразится уравнение:

$$\#2: 6.2 = a_0 + a_1 \cdot 1 + a_2 \cdot 1^2 + a_3 \cdot 1^3.$$

После щелчка мышью по уравнению #1 производится подстановка значений $x = 2$, $y = 4.1$ аналогично предыдущему. Таким же способом образуются уравнения при $x = 3$, $y = 1.9$ и $x = 4$, $y = 0.6$. На экране с метками #2, #3, #4, #5 увидим следующие четыре уравнения:

$$\#2: 6.2 = a_0 + a_1 \cdot 1 + a_2 \cdot 1^2 + a_3 \cdot 1^3;$$

$$\#3: 4.1 = a_0 + a_1 \cdot 2 + a_2 \cdot 2^2 + a_3 \cdot 2^3;$$

$$\#4: 1.9 = a_0 + a_1 \cdot 3 + a_2 \cdot 3^2 + a_3 \cdot 3^3;$$

$$\#5: 0.6 = a_0 + a_1 \cdot 4 + a_2 \cdot 4^2 + a_3 \cdot 4^3.$$

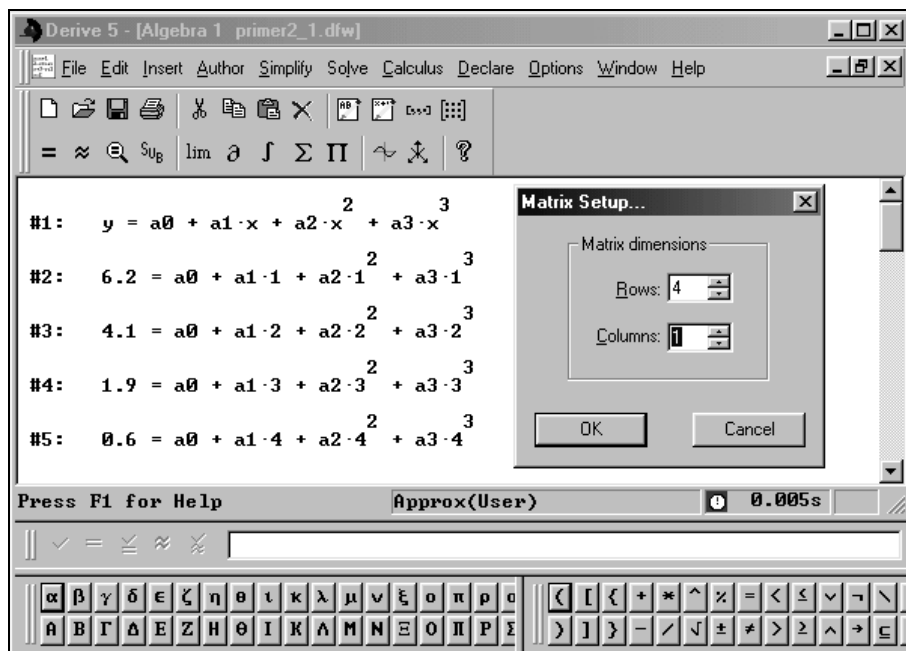


Рис. 2.4

3. Образование системы уравнений.

Выбираем (щелкая мышью) в главном меню пункт **Author | Matrix**. На экране откроется окно **Matrix Setup** (рис. 2.4). В поле **Rows** устанавливаем число уравнений (в данном случае 4), в поле **Columns** устанавливаем

число столбцов (в нашем случае равное 1). Щелкаем по кнопке **ОК**. На экране активизируется окно **Author 4 x 1 matrix** (рис. 2.5). Последовательно вводим метки (номера) уравнений: #2, #3, #4, #5, перемещая курсор клавишей <Tab> или щелчком левой кнопки мыши. После заполнения всех строк таблицы надо нажать клавишу <Enter> или щелкнуть мышью по кнопке **ОК**. На экране появится система уравнений, отмеченная матричными линиями и имеющая метку #6 (рис. 2.6).

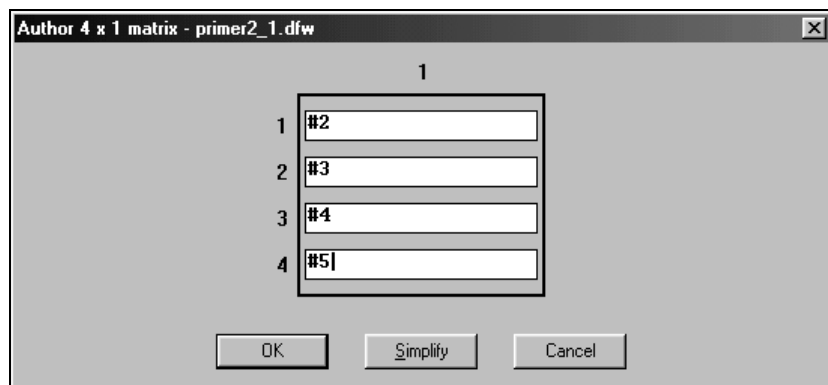


Рис. 2.5

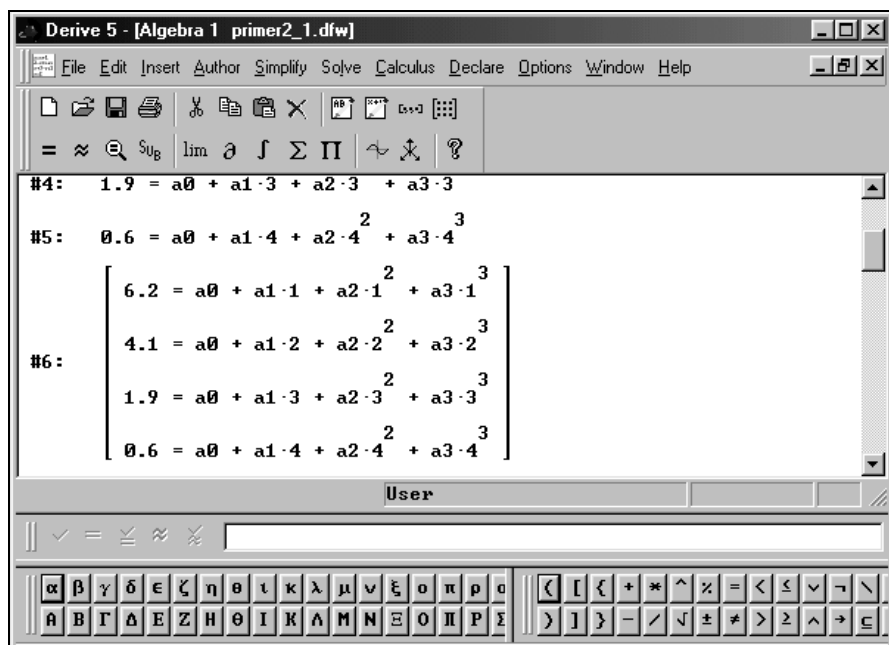


Рис. 2.6

4. Решение системы уравнений:

- выполняем пункт главного меню **Solve | Expression**. На экране активируется диалоговое окно **Solve Expression** (рис. 2.7).
- в поле **Solution Variables** активизируем щелчком мыши все неизвестные a_0, a_1, a_2, a_3 ; в поле **Solution Method** активизируем переключатель **Either**, а в поле **Solution Domain** — переключатель **Real**.
- щелкаем мышью по кнопке **Solve**.

На экране увидим значения коэффициентов:

$$a_0 = 7.2; a_1 = -0.116666; a_2 = -1.05; a_3 = 0.166666.$$

Тогда интерполяционная формула будет иметь вид:

$$(x) = 7.2 - 0.116666x - 1.05x^2 + 0.166666x^3.$$

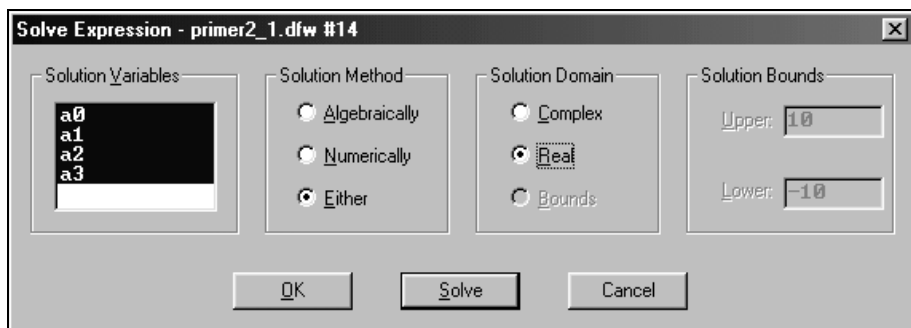


Рис. 2.7

5. Проверка правильности решения задачи интерполяции.

Выполним табулирование функции (x) и сравним с исходными данными, приведенными в табл. 2.1. Табулирование функции в Derive осуществляется с помощью функции $\text{VECTOR}([x, (x)], x, x_n, x_k, h)$ (где x_n, x_k — начальное и конечное значения узлов интерполяции, h — шаг таблицы функции).

Последовательность процедуры проверки решения в Derive:

- ввод функции $\varphi(x)$. На экране появится выражение:

$$\#1: 7.2 - 0.116666x - 1.05x^2 + 0.166666x^3.$$

- ввод в строке пользователя команды:

`VECTOR([x, #1], x, 1, 4, 1).`

На экране отобразится эта команда:

$$\#2: \text{VECTOR}([x, 7.2 - 0.116666x^3 - 1.05x^2 + 0.166666x^3], x, 1, 4, 1).$$

- выполнение команды Approximate (кнопка  на панели инструментов). На экране (рис. 2.8) увидим ответ в виде следующей матрицы:

$$\#3: \begin{bmatrix} 1 & 6.2 \\ 2 & 4.09999 \\ 3 & 1.89998 \\ 4 & 0.59996 \end{bmatrix}$$

Сравнение результата с исходной таблицей показывает, что решение верно.

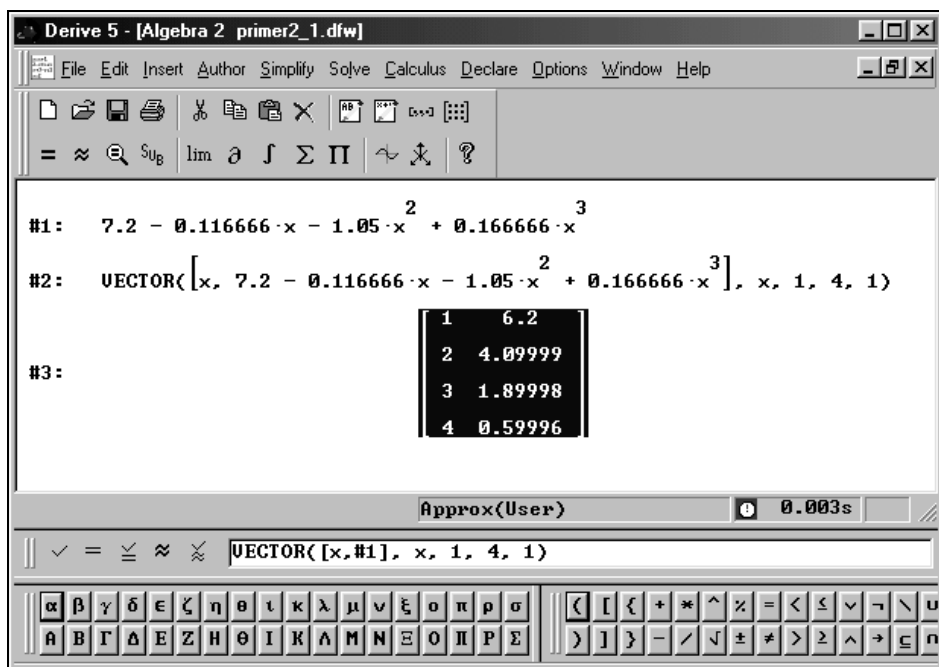


Рис. 2.8

Более рациональным может оказаться матричный способ решения системы уравнений. Пусть A — матрица коэффициентов, X — вектор неизвестных, B — вектор правых частей системы уравнений. В нашем случае:

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \\ 1 & x_4 & x_4^2 & x_4^3 \end{bmatrix}, \quad X = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}, \quad B = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}.$$

Тогда решение ищется в виде $X = A^{-1} \cdot B$.

В Derive матричный метод реализуется следующим образом.

Для ввода векторов X и B выполняем команду главного меню **Author | Vector**, в результате чего на экране активизируется окно **Vector Setup**. В поле **Elements** устанавливаем размер вектора (в нашем случае 4), щелкаем по кнопке **OK**. На экране появится таблица с четырьмя пустыми строками для будущего вектора. Последовательно вводим элементы вектора B : 6.2, 4.1, 1.9, 0.6 и щелкаем по кнопке **OK**. На экране увидим вектор: [6.2, 4.1, 1.9, 0.6].

Аналогично образуется вектор неизвестных, который на экране будет иметь вид:

$[a_0, a_1, a_2, a_3]$.

Создаем матрицу коэффициентов A таким образом, как это описано ранее при создании системы уравнений. Размер матрицы 4×4 .

Пусть метками векторов и матрицы являются:

- #1 — вектор правых частей системы уравнений;
- #2 — вектор неизвестных;
- #3 — матрица коэффициентов.

В строке пользователя вводим выражение: $\#2 = \#3^{(-1)} \cdot \#1$.

На экране появится выражение $X = A^{-1} \cdot B$, представленное в матричном виде.

Щелкаем по пункту главного меню **Simplify** (а можно по любой из двух кнопок панели инструментов: по кнопке  или по кнопке  (команда Approximate)). На экране увидим приближенное решение в виде чисел, представленных в естественной форме (рис. 2.9.). (На экране промежуточное выражение с меткой #4 не появляется. Мы его приводим в тексте для большей наглядности процесса решения системы уравнений)

#1: [6.2, 4.1, 1.9, 0.6]

#2: $[a_0, a_1, a_2, a_3]$

#3:
$$\begin{vmatrix} 1 & 1 & 1^2 & 1^3 \\ 1 & 2 & 2^2 & 2^3 \\ 1 & 3 & 3^2 & 3^3 \\ 1 & 4 & 4^2 & 4^3 \end{vmatrix}$$

$$\#4 : [a_0, a_1, a_2, a_3] = \begin{bmatrix} 1 & 1 & 1^2 & 1^3 \\ 1 & 2 & 2^2 & 2^3 \\ 1 & 3 & 3^2 & 3^3 \\ 1 & 4 & 4^2 & 4^3 \end{bmatrix}^{-1} \cdot [6.2, 4.1, 1.9, 0.6]$$

$$\#5 : [a_0 = 7.2, a_1 = -0.116666, a_2 = -1.05, a_3 = 0.166666]$$

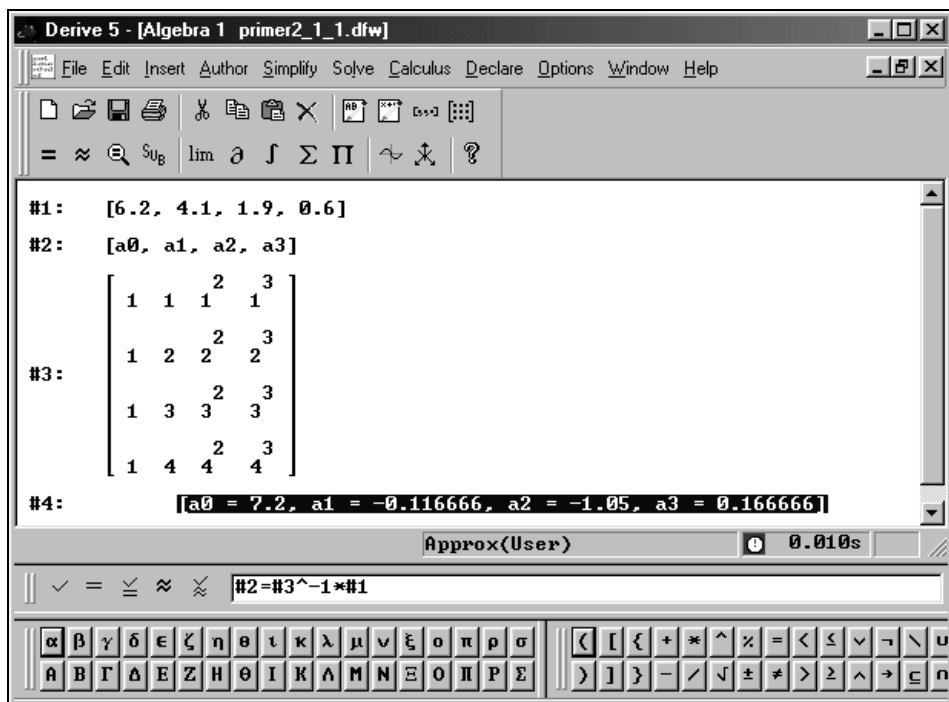


Рис. 2.9

Замечание

При вводе выражения $X = A^{-1} \cdot B$ необходимо соблюдать последовательность символов A и B . Нельзя писать $X = B \cdot A^{-1}$, т. к. решение будет ошибочным.

При большой размерности исходных данных интерполяция точная в узлах применяется для ограниченного диапазона изменения аргумента. При этом интерполяционные формулы могут быть различны для различных участков функции $y = f(x)$. Высокая точность интерполяции для всего диапазона исходных данных дает положительные результаты лишь в случае высокой точности этих данных и при наличии в изучаемом явлении физических зако-

номерностей. При этом решение задачи интерполяции требует вычисления погрешности функции $\varphi(x)$. Рассмотрим типичный для этого случая пример.

Пример 2.2

Пусть зависимость высоты сосны h от ее возраста L , полученная в результате статистической обработки большого количества измерений, приведена в табл. 2.2.

Таблица 2.2

	Значения переменных											
L , лет	20	30	40	50	60	70	80	90	100	110	120	130
h , м	18	22	22.5	28.5	30.5	32	33	34	35	35.5	36	36.3

Необходимо найти математическую модель роста сосны в виде функции $h = \varphi(L)$ и определить высоту сосны для возраста 45 и 67 лет.

Будем предполагать, что существует объективный физический закон, определяющий зависимость параметров сосны от ее возраста, а данные таблицы получены с высокой точностью. Тогда для получения зависимости $h = \varphi(L)$ можно воспользоваться методом интерполяции точной в узлах при ограниченном числе узлов. Из таблицы видно, что функция $h = \varphi(L)$ нелинейна (табличные разности не постоянны). Предположим для начала, что функция $h = \varphi(L)$ является многочленом второй степени: $h = a + bL + cL^2$. Поскольку многочлен содержит три неизвестных — a, b, c , то составим следующую систему из трех уравнений для данных (20, 18), (70, 32), (130, 36.3), взятых из начала, конца и из средней части таблицы:

$$18 = a + 20b + 400c$$

$$32 = a + 70b + 4900c$$

$$36.3 = a + 130b + 16900c.$$

Последовательность решения задачи с помощью Derive описана ранее. В результате решения получим следующие значения коэффициентов:

- $\square \ a = 9.748484848;$
- $\square \ b = 0.4504545454;$
- $\square \ c = -0.001893939393.$

Тогда функция $h = (L)$ будет иметь вид:

$$h = 9.75 + 0.45L - 0.0019L^2 .$$

Так как решение является приближенным и получено лишь по трем значениям исходной функции, то необходимо оценить точность интерполяционной формулы. За критерий близости функций выберем среднеквадратичную

погрешность. Абсолютную среднеквадратичную погрешность вычислим по формуле:

$$\varepsilon = \frac{\sqrt{\sum_{i=1}^n \Delta_i^2}}{n} . \quad (2.1)$$

Здесь:

- $\Delta_i = (f(L_i) - (L_i))$ — разность между i -тым значением исходной функции $f(L)$ и значением функции (L) , полученной в результате интерполяции;
- n — число точек таблицы (в нашем случае $n = 12$).

Рассмотрим технологию определения погрешности в Derive.

1. Образование векторов функций $f(L)$ и (L) .

Вектор функции $f(L)$ образуется так, как описано ранее. Вектор функции (L) образуется путем табулирования. Функция табулирования для нашего случая имеет вид:

VECTOR(#2, x , 20, 130, 10), где:

- #2 — метка функции (L) на экране;
- 20, 130 — начальное и конечное значения;
- 10 — шаг таблицы.

После ввода функции и выполнения команды Approximate (которая выполняется нажатием кнопки  на панели инструментов), на экране появится вектор значений (L) с меткой #4.

#4: [17.99, 21.54, 24.71, 27.5, 29.91, 31.94, 33.59, 34.86, 35.75, 36.26, 36.39, 36.14]

2. Образование суммы квадратов разностей векторов $f(L)$ и (L) .

Пусть #1 и #4 — метки векторов $f(L)$ и (L) на экране монитора. Тогда вектор определяется как разность (#1 — #4). После ввода этого выражения на экране обозначится разность векторов $f(L) - (L)$. Щелкаем по кнопке . На экране отобразится вектор разностей векторов с меткой (например, #6).

#6: [0.01, 0.46, -2.21, 1, 0.59, 0.06, -0.59, -0.86, -0.75, -0.76, 0.39, 0.16]

После ввода выражения #6² на экране увидим квадрат вектора разностей векторов $f(L)$ и (L) . После щелчка мышью по кнопке  на экране отобразится значение суммы квадратов разностей векторов в виде числа с меткой (например, #8).

В нашем случае значение $\sum_{i=1}^{12} \Delta_i^2 = 8.853$, представленное в виде: #8: 8.853.

3. Вычисление погрешностей.

Введем выражение: `sqrt(#8)/12`, затем нажмем клавишу <Enter>. На экране отобразится это выражение. После щелчка мыши по кнопке  (Approximate) на экране увидим ответ: значение среднеквадратичной погрешности. В нашем случае $= 0.247949$ (с меткой #9).

Вычисление максимальной относительной погрешности осуществляется по формуле:

$$\delta = \frac{\varepsilon}{y_{\min}} \cdot 100 \% . \quad (2.2)$$

В нашем случае $y_{\min} = h_{\min} = 18$ м.

Затем введем выражение `δ=#9/18*100`.

После щелчка мыши по кнопке  на экране увидим ответ:
 $= 1.38 \%$.

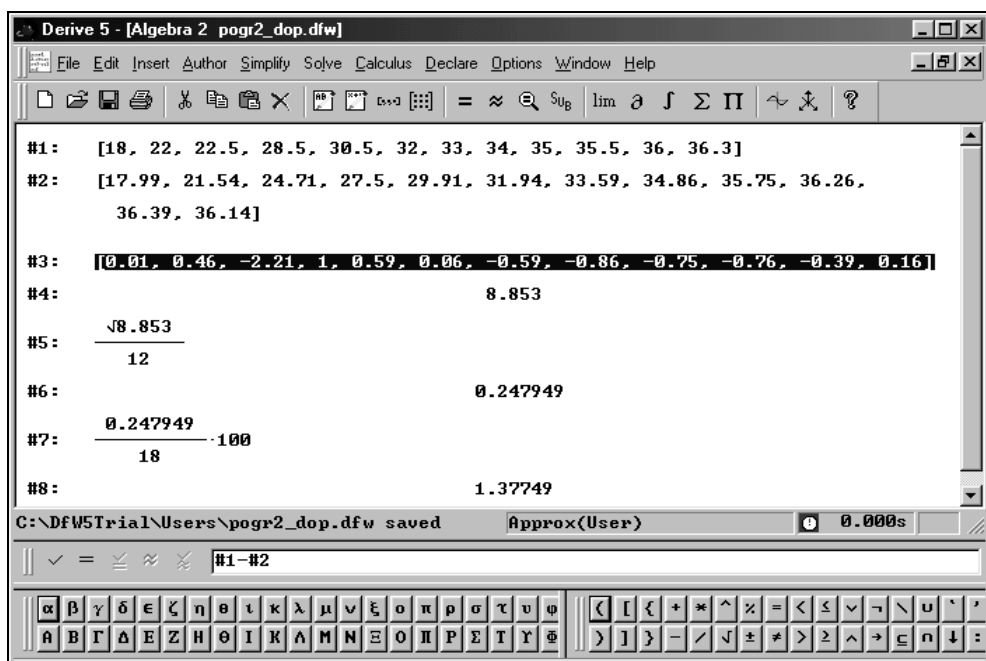


Рис. 2.10

Поскольку максимальная относительная погрешность мала, то полученная в результате интерполяции квадратичная функция может быть принята в качестве модели роста сосны. Процедуры и результаты вычисления погрешности показаны на рис. 2.10.

2.2. Интерполяция по методу Лагранжа. Функция POLY_INTERPOLATE

Технология интерполяции по методу Лагранжа на Derive заключается в следующих действиях:

1. Вызовем утилиту решения задач интерполяции командами: **File | Load | Utility | File | Hermite**. В результате на экране отобразится:
#1: LOAD (Hermite.mth).
2. Введем матрицу исходных данных по методике, описанной в *разд. 2.1 данной главы*. На экране увидим матрицу с меткой #2.
3. Введем функцию: `LAGRANGE (#2, x)`. На экране увидим функцию с меткой #3 и матрицей исходных данных.
4. Для решения задачи выполним пункт **Simplify** (главное меню) или нажмем кнопку  на панели инструментов. На экране отобразится ответ в виде полинома степени $n - 1$, где n — число узлов функции $f(x)$.

Кроме функции `LAGRANGE`, в Derive имеется функция `POLY_INTERPOLATE` решения задач интерполяции многочленами степени n . Функция имеет вид:

`POLY_INTERPOLATE(A, x)`.

Здесь:

- A — матрица исходных данных функции $f(x)$, заданной в виде таблицы;
- x — аргумент функции $f(x)$.

Рассмотрим технологию решения задачи с помощью функции `POLY_INTERPOLATE(A, x)` в системе Derive:

1. Создадим с помощью пунктов меню **Author | Matrix** матрицу исходных данных способом, описанным в *разд. 2.1 данной главы*. Пусть матрица имеет метку #1.
2. Введем в строке пользователя функцию: `POLY_INTERPOLATE (#1, x)`. На экране появится выражение функции, в котором вместо #1 находится матрица.
3. Выполним пункт главного меню **Simplify**. На экране отобразится ответ.

Пример 2.3 Пусть функция $f(x)$ задана в виде таблицы (табл. 2.3).

Таблица 2.3

	Значения переменных				
x	1	2	3	4	5
y	12	9	7	2	1

Необходимо найти функцию интерполяции по методу Лагранжа и с помощью функции POLY_INTERPOLATE. Строго следуя описанным выше технологиям, получим решение задачи интерполяции обоими методами. На рис. 2.11 приведено решение с помощью функции POLY_INTERPOLATE.

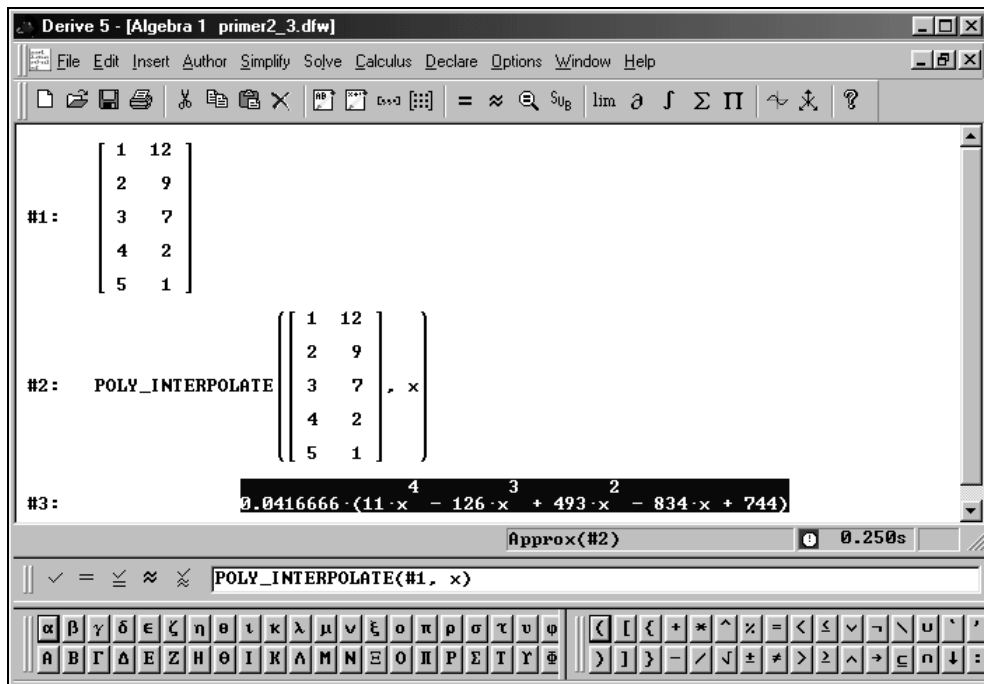


Рис. 2.11

2.3. Интерполяция точная в узлах нелинейными функциями

В Derive задача интерполяции в данном случае сводится к решению систем линейных или нелинейных уравнений. Сведение задачи интерполяции к

решению систем линейных уравнений возможно лишь в том случае, если удастся преобразовать нелинейную функцию в линейную, представив ее в иной системе координат (например, логарифмической).

Таковыми функциями являются:

□ степенная $y = ax^b$;

□ логарифмическая $y = a + b \ln(x)$;

□ дробно-линейные: $y = a + \frac{b}{x}$, $y = \frac{1}{a+bx}$, $y = \frac{x}{a+bx}$.

Способы их линеаризации подробно описаны в *главе 1*.

Во всех этих случаях задача интерполяции точной в узлах сводится к решению системы линейных уравнений. Технология решения ничем не отличается от рассмотренной в *разд. 2.1*, если только не считать того, что функцию $y = f(x)$, показанную в виде таблицы, необходимо представить в иной системе координат.

Если функцию интерполяции не удастся линеаризовать, то задача сводится к решению систем нелинейных уравнений, обычно невысокого порядка. К классу таких функций относится показательная функция $y = ab^x + c$.

В данном случае свести задачу к решению систем линейных уравнений нельзя. Для определения неизвестных a , b и c составляется следующая система нелинейных уравнений:

$$y_1 = ab^{x_1} + c$$

$$y_2 = ab^{x_2} + c$$

$$y_3 = ab^{x_3} + c.$$

Решение возможно численными методами. Наиболее известными из них являются методы Ньютона, а также методы итераций, реализованные в системе Derive.

Рассмотрим технологию решения задач в системе Derive на примере.

Пример 2.4

Пусть функция $y = f(x)$ задана в виде таблицы (табл. 2.4).

Таблица 2.4

	Значения переменных					
x	1	2	3	4	5	6
y	6.5	20	53.5	167	473	1470

Число узлов функции равно шести, т. е. в два раза больше числа неизвестных функции $y = ab^x + c$.

Выберем следующие три узла интерполяции: 1, 3, 6. Тогда система уравнений будет иметь вид:

$$6.5 = ab^1 + c$$

$$53.5 = ab^3 + c$$

$$1470 = ab^6 + c.$$

Метод Ньютона

Функция решения систем нелинейных уравнений методом Ньютона в системе Derive имеет вид:

NEWTONS($[f_1(x, y, \dots), f_2(x, y, \dots), \dots, f_m(x, y, \dots)]$, $[x, y, \dots]$, $[x_0, y_0, \dots]$, n)

Здесь:

- $f_i(x, y, \dots)$ — i -тое уравнение системы из m уравнений, записанное в виде $ab^x + c - y_i$, т. е. без правой части, равной нулю;
- $[x, y, \dots]$ — вектор неизвестных;
- $[x_0, y_0, \dots]$ — вектор начальных значений;
- n — число итераций, устанавливаемое пользователем.

Рациональная последовательность решения задачи в системе Derive:

1. Подбор вида функции интерполяции:

- ввод данных функции $y = f(x)$ в виде матрицы;
- построение графика в виде точек: щелкнуть мышью по пункту главного меню **Window | 2D-plot**, в появившемся окне **2D-plot** щелкнуть мышью по кнопке **Plot Expression** (наглядное расположение точек на плоскости (x, y) регулируется кнопками **Zoom**). По расположению точек можно предположить, что искомая функция является показательной (см. рис. 1.9, $b > 1$).

2. Выход из окна **2D-plot** (производится нажатием кнопки **Algebra Window**). На экране отобразится матрица данных.

3. Решение системы нелинейных уравнений:

- ввод уравнений (как описано в *разд. 2.1 данной главы*) в виде:


```
a * b + c - 6.5
a * b^3 + c - 53.5
a * b^6 + c - 1470
```
- ввод функции:


```
NEWTONS([#1, #2, #3 ], [a, b, c], [1.8, 3.5, 0.9], 6).
```

Здесь: #1, #2, #3 — метки первого, второго и третьего уравнений. После нажатия клавиши <Enter> на экране увидим функцию NEWTONS, в которой вместо меток находятся уравнения.

4. Получение решения. После щелчка по кнопке  (Approximate) на экране отобразится решение в виде таблицы. За шесть итераций получен следующий ответ: $a = 1.89307$, $b = 3.03147$, $c = 0.761182$. Функция интерполяции имеет вид:

$$y = 1.89307 \cdot 3.03147^x + 0.761182.$$

Метод итераций

Для решения нелинейных уравнений методом итераций в системе Derive имеется функция:

`FIXED_POINT([f1(x, y, ), f2(x, y, ), , fm(x, y, )], [x, y, ], [x0, y0, ], n).`

Как видно из этого выражения, она отличается от функции NEWTONS лишь тем, что вместо слова NEWTONS пишется FIXED_POINT. Технология решения та же, что и в методе Ньютона.

Следует иметь в виду, что для получения решения с помощью методов Ньютона и итераций необходимо, чтобы итерационный процесс сходиллся. Для этого нужно надлежащим образом выбрать начальные приближения и обеспечить условия сходимости итераций. Поэтому при решении задач интерполяции нелинейными функциями могут возникнуть сложности реализации функций NEWTONS и FIXED_POINT.

Определение погрешности интерполяции

Среднеквадратичная погрешность вычисляется по формуле (2.1) по методике, изложенной в *разд. 2.1 данной главы*. Для нашего примера среднеквадратичная погрешность равна 2.35.

2.4. Интерполяция приближенная в узлах

Интерполяция приближенная в узлах (аппроксимация) в системе Derive осуществляется функцией `FIT(A)`, где A — матрица функции $y = f(x)$, имеющая вид:

$$\begin{pmatrix} x & \varphi(x) \\ x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{pmatrix},$$

Здесь:

- x — аргумент функции $f(x)$;
- $\varphi(x)$ — выражение функции интерполяции в аналитическом виде;
- x_i, y_i — узлы интерполяции и значения функции в соответствующих узлах.

Технологию решения задачи аппроксимации рассмотрим на примере.

Пример 2.5

Пусть функция $y = f(x)$ представлена в виде таблицы (табл. 2.5):

Таблица 2.5

	Значения переменных			
x	1	2	3	4
y	2	4	8	12

Необходимо найти функцию аппроксимации $y = a + bx + cx^2$ и найти значения функции при $x = 2.5$ и при $x = 3.5$.

Последовательность решения задачи в системе Derive (рис. 2.12):

1. Введем матрицу A размером 5 x 2. После нажатия клавиши <Enter> или кнопки **ОК** на экране отобразится матрица в виде:

#1:
$$\begin{vmatrix} x & a+b \cdot x+c \cdot x^2 \\ 1 & 2 \\ 2 & 4 \\ 3 & 8 \\ 4 & 12 \end{vmatrix} .$$

2. Введем выражение: `FIT (#1)`. На экране увидим:

#2: `FIT`
$$\begin{vmatrix} x & a+b \cdot x+c \cdot x^2 \\ 1 & 2 \\ 2 & 4 \\ 3 & 8 \\ 4 & 12 \end{vmatrix} .$$

3. Выполним команду `Approximate`. На экране увидим ответ:

#3: $0.5x^2 + 0.9x + 0.5$.

4. Выполним команду Variable Substitution (щелчком мыши по кнопке **Sub**). На экране активизируется окно **Substitute for variables in #3** (рис. 2.13).
5. Введем в поле **New Value** значение x , равное 2.5, после чего щелкнем мышью по кнопке **OK**. На экране появится выражение:
#4: $0.5 \cdot 2.5^2 + 0.9 \cdot 2.5 + 0.5$.
6. Щелкнем мышью по кнопке  (Approximate). На экране увидим ответ:
#5: 5.875;
7. Аналогично вычислим значение функции при $x = 3.5$. Получим ответ:
 $y = 9.775$.

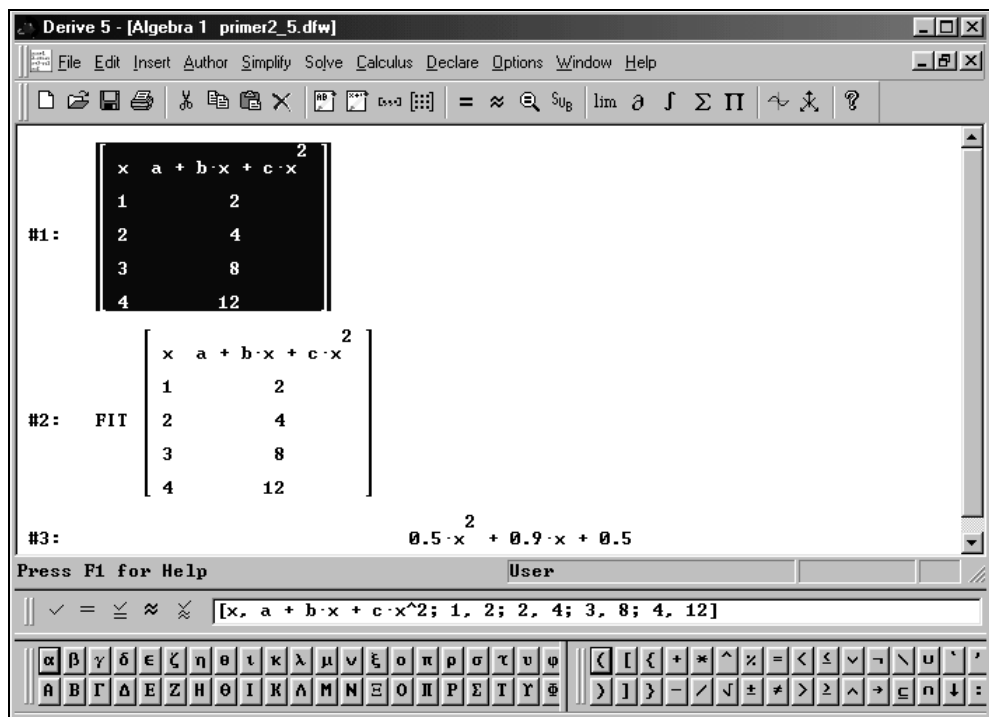


Рис. 2.12

Если число неизвестных функции интерполяции (x) равно числу узлов интерполяции (числу строк таблицы), то функция FIT решает задачу интерполяции точную в узлах. Если число неизвестных меньше числа строк таблицы, то решается задача интерполяции приближенная в узлах (аппроксимация). Задача решается методом наименьших квадратов, при этом узлы интерполяции могут быть неравноотстоящими. Если число строк таблицы функции $y = f(x)$ меньше числа неизвестных, то задача не решается.

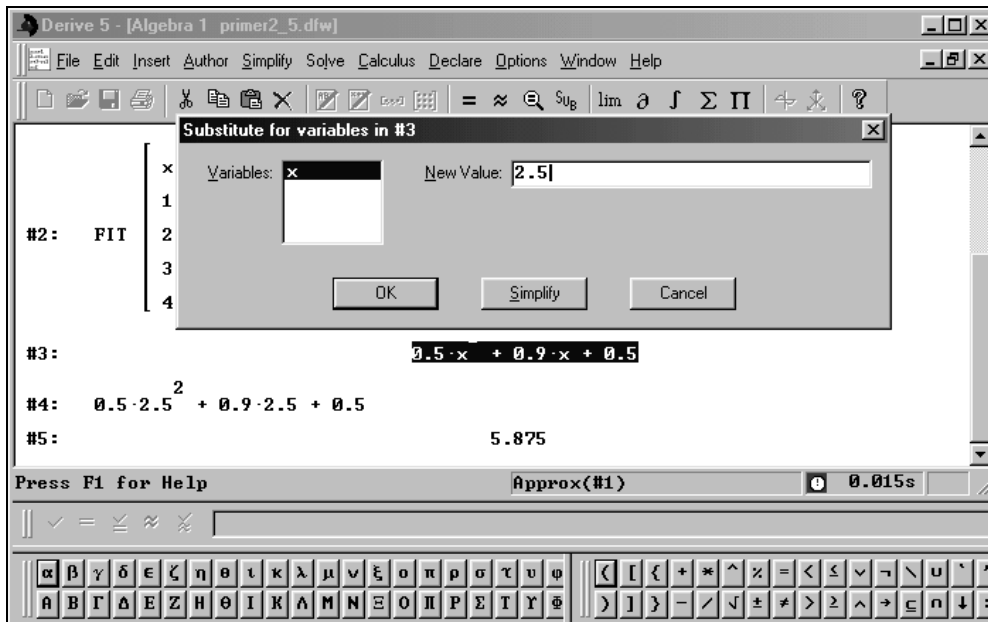


Рис. 2.13

В качестве функции $\varphi(x)$ может быть многочлен степени n , степенная функция, логарифмическая, показательная и др. При этом аргумент и неизвестные не должны входить в выражение для $\varphi(x)$ в виде сомножителей. Например, Derive решает задачу, если функция интерполяции имеет вид:

$$\varphi(x) = a + b^x + c \ln(x) + d e^x + \frac{e}{x} + \frac{f}{x^2}.$$

Но задача не будет решена, если функция будет задана в виде:

$$\varphi(x) = a + b^{cx} \text{ или } y = a + b \ln(cx).$$

Функция FIT позволяет решать задачу интерполяции многих переменных.

Пусть, например, $\varphi(x) = a + b x + c y + d z$. Тогда функция FIT будет иметь вид:

$$\text{FIT} \begin{vmatrix} x & y & z & a + bx + cy + dz \\ 1 & 1.2 & 3.1 & 1.15 \\ 2 & 3.5 & 1.8 & 6.8 \\ \vdots & \vdots & \vdots & \vdots \\ n & 8.4 & 0.2 & 12.3 \end{vmatrix}.$$

Функция интерполяции $\varphi(x)$ при этом может быть линейной, квадратичной или более сложного вида.

2.5. Аппроксимация многими функциями

Derive позволяет решить задачу аппроксимации одновременно семью функциями вида:

$$a + bx, \quad ae^{bx}, \quad ax^b, \quad a + b \ln(x), \quad a + \frac{b}{x}, \quad \frac{a}{b+x}, \quad \frac{ax}{b+cx}.$$

Для этого служит функция $\text{ALL_SEVEN}(A)$, где A — матрица исходных данных.

Технологию решения задачи аппроксимации рассмотрим на примере.

Пример 2.6

Пусть функция задана в виде таблицы (см. табл. 2.5). Необходимо решить многофункциональную задачу аппроксимации, что можно произвести следующими действиями:

1. Вызовем утилиту **Curv_fit**: **File** | **Load** | **Utility File**. На экране активизируется окно **Load Utility**.
2. Щелкнем мышью по **Curv_fit**, а затем по кнопке **Открыть**. На экране увидим: #1: **LOAD(Curv_fit.mth)**.
3. Введем матрицу исходных данных (в нашем примере размер матрицы 4×2). На экране появится матрица с меткой (например, #2).
4. Введем функцию: **ALL_SEVEN**(#2). На экране отобразится функция с матрицей исходных данных.
5. Вызовем команду **Approximate** (кнопка ). На экране увидим решение в виде совокупности семи функций.

В конечном итоге решение на экране монитора будет иметь вид:

#1: **LOAD(Curv_fit.mth)**

$$\#2: \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 8 \\ 4 & 12 \end{bmatrix}$$

$$\#3: \text{ALL_SEVEN} \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 8 \\ 4 & 12 \end{bmatrix}$$

$$\#4: [3.4 \cdot t - 2, 1.15470 \cdot e^{0.606842 t}, 1.87103 \cdot t^{1.30199}, 6.95408 \cdot \ln(t) + 0.974887,$$

$$12.4615 - \frac{11.4461}{t}, \frac{2680}{1267 - 271 \cdot t}, \frac{43.3333 \cdot t}{26 - 3 \cdot t}]$$

2.6. Аппроксимация Паде

Аппроксимация Паде применяется для аппроксимации нелинейных функций $y = f(x)$ дробно-рациональными функциями вида:

$$y = \frac{b_0 + b_1 x + \dots + b_m x^m}{a_0 + a_1 x + \dots + a_n x^n}.$$

Метод является наиболее точным. Его реализация в Derive возможна, если нелинейная функция представлена в аналитическом виде.

Функция Паде имеет вид:

$\text{PADE}(y, x, x_0, m, n)$.

Здесь:

- y — функция, представленная в аналитическом виде;
- x — аргумент функции;
- x_0 — значение аргумента, вблизи которого обеспечивается высокая точность аппроксимации;
- m — степень многочлена числителя;
- n — степень многочлена знаменателя.

Реализация метода возможна только после загрузки файла `approx.mth`.

Технологию аппроксимации Паде в Derive рассмотрим на примере.

Пример 2.7

Пусть функция имеет вид $y = e^{-x} \cos(x)$. Необходимо получить функцию аппроксимации $\varphi(x)$ в виде дробно-рациональной функции вблизи значения аргумента $x = 0$ при степени числителя $m = 2$ и степени знаменателя $n = 3$. Нужно выполнить сравнение значений исходной функции и функции аппроксимации путем их табулирования.

Рассмотрим технологию решения задачи в системе Derive:

1. Введем выражение: `exp(-x)*cos(x)`. На экране отобразится выражение с меткой #1:.
2. Введем функцию: `y = PADE(#1, x, 0, 2, 3)`. На экране увидим функцию с меткой #2:.
3. Щелкнем мышью по кнопке  (Approximate). На экране появится решение с присвоенной ему меткой #3:.
4. Введем функцию табулирования:
`VECTOR([x, #1, #3], x, -0.5, 0.5, 0.1)`.

На экране отобразится функция VECTOR, в которой вместо #1 и #3 — табулируемые функции.

5. Выполним команду Approximate. На экране увидим ответ в виде таблицы:

−0.5	1.44688	1.44303
−0.4	1.37406	1.37336
−0.3	1.28956	1.28947
−0.2	1.19705	1.19704
−0.1	1.09964	1.09964
#5: 0	1	1
0.1	0.900316	0.900316
0.2	0.802410	0.802408
0.3	0.707730	0.707708
0.4	0.617405	0.617302
0.5	0.532280	0.531958

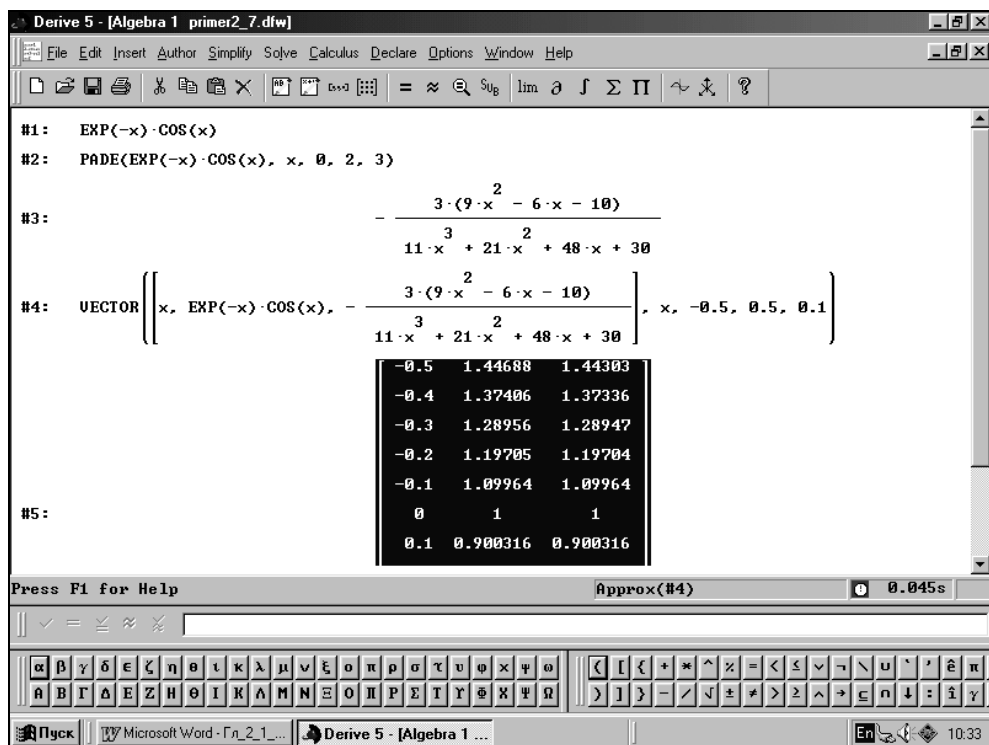


Рис. 2.14

Процедуры решения задачи аппроксимации Паде на экране монитора имеют вид, показанный на рис. 2.14.

Из таблицы видно, что в диапазоне $-0.5 \leq x \leq 0.5$ значения функции интерполяции совпадают со значениями функции $e^{-x} \cos(x)$ с точностью не менее двух значащих цифр после запятой.

2.7. Решение задач интерполяции путем прямых вычислений

Derive не имеет функций для реализации ряда методов интерполяции. К таким относятся методы Ньютона, Гаусса, Стирлинга, Бесселя и др. Для их реализации можно воспользоваться методом прямых вычислений. Сущность этого метода состоит в следующем: вводится расчетная формула и выполняются вычисления с использованием символьных преобразований, которые в Derive выполняются весьма эффективно.

Технологию прямых методов вычислений покажем на примере.

Пример 2.8

Пусть исходные данные представлены в виде таблицы (см. табл. 2.3). Необходимо найти интерполяционный многочлен $(n - 1)$ -ой степени методом Ньютона для случаев интерполирования вперед и назад.

Решение можно найти, воспользовавшись формулами (1.11) и (1.15). Для этого необходимо знать табличные разности. В нашем случае вычисления табличных разностей не требуют обращения к компьютеру или калькулятору ввиду их простоты. Они приведены в табл. 2.6.

Таблица 2.6

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
1	12	-3			
2	9	-2	1	-4	
3	7	-5	-3	7	11
4	2	-1	4		
5	1				

В примере число узлов $n = 5$. Тогда полином будет степени $n - 1 = 4$. На основании (1.11) и (1.15) интерполяционные формулы Ньютона будут иметь вид:

$$\begin{aligned}
 y(x) &= y_0 + \frac{\Delta y_0}{1! h} (x - x_0) + \frac{\Delta^2 y_0}{2! h^2} (x - x_0)(x - x_1) + \\
 &+ \frac{\Delta^3 y_0}{3! h^3} (x - x_0)(x - x_1)(x - x_2) + \frac{\Delta^4 y_0}{4! h^4} (x - x_0)(x - x_1)(x - x_2)(x - x_3); \\
 y(x) &= y_5 + \frac{\Delta y_{n-1}}{1! h} (x - x_5) + \frac{\Delta^2 y_{n-2}}{2! h^2} (x - x_5)(x - x_4) + \\
 &+ \frac{\Delta^3 y_{n-3}}{3! h^3} (x - x_5)(x - x_4)(x - x_3) + \frac{\Delta^4 y_{n-4}}{4! h^4} (x - x_5)(x - x_4)(x - x_3)(x - x_2).
 \end{aligned}$$

Первая формула используется для интерполирования вперед, вторая — для интерполирования назад.

Рассмотрим технологию решения задачи интерполяции по Ньютону с помощью Derive:

1. Введем формулу Ньютона. На экране появится формула с меткой #1.
2. Нажмем кнопку **Sub** (на панели инструментов), после чего выполним подстановку в формулу значений h , y_0 , табличных разностей, узлов интерполирования x_0 , x_1 , x_2 , x_3 . На экране отобразится формула с численными значениями этих переменных;
3. Выполним команду **Simplify | Factor**, после чего произойдет упрощение выражения. На экране отобразится ответ:

$$y = 0.0416666 (11x^4 - 126x^3 + 493x^2 - 834x + 744),$$

что совпадает с решением по методу, изложенному в *разд. 2.2 данной главы*.

Решение по второй формуле Ньютона совпадает с полученным ранее.

Реализовать метод Ньютона на Derive можно без вычисления табличных разностей. Для этого необходимо использовать исходную интерполяционную формулу Ньютона:

$$\begin{aligned}
 y(x) &= c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + c_3(x - x_0)(x - x_1)(x - x_2) + \\
 &+ c_4(x - x_0)(x - x_1)(x - x_2)(x - x_3)
 \end{aligned}$$

и функцию FIT.

Рассмотрим технологию решения задачи на Derive:

1. Введем исходную формулу Ньютона, в результате чего на экране отобразится формула с меткой #1:.
2. Нажмем кнопку **Sub**, после чего произойдет подстановка в формулу значений узлов интерполирования $x_0 = 1$, $x_1 = 2$, $x_2 = 3$, $x_3 = 4$. На экране появится формула с численными значениями узлов интерполирования и присвоенной ей меткой #2:.

3. Введем матрицу:

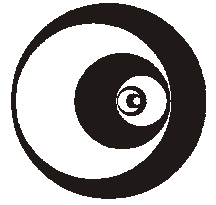
$$\#3: \begin{vmatrix} x & \#2 \\ 1 & 12 \\ 2 & 9 \\ 3 & 7 \\ 4 & 2 \\ 5 & 1 \end{vmatrix}$$

На экране увидим матрицу с меткой #3: и формулой, соответствующей метке #2:.

4. Введем функцию: `FIT(#3)`. На экране появится матрица с функцией FIT.

5. Выполним команду главного меню **Simplify | Factor**. На экране увидим ответ, совпадающий с предыдущим.

Аналогично решаются задачи интерполяции методов Ньютона в случае не равноотстоящих узлов, а также методами Гаусса, Стирлинга, Бесселя и др.



Глава 3

Компьютерные технологии решения задач интерполяции в системе Mathematica

3.1. Интерполяция точная в узлах. Универсальный метод

Универсальный метод требует решения систем алгебраических уравнений. Система символьной математики Mathematica имеет богатые средства решения уравнений с получением ответов в желаемом для пользователя виде. Рассмотрим основные из этих средств, необходимые для решения задач интерполяции.

3.1.1. Функция Solve

Функция записывается в следующем виде:

$\text{Solve}[\{f_1(x, y,) = = b_1, f_2(x, y,) = = b_2, \dots, f_n(x, y,) = = b_n\}, \{x, y, \}]$

Здесь:

- $f_i(x, y,) = = b_i$ — i -ое уравнение системы из n уравнений;
- $\{x, y, \}$ — неизвестные.

Mathematica выдает решение в символьном и числовом виде в зависимости от того, в каком виде заданы коэффициенты уравнений: цифровом, символьном или комбинированном.

Если заданы коэффициенты в виде чисел, то решение будет точным, представленным в виде дроби с числителем и знаменателем. При необходимости

получения решения в естественной форме представления чисел достаточно вслед за решением ввести оператор: `N[%]` и нажать одновременно клавиши `<Shift>+<Enter>`.

Пример 3.1

Пусть необходимо решить следующую систему уравнений:

$$2x - 3y + 5z = 1$$

$$-x + y + 2z = 3$$

$$3x + 4y - z = -1$$

Решение:

Ввод в строке пользователя:

```
In[1]: = Solve[{2 * x - 3 * y + 5 * z == 1, - x + y + 2 * z == 3,
               3 * x + 4 * y - z == -1}, {x, y, z}]
```

Ответ: Out[1] = $\left\{ \left\{ x \frac{53}{68}, y \frac{37}{68}, z \frac{57}{68} \right\} \right\}$.

Ввод в строке пользователя:

```
In[2]: = N[%]
```

Ответ: Out[2] = $\left\{ \left\{ x \ 0.779412, y \ 0.544118, z \ 0.838235 \right\} \right\}$.

Представление уравнений в удобной для пользователя форме.

Mathematica допускает следующую форму представления системы уравнений функции Solve:

$\{\{2, -3, 5\}, \{-1, 1, 2\}, \{3, 4, -1\}\} \cdot \{x, y, z\} = \{1, 3, -1\}$

`Solve[%, {x, y, z}]`, т. е. в матричном виде $A \cdot X = B$, где:

- A — матрица коэффициентов;
- X — вектор неизвестных;
- B — вектор правых частей уравнений.

После нажатия клавиш `<Shift>+<Enter>` на экране появится система уравнений и ее решение.

На рис. 3.1 приведено решение системы уравнений двумя способами.

3.1.2. Функция NSolve

Функция записывается в том же виде, что и Solve. Разница лишь в том, что решение здесь выдается в численном виде, с представлением чисел в естественной форме.

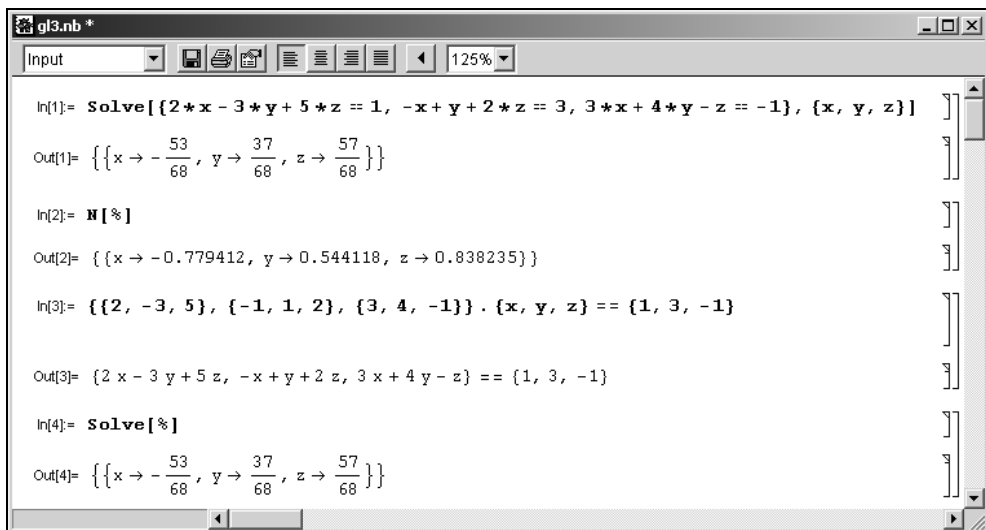


Рис. 3.1.

Пример 3.2

Пусть необходимо решить следующую систему уравнений:

$$x + 1.5y - 2z = 1$$

$$3x - y + 7z = -2$$

$$-2x + 6y + z = 5$$

Решение:

```
NSolve[{x + 1.5 * y - 2 * z == 1, 3 * x - y + 7 * z == -2,
-2 * x + 6 * y + z == 5}, {x, y, z}]
```

После нажатия клавиш <Shift>+<Enter> получится ответ:

```
{{x -> -0.263682, y -> 0.756219, z -> -0.0646766}}
```

Удобной технологией решения системы уравнений является следующая:

1. Ввод уравнений в виде:

```
{x + 1.5 * y - 2 * z == 1, 3 * x - y + 7 * z == -2,
-2 * x + 6 * y + z == 5}
```

2. Ввод функции:

```
Solve[%, {x, y, z}]
```

После нажатия клавиш <Shift>+<Enter> на экране отобразится система уравнений и ответ.

При решении рассмотренными методами системы уравнений примера 3.2 экран имеет вид, показанный на рис. 3.2.

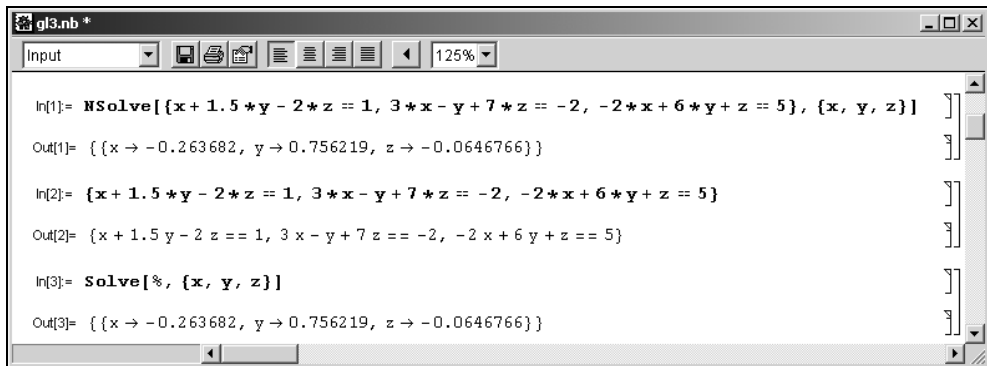


Рис. 3.2

Систему уравнений можно также вводить поочередно (уравнение за уравнением), нажимая после каждого введенного уравнения клавишу <Enter>, что особенно удобно при большом объеме ввода.

```
{x + 1.5 * y - 2 * z == 1,
3 * x - y + 7 * z == -2,
-2 * x + 6 * y + z == 5}.
```

Затем производится ввод функции: `Solve[%, {x, y, z}]`.

После нажатия клавиш <Shift>+<Enter> на экране отобразится ответ в виде системы уравнений и ее решения.

3.1.3. Функция FindRoot

Функция предназначена для численного решения систем нелинейных уравнений методом Ньютона. Она записывается в следующем виде:

```
FindRoot[{f1(x, y, ) == b1, f2(x, y, ) == b2, ...,
fn(x, y, ) == bn}, {x, x0}, {y, y0}, {z, z0},
```

где $x_0, y_0, \dots, z_0$ — числа, представляющие собой начальные приближения.

Пример 3.3

Пусть система уравнений имеет вид:

$$\sin[x] - x + y = 0$$

$$\cos[x] - x - y = 0$$

и пусть имеются начальные приближения: $x_0 = 1, y_0 = 0$.

Тогда функция FindRoot будет иметь вид:

```
FindRoot[{Sin[x] == x - y, Cos[x] == x + y}, {x, 1}, {y, 0}]
```

Ответ будет получен в виде:

$x \rightarrow 0.704812$, $y \rightarrow 0.0569213$ (рис. 3.3).

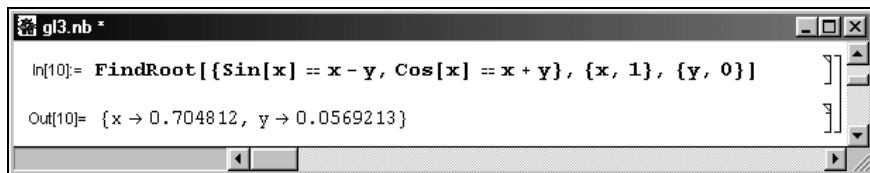


Рис. 3.3

3.1.4. Решение задачи интерполяции универсальным методом

Решение задач интерполяции универсальным методом в среде Mathematica рассмотрим на примерах.

Пример 3.4

Пусть функция $y = f(x)$ задана в виде таблицы (табл. 3.1).

Таблица 3.1

	Значения переменных			
x	1	2	3	4
y	6.2	4.1	1.9	0.6

Необходимо решить задачу интерполяции точную в узлах, если функция $y = f(x)$ является полиномом.

Так как число узлов $n = 4$, то полином должен иметь степень, равную $n - 1$, т. е. $y = a_0 + a_1x + a_2x^2 + a_3x^3$.

Составим систему уравнений:

$$a_0 + a_1 \cdot 1 + a_2 \cdot 1^2 + a_3 \cdot 1^3 = 6.2$$

$$a_0 + a_1 \cdot 2 + a_2 \cdot 2^2 + a_3 \cdot 2^3 = 4.1$$

$$a_0 + a_1 \cdot 3 + a_2 \cdot 3^2 + a_3 \cdot 3^3 = 1.9$$

$$a_0 + a_1 \cdot 4 + a_2 \cdot 4^2 + a_3 \cdot 4^3 = 0.6$$

Порядок решения задачи в среде Mathematica:

1. Ввод системы уравнений:

$$\{a_0 + a_1 + a_2 + a_3 == 6.2,$$

$$a_0 + a_1 * 2 + a_2 * 2^2 + a_3 * 2^3 == 4.1,$$

$$a_0 + a_1 * 3 + a_2 * 3^2 + a_3 * 3^3 == 1.9,$$

$$a_0 + a_1 * 4 + a_2 * 4^2 + a_3 * 4^3 == 0.6\}$$

2. Ввод функции:

```
Solve[%, {a0, a1, a2, a3}]
```

3. Нажатие клавиш <Shift>+<Enter>, в результате чего на экране появится решение (рис. 3.4).

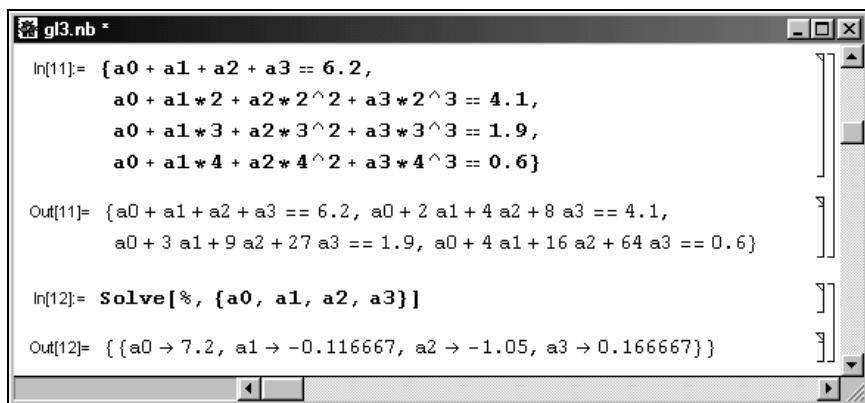


Рис. 3.4

В результате полученного решения интерполяционная формула будет иметь вид:

$$y = 7.2 - 0.116667x - 1.05x^2 + 0.166667x^3.$$

Рассмотрим порядок решение в матричной форме $A \cdot X = B$:

1. Ввод выражения:

```
{{1, 1, 1, 1}, {1, 2, 2^2, 2^3}, {1, 3, 3^2, 3^3}, {1, 4, 4^2, 4^3}} .  
{a0, a1, a2, a3} == {6.2, 4.1, 1.9, 0.6}.
```

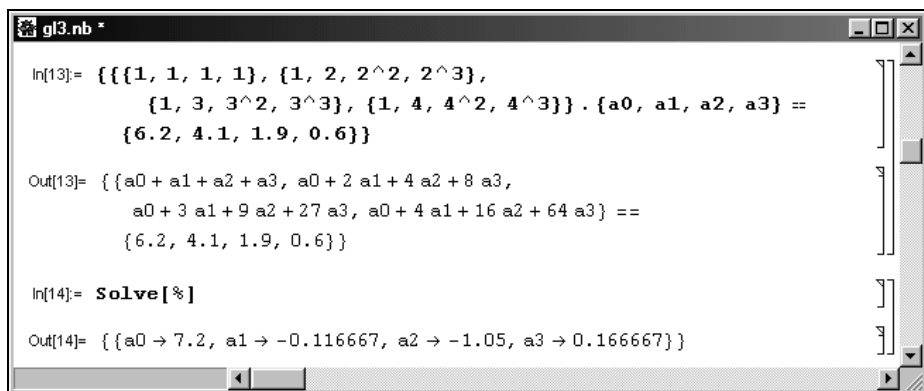


Рис. 3.5

2. Ввод функции:

```
Solve[%].
```

3. Нажатие клавиш <Shift>+<Enter>. На экране появится ответ (рис. 3.5).

Проверка правильности решения задачи интерполяции

Для проверки правильности решения задачи выполним табулирование полученной формулы и результаты табулирования сравним с исходными данными. Функция табуляции в Mathematica имеет вид:

```
Table[f(x), {x, xn, xk, h}]
```

Здесь:

- $f(x)$ — табулируемая функция;
- x — аргумент функции $f(x)$;
- x_n, x_k — начальное и конечное значения аргумента;
- h — шаг таблицы (если $h = 1$, то его можно опустить).

Функция Table дает ответ в виде вектора-строки.

Другая функция табулирования записывается в виде:

```
Do[Print[f(x)], {x, xn, xk, h }].
```

Эта функция выдает решение в виде вектора-столбца.

Выполним табулирование функции, полученной в результате решения предыдущего примера:

1. Введем функцию:

```
Table[7.2 - 0.116667 * x - 1.05 * x^2 + 0.166667 * x^3,  
{x, 1, 4}].
```

2. Нажмем клавиши <Shift>+<Enter>.

3. На экране появится ответ: {6.2, 4.1, 1.9, 0.6}.

Сравнивая ответ с исходными данными, убеждаемся, что функция интерполяции получена верно.

Теперь введем функцию:

```
Do[Print[7.2 - 0.116667 * x - 1.05 * x^2 + 0.166667 * x^3],  
{x, 1, 4}].
```

После нажатия клавиш <Shift>+<Enter> получим ответ в виде:

6.2

4.1

1.90001

0.60002

Mathematica позволяет получить решение в виде матрицы, элементами которой являются x и $f(x)$. Для этого служит функция:

`Table[{x, f(x)}, {x, xn, xk, h }].`

Для нашего примера функция имеет вид:

```
Table[{x, 7.2 - 0.116667 * x - 1.05 * x^2 + 0.166667 * x^3},
      {x, 1, 4, 1}].
```

После нажатия клавиш <Shift>+<Enter> получим следующий ответ:

```
{{1, 6.2}, {2, 4.1}, {3, 1.9}, {4, 0.6}}.
```

Для получения решения в столбик служит функция `MatrixForm[%]`. После ее исполнения получим ответ в виде матрицы:

```
1      6.2
2      4.1
3      1.90001
4      0.60002
```

Процедуры табуляции показаны на рис. 3.6.

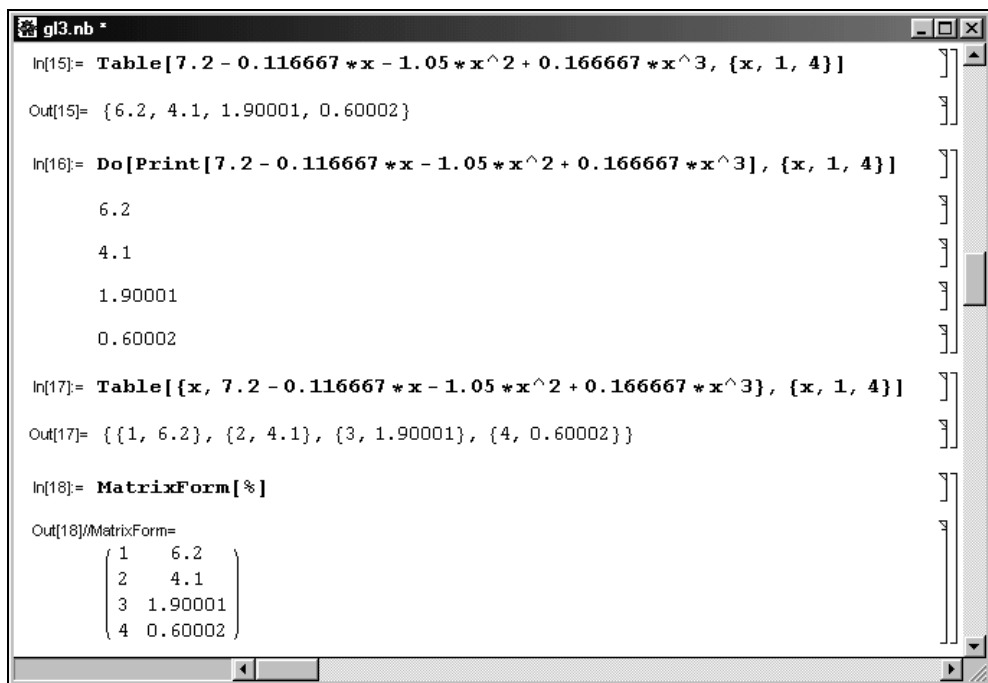


Рис. 3.6

3.2. Интерполяция точная в узлах.

Функция InterpolatingPolynomial

Интерполяция полиномами хорошо и достаточно просто реализуется в среде Mathematica функцией InterpolatingPolynomial. Функция записывается в следующем виде:

InterpolatingPolynomial[z, x],

где:

□ z — матрица исходных данных;

□ x — аргумент функции z .

Пример 3.5

Пусть функция z задана в виде таблицы (табл. 3.2).

Таблица 3.2

	Значения переменных				
x	1	2	3	4	5
y	1	8	27	64	125

Тогда решение задачи сводится к следующим действиям:

1. Ввод матрицы: $z = \{\{1, 1\}, \{2, 8\}, \{3, 27\}, \{4, 64\}, \{5, 125\}\}$.
2. Ввод функции: InterpolatingPolynomial[z, x].
3. Нажатие клавиш <Shift>+<Enter>.

Ответом будет следующее выражение:

$1 + (-1 + x)(7 + (-2 + x)(3 + x))$.

От скобочных форм легко избавиться, применяя функции:

- Simplify;
- Expand;
- Factor.

В нашем случае:

ввод: Simplify[%]

На экране будет: x^3 .

При этом решение на экране будет иметь вид, показанный на рис. 3.7.

Функция InterpolatingPolynomial решает задачи интерполяции также в случае неравноотстоящих узлов.

Функция z может быть задана также в виде набора функций $f_1(x), f_2(x), \dots$. Тогда решение будет выдаваться в виде многочлена с сохранением вида функций:
 $f_1(x), f_2(x), \dots$.

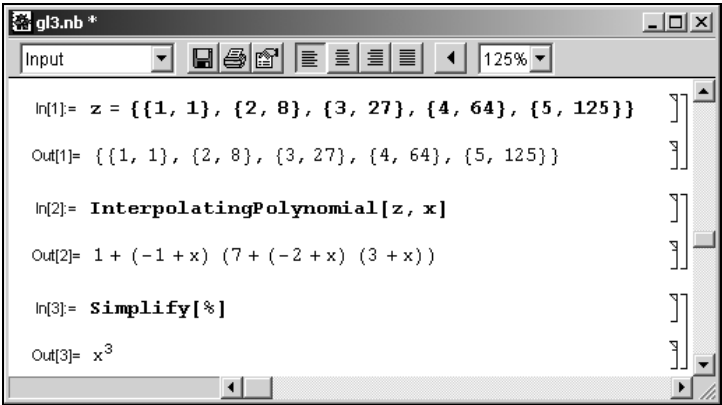


Рис. 3.7

Пример 3.6

Пусть функция z задана в виде таблицы (табл. 3.3).

Таблица 3.3

	Значения переменных			
x	1	2	3	4
y	1	$\text{Sin}[x]$	$1 / x$	e^{-x}
$y(x)$	1	0.909297	0.333333	0.0183156

В последней строке таблицы приведены значения функции при заданных значениях аргумента x .
Процедуры решения задачи с помощью функции InterpolatePolynomial будут иметь вид (при $h = \text{const}$):
 $\text{In}[1] := z = \{1, \text{Sin}[x], 1 / x, \text{Exp}[-x]\},$
 $\text{InterpolatingPolynomial}[z, x],$
После нажатия клавиш <Shift>+<Enter> получится ответ:
 $\text{Out}[1] = \{1, \text{Sin}[x], \frac{1}{x}, e^{-x}\}$

$$\begin{aligned} \text{Out}[2] = & 1 + (-1 + x)(-1 + \sin[x]) + (-2 + x)\left(\frac{1}{2}\left(1 + \frac{1}{x} - 2 \sin[x]\right) + \right. \\ & \left. + \frac{1}{3}(-3 + x)\left(\frac{1}{2}\left(e^{-x} - \frac{2}{x} + \sin[x]\right) + \frac{1}{2}\left(-1 - \frac{1}{x} + 2 \sin[x]\right)\right)\right) \end{aligned}$$

После ввода функций:

```
In[3]: = Table[Out[2], {x, 1, 4}]
```

```
In[4]: = N[%]
```

на экране отобразится ответ:

Out[4] = {1, 0.909297, 0.333333, 0.0183156}, что совпадает с исходными данными.

Функция Collect

Функция Collect представляет решение в бесскобочной форме.

Например, $w = x(1 + x) - x^3 + (x - 1)(x + 12)$.

Тогда Collect[w, x] представит ответ в виде многочлена: $12 + 12x + 2x^2 - x^3$.

Эту функцию целесообразно применять при решении задач интерполяции совместно с функцией InterpolatingPolynomial, дающей ответ обычно в сложных скобочных формах.

Для примера 3.3 функция запишется в виде:

```
Collect[InterpolatingPolynomial[z, x], x]
```

Последует ответ: x^3 .

Такой же ответ будет получен при реализации следующих функций:

- ☐ Factor[InterpolatingPolynomial[z, x]];
- ☐ Expand[InterpolatingPolynomial[z, x]];
- ☐ Simplify[InterpolatingPolynomial[z, x]].

3.3. Интерполяция нелинейными функциями

В ряде случаев интерполяция нелинейными функциями сводится к решению линейных уравнений. Этот случай имеет место тогда, когда удастся представить функцию в новой системе координат, в которой функция становится линейной. В главе 1 показано, что такими являются нелинейные функции: степенная, логарифмическая, дробно-линейная. В системе Mathematica задачи аппроксимации нелинейными функциями, приводящимися к линейным путем преобразования координат, решаются легко. Это обеспечивается возможностью операций с функциями, представленными в виде векторов и матриц.

Пусть функция задана в виде таблицы (табл. 3.4).

Таблица 3.4

	Значения переменных			
x	1	2	3	4
y	2	4	8	16

Необходимо представить функцию в логарифмическом масштабе по осям x и y .

Порядок действий в системе Mathematica:

1. Ввод матрицы: $\{\{1, 2\}, \{2, 4\}, \{3, 8\}, \{4, 16\}\}$.
2. Ввод команды: `N[Log[%]]`.
3. Нажатие клавиш `<Shift>+<Enter>`.
4. На экране ответ:
 $\text{Out}[1] = \{\{1, 2\}, \{2, 4\}, \{3, 8\}, \{4, 16\}\}$
 $\text{Out}[2] = \{\{0, 0.693147\}, \{0.693147, 1.38629\}, \{1.09861, 2.07944\},$
 $\{1.38629, 2.77259\}\}$

Пример 3.7

Пусть функция $y = f(x)$ задана в виде таблицы (табл. 3.5).

Таблица 3.5

	Значения переменных			
x	1	2	3	4
y	5	8	10	18

И пусть функция интерполяции имеет вид дробно-линейной функции

$$y = \frac{x}{a + bx}.$$

Решим задачу интерполяции путем решения системы линейных уравнений.

Запишем дробно-линейную функцию в виде $Y = a + bx$, где $Y = \frac{x}{y}$ и пред-

ставим ее в табличной форме, воспользовавшись векторными преобразованиями системы Mathematica.

В результате преобразований исходная таблица примет вид, представленный в табл. 3.6.

Таблица 3.6

	Значения переменных			
x	1	2	3	4
y	0.2	0.25	0.3	0.222222

Составим систему линейных уравнений. Число неизвестных дробно-линейной функции равно двум. Выберем следующие две координаты $\{1, 0.2\}$ и $\{4, 0.222222\}$. Тогда система уравнений будет выглядеть так:

$$a + b \cdot 1 = 0.2$$

$$a + b \cdot 4 = 0.222222$$

Решим эту систему линейных уравнений с помощью функции `Solve`. Для нашего случая процедуры решения будут заключаться в следующем:

1. Ввод: `{a + b == 0.2,`
`a + 4 * b == 0.222222}`

2. Ввод: `Solve[%, {a, b}]`

На экране отобразится ответ в следующем виде:

$$\text{Out}[1] = \{a + b == 0.2, a + 4b == 0.222222\}$$

$$\text{Out}[2] = \{\{a \ 0.192593, b \ 0.00740733\}\}$$

Тогда функция интерполяции будет выглядеть так:

$$y = \frac{x}{0.192593 + 0.00740733 x}$$

Процедуры преобразования на экране монитора имеют вид, показанный на рис. 3.8.

Решим теперь сформулированную задачу интерполяции методом решения нелинейных уравнений. В нашем случае система нелинейных уравнений будет иметь вид:

$$\frac{1}{a+b} = 5$$

$$\frac{4}{a+4b} = 18$$

Для ее решения в системе Mathematica воспользуемся функцией `FindRoot`.

Для нашего примера функция будет выглядеть следующим образом:

`FindRoot[{1 / (a + b) == 5, 4 / (a + 4b) == 18}, {a, 0.5}, {b, 0.1}]`

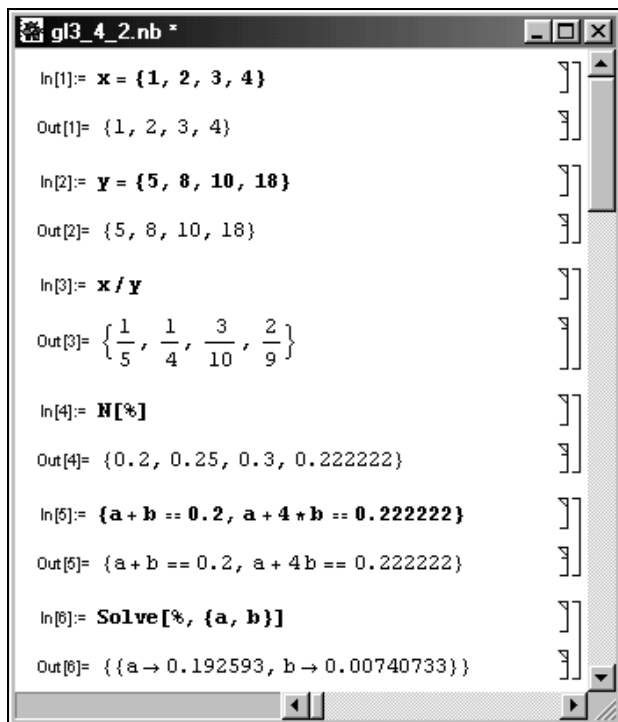


Рис. 3.8

После нажатия клавиш <Shift>+<Enter> на экране увидим решение:

$\text{Out}[1] = \{a \rightarrow 0.192593, b \rightarrow 0.00740741\},$

что практически совпадает с решением, полученным ранее.

Проверим правильность решения путем табулирования формулы интерполяции. Функция Table для нашего случая имеет вид:

$\text{Table}\{x, x / (0.192593 + 0.00740733x)\}, \{x, 1, 4, 1\}$

После реализации этой функции на ПК получим следующий ответ (рис. 3.9):

$\{\{1, 4.99999\}, \{2, 9.64285\}, \{3, 13.9655\}, \{4, 18\}\}$

Сравнение полученных результатов с исходными данными показывает, что при значениях аргументов $x = 2$ и $x = 3$ значения функций существенно различны.

Это является признаком того, что дробно-линейная функция $y = \frac{x}{a + bx}$ плохо воспроизводит функцию, заданную таблицей. Необходимо подобрать иную функцию.

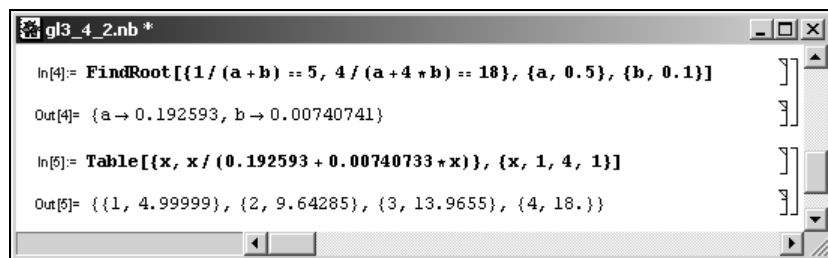


Рис. 3.9

3.4. Функция Interpolation [data]

Эта функция позволяет решать задачу интерполяции над данными (data) в диапазоне аргументов, заданных этими данными. При этом функция аппроксимации пользователю неизвестна. Данные вводятся в виде матрицы функции $y = f(x)$. При вводе этой функции Mathematica выдает не функцию интерполяции, а новую функцию `InterpolatingFunction[{{xn, xk}}, < >]` (где $x_n, \dots, x_k$ — диапазон аргументов функции интерполяции). Если теперь ввести значение аргумента из диапазона $x_n, \dots, x_k$, то откликом будет значение функции при заданном значении аргумента.

Пример 3.8

Пусть функция $y = f(x)$ задана в виде таблицы (табл. 3.7).

Таблица 3.7

	Значения переменных				
x	2	3	8	12	20
y	1	2.5	4.6	3.2	1.6

Необходимо определить значение функции при $x = 5.8$ и при $x = 18.5$.

Возможны следующие способы решения задачи.

□ Способ 1.

Ввод функции:

```
y = Interpolation[{{2, 1}, {3, 2.5}, {8, 4.6}, {12, 3.2}, {20, 1.6}}]
```

После ввода на экране появится отклик в виде функции:

```
InterpolatingFunction[{{2, 20}}, < > ]
```

Для определения значения функции вводится `y[x]`. На экране увидим ответ в виде числа в естественной форме.

В нашем случае поочередно вводятся выражения: $y[5.8]$, $y[18.5]$

На экране появится ответ: 4.56372, 1.18763

□ Способ 2.

Ввод данных и функции:

```
{ {2, 1}, {3, 2.5}, {8, 4.6}, {12, 3.2}, {20, 1.6} }
```

```
y = Interpolation[%]
```

На экране увидим отклик:

```
InterpolatingFunction[{ {2, 20} }, < > ]
```

Ввод: $y[5.8]$, $y[18.5]$.

Откликом является числовой ответ 4.56372, 1.18763.

□ Способ 3.

Ввод данных и функции в виде:

```
y = { {2, 1}, {3, 2.5}, {8, 4.6}, {12, 3.2}, {20, 1.6} }
```

```
w = Interpolation[y]
```

На экране появится отклик:

```
InterpolatingFunction [ { {2, 20} }, < > ].
```

Ввод: $w[5.8]$, $w[18.5]$.

На экране появится ответ в виде чисел в естественной форме: 4.56372, 1.18763.

3.5. Интерполяция приближенная в узлах

Интерполяция приближенная в узлах (аппроксимация) осуществляется по критерию минимума среднеквадратической ошибки (метод наименьших квадратов). Реализуется системой Mathematica с помощью функции Fit. Функция Fit имеет вид:

$\text{Fit}[\{\{M\}\}, \{X\}, x]$.

Здесь:

- M — матрица исходных данных;
- X — перечень базисных переменных;
- x — аргумент функции.

Рассмотрим решение задачи аппроксимации на примерах.

Пример 3.9

Пусть функция $y(x)$ задана в виде таблицы (табл. 3.8).

Таблица 3.8

	Значения переменных				
x	1	3	4	7	10
y	3.5	6.7	4.2	2.8	1.2

Необходимо найти функцию интерполяции в базисе $a, x, x^2, \frac{x}{1+x}, e^x$. В данном примере функция Fit будет иметь вид:

Fit[{{1, 3.5}, {3, 6.7}, {4, 4.2}, {7, 2.8}, {10, 1.2}}, {a, x, x^2, x / (1+x), Exp[x]}, x]

После нажатия клавиш <Shift>+<Enter> на экране увидим ответ (рис. 3.10):

$$-33.1913 - 0.00090638E^x - 16.2137x + 1.22525x^2 + \frac{103.364x}{1+x}.$$

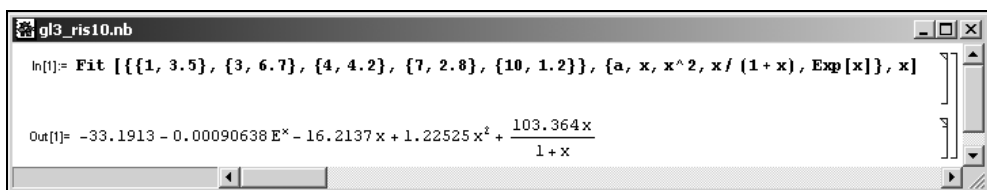


Рис. 3.10

Пример 3.10

Пусть надо найти функцию интерполяции, определяющую зависимость высоты сосны от ее возраста по данным, приведенным в *главе 2* (см. табл. 2.2). Выполним полиномиальную интерполяцию с базисом a, x, x^2, x^3 .

Решение задачи с помощью функции Fit в системе Mathematica удобно свести к следующим процедурам:

□ — ввод матрицы исходных данных:

```
M = {{20, 18}, {30, 22}, {40, 22.5}, {50, 28.5}, {60, 30.5},
      {70, 32}, {80, 33}, {90, 34}, {100, 35}, {110, 35.5}, {120, 36},
      {130, 36.3}};
```

□ — ввод вектора базисных переменных:

```
X = {a, x, x ^ 2, x ^ 3};
```

□ — ввод функции Fit:

```
Fit[M, X, x].
```

После нажатия клавиш <Shift>+<Enter> на экране отобразится ответ:

$$8.26519 + 0.536353x - 0.0033658x^2 + 6.85056 \cdot 10^{-6}x^3.$$

Решение задачи интерполяции на экране монитора имеет вид, показанный на рис. 3.11.

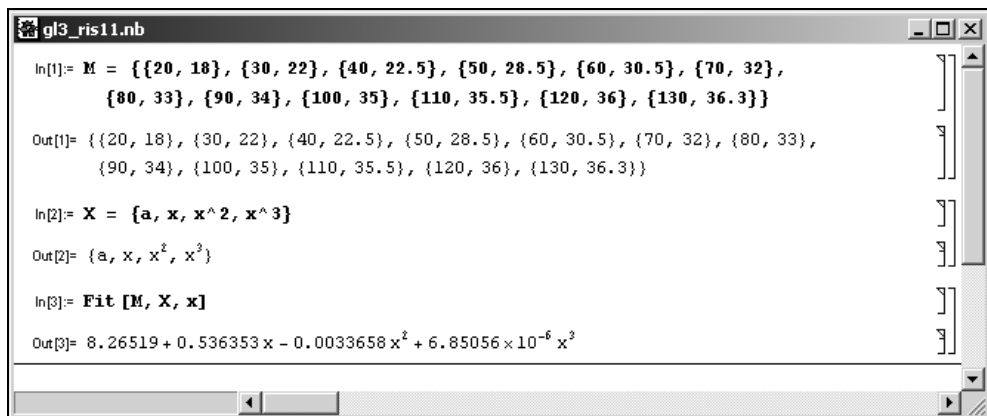


Рис. 3.11

В базисе a, x, x^2 функция интерполяции имеет вид:

$$\varphi(x) = 10.0635 + 0.435307x - 0.00182443x^2,$$

что близко к функции $h = 9.75 + 0.45L - 0.0019L^2$, полученной методом точной интерполяции (см. главу 2).

С целью сравнения результатов сглаживания протабулируем полученную функцию интерполяции и сравним с исходными данными:

```
Table[{x, 8.26519 + 0.536353 * x - 0.0033658 * x^2 +
+ 6.85056 * 10^-6 * x^3}, {x, 20, 130, 10}].
```

В результате табулирования получится следующий ответ (приводится с округлением результатов):

{17.7, 21.5, 24.77, 27.5, 29.8, 31.67, 33.14, 34.27, 35.1, 35.66, 36, 36.16}.

Из сравнения результатов интерполяции видно, что полином третьей степени хорошо описывает зависимость высоты сосны от ее возраста.

Определим погрешность интерполяции. Абсолютную погрешность вычислим по формуле:

$$\varepsilon = \frac{\sqrt{\sum_{i=1}^n \Delta_i^2}}{n}.$$

Здесь:

$$\square \Delta_i = y(x_i) - (x_i);$$

□ $i = 1, 2, \dots, n$ (разность значений исходной функции и функции интерполяции);

□ n — число узлов функции.

Максимальная относительная погрешность вычисляется по формуле:

$$\delta = \frac{\varepsilon}{y_{\min}} 100 \%, \text{ где } y_{\min} — \text{минимальное значение функции } y = f(x).$$

Для вычисления абсолютной погрешности необходимо:

1. Образовать два вектора функций $y(x)$ и (x) .
2. Выполнить вычитание векторов.
3. Выполнить операцию возведения вектора в квадрат.
4. Найти сумму элементов вектора Δ^2 .
5. Извлечь квадратный корень из суммы элементов вектора Δ^2 и разделить на n (в нашем примере равное 12).

Покажем процедуры получения и в среде Mathematica:

```
In[1]: = Table[{8.26519 + 0.536353 * x - 0.0033658 * x^2 +
+ 6.85056 * 10^-6 * x^3}, {x, 20, 130, 10}]
```

```
Out[1] = {{17.7007}, {21.5115}, {24.7725}, {27.5247}, {29.8092}, {31.6672},
{33.1398}, {34.268}, {35.093}, {35.6559}, {35.9976}, {36.1597}}
```

```
In[2]: = {18, 22, 22.5, 28.5, 30.5, 32, 33, 34, 35, 35.5, 36, 36.3}
- Out[1]
```

```
Out[2] = {{0.299267}, {0.488478}, {-2.27246}, {0.975347}, {0.69798}, {0.332789},
{-0.139782}, {-0.268021}, {-0.0930292}, {-0.155911}, {0.0022311},
{0.140293}}
```

```
In[3]: = Out[2]^2
```

```
Out[3] = {{0.0895599}, {0.238608}, {5.1641}, {0.951288}, {0.477189}, {0.11074},
{0.019543}, {0.0718445}, {0.00865831}, {0.0243158}, {4.85021*10^-6},
{0.0196728}}
```

Далее необходимо образовать функцию:

Plus[$x_1, x_2, \dots$], где $x_1, x_2, \dots$ — элементы вектора Out[3].

Для этого необходимо установить курсор в конце вектора Out[3] и нажать клавишу <Enter>. На экране появится вектор {0.089559, ..., 0.0196727}, не имеющий номера I_n . Вектор необходимо отредактировать (убрать фигурные скобки, написать Plus и поставить квадратные скобки). На экране появится выражение:

```
Plus[0.089561, ..., 0.0196821].
```

После нажатия клавиш <Shift>+<Enter> на экране увидим ответ:

Out[4] = 7.17553.

Это и есть искомая сумма квадратов разностей.

После ввода выражения: `Sqrt[Out[4]] / 12` на экране появится отклик:

Out[5] = 0.223226. Это и есть абсолютная среднеквадратическая ошибка.

Далее вычисляется максимальная относительная погрешность:

Out[5] / 18 * 100

Out[6] = 1.24015.

Таким образом, максимальная относительная погрешность функции интерполяции составляет 1.24%.

3.6. Аппроксимация Паде

Для реализации функции необходимо обратиться к пакету расширения <<calculus\pade.m.

Функция Паде имеет вид:

`Pade[f(x), {x, r, n1, n2}]`.

Здесь:

- $f(x)$ — функция, заданная в аналитическом виде;
- x — аргумент функции $f(x)$;
- r — точка, вблизи которой справедлива функция аппроксимации;
- n_1 — степень многочлена числителя;
- n_2 — степень многочлена знаменателя.

Пример 3.11

Пусть необходимо представить функцию $y = \text{Sin}[x]$ в виде, реализуемом аппроксимацией Паде. Процедура аппроксимации в среде Mathematica имеет вид:

```
<<calculus\pade.m
```

```
Pade[Sin[x], {x, 0, 3, 4}]
```

Ответ:

$$\frac{x - \frac{31x^3}{294}}{1 + \frac{3x^2}{49} + \frac{11x^4}{5880}}.$$

Решение приведено на рис. 3.12.

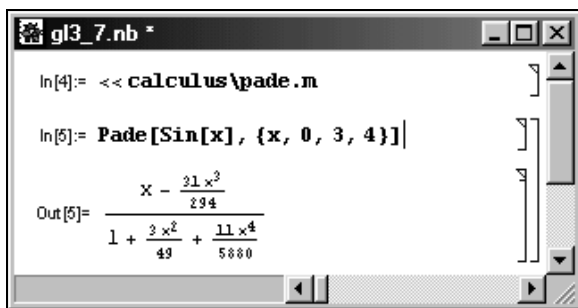


Рис. 3.12

3.7. Компьютерные технологии интерполяции и графическое представление функций

Графическое представление функций при решении задач интерполяции полезно в следующих случаях:

- подбор вида функции интерполяции;
- визуальная проверка правильности решения задачи интерполяции;
- документирование результатов решения задачи.

Система Mathematica является лидером по своим графическим возможностям в сравнении с другими универсальными средствами символьной математики. Большое число функций, позволяющих строить графики, и их модификации дают возможность воспроизводить графические образы любых функций на плоскости и в пространстве трехмерной графики.

При решении задач интерполяции полезными являются следующие графические функции:

- $\text{Plot}\{f_1(x), f_2(x), \dots, \{x, x_{\min}, x_{\max}\}\}$ — строит на одном графике функции $f_1(x), f_2(x), \dots$ в диапазоне изменения аргумента от $x_{\min}$ до $x_{\max}$. Масштабы по осям устанавливаются автоматически;
- $\text{Plot}\{f_1(x), f_2(x), \dots, \{x, x_{\min}, x_{\max}\}, \text{PlotRange} \{Y_1, Y_2\}\}$.
Здесь опция $\text{PlotRange} \{Y_1, Y_2\}$ устанавливает масштаб по оси y ;
- $\text{Plot}[f(x), \{x, x_{\min}, x_{\max}\}, \text{AxesLabel} \{“x, имя”, “y, имя”\}]$.
Здесь опция AxesLabel устанавливает текст по осям x и y , который заключается в кавычки;
- $\text{ListPlot}\{\{x_1, y_1\}, \{x_2, y_2\}, \dots\}$ — выводит точки графика, образующие матрицу исходных данных функции $y = f(x)$, заданной в виде таблицы.

PlotJoined — опция функции ListPlot, указывающая на то, что точки необходимо соединять отрезками прямой;

□ Show[$f_1(x)$, $f_2(x)$,] — строит графики функций $f_1(x)$, $f_2(x)$, с наложением их друг на друга.

Рассмотрим пример решения задачи интерполяции с применением графики.

Пример 3.12

Пусть функция $y = f(x)$ задана в виде таблицы (табл. 3.9).

Таблица 3.9

	Значения переменных						
x	1	2	4	8	10	15	20
y	0.4	0.7	1.2	1.8	2.2	2.6	3.2

Необходимо найти функцию интерполяции $y = \varphi(x)$, проверить правильность решения задачи и вычислить погрешность интерполяции.

Задачу будем решать в такой последовательности:

1. Выбор вида функции $y = \varphi(x)$.
2. Определение коэффициентов функции интерполяции.
3. Проверка правильности решения путем сопоставления графиков функций $f(x)$ и $\varphi(x)$.
4. Вычисление среднеквадратической погрешности.

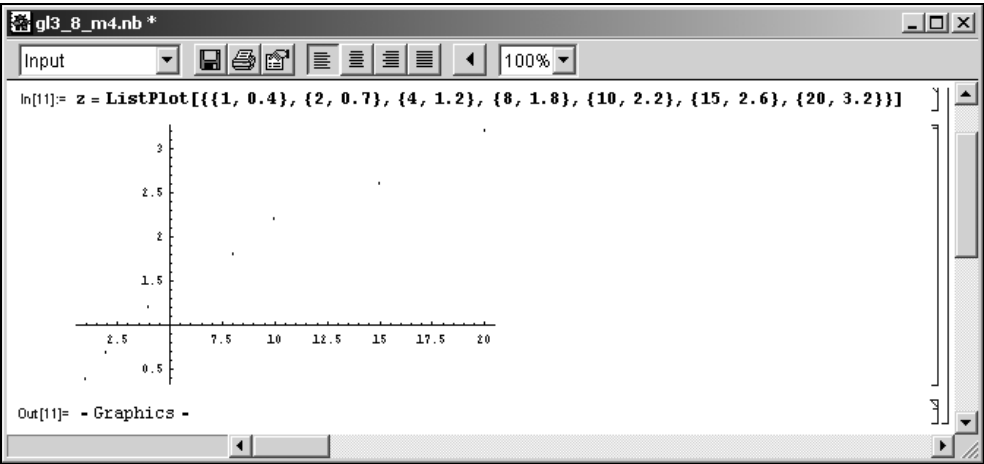


Рис. 3.13

Опишем решение задачи в указанной последовательности.

Выбор вида функции $y = f(x)$.

Выведем точки графика на экран дисплея, воспользовавшись функцией ListPlot:

```
z = ListPlot[{{1, 0.4}, {2, 0.7}, {4, 1.2}, {8, 1.8}, {10, 2.2}, {15, 2.6}, {20, 3.2}}].
```

На рис. 3.13 функция представлена на экране в виде точек.

По расположению точек можно предположить, что функцией интерполяции может быть дробно-линейная функция $y = \frac{x}{a + bx}$.

Определение коэффициентов функции интерполяции.

Дробно-линейная функция содержит два неизвестных коэффициента: a и b . Выберем две точки с координатами: $\{2, 0.7\}$ и $\{15, 2.6\}$. Составим следующую систему уравнений:

$$0.7 = \frac{2}{a + 2b}$$

$$2.6 = \frac{15}{a + 15b}.$$

Решим эту систему нелинейных уравнений методом Ньютона, воспользовавшись функцией FindRoot:

```
FindRoot[{2 / (a + 2b) == 0.7, 15 / (a + 15b) == 2.6}, {a, 1}, {b, 1}]
```

Ответ:

{a 2.40913, b 0.224007}.

Таким образом, дробно-линейная функция после округления имеет вид:

$$y = \frac{x}{2.41 + 0.22 x}.$$

Для сравнения результатов интерполяции и оценки эффективности сглаживания решим задачу интерполяции с применением функции Fit и базиса a, x, x^2 :

```
Fit[{{1, 0.4}, {2, 0.7}, {4, 1.2}, {8, 1.8}, {10, 2.2}, {15, 2.6}, {20, 3.2}}, {a, x, x^2}, x].
```

Ответ:

$$y = 0.250522 + 0.230486 x - 0.00429974 x^2.$$

Проверка правильности решения путем сопоставления графиков функций $f(x)$ и (x) .

На рис. 3.14 показаны процедуры получения графиков с помощью функций Plot, ListPlot, Show. Изображение графиков говорит о том, что решение по-

лучено верное, поскольку они хорошо согласуются между собой (практически совпадают).

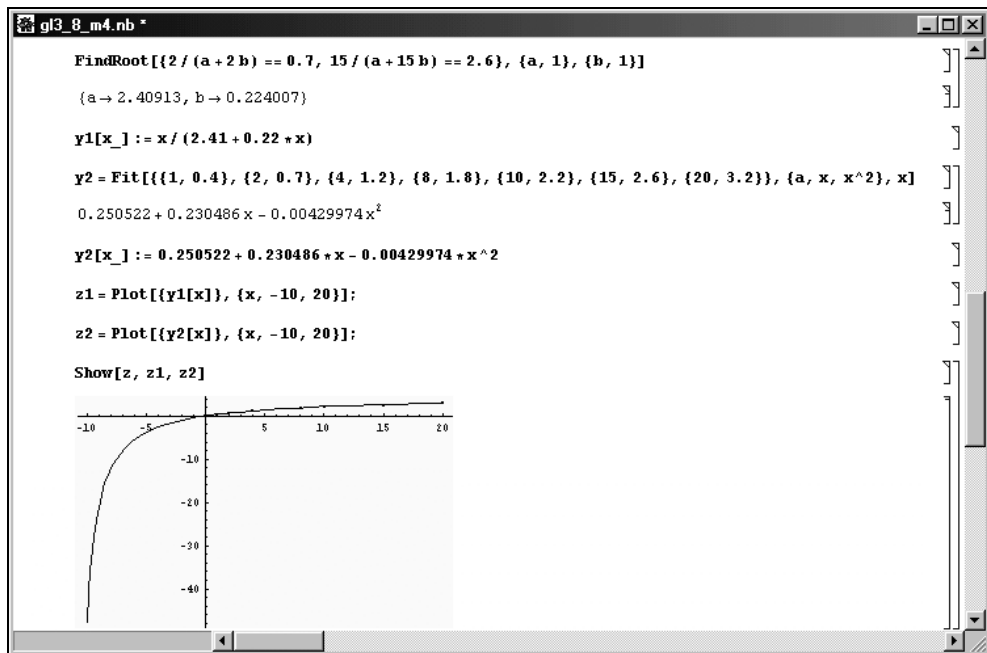


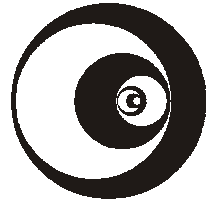
Рис. 3.14

Вычисление погрешностей.

Вычислим абсолютную и относительную погрешности функций интерполяции, полученные путем решения системы нелинейных уравнений (дробно-линейная функция) и с помощью функции Fit (многочлен второй степени). Вычисления выполним, используя векторные процедуры вычитания, возведения в квадрат и суммирования. В результате расчетов получим следующие значения погрешностей:

- = 0.0456 — абсолютная погрешность дробно-линейной функции;
- = 11.4 % — максимальная относительная погрешность дробно-линейной функции;
- = 0.0301 — абсолютная погрешность квадратичной функции;
- = 7.5 % — максимальная относительная погрешность квадратичной функции.

Из сравнения погрешностей видим, что более предпочтительным является полином второй степени.



Глава 4

Компьютерные технологии решения задач интерполяции в среде Maple

4.1. Интерполяция точная в узлах. Универсальный метод

Решение задачи интерполяции точной в узлах универсальным методом в среде Maple сводится к выполнению следующих действий:

1. Выбор вида функции интерполяции.
2. Решение системы уравнений.
3. Проверка правильности решения задачи.
4. Определение погрешности интерполяции.

Для реализации этих действий на ПК в среде Maple имеется большое число разнообразных функций. Приведем основные из них, достаточные для решения задач интерполяции универсальным методом.

Функция solve решения систем уравнений

Эта функция позволяет решать системы уравнений в численном виде. Она имеет следующий вид:

$\text{solve}(\{y_1, y_2, \dots\}, \{a, b, \dots\})$.

Здесь:

- $y_1, y_2, \dots$ — уравнения системы;
- $a, b, \dots$ — неизвестные.

При решении систем линейных уравнений решение является точным, представленным в виде дробей. Для получения решения в естественной форме используется команда **evalf** (<выражение>). Откликом будет представление чисел в естественной форме.

Решение появляется на экране дисплея только тогда, когда выражение заканчивается символом ";" (точка с запятой) и нажимается клавиша <Enter>. Если нет необходимости в получении промежуточных результатов, то в конце выражения ставится символ ":" (двоеточие), а перевод строки осуществляется нажатием клавиши <Enter>.

Перевод строки без выдачи результата (при наличии в конце выражения символа ";") также возможен нажатием клавиши <Enter>, но только при нажатой клавише <Shift>.

Приведем примеры использования команды **evalf**.

❑ **evalf**(-1 / 2 * cos(6) + 1 / 2);

Ответ: .0199148566

❑ **evalf**(3 / 7);

Ответ: 0.4285714286

Если команду в среде Maple 5 записать в виде: **evalf**(''), то решение преобразуется в естественную форму, например:

❑ **exp**(5) / 2 + 1 / 4;

Ответ: $\frac{1}{2}e^5 + \frac{1}{4}$;

evalf();

Ответ:

74.45657955

❑ **sqrt**(2);

Ответ: $\sqrt{2}$

evalf('');

Ответ: 1.414213562

На рис. 4.1 показан экран решения задач в среде Maple 6.

Проверку правильности решения задачи можно осуществить тремя способами:

❑ методом подстановки;

❑ табулированием функции аппроксимации;

❑ использованием графических образов.

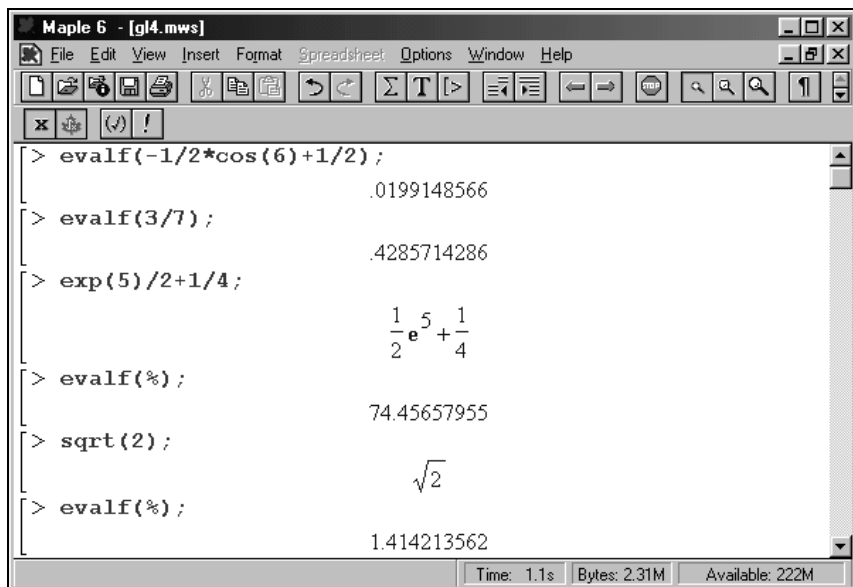


Рис. 4.1

Для проверки правильности решения методом подстановки используется функция `subs`, имеющая вид:

`subs(z, s).`

Здесь:

- ☐ z — имя, присвоенное функции `solve` решения системы уравнений,
- ☐ s — имя, присвоенное системе уравнений.

Пример 4.1

Пусть необходимо решить следующую систему уравнений:

$$x + 3y = 2$$

$$-2x + y = 7$$

Решение системы с помощью функций `solve` и `subs` имеет вид:

```
s := {x + 3 * y = 2, -2 * x + y = 7};
```

```
z := solve(s, {x, y});
```

```
subs(z, s);
```

Решение показано на рис. 4.2.

Из ответа видно, что система уравнений имеет решение $x = -\frac{19}{7}$, $y = \frac{11}{7}$ и что решение верное, т. к. подстановка значений x и y в исходные уравнения приводит к тождеству $\{2 = 2, 7 = 7\}$.

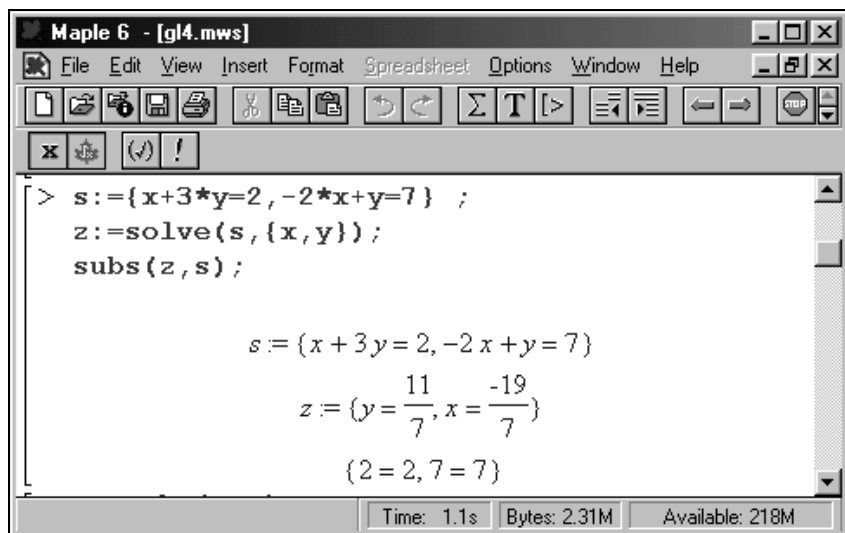


Рис. 4.2

В этом решении переменным x и y не присвоены никакие значения. Присвоение им имени осуществляется с помощью функции `subs`.

```
a: = subs(z, x);
```

```
b: = subs(z, y);
```

Решение показано на рис. 4.3.

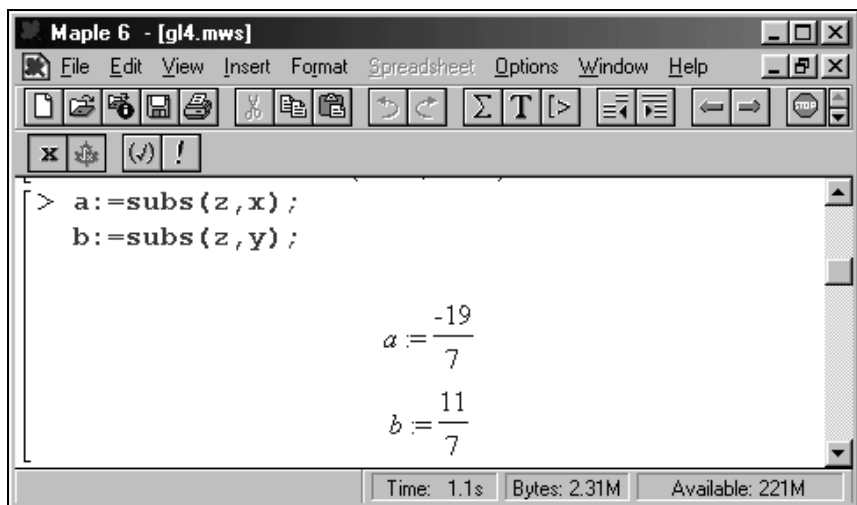


Рис. 4.3

Проверка правильности решения задачи интерполяции может быть осуществлена путем табулирования функции $\varphi(x)$ и сравнения ее с исходными данными. В среде Maple табулирование осуществляется с помощью функции `seq`, имеющей вид:

`seq( $\varphi(x)$ ,  $x = \{a, b, \}$ ).`

Здесь:

- $\varphi(x)$ — функция интерполяции;
- a, b , — значения аргумента x .

Пусть необходимо протабулировать функцию интерполяции:

$y = 2.3x^2 + 1.35x + 3.6$ при $x = 1, 2, 3, 4, 5$.

Функция будет иметь вид:

`seq(2.3 * x^2 + 1.35 * x + 3.6, x = {1,2,3,4,5});`

После нажатия клавиши <Enter> получится ответ:

7.25, 15.50, 28.35, 45.80, 67.85

Решение показано на рис. 4.4.

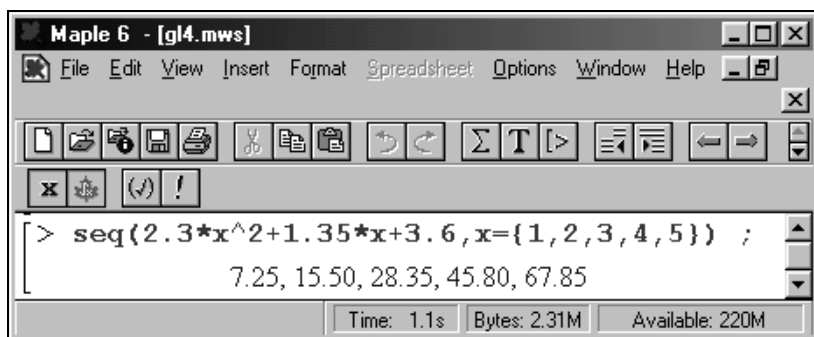


Рис. 4.4

Сравнивая эти данные с исходными, можно судить о правильности решения задачи интерполяции.

Наиболее наглядным может оказаться проверка правильности решения путем использования графических образов. Для этого служит функция `plot`, которая для данного случая имеет вид:

`plot( $\{f_1(x), f_2(x), \dots, f_n(x)\}$ ,  $x = a \dots n$ ,  $y = b \dots m$ ).`

Здесь:

- $f_i(x)$ — i -ая функция;
- $x = a \dots n$ — диапазон изменения x ;
- $y = b \dots m$ — диапазон изменения y .

При определении точности интерполяции имеются лишь две функции: $f(x)$ и $\varphi(x)$.

Функция $f(x)$ задана в виде матрицы. В функции plot она представляется так:

$[[x_1, y_1], [x_2, y_2], \dots, [x_n, y_n]]$.

Функция $\varphi(x)$ является функцией интерполяции, представленной в виде математического выражения.

Пример 4.2

Пусть функция $f(x)$ задана в виде таблицы (табл. 4.1).

Таблица 4.1

	Значения переменных					
x	1	2	3	4	5	6
y	2.5	3.7	6.2	4.1	2.9	2

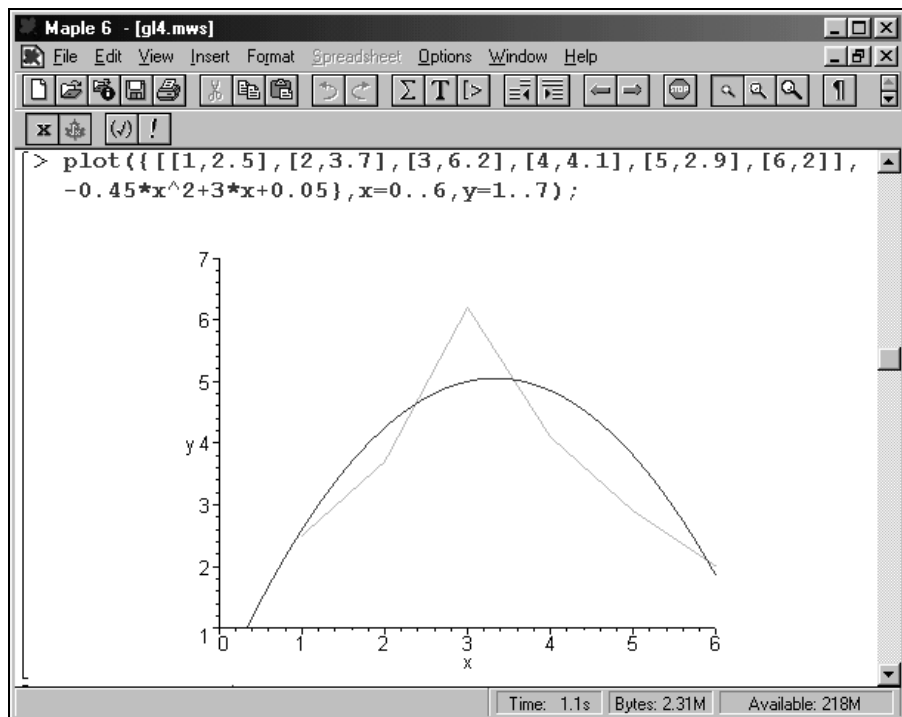


Рис. 4.5

И пусть функция интерполяции является полиномом:

$$y = -0.45x^2 + 3x + 0.05.$$

Тогда функция `plot` будет иметь вид:

```
plot([[1, 2.5], [2, 3.7], [3, 6.2], [4, 4.1], [5, 2.9], [6, 2]], -
0.45 * x^2 + 3 * x + 0.05}, x = 0, ..., 6, y = 1, ..., 7);
```

После нажатия клавиши <Enter> на экране монитора увидим графики двух функций (рис. 4.5), из которых видно, что решение ошибочно, т. к. графики сильно отличаются друг от друга.

Наиболее вероятно, что полином второй степени плохо описывает функцию, заданную таблично.

Определим погрешность интерполяции. Неточность интерполяционной формулы будем оценивать абсолютной среднеквадратической погрешностью, которая вычисляется по формуле:

$$\varepsilon = \frac{\sqrt{\sum_{i=1}^n \Delta_i^2}}{n}.$$

Здесь:

□ n — число узлов исходной функции, заданной таблично;

□ $\Delta_i = y(x_i) - (x_i)$.

Из формулы видно, что для вычисления необходимо выполнить следующие действия:

1. Представить в векторной форме функции $y(x)$ и (x) .
2. Для представления (x) в векторной форме необходимо выполнить табулирование функции с помощью функции `seq`.
3. Найти разность векторов.
4. Вычислить сумму квадратов разностей.
5. Извлечь корень квадратный из суммы квадратов разностей векторов и разделить на n .

Сумму квадратов разностей в Maple можно получить путем вычисления векторного произведения однотипных векторов i . Для этого служит функция `evalm`, которая имеет вид:

`evalm (A & * B)`, где A , B — векторы.

Функция `evalm` находится в пакете линейной алгебры (`linalg`), который загружается с помощью команды `with(linalg)`.

Покажем процедуры вычисления погрешности на предыдущем примере.

Введем вектор функции $(x) = 0.45x^2 + 3x + 0.05$:

```
seq(-0.45*x^2 + 3*x + 0.05, x = {1, 2, 3, 4, 5, 6});
```

Ответ: 2.6, 4.25, 5, 4.85, 3.8, 1.85.

Введем векторы функций $f(x)$ и (x)

```
with(linalg):
```

```
y1:= [2.5, 3.7, 6.2, 4.1, 2.9, 2];
```

```
y2:= [2.6, 4.25, 5, 4.85, 3.8, 1.85];
```

Вычислим сумму квадратов разностей, т. е. введем выражение:

```
e:= evalm((y1 - y2) & * (y1 - y2));
```

Ответ:

```
e:= 3.1475
```

Вычислим среднеквадратическую погрешность, т. е. введем выражение:

```
E:= sqrt(e) / 6;
```

Ответ:

```
E:= 0.2956865833
```

Процедуры вычисления погрешностей интерполяции в среде Maple 6 приведены на рис. 4.6.

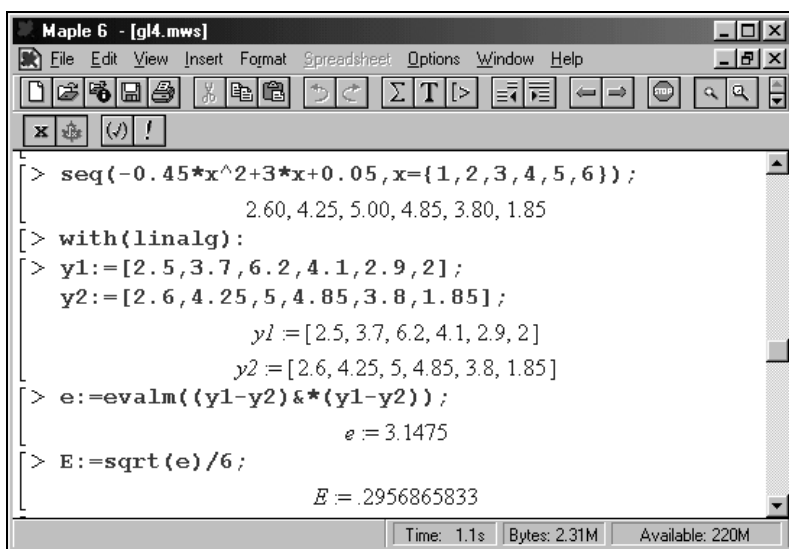


Рис. 4.6

Максимальная относительная погрешность будет:

$$\delta = \frac{E}{y_{\min}} \cdot 100\% = \frac{0.295}{2.5} \cdot 100\% = 11.8\%.$$

Процедуры решения задачи интерполяции универсальным методом покажем на примере.

Пример 4.3

Пусть зависимость давления P насыщенного водяного пара от температуры t приведена в табл. 4.2.

Таблица 4.2

	Значения переменных							
$t, ^\circ\text{C}$	0	5	10	15	20	25	30	35
$P, \text{мм.рт.ст.}$	4.58	6.54	9.21	12.79	17.54	23.76	31.82	42.18

Необходимо установить закон изменения давления насыщенного водяного пара от температуры, т. е. найти функцию $P = f(t)$.

Предположим, что такой закон существует, а данные таблицы достоверны. Тогда для отыскания функции $P = f(t)$ можно воспользоваться интерполяцией точной в узлах при ограниченном их числе.

При решении задач интерполяции наиболее трудным этапом является выбор вида функции интерполяции.

Предположим первоначально, что функцией интерполяции является полином n -ой степени. Определим степень полинома, для чего вычислим табличные разности (табл. 4.3).

Таблица 4.3

	Значения переменных							
t	0	5	10	15	20	25	30	35
P	4.58	6.54	9.21	12.79	17.54	23.76	31.82	42.18
ΔP		1.96	2.67	3.58	4.75	6.22	8.06	10.36
$\Delta^2 P$			0.71	0.98	1.17	1.47	1.84	2.3
$\Delta^3 P$				0.27	0.19	0.3	0.37	0.46

Из табл. 4.3 видно, что табличные разности третьего порядка почти одинаковы. Это позволяет нам предположить, что интерполяционный многочлен является многочленом третьей степени:

$$P(t) = a_0 + a_1t + a_2t^2 + a_3t^3.$$

Для определения коэффициентов многочлена составим систему уравнений. Выберем из таблицы координаты: {0, 4.58}, {10, 9.21}, {25, 23.76}, {35, 42.18}.

Тогда система уравнений будет иметь вид:

$$a_0 = 4.58$$

$$a_0 + a_1 \cdot 10 + a_2 \cdot 10^2 + a_3 \cdot 10^3 = 9.21$$

$$a_0 + a_1 \cdot 25 + a_2 \cdot 25^2 + a_3 \cdot 25^3 = 23.76$$

$$a_0 + a_1 \cdot 35 + a_2 \cdot 35^2 + a_3 \cdot 35^3 = 42.18$$

Решение системы уравнений в среде Maple:

```
y1: = a0 = 4.58:
```

```
y2: = a0 + a1 * 10 + a2 * 10^2 + a3 * 10^3 = 9.21:
```

```
y3: = a0 + a1 * 25 + a2 * 25^2 + a3 * 25^3 = 23.76:
```

```
y4: = a0 + a1 * 35 + a2 * 35^2 + a3 * 35^3 = 42.18:
```

```
s: = {y1, y2, y3, y4}:
```

```
z: = solve(s, {a0, a1, a2, a3});
```

Ответ в среде Maple 5:

$$\{a_1 = .36448, a_2 = .00568,$$

$$a_3 = .00041714, a_0 = 4.58\}$$

Проверим правильность решения системы уравнений:

```
subs(z, s);
```

Ответ в среде Maple 5:

$$\{42.18 = 42.18, 23.76 = 23.76, 9.21 = 9.21, 4.58 = 4.58\}$$

Решение верное. Таким образом, интерполяционный полином имеет вид:

$$P(t) = 4.58 + 0.36448 t + 0.00568 t^2 + 0.000417 t^3.$$

Вычислим теперь абсолютную среднеквадратическую и максимальную относительную погрешности. Для этого введем векторы исходной функции и функции интерполяции, а также выполним операции в соответствии с выражением для абсолютной и относительной погрешности. Вычислительные процедуры приводятся в среде Maple 5.

Образуем вектор функции интерполяции:

```
seq(4.58 + 0.36448 * t + 0.00568 * t^2 + 0.000417 * t^3,
t = {0, 5, 10, 15, 20, 25, 30, 35});
```

Ответ:

$$4.58, 6.597, 9.21, 12.73, 17.48, 23.76, 31.89, 42.18.$$

Дальнейшие процедуры объединим в единый блок:

```
P: = [4.58, 6.54, 9.21, 12.79, 17.54, 23.76, 31.82, 42.18]:
```

```
P1: = [4.58, 6.597, 9.21, 12.73, 17.48, 23.76, 31.89, 42.18]:
```

```
e: = P - P1:
```

```
s: = evalm((e&*e)):
```

```
E: = sqrt(s) / 8;
d: = E / 4.58*100;
```

Ответ:

$E = .01549$

$d = .338$.

Для среды Maple 6 эти процедуры показаны на рис. 4.7.

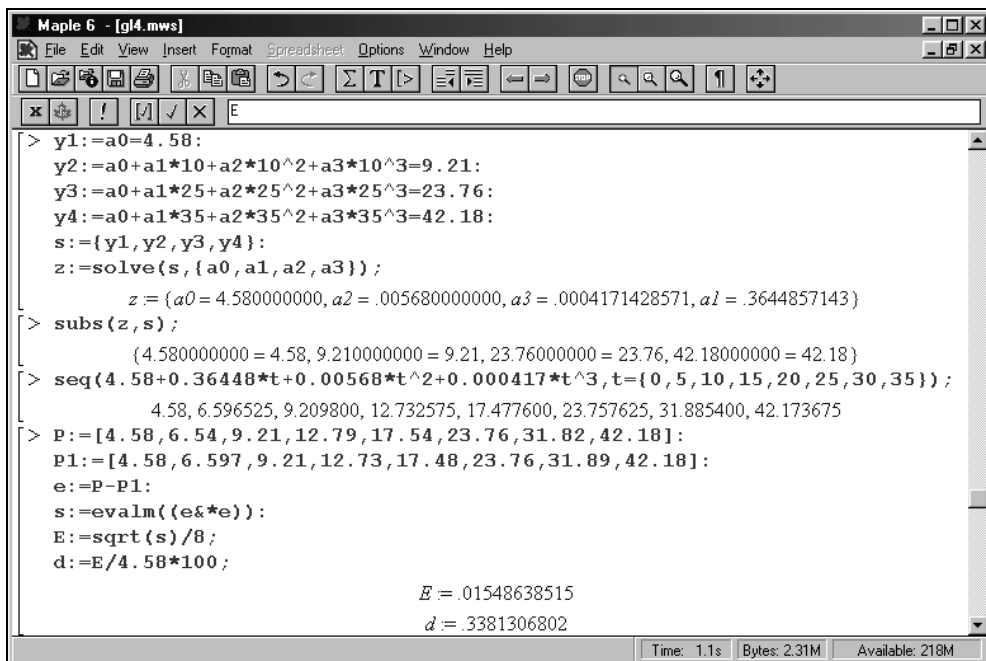


Рис. 4.7

Малая относительная погрешность (примерно 0.34 %) свидетельствует о том, что полученная в результате интерполяции функция хорошо отражает зависимость давления насыщенного водяного пара от температуры.

4.2. Интерполяция точная в узлах (функция `interp`)

Воспользуемся функцией `interp` для определения зависимости давления насыщенного водяного пара от температуры. Эта функция имеет вид:

`interp(x, y, x).`

Здесь:

□ x — вектор аргументов функции $y(x)$;

□ y — вектор значений функции.

Символы (имена векторов) x , y могут быть любыми.

Рассмотрим случай многочлена третьей степени, взяв из табл. 4.2 следующие координаты: $\{0, 4.58\}$, $\{10, 9.21\}$, $\{25, 23.76\}$, $\{35, 42.18\}$.

Процедуры решения задачи интерполяции в среде Maple 6:

```
t := vector([0, 10, 25, 35]):
```

```
P := vector([4.58, 9.21, 23.76, 42.18]):
```

```
interp(t, P, t);
```

Ответ:

$$.000417t^3 + .005679t^2 + .36448t + 4.58$$

Решение показано на рис. 4.8. Ответ, естественно, совпадает с ранее полученным решением.

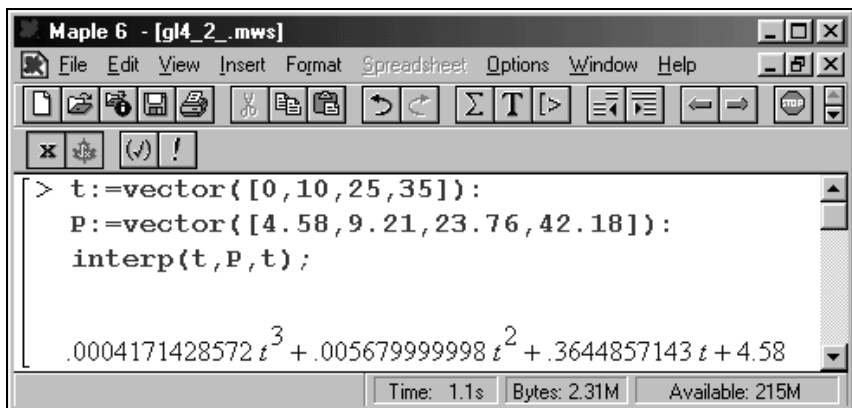


Рис. 4.8

4.3. Интерполяция приближенная в узлах (аппроксимация)

Естественно предположить, что погрешность, возникающая при интерполяции точной в узлах, происходит за счет неточности исходных данных. Тогда имеет смысл осуществить сглаживание неточностей, используя функции аппроксимации. Одной из таких функций в Maple является функция `fit`, которая имеет вид:

`fit[leastsquare]([x, y], y =  $\varphi(x)$ ) ([a, b]).`

Здесь:

- x, y — символы (имена) аргумента и функции аппроксимации;
- $\varphi(x)$ — функция аппроксимации, задаваемая пользователем;
- a, b — векторы узлов интерполяции и значений функции.

Решим задачу аппроксимации полиномом третьей степени, используя функцию `fit`. В качестве примера найдем зависимость давления насыщенного водяного пара от температуры, которая получена методами точными в узлах интерполяции (см. *разделы 4.1 и 4.2*).

Чтобы воспользоваться функцией `fit`, необходимо предварительно загрузить статистический пакет `stats` с помощью команды `with(stats)`.

Приведем процедуры решения задачи аппроксимации в среде Maple 5:

`with(stats):`

`a := [0, 5, 10, 15, 20, 25, 30, 35]:`

`b := [4.58, 6.54, 9.21, 12.79, 17.54, 23.76, 31.82, 42.18]:`

`fit[leastsquare]([t, P], P = c + d * t + e * t^2 + f * t^3) ([a, b]);`

Ответ:

$$P = 4.55 + .371t + .00545t^2 + .000418t^3$$

Ответ записан с округлением до трех значащих цифр. Количество значащих цифр в ответе можно задать командой:

`Digits := <цифра>.`

Решение в среде Maple 6 приведено на рис. 4.9.

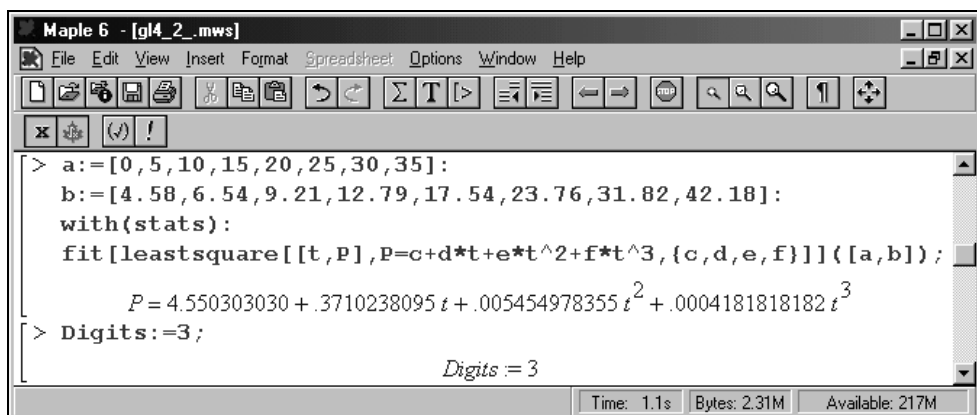


Рис. 4.9

Вычислим среднеквадратическую погрешность для случая многочлена третьей степени. Создадим вектор функции P :

```
y := seq(4.55 + 0.371 * t + 0.00545 * t^2 + 0.000418 * t^3,
x = {0,5,10,15,20,25,30,35});
```

Ответ:

```
y = 4.55, 6.5935, 9.223, 12.752, 17.494, 23.7626, 31.871, 42.133
```

Дальнейшие процедуры понятны из предыдущих примеров.

```
f1:= [4.55, 6.5935, 9.223, 12.752, 17.494, 23.7626, 31.871,
42.133]:
```

```
f2:= [4.58, 6.54, 9.21, 12.79, 17.54, 23.76, 31.82, 42.18]:
```

```
e:= f1 - f2:
```

```
s:= evalm((e&*e)):
```

```
E:= sqrt(s) / 8;
```

```
d:= E / 4.58*100;
```

Ответ:

```
E = .013867
```

```
d = .302787
```

Из сравнения с данными прежних расчетов погрешностей видно, что сглаживание привело к уменьшению максимальной относительной погрешности с 0.34 % до 0.30 %. Решение в среде Maple 6 приведено на рис. 4.10.

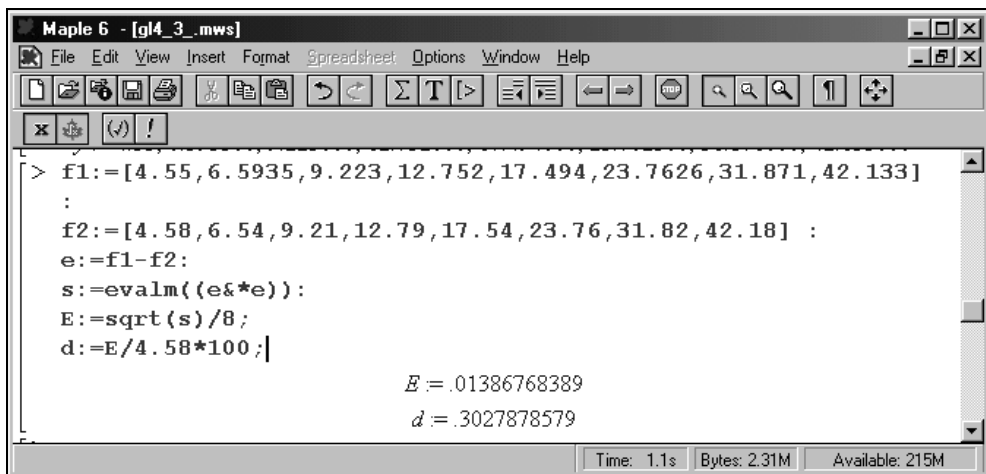


Рис. 4.10

4.4. Интерполяция нелинейными функциями

Задача интерполяции точной в узлах нелинейными функциями сводится к решению систем нелинейных уравнений.

Решение систем нелинейных уравнений осуществляется с помощью функции `solve`, имеющей такой же вид, как и в случае решения систем линейных уравнений:

`solve({y1, y2, }, {a, b,}).`

Существенное отличие этой функции при решении системы нелинейных уравнений в ответе. Система линейных уравнений имеет единственное и при том точное решение. Система нелинейных уравнений может иметь множество решений как вещественных, так и комплексных. Функция `solve` в системе Maple 5 имеет следующие особенности:

- может выдавать решение в неявном виде `RootOf(poly)` (где `poly` — полином n -ой степени);
- решение в явном виде может быть получено, но при этом некоторые решения будут потеряны.

Чтобы получить решения в явном виде, можно использовать следующие способы.

□ Способ 1.

Вслед за строкой `solve` ввести команду `evalf (``);`

При исполнении этой команды будет выдано решение в явном виде, однако некоторые решения могут быть потеряны.

□ Способ 2.

Использовать функцию:

`[allvalues(s[n])].`

Здесь:

- s — переменная, присвоенная функции `solve`;
- n — число найденных решений функцией `solve`.

Для численного решения системы нелинейных уравнений используется функция `fsolve({y1, y2, }, {a, b, })`. С помощью этой функции находятся решения в числах, представленных в естественной форме. Функция имеет ряд опций и записывается в виде:

`fsolve({y1, y2, }, {a, b, }, option).`

Опциями могут быть:

- `complex` — находятся комплексные корни;
- `maxsols = n` — имеется n решений системы уравнений.

Опция позволяет получить все вещественные корни.

Кроме функций `solve` и `fsolve`, в системе Maple имеются также следующие функции:

- `isolve` — решение систем уравнений в целых числах;
- `msolve` — решение уравнений по модулю m ;
- `rsolve` — решение систем линейных разностных уравнений.

Из этих функций при решении задач интерполяции чаще всего может понадобиться функция `isolve`, имеющая вид:

`isolve({y1, y2, }, {a, b, }).`

Покажем процедуры решения следующей системы нелинейных уравнений:

$$x + 3y = 1$$

$$x^2 - y^3 = 3$$

Используются функции `solve`, `fsolve`, `isolve`.

```
R := {x + 3 * y - 1, x^2 - y^3 - 3}:
```

```
S := solve (R, {x, y});
```

Ответ (в среде Maple 5):

```
{y = 1, x = -2}, {y = RootOf(_Z^2 - 8_Z - 2),
```

```
x = -3RootOf(_Z^2 - 8_Z - 2) + 1}
```

Решение в среде Maple 6 показано на рис. 4.11.

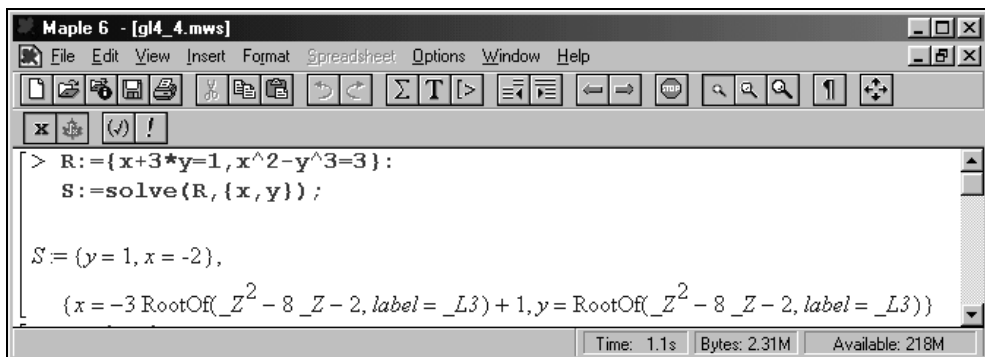


Рис. 4.11

Из ответа видно, что система имеет три решения, причем одно из них выдано в явном виде: $x = -2$, $y = 1$. Два других решения можно получить, решая следующие уравнения:

$$z^2 - 8z - 2 = 0$$

$$-3(z^2 - 8z - 2) + 1 = 0$$

При этом первое из них даст два значения y , а второе — два соответствующих значения x .

Для получения решения в радикалах воспользуемся функцией `allvalues`:

```
[allvalues(S [2])];
```

Ответ:

$$\{x = -11 - 9\sqrt{2}, y = 4 + 3\sqrt{2}\}, \{x = -11 + 9\sqrt{2}, y = 4 - 3\sqrt{2}\}$$

Для получения решения в естественной форме воспользуемся командой:

```
evalf('');
```

Ответ:

$$\{x = -23.7279, y = 8.2426\},$$

$$\{x = 1.7279, y = -.242\}.$$

Решение приведено на рис. 4.12.

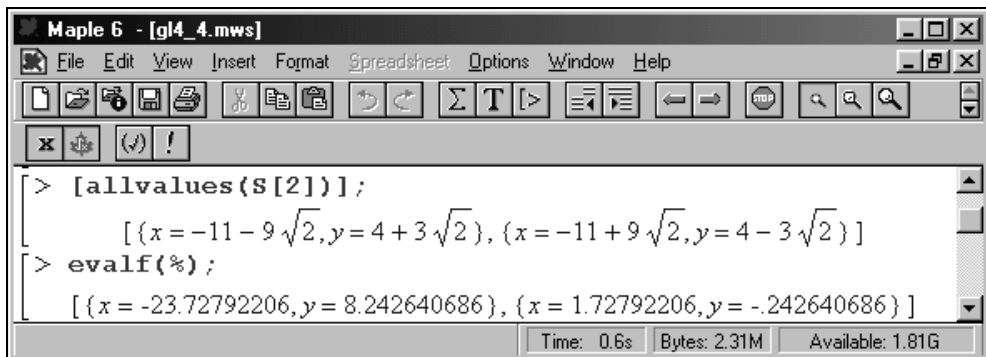


Рис. 4.12

Теперь найдены все решения нелинейного уравнения.

Если воспользоваться командой `evalf('')`, не вводя функции `allvalues`, то решение будет выдано в виде:

$$\{y = 1, x = -2\}, \{y = -.2426, x = 1.72\}$$

Решение $\{y = 8.2426, x = -23.7279\}$ будет потеряно.

Воспользуемся теперь функцией `fsolve`:

```
fsolve(R, {x, y});
```

Ответ: $\{y = 1, x = -2\}$

Здесь потеряны два решения.

Применим теперь опцию `maxsols = n` (в нашем случае $n = 3$):

```
fsolve(R, {x, y}, maxsols = 3);
```

Ответ:

$$\{y = 1, x = -2\}, \{y = -.242, x = 1.7279\}, \{y = 8.2426, x = -23.7279\}$$

Найдем теперь решение в целых числах:

```
isolve(R, {x, y});
```

Ответ: $\{y = 1, x = -2\}$.

Решение приведено для среды Maple 5 (для среды Maple 6 решение иллюстрируется на рис. 4.13).

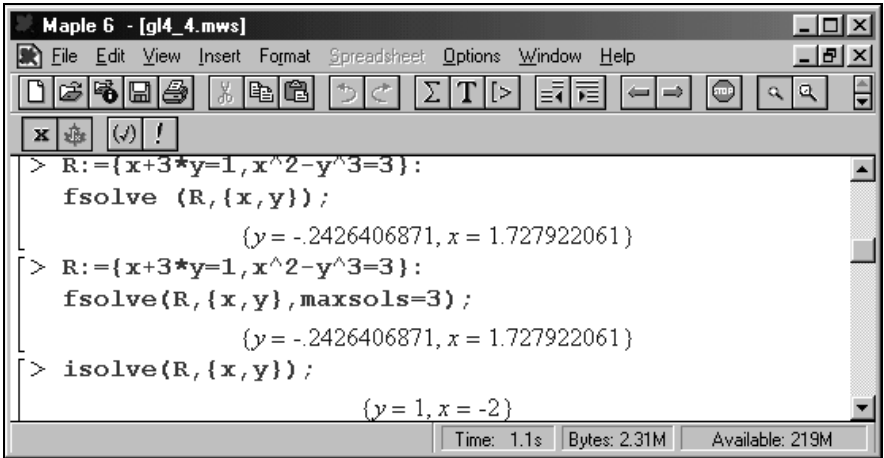


Рис. 4.13

Рассмотрим пример решения задачи интерполяции нелинейными функциями.

Пример 4.4

Пусть данные эксперимента приведены в табл. 4.4.

Таблица 4.4

	Значения переменных					
x	1	2	3	4	5	6
y	6.5	20	53.5	167	473	1470

Выберем функцию интерполяции в виде показательной функции $y = ab^x + c$ (согласно разд. 2.3 главы 2).

Составим систему нелинейных уравнений для случая интерполяции точной в узлах ($x = 1, 4, 6$):

$$ab + c = 6.5$$

$$ab^4 + c = 167$$

$$ab^6 + c = 1470$$

Воспользуемся функцией solve.

```
R:= {a * b + c - 6.5, a * b^4 + c - 167, a * b^6 + c - 1470};
z:= solve(R, {a, b, c});
```

Система Maple 5 выдаст следующие 4 решения:

```
{a = 1.75549, b = -3.06595, c = 11.88226},
{a = 762.5296 - 621.907 I, b = -.45092 -.82999 I, c = 866.518 + 352.46285 I},
{a = 762.5296 + 621.907 I, b = -.45092 + .82999 I, c = 866.518 - 352.46285 I},
{a = 2.151197, b = 2.96779, c = .115688859}
```

Из решения видно, что существуют два вещественных решения. Остальные два решения комплексные и должны быть отброшены, т. к. эти решения в нашей постановке задачи смысла не имеют.

Можно было исключить комплексные решения, воспользовавшись функцией

```
fsolve(R, {a, b, c}, maxsols = 4);
```

В таком случае ответом будут два вещественных решения.

Таким образом, функциями интерполяции $\varphi(x)$ могут быть следующие две функции:

$$y = 1.755 \cdot (-3.06595)^x + 11.88,$$

$$y = 2.15 \cdot 2.9678^x + 0.1157.$$

Какая же из этих функций наиболее предпочтительна? Для ответа на этот вопрос вычислим погрешности интерполяции, для чего первоначально осуществим табулирование этих функций:

```
with(linalg):
```

```
Y1:= 1.755 * (-3.06595)^x + 11.88:
```

```
Y2:= 2.15 * (2.9678)^x + 0.1157:
```

```
seq(Y1, x = {1, 2, 3, 4, 5, 6});
```

```
seq(Y2, x = {1, 2, 3, 4, 5, 6});
```

Ответ:

6.499, 28.377, -38.699, 166.953, -463.5678, 1469.5779

6.5, 19.053, 56.3165, 166.908, 495.123, 1469.198

Решение в среде Maple 6 приведено на рис. 4.14.

Из результатов расчетов видно, что решение $Y_1 = f(x)$ содержит отрицательное значение y , что не физично. Это решение должно быть отвергнуто. Тогда функцией интерполяции будет следующая нелинейная функция:

$$y = 2.15 \cdot 2.97^x + 0.12.$$

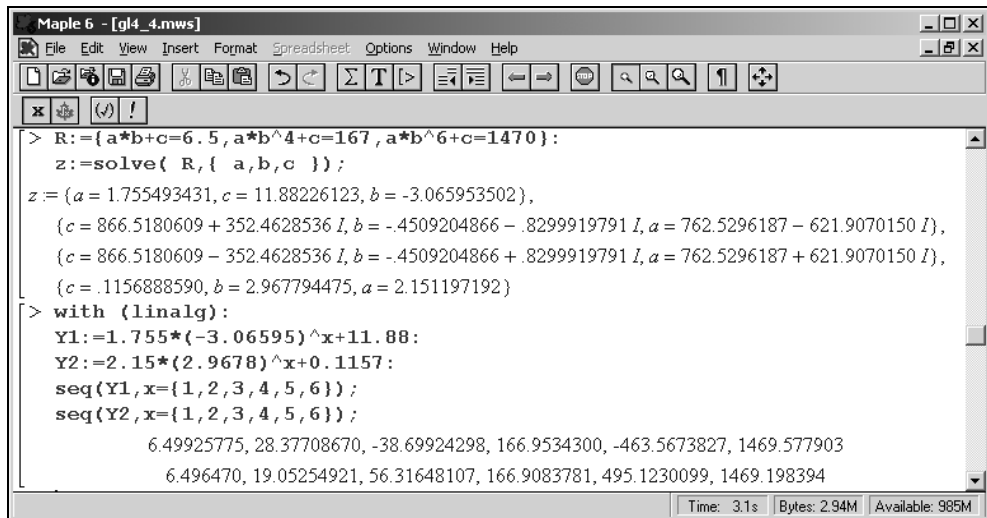


Рис. 4.14

Коэффициенты здесь представлены с точностью до двух знаков после запятой.

Среднеквадратическая относительная погрешность интерполяции, вычисленная по методике, описанной в *разд. 4.3*, равна 2.14 %, что практически совпадает с погрешностью интерполяции, равной 2.35 %, полученной для этой задачи в *разд. 2.3*. Система Derive выдала следующую нелинейную функцию интерполяции:

$$y = 1.89 \cdot 3.03^x + 0.76.$$

Некоторое отличие функций интерполяции, полученных системами Derive и Maple, объясняется применением различных численных методов и различных точностей вычислений.

4.5. Аппроксимация Паде

Аппроксимация Паде осуществляется в Maple с помощью функции `pade`, имеющей вид:

$$\text{pade}(f(x), x, [m, n]).$$

Здесь:

- ☐ $f(x)$ — функция, заданная в виде математического выражения;
- ☐ x — аргумент функции;
- ☐ m — степень многочлена числителя;
- ☐ n — степень многочлена знаменателя.

Функция `pade` находится в пакете `numapprox`, который необходимо первоначально загрузить командой `with(numapprox)`.

В качестве примера выполним аппроксимацию Паде функции e^{-x} при $n = m = 3$:

```
with(numapprox):
```

```
z:=pade(exp(-x), x, [3, 3]);
```

```
evalf('');
```

Решение приведено на рис. 4.15.

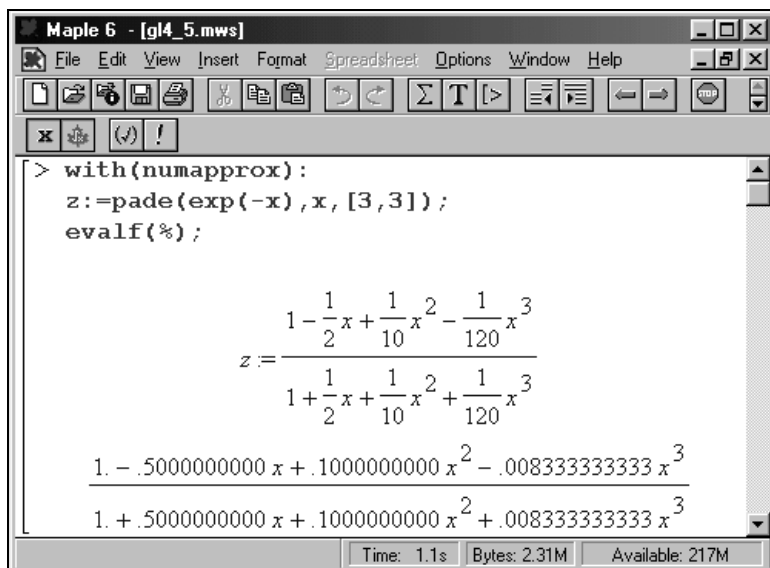


Рис. 4.15

4.6. Аппроксимация Паде при помощи полиномов Чебышева

Аппроксимация при помощи полиномов Чебышева осуществляется функцией `chebade`. Функция имеет вид:

`chebade(f(x), x, [m, n]).`

Здесь:

- ☐ $f(x)$ — функция, представленная в виде математического выражения;
- ☐ x — аргумент функции $f(x)$;
- ☐ m — порядок полинома в числителе;
- ☐ n — порядок полинома в знаменателе.

Решение выдается в виде суммы полиномов Чебышева вида $T(n, x)$ (где n — порядок полинома). Полиномы Чебышева можно получить с помощью команд:

`with(orthopoly) :`

$T(n, x)$,

или

`with(ortpolonly)`

$U(n, x)$.

Здесь:

□ $T(n, x)$ — полиномы первого рода;

□ $U(n, x)$ — полиномы второго рода.

В качестве примера выполним аппроксимацию Паде функции $y = e^{-x}$ полиномами Чебышева третьего порядка:

`with(numapprox) :`

`z: = chebpade(exp(-x), x, [3,3]) ;`

Ответ:

$$z = (1.000 \, T(0, x) - .484 \, T(1, x) + .0479 \, T(2, x) - .00190 \, T(3, x)) /$$

$$/ (T(0, x) + .480 \, T(1, x) + .0470 \, T(2, x) + .00192 \, T(3, x))$$

Развернутый ответ можно получить с помощью команды `with(orthopoly)`.

`with(numapprox) :`

`with(orthopoly) :`

`z: = chebpade(exp(-x), x, [3,3]) ;`

Ответ:

$$z = (.952 - .478x + .0959x^2 - .00798x^3) /$$

$$/ (.952 + .474x + .094x^2 + .0077x^3)$$

Решение приведено на рис. 4.16.

4.7. Сплайн-интерполяция

Интерполяция сплайнами в среде Maple 5 осуществляется с помощью функции `spline`, которая имеет вид:

`spline(x, y, t, d).`

Здесь:

□ x — вектор узлов интерполяции;

□ y — вектор значений функции;

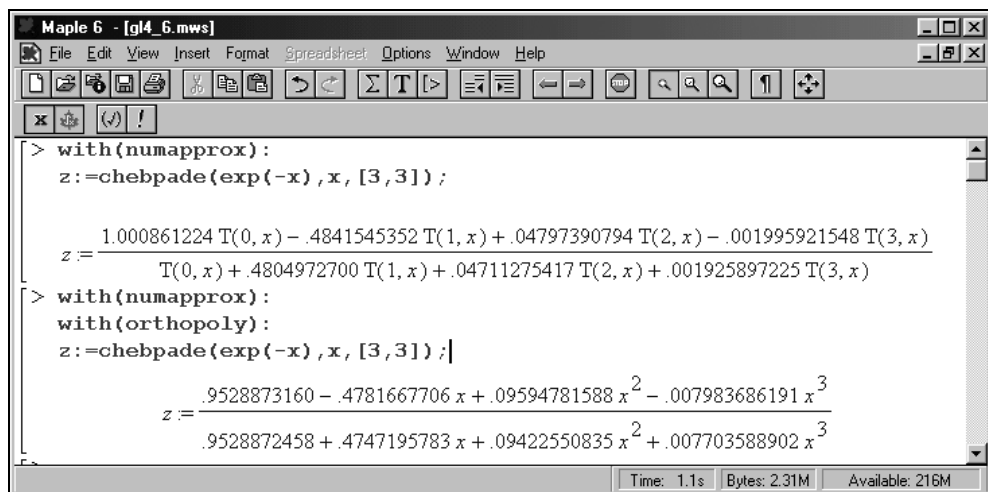


Рис. 4.16

- t — аргумент получаемой функции интерполяции (символ произвольный);
- d — порядок сплайна:
 - linear — линейный;
 - quadratic — квадратичный;
 - cubic — кубический (если сплайн кубический, то d можно опустить);
 - quartic — четвертой степени.

Перед использованием функции spline требуется ее вызов с помощью функции readlib(spline).

Пример реализации сплайн-интерполяции в среде Maple 6:

```
readlib(spline):
x:= [0, 1, 2, 3, 4]:
y:= [1, 2, 8, 27, 64]:
res:= spline(x, y, t, quadratic);
```

Ответ:

$$\begin{aligned} 1 + t^2 & \quad t < 1 \\ 4 - 6t + 4t^2 & \quad t < 2 \\ 24 - 26t + 9t^2 & \quad t < 3 \\ 24 - 26t + 9t^2 & \quad \text{otherwise} \end{aligned}$$

Решение показано на рис. 4.17.

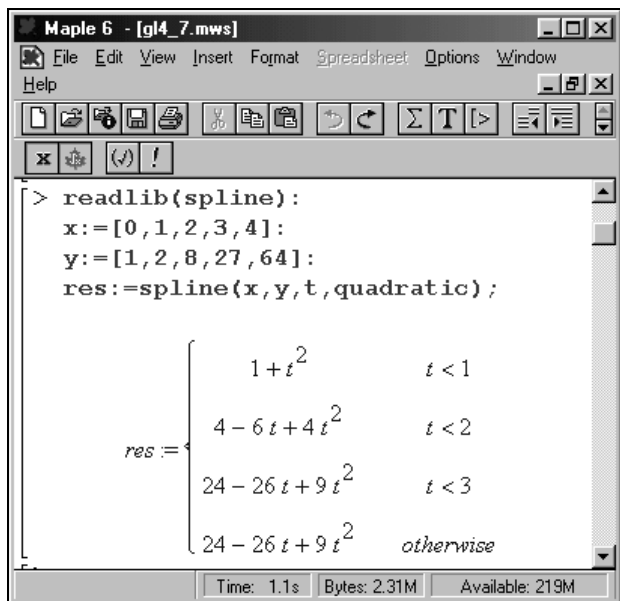


Рис. 4.17

Как видно из примера, результатом функции `readlib(spline)` является создание сплайна в виде условного выражения для $t < k$. Для получения сплайна в виде процедуры s нужно использовать следующую функцию:

$s := \text{'spline / makeproc'}(res, t).$

Здесь:

- ☐ res — имя результата;
- ☐ t — независимая переменная.

4.8. Представление функции интерполяции полиномами

При решении задач функциями интерполяции могут быть функции с различными базисами. Пользователю же часто полезно иметь решение в виде полиномов. В этом случае Maple представляет следующие возможности:

- ☐ разложение функции в ряды (Тейлора, Лорана);
- ☐ представление многочлена в виде схемы Горнера;
- ☐ представление функции в виде цепной дроби;
- ☐ наилучшая минимаксная рациональная аппроксимация;
- ☐ наилучшая минимаксная аппроксимация по алгоритму Ремеза.

В Maple 5 имеются функции, позволяющие выполнить перечисленные приближения функций. Для использования соответствующих функций вначале необходимо загрузить пакет numapprox, в котором находятся все функции. Пакет загружается командой **with(numapprox)**.

Рассмотрим некоторые из функций:

- **hornerform($f(x)$, x)** – схема Горнера;
- **confracform($f(x)$)** – преобразование функции $f(x)$ в цепную дробь;
- **taylor($f(x)$, $x = x_0$, n)** (где n – число членов ряда, x_0 – значение аргумента, вблизи которого ведется разложение в ряд Тейлора);
- **laurent($f(x)$, x , n)** – разложение в ряд Лорана;
- **minimax($f(x)$, $x = ab$, n)** (где ab – область разложения наилучшей минимаксной аппроксимации).

Примеры приведем без комментариев в связи с их очевидностью:

with(numapprox) :

hornerform(a * x + b * x^2 + c * x^3 + d * x^4, x) ;

Ответ: $(a + (b + (c + d \cdot x) \cdot x) \cdot x) \cdot x$.

confracform((7/60 * x^3 + x) / (1 + 1/20 * x^2)) ;

Ответ: $-\frac{7}{3}x + \frac{80}{3} \cdot \frac{1}{x + \frac{20}{x}}$

minimax((x-1) / (x+1), x = 0..1, 3) ;

Ответ: $-0.997 + (1.901 + (-1.372 + 0.470x) \cdot x) \cdot x$. (Здесь выписаны лишь три знака после запятой).

taylor(exp(-x), x = 0.4)

Ответ: $1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + 0(x^4)$

laurent(cos(x), x, 5) ;

Ответ: $1 - \frac{1}{2}x^2 + \frac{1}{24}x^4 + 0(x^5)$

Maple 5 имеет функцию разложения функции многих переменных в ряд Тейлора. Загрузка функции осуществляется командой **readlib(mtaylor)**. Приведем процедуры разложения в ряд Тейлора функции $e^x \cos(y)$ по степеням x и y вокруг $x = 0$ при $n = 5$:

readlib(mtaylor) :

mtaylor(exp(x) * cos(y), [x = 0, y], 5) ;

Ответ:

$$1 + x - \frac{1}{2}y^2 + \frac{1}{2}x^2 + \frac{1}{6}x^3 - \frac{1}{2}xy^2 + \frac{1}{24}y^4 - \frac{1}{4}x^2y^2 + \frac{1}{24}x^4.$$

Процедуры представления функции полиномами приведены на рис. 4.18.

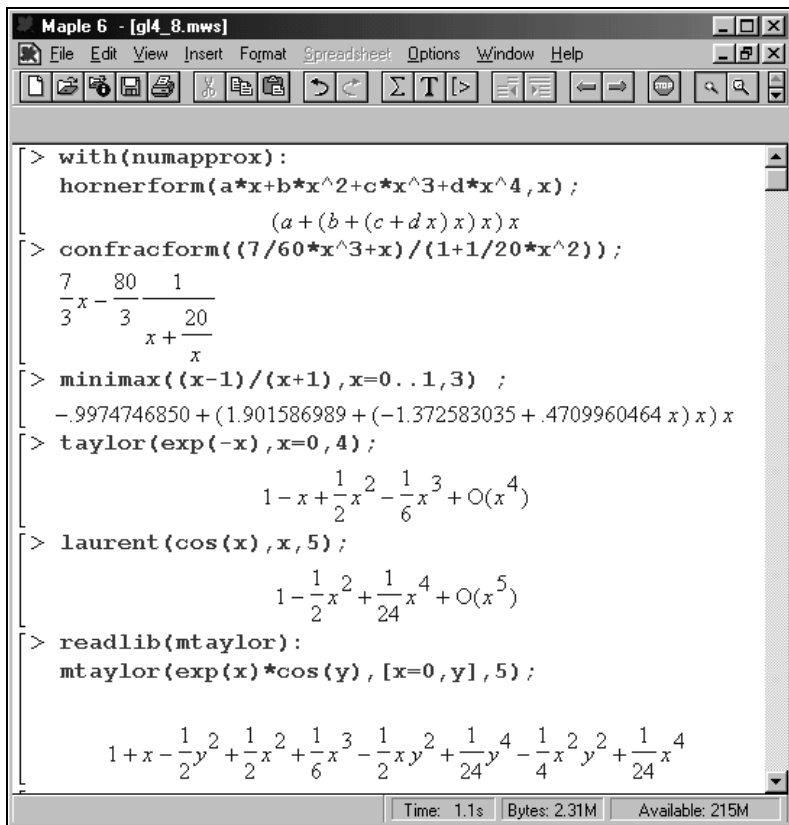
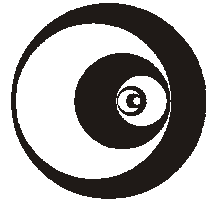


Рис. 4.18



Глава 5

Компьютерные технологии решения задач интерполяции в системе Mathcad

5.1. Интерполяция точная в узлах. Универсальный метод

Решение задачи интерполяции точным универсальным методом в системе Mathcad, как и во всех других математических системах, требует выбора вида функции интерполяции $\varphi(x)$, составления системы уравнений, решения системы и проверки правильности результатов.

В системе Mathcad эти процедуры осуществляются следующим образом:

1. Исходные данные, представленные в табличной форме, вводятся в виде векторов V_x и V_y .
2. Выполняется визуализация функции $f(x)$ в виде графика.
3. Подбирается подходящая функция $\varphi(x)$ путем сравнения с графиками типичных функций.
4. Составляется система уравнений, которая решается либо методами символьной математики, либо численными методами (выбор метода зависит от вида функции, необходимой точности получения результата, возможностей данного компьютера).
5. Определяется погрешность интерполяции.

Опишем процедуры, позволяющие достаточно просто получить решение задачи интерполяции в системе Mathcad. Описание процедур выполнено применительно к версиям Mathcad 5 и Mathcad 6. Примеры решены в Mathcad версии Mathcad 2000 Professional.

Ввод исходных данных

Для создания векторов V_x и V_y исходных данных необходимо:

1. Щелкнуть мышью в свободном месте экрана, где будет расположен вектор.
2. Ввести символ V_x и нажать клавишу присвоения значения $<:=>$ (двоеточие). На экране появятся символы: " $V_x :=$ ".
3. Выполнить команду главного меню **Math | Matrices** (или нажать клавиши $<Ctrl>+<M>$). На экране появится диалоговое окно **Matrices**.
4. Указать число строк (**Rows**), число столбцов (**Columns**), затем щелкнуть мышью по кнопке **Create** (Создать) (например, если вектор состоит из 10 элементов, то необходимо ввести в поле **Rows** число 10, а в поле **Columns** — число 1. На экране появится вектор 10×1 с пустыми полями).
5. Заполнить поля исходными данными аргументов x функции $f(x)$, пользуясь клавишей $<Tab>$ или щелкая мышью по пустым полям.

Аналогично образуется вектор V_y , который должен иметь ту же размерность, что и вектор V_x .

Визуализация функции

Представление функции $y = f(x)$ в виде графика осуществляется с помощью векторов V_x , V_y . Методика состоит в следующем:

1. Создать векторы V_x и V_y (при этом символ V опускается, векторам дается имя x и y).
2. Ввести диапазон изменения индексной переменной i (или j), соответствующий числу узлов функции, при этом первым индексом в Matcad считается ноль (например, если число узлов равно 12, то $i := 0..11$).

Символ присвоения ($:=$) образуется нажатием клавиши $<Shift>+<:=>$ (точка с запятой).

Многоточие ($..$) образуется нажатием клавиши $<:=>$ (точка с запятой).

3. Создать пустой график с помощью пункта меню **Graphics | Create x-y Plot**.
4. Заполнить пустые поля графика по осям координат (индексы i при x_i и y_i устанавливаются нажатием клавиши $<[>$ (квадратная скобка)).
5. Щелкнуть мышью вне графика или нажать клавишу $<F9>$, что приведет к появлению графика. График является основанием для выбора вида функции интерполяции $\phi(x)$.

Пример 5.1

Пусть функция $y = f(x)$ представлена в виде таблицы (табл. 5.1).

Таблица 5.1

	Значения переменных					
X	1	2.4	3.6	7.8	11.5	19
y	1	2.5	4.6	3.2	1.6	1

Необходимо функцию представить в виде графика. Процедуры решения представлены на рис. 5.1.

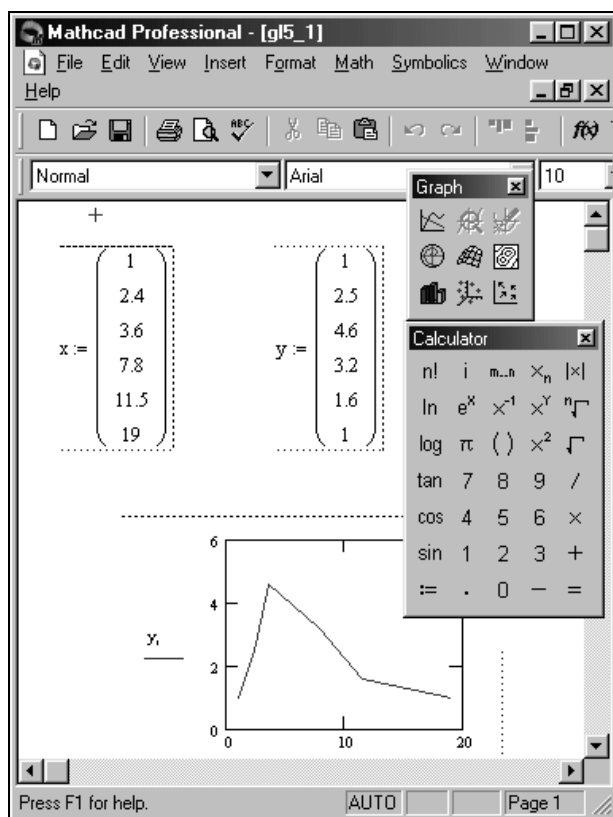


Рис. 5.1

Решение систем уравнений

Системы уравнений в Mathcad могут решаться с помощью функций: Isolve, Find, Minerr. Функции представляются в следующем виде:

□ Isolve(M , V).

Здесь:

- M — матрица коэффициентов системы линейных алгебраических уравнений;
- V — вектор правых частей системы уравнений.

□ Find($x, y, z, \dots$).

Здесь:

- $x, y, z, \dots$ — искомые неизвестные.

Решение системы уравнений с помощью функции `lsolve` рассмотрим на примере.

Пример 5.2

Пусть необходимо решить следующую систему линейных уравнений:

$$3x + 6y = 9$$

$$2x + 0.5y = 4$$

Решение системы в Mathcad осуществляется с помощью следующих процедур:

$$M := \begin{vmatrix} 3 & 6 \\ 2 & 0.5 \end{vmatrix}$$

$$V := \begin{vmatrix} 9 \\ 4 \end{vmatrix}$$

`lsolve(M, V)`

После нажатия клавиши `<=>` (равно) на экране отобразится ответ в виде вектора:

$$\begin{vmatrix} 1.857 \\ 0.571 \end{vmatrix}$$

Таким образом, в нашем примере $x = 1.857$, $y = 0.571$.

Решение можно получить еще проще, не используя функцию `lsolve`:

$$M^{-1} \cdot V =$$

После нажатия клавиши `<=>` (равно) получаем тот же ответ.

Решение с помощью функции `lsolve`, а также матричным способом в системе Mathcad 2000 показано на рис. 5.2. Стрелка $\rightarrow$ является знаком символьных вычислений (реализуется нажатием клавиш `<Ctrl>+<•>` (точка)). Решение появляется после нажатия клавиши `<Enter>`.

Рассмотрим решение системы уравнений с помощью функции `Find(x, y, z, ...)`.

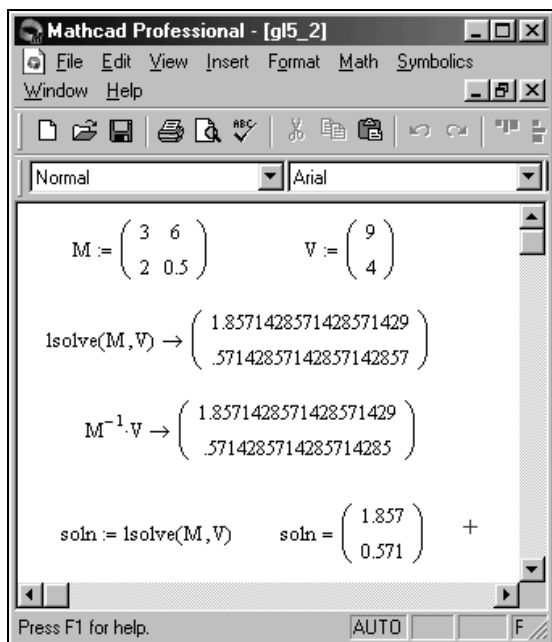


Рис. 5.2

Функция Find дает возможность решать как линейные, так и нелинейные системы уравнений. Используется метод итераций. Решение осуществляется следующим образом:

1. Задаем начальные приближения для всех искомых неизвестных:

$$x := x_0, y := y_0, z := z_0, \dots$$

2. Вводим слово **Given**, указывающее на то, что далее следует система уравнений.

3. Вводим систему уравнений:

$$f1(x, y, z, \dots) = b1$$

$$f2(x, y, z, \dots) = b2$$

$$f_n(x, y, z, \dots) = b_n$$

знак = (равно) вводится путем нажатия клавиш <Ctrl>+<=>.

4. Вводится: $z := \text{Find}(x, y, z, \dots)$

В результате получаем: $z = \text{ответ}$

Пример 5.3

Пусть требуется решить следующую систему нелинейных уравнений:

$$2x^2 + 3y + z = 18$$

$$-3x + 7y^2 + 5z^2 = 525$$

$$x + y + z = 13$$

За начальные приближения возьмем $x = 1.5$, $y = 1.5$, $z = 8$.

Процедуры в Mathcad:

$x := 1.5$ $y := 1.5$ $z := 8$

Given

$$2 \cdot x^2 + 3 \cdot y + z = 18$$

$$-3 \cdot x + 7 \cdot y^2 + 5 \cdot z^2 = 525$$

$$x + y + z = 13$$

$R := \text{Find}(x, y, z)$

Ответ:

$$R = \begin{pmatrix} 1 \\ 2 \\ 10 \end{pmatrix}$$

Ответ в этом примере является точным.

Процедуры решения показаны на рис. 5.3.

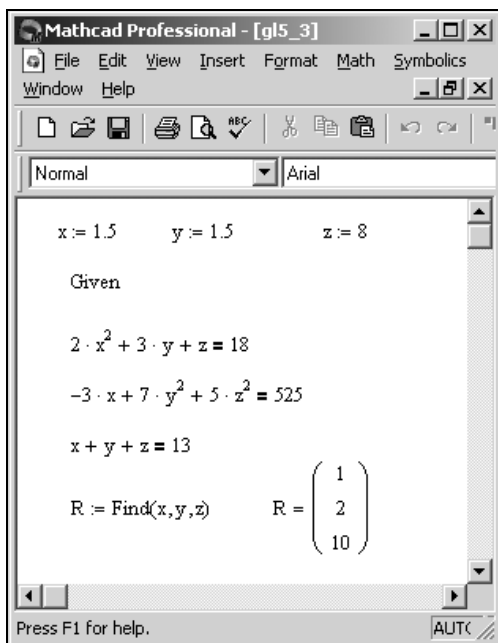


Рис. 5.3

Если функция Find не дает решения, то следует попытаться решить систему уравнений с помощью функции Minerr (x, y, z,). Эта функция может выдать решение, не достигнув требуемой точности итераций, задаваемой функцией TOL.

При использовании функции Minerr необходимо осуществлять проверку правильности решения системы уравнений.

Решение системы линейных уравнений в Mathcad с помощью функции Find(x, y, z) можно получить символьным методом. Этот метод дает возможность решить систему уравнений как в аналитическом, так и в численном виде. Решение получаем с помощью последовательности действий:

1. Установить автоматический режим в меню **Math** путем активизации команды **Automatic Mode**.
2. Выполнить команду **StartMath** в меню **Math**.
3. Ввести команду **Given**.
4. Ввести систему уравнений.
5. Ввести функцию: Find (x, y, z, ...).
6. Нажать клавиши <Ctrl>+<. > (точка). На экране появится знак символьных вычислений: Find(x, y, z,) ;
7. Щелкнуть мышью вне Find. На экране отобразится ответ.

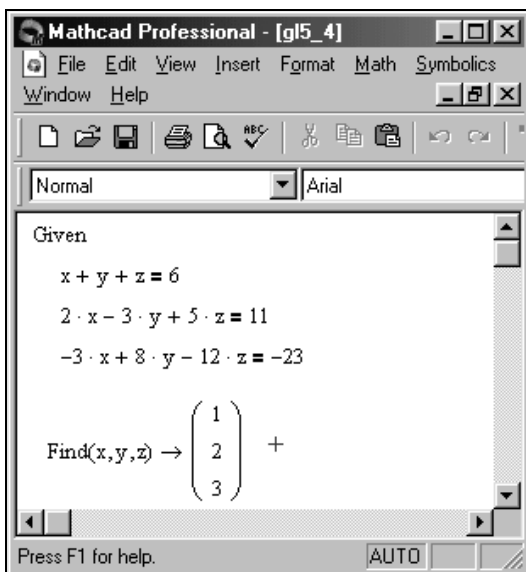


Рис. 5.4

Пример 5.4

Пусть требуется решить следующую систему уравнений:

$$x + y + z = 6$$

$$2x - 3y + 5z = 11$$

$$-3x + 8y - 12z = -23.$$

Выполняя описанные действия, получим решение в виде, показанном на рис. 5.4.

Вместо функции Find(x, y, z) можно применить функцию Minerr(x, y, z).

Определение погрешностей интерполяции

Прежде чем определять погрешность интерполяции, необходимо убедиться в правильности полученного решения. В системе Mathcad проверка правильности решения системы уравнений осуществляется следующим образом:

1. Переменным x , y , z , присваиваются значения, полученные в результате решения системы уравнений.
2. Вводится левая часть каждого из уравнений системы и ставится знак "=" (равно).

Получаемые при этом ответы должны быть равны значениям правых частей уравнений.

Приведем процедуры проверки правильности решения системы уравнений предыдущего примера:

$$x := 1 \quad y := 2 \quad z := 3$$

$$x + y + z =$$

$$2 * x - 3 * y + 5 * z =$$

$$-3 * x + 8 * y - 12 * z =$$

Mathcad выдаст поочередно следующие ответы: 6, 11, -23.

Теперь можно определять погрешность интерполяции. За критерий точности часто принимается абсолютная среднеквадратическая погрешность, вычисляемая по формуле:

$$\varepsilon = \frac{\sqrt{\sum_{i=1}^n \Delta_i^2}}{n}.$$

Здесь:

□ n — число узлов исходной функции;

□ $\Delta_i = y(x_i) - \varphi(x_i)$ — разность значений исходной и интерполяционной функций в i -ом узле интерполяции.

В системе Mathcad сумму квадратов разностей $\sum_{i=1}^n \Delta_i^2$ легко вычислить,

пользуясь матричными операциями вычитания и скалярного произведения двух векторов. При этом предварительно для получения вектора функции $\varphi(x)$ необходимо ее протабулировать на всех значениях аргумента.

Табулирование функции $\varphi(x)$ осуществляется следующим образом:

1. Устанавливаем диапазон изменения аргумента в виде $x := x_n, \Delta_x \dots x_k$, где:
 - x_n, x_k — начальное и конечное значения аргумента соответственно;
 - Δ_x — шаг таблицы (например, $x := 0, 0.1..5$ или $x := 1, 1..7$. Если шаг равен 1, то его можно опустить: $x := 1..7$).
2. Вводим функцию $\varphi(x)$.
3. Щелкаем мышью вне функции. На экране монитора появится ответ в векторной форме.

Если шаг функции переменный, то ее табулирование осуществляется путем вычисления значений $\varphi(x)$ для всех значений аргумента x .

После табулирования функции вычисляются абсолютная и максимальная относительная погрешности интерполяции. Вычислительные процедуры состоят в выполнении следующих операций:

1. Создаем векторы $y(x)$ и $\varphi(x)$.
2. Вычисляем разность векторов $D = y(x) - \varphi(x)$.
3. Находим скалярное произведение разности векторов $z = D \cdot D$.
4. Вычисляем абсолютную среднеквадратическую погрешность E в режиме калькулятора.
5. Вычисляем максимальную относительную погрешность интерполяции.

Пример 5.5

Пусть функции $y(x)$ и $\varphi(x)$ имеют в узлах интерполяции значения, приведенные в табл. 5.2.

Таблица 5.2

	Значения переменных					
$y(x)$	1.2	2.6	3.8	5	6.8	8.3
$\varphi(x)$	1.4	2.5	3.5	5.1	7	8.1

Тогда вычисление абсолютной среднеквадратической погрешности в системе Mathcad осуществляется путем реализации следующих процедур:

$$y := \begin{vmatrix} 1.2 \\ 2.6 \\ 3.8 \\ 5 \\ 6.8 \\ 8.3 \end{vmatrix}$$

$$\varphi := \begin{vmatrix} 1.4 \\ 2.5 \\ 3.5 \\ 5.1 \\ 7 \\ 8.1 \end{vmatrix}$$

$$D := y - \varphi$$

$$z := D * D$$

$$E := \sqrt{z} / 6$$

$$E = 0.08$$

Рассмотрим технологию решения задачи интерполяции точной в узлах на примере.

Пример 5.6

Пусть необходимо определить возраст дерева L по его высоте h . Зависимость $L = f(h)$, полученная для сосны в результате натурного эксперимента и его статистической обработки, приведена в табл. 5.3.

Таблица 5.3

	Значения переменных											
$h, \text{ м}$	18	22	22.5	28.5	30.5	32	33	34	35	35.5	36	36.3
$L, \text{ лет}$	20	30	40	50	60	70	80	90	100	110	120	130

В данном примере необходимо найти функцию интерполяции $L = \varphi(h)$, когда узлы интерполяции неравноотстоящие.

Будем решать задачу в такой последовательности:

1. Выбор вида функции интерполяции.
2. Составление и решение системы уравнений.
3. Оценка погрешностей интерполяции.

Выбор вида функции интерполяции

Представим функцию $L = f(h)$ в виде графика. Для этого необходимо представить на экране монитора аргумент h и функцию $L(h)$ в виде векторов. Векторное представление функции в системе Mathcad описано в разд. 5.1.

После создания векторов h и L функция $L = f(h)$ представляется графически.

Выполним следующие действия:

1. Введем индексную переменную $i := 0..n$, где $n + 1$ — число аргументов h (в нашем случае $n = 11$, т. к. первый элемент вектора считается в Mathcad нулевым), символ многоточия (..) образуется нажатием клавиши <;> (точка с запятой).
2. Щелчком мышью в месте экрана, где предполагается поместить график.
3. Выполним из меню **Graphics** (Графика) пункт **Create x-y Plot** (активируем Декартов график) (можно нажать клавишу <@>). На экране появится пустой график с шестью полями ввода.
4. Среднее поле по оси абсцисс заполним знаком h_i , а по оси ординат — знаком L_i .
5. Щелчком мышью вне графика или нажмем клавишу <F9>. На экране появится график функции $L = f(h)$ с автоматической установкой масштабов. Если масштабы необходимо изменить, то следует заполнить пустые поля ввода по обеим осям графика.

Для нашего примера образование графика показано на рис. 5.5.

Из графика видно, что функция $L = f(h)$ достаточно гладкая, однако в области $h = 20$ — 25 м имеется локальный максимум. Вполне возможно, что его появление обусловлено недостоверностью исходных данных. Полученная зависимость не описывается какой-либо известной функцией. Предположим, что такой функцией является полином n -ой степени. Выберем в качестве интерполяционного полином второй степени $L = a + bh + ch^2$.

Составление и решение системы уравнений

Выберем в качестве аппроксимирующих следующие координаты функции $L = f(h)$:

[18, 20], [28.5, 50], [36.3, 130]. Тогда система уравнений будет иметь вид:

$$a + 18b + 18^2c = 20$$

$$a + 28.5b + 28.5^2c = 50$$

$$a + 36.3b + 36.3^2c = 130$$

Решим эту систему, воспользовавшись матричными процедурами. В системе Mathcad решение будет иметь вид:

$$M := \begin{vmatrix} 1 & 18 & 18^2 \\ 1 & 28.5 & 28.5^2 \\ 1 & 36.3 & 36.3^2 \end{vmatrix} \quad B := \begin{vmatrix} 20 \\ 50 \\ 130 \end{vmatrix} \quad M^{-1} \cdot B = \begin{vmatrix} 175.994 \\ -15.944 \\ 0.404 \end{vmatrix}$$

Из решения получаем следующую функцию интерполяции:

$$L = 176 - 16h + 0.404h^2.$$

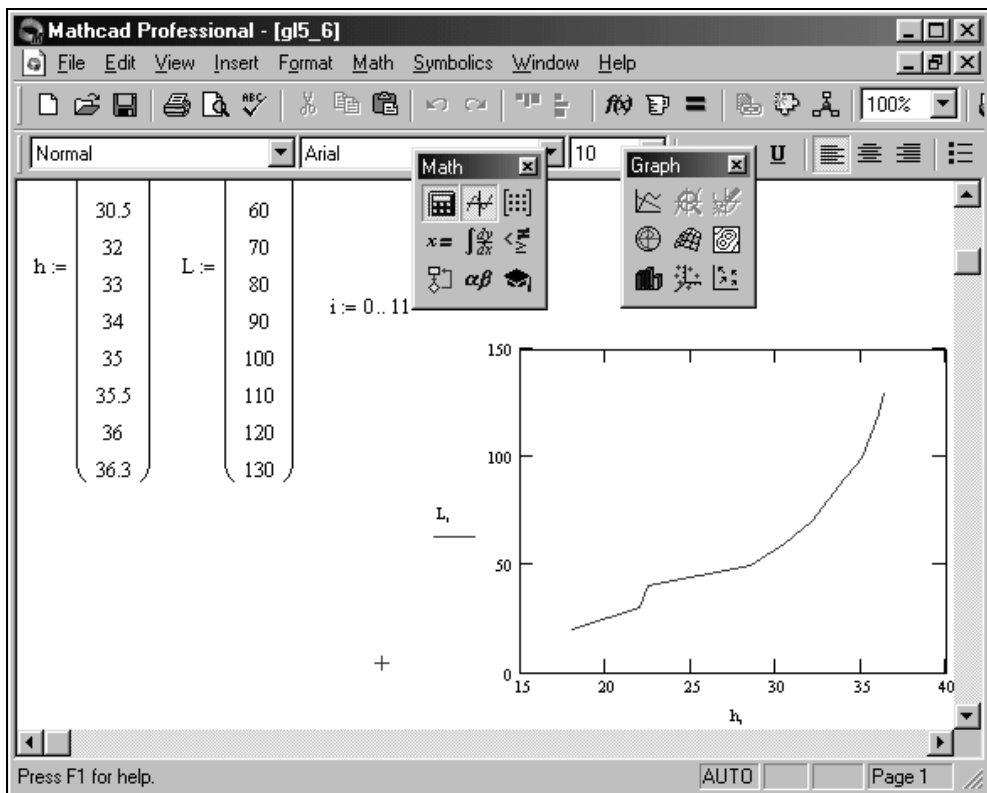


Рис. 5.5

На рис. 5.6 приведены графики исходной функции, заданной таблично, и функции интерполяции. Внешне они близки друг другу.

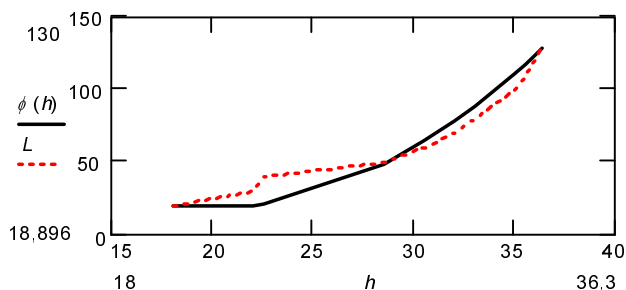


Рис. 5.6

Оценка погрешности интерполяции

Протабулируем функцию $L = \varphi(h)$

Функция $L = \varphi(h)$ с переменным шагом, поэтому табулирование осуществим путем вычисления значения L по формуле:

$L = 176 - 16h + 0.404h^2$ для всех значений h . Результаты вычислений приведены в табл. 5.4.

Таблица 5.4

	Значения переменных											
$h, м$	18	22	22.5	28.5	30.5	32	33	34	35	35.5	36	36.3
$L, лет$	18.896	19.536	20.525	48.149	63.821	77.696	87.956	99.024	110.9	117.141	123.584	127.547

Теперь можно вычислить абсолютную среднеквадратическую погрешность.

$L :=$	20	$\varphi :=$	18.896
	30		19.536
	40		20.525
	50		48.149
	60		63.821
	70		77.696
	80		87.956
	90		99.024
	100		110.9
	110		117.141
	120		123.584
	130		127.547

$D := L - \varphi \quad z := D \cdot D$

$E := \sqrt{z / 12} \quad E = 2.5$

Относительная максимальная погрешность:

$\delta = E / L_{\min} \cdot 100 \% = 2.5 / 20 \cdot 100 \% = 12.5 \%$.

Большая относительная погрешность объясняется либо неточностью исходных данных, либо неудачным выбором функции интерполяции. Наиболь-

шие погрешности интерполяции находятся в диапазоне L , равном 30—50 лет.

Повысим степень полинома на единицу, т. е. предположим, что $L(h) = a + bh + ch^2 + dh^3$.

Выберем следующие координаты функции $L = f(h)$:

[18, 20], [22.5, 40], [30.5, 60], [36, 120]. Тогда система уравнений будет иметь вид:

$$a + 18b + 18^2c + 18^3d = 20$$
$$a + 22.5b + 22.5^2c + 22.5^3d = 40$$
$$a + 30.5b + 30.5^2c + 30.5^3d = 60$$
$$a + 36b + 36^2c + 36^3d = 120$$

В данном случае матрица коэффициентов и вектор правых частей системы уравнений будут иметь вид:

$$M := \begin{vmatrix} 1 & 18 & 18^2 & 18^3 \\ 1 & 22.5 & 22.5^2 & 22.5^3 \\ 1 & 30.5 & 30.5^2 & 30.5^3 \\ 1 & 36 & 36^2 & 36^3 \end{vmatrix} \qquad B := \begin{vmatrix} 20 \\ 40 \\ 60 \\ 120 \end{vmatrix}$$

Тогда

$$M^{-1} \cdot B = \begin{vmatrix} -657.212 \\ 81.681 \\ -3.226 \\ 0.043 \end{vmatrix}$$

Функция интерполяции примет вид:

$$L(h) = -657.212 + 81.681h - 3.226h^2 + 0.043h^3.$$

Оценим погрешность интерполяции. Результаты табулирования интерполяционного полинома приведены в табл. 5.5.

Таблица 5.5

	Значения переменных											
$h, м$	18	22	22.5	28.5	30.5	32	33	34	35	35.5	36	36.3
$L, лет$	18.598	36.25	37.245	45.79	53.095	62.18	70.438	80.758	93.398	100.669	108.616	113.723

Вычисления погрешностей дают следующие результаты:

$$\varepsilon = 0.988, \quad \delta_{\max} = 4.94 \, \%.$$

Ввиду очевидности расчетов промежуточные результаты вычислений не приводятся.

Как видно из результатов расчетов, увеличение степени полинома привело к повышению точности интерполяции более чем в два раза.

5.2. Кусочно-линейная интерполяция

Кусочно-линейная интерполяция реализуется в Mathcad функцией `linterp`. Эта функция имеет вид

`linterp( $V_x$ ,  $V_y$ ,  $x$ ).`

Здесь:

- V_x — вектор значений аргументов x (узлов интерполяции);
- V_y — вектор значений функции $y(x)$;
- x — значение аргумента, для которого вычисляется значение функции. Аргументу x можно присвоить множество значений. Тогда функция `linterp` выдаст вектор значений функции $y(x)$.

Процедуры решения задачи интерполяции в Mathcad с помощью функции `linterp` рассмотрим на примере.

Пример 5.7

Пусть функция $y = f(x)$ задана в виде таблицы (табл. 5.6).

Таблица 5.6

	Значения переменных				
x	1	2	3	4	5
y	3	6	11	18	27

Необходимо решить задачу интерполяции для случаев всего диапазона изменения аргумента x и для диапазона $x = 3 \text{—} 4$ с шагом 0.2.

Приведем решение задачи в Mathcad с помощью функции `linterp`:

$$V_x := \begin{vmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{vmatrix} \quad V_y := \begin{vmatrix} 3 \\ 6 \\ 11 \\ 18 \\ 27 \end{vmatrix} \quad x := 1..5$$

$$\text{linterp}(V_x, V_y, x) = \begin{vmatrix} 3 \\ 6 \\ 11 \\ 18 \\ 27 \end{vmatrix}$$

$x := 3, 3.2..4$

$$\text{linterp}(V_x, V_y, x) = \begin{vmatrix} 11 \\ 12.4 \\ 13.8 \\ 15.2 \\ 16.6 \\ 18 \end{vmatrix}$$

Функция `linterp` не позволяет получить интерполяционную формулу, ее результатом является число или таблица $y = \varphi(x)$.

5.3. Сплайн-интерполяция

Интерполяция кубическими сплайнами осуществляется в Mathcad функцией:

`interp( $V_s, V_x, V_y, x$ ).`

Здесь:

□ $V_s := \text{cspline}(V_x, V_y)$;

□ V_x, V_y — векторы аргумента и функции $y = f(x)$;

□ x — значение аргумента.

Функцию `interp` можно представить также в виде:

`interp(cspline( $V_x, V_y$ ),  $V_x, V_y, x$ ).`

Решим задачу интерполяции с помощью функции `interp`. В качестве примера возьмем задачу из *разд. 5.2* (пример 5.7).

Процедуры решения задачи имеют вид:

$$V_x := \begin{vmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{vmatrix} \quad V_y := \begin{vmatrix} 3 \\ 6 \\ 11 \\ 18 \\ 27 \end{vmatrix} \quad V_s := \text{cspline}(V_x, V_y) \quad x := 3, 3.2..4$$

`interp(Vs, Vx, Vy, x)`

После нажатия клавиши <=> (равно) ответ получится в виде вектора:

$$\begin{bmatrix} 11 \\ 12.24 \\ 13.56 \\ 14.96 \\ 16.44 \\ 18 \end{bmatrix}$$

Процедуры при использовании функции `interp(cspline( $V_x$ ,  $V_y$ ),  $V_x$ ,  $V_y$ ,  $x$ )` имеют вид:

$$V_x := \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} \quad V_y := \begin{bmatrix} 3 \\ 6 \\ 11 \\ 18 \\ 27 \end{bmatrix} \quad x := 3, 3.2..4$$

$$\text{interp}(\text{cspline}(V_x, V_y), V_x, V_y, x) = \begin{bmatrix} 11 \\ 12.24 \\ 13.56 \\ 14.96 \\ 16.44 \\ 18 \end{bmatrix}$$

Сравнивая данные с результатами, приведенными в *разд. 5.2*, видим, что они отличаются друг от друга во всем диапазоне x , за исключением крайних точек.

Интерполяция сплайнами обладает высокой точностью результатов и по этому критерию превосходит кусочно-линейную интерполяцию.

Решим теперь с помощью сплайн-интерполяции задачу определения зависимости высоты сосны от ее возраста, сформулированную в *разд. 5.1* и решенную универсальным методом (пример 5.6).

Для сравнения результатов выберем точки интерполяции те же, что и в примере 5.6 для случая полинома третьей степени. Тогда исходная функция будет иметь вид, приведенный в табл. 5.7.

Таблица 5.7

	Значения переменных			
h , м	18	22.5	30.5	36
L , лет	20	40	60	120

Представим данные в виде векторов, а значение L в диапазоне 20—130 лет:

$$V_l := \begin{vmatrix} 20 \\ 40 \\ 60 \\ 120 \end{vmatrix} \quad V_h := \begin{vmatrix} 18 \\ 22.5 \\ 30.5 \\ 36 \end{vmatrix} \quad L := 20, 30 \dots 130$$

Тогда функция сплайн-интерполяции Mathcad будет иметь вид:

`interp(cspline( $V_l$ ,  $V_h$ ),  $V_l$ ,  $V_h$ ,  $L$  )`

После нажатия клавиши `<=>` (равно) получим решение в виде следующего вектора:

$$\begin{vmatrix} 18 \\ 19.566 \\ 22.5 \\ 26.309 \\ 30.5 \\ 34.578 \\ 38.05 \\ 40.422 \\ 41.2 \\ 39.891 \\ 36 \\ 29.034 \end{vmatrix}$$

Погрешности интерполяции, вычисленные по описанной выше методике, имеют значения:

□ абсолютная среднеквадратическая $E = 0.91$;

□ максимальная относительная $= 4.55 \%$.

Таким образом, погрешности сплайн-интерполяции оказались несколько ниже, чем полиномиальной.

5.4. Линейная аппроксимация

Сглаживание неточности исходных данных требует решения задач интерполяции приближенной в узлах (аппроксимации). В системе Mathcad имеется ряд функций, решающих задачи аппроксимации. Такими являются функции `slope` и `intercept`, позволяющие определить коэффициенты a и b линейной функции $y = ax + b$, заданной в виде таблицы. Указанные функции реализуются в Mathcad для определения коэффициентов a и b в виде следующих команд:

$a := \text{slope}(V_x, V_y)$
 $b := \text{intercept}(V_x, V_y).$

Здесь V_x, V_y — векторы аргументов и функции соответственно.

Пример 5.8

Пусть функция $y = f(x)$ задана в виде таблицы (табл. 5.8).

Таблица 5.8

	Значения переменных							
x	1.2	2.6	3.8	4.3	5	6.7	8	9.3
y	2	3.5	5.4	8.2	10.5	12	14.2	16.5

Необходимо установить, можно ли моделировать данный изучаемый процесс линейной функцией, если абсолютная среднеквадратичная погрешность не должна превышать 0.2. Рассмотрим процесс решения задачи.

Создание векторов V_x и V_y :

$V_x :=$

1.2

2.6

3.8

4.3

5

6.7

8

9.3

$V_y :=$

2

3.5

5.4

8.2

10.5

12

14.2

16.5

Определение коэффициентов линейной функции $y = a + bx$:

$a := \text{slope}(V_x, V_y)$
 $b := \text{intercept}(V_x, V_y)$
 $a = 1.862 \quad b = - 0.483$

Таким образом, линейная аппроксимирующая функция имеет вид:

$\varphi(x) = 1.862x - 0.483.$

Вычислим теперь абсолютную среднеквадратичную погрешность аппроксимации.

Представим функцию $\varphi(x)$ в виде вектора, вычислив ее значения для всех исходных значений аргумента x .

Исходные значения функции $y(x)$ и расчетные $\varphi(x)$ сведены в табл. 5.9.

Таблица 5.9

	Значения переменных							
$y(x)$	2	3.5	5.4	8.2	10.5	12	14.2	16.5
$\phi(x)$	1.751	4.358	6.593	7.524	8.827	11.992	14.413	16.834

Приведем процедуры определения погрешности E :

$$y := \begin{vmatrix} 2 \\ 3.5 \\ 5.4 \\ 8.2 \\ 10.5 \\ 12 \\ 14.2 \\ 16.5 \end{vmatrix} \quad \phi := \begin{vmatrix} 1.751 \\ 4.358 \\ 6.593 \\ 7.524 \\ 8.827 \\ 11.992 \\ 14.413 \\ 16.834 \end{vmatrix}$$

$$D := y - \phi$$

$$z := D \cdot D$$

$$E := \sqrt{z} / 8$$

$$E = 0.297$$

Из расчета следует, что погрешность интерполяции превышает допустимую, а полученная линейная функция не может быть моделью изучаемого явления.

5.5. Полиномиальная аппроксимация

Аппроксимация полиномами в Mathcad осуществляется с помощью функции `interp`. Функция имеет вид:

`interp( $V_s$ ,  $V_x$ ,  $V_y$ ,  $x$ ).`

Здесь:

- V_x , V_y — векторы соответственно аргумента x и функции $y(x)$;
- V_s — вектор, вычисляемый функциями `loess` или `regress`;
- x — аргумент, для которого вычисляется y .

Функция `regress` имеет вид:

`regress( $V_x$ ,  $V_y$ ,  $n$ )`, где n — степень полинома, рекомендуемая условием $n \leq 4$.

Функция loess имеет вид:
loess(V_x , V_y , span), где span — программа, которая подбирает диапазон значений x для интерполяции функции в этом диапазоне полиномом второй степени. По умолчанию span = 0.75.

Пример 5.9

Пусть функция $y = f(x)$ задана в виде таблицы (табл. 5.10).

Таблица 5.10

	Значения переменных							
x	0	1	2	3	4	5	6	7
y	−3.5	2.2	21	76	170	350	600	970

Необходимо найти значения функции в диапазоне аргументов $x = 2\text{--}4$ с шагом 0.2. При этом функция $\varphi(x)$ является полиномом третьей степени.

Приведем решение задачи с помощью функции interp.

$V_x :=$

0

1

2

3

4

5

6

7

$V_y :=$

−3.5

2.2

21

76

170

350

600

970

$x := 0,1..7$

$V_s := \text{regress}(V_x, V_y, 3)$

$\text{interp}(V_s, V_x, V_y, x) =$

После нажатия клавиши <=> (равно) последует ответ в виде вектора:

−3.838

2.672

22.012

72.976

174.36

344.96

603.571

968.988

Вычислим теперь значение функции в диапазоне 2—4 с шагом $h = 0.2$:

$x: = 2, 2.2..4$

$\text{interp}(V_s, V_x, V_y, x)$

После нажатия клавиши $\langle = \rangle$ (равно) ответ получим в виде вектора:

22.012
29.073
37.55
47.592
59.351
72.976
88.618
106.426
126.553
149.147
174.36

Вместо функции `regress` можно применить функцию `loess` в следующем виде:

$V_s = \text{loess}(V_x, V_y, 0.75)$.

При этом ответ не будет совпадать с предыдущим.

Недостаток функции `interp` в том, что откликом этой функции не является искомая функция $\phi(x)$ в виде формулы.

5.6. Аппроксимация линейной комбинацией функций

В ряде случаев трудно подобрать функцию аппроксимации вида полинома или простой нелинейной функции. Такие случаи возникают всякий раз, когда функция $y = f(x)$ имеет сложную форму. Mathcad имеет функцию, аппроксимирующую $y = f(x)$, заданную в виде таблицы линейной комбинацией функций:

$$y(x) = a_0 f_0(x) + a_1 f_1(x) + a_2 f_2(x) + \dots$$

Эта функция имеет вид:

$\text{linfit}(V_x, V_y, F)$.

Здесь:

- V_x, V_y — векторы аргумента и функции $y = f(x)$;
- F — вектор функций $f_i(x)$.

Пример 5.10

Пусть функция $y = f(x)$ задана в виде таблицы (табл. 5.11).

Таблица 5.11

	Значения переменных						
x	0	1	2	3	4	5	6
y	4	3.9	9	20	32	50	73

Пусть также аппроксимирующая функция имеет вид:

$$F(x) = a_0 x^2 + a_1 e^{-x} + a_2 \frac{1}{1+x}.$$

Необходимо найти коэффициенты a_0 , a_1 , a_2 и определить погрешность аппроксимации.

Вектор функции $F(x)$ представим в следующем виде:

$$F(x) := \begin{bmatrix} x^2 \\ e^{-x} \\ \frac{1}{1+x} \end{bmatrix}$$

Векторы V_x и V_y исходной функции запишем в виде:

$$V_x := \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \quad V_y := \begin{bmatrix} 4 \\ 3.9 \\ 9 \\ 20 \\ 32 \\ 50 \\ 73 \end{bmatrix}$$

Тогда функция `linfit` запишется в следующем виде:

$$s := \text{linfit}(V_x, V_y, F)$$

Введем: $s =$.

Получим решение в виде следующего вектора:

$$s = \begin{bmatrix} 1.999 \\ 0.331 \\ 3.653 \end{bmatrix}$$

Таким образом, аппроксимирующая функция будет иметь вид:

$$\varphi(x) = 2x^2 + 0.33 e^{-x} + 3.65 \frac{1}{1+x} .$$

Погрешность аппроксимации, вычисленная по приведенной выше методике, составляет $E = 0.219$.

5.7. Аппроксимация нелинейной функцией

Аппроксимация нелинейной функцией осуществляется в Mathcad с помощью функции `genfit`. Функция имеет вид:

`genfit( $V_x, V_y, V_q, F$ ).`

Здесь:

- V_x, V_y — векторы аргументов и функции соответственно;
- V_q — вектор начальных приближений для всех неизвестных;
- F — вектор, образованный аппроксимирующей функцией и ее частными производными по всем неизвестным параметрам.

Функция `genfit` возвращает вектор неизвестных, позволяющий представить функцию аппроксимации в виде формулы.

Рассмотрим технологию решения задачи аппроксимации с помощью функции `genfit` на примере.

Пример 5.11

Пусть функция $y = f(x)$, полученная экспериментально, представлена в табл. 5.12.

Таблица 5.12

	Значения переменных					
x	0	1	2	3	4	5
y	8	7	4	−1	−8	−17

В качестве аппроксимирующей выберем показательную функцию вида $\varphi(x) = a + bx^c$. Необходимо определить неизвестные a, b, c и вычислить погрешность аппроксимации.

Приведем процедуры решения задачи в системе Mathcad с помощью функции `genfit`. За начальные приближения выбраны $a_0 = 5, b_0 = -0.5, c_0 = 1.5$.

Обратим внимание на способ образования вектора F . При образовании вектора F неизвестные обозначаются одним символом с индексами, т. е. $a = a_0$,

$b = a_1$, $c = a_2$. Тогда функция и ее производные по неизвестным параметрам будут иметь вид:

$$f(x) = a_0 + a_1 x^{a_2}$$

$$\frac{d f(x)}{a_0} = 1$$

$$\frac{d f(x)}{a_1} = x^{a_2}$$

$$\frac{d f(x)}{a_2} = a_1 x^{a_2} \ln(x)$$

С учетом этих замечаний процедуры решения задачи аппроксимации будут иметь вид:

$$V_x := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} \quad V_y := \begin{pmatrix} 8 \\ 7 \\ 4 \\ -1 \\ -8 \\ -17 \end{pmatrix} \quad V_q := \begin{pmatrix} 5 \\ -0.5 \\ 1.5 \end{pmatrix} \quad F(x, a) := \begin{pmatrix} a_0 + a_1 \cdot x^{a_2} \\ 1 \\ x^{a_2} \\ a_1 \cdot x^{a_2} \ln(x) \end{pmatrix}$$

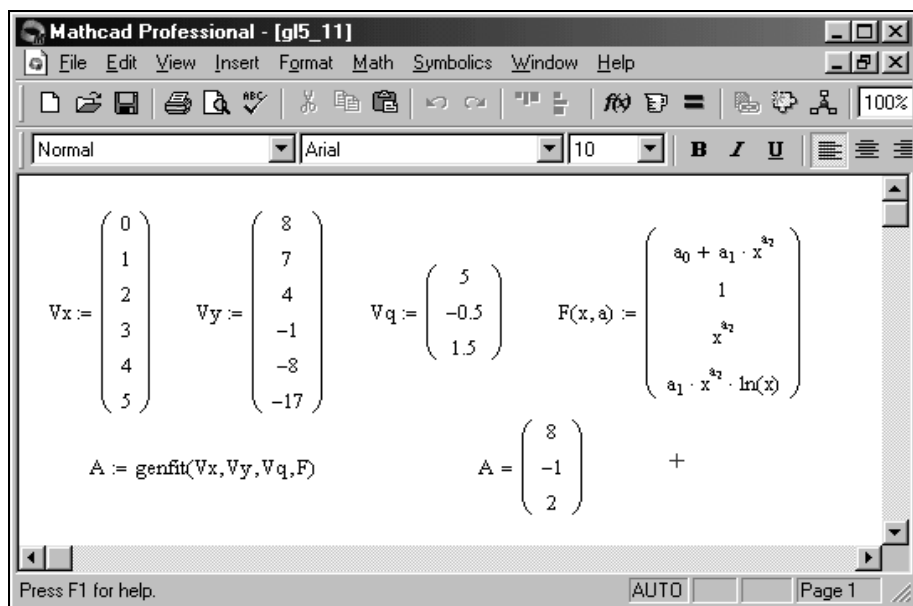


Рис. 5.7

$A := \text{genfit}(V_x, V_y, V_q, F)$

$$A = \begin{vmatrix} 7.999 \\ -0.999 \\ 2 \end{vmatrix}$$

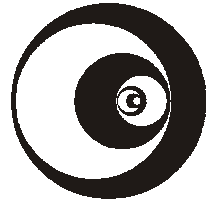
Процедуры решения на компьютере показаны на рис. 5.7.

В результате решения задачи найдены неизвестные a , b , c , которые имеют значения:

$a = 8$, $b = -1$, $c = 2$, т. е. функция аппроксимации $\varphi(x)$ имеет вид:

$$\varphi(x) = 8 - x^2.$$

Исходная таблица в настоящей задаче получена по формуле $y = 8 - x^2$. Сравнивая эту функцию с $\varphi(x)$ видим, что задача аппроксимации решена с нулевой погрешностью.



Глава 6

Компьютерные технологии решения задач интерполяции в системе Matlab

6.1. Начальные сведения о системе Matlab

Технология решения задач интерполяции состоит в выполнении на компьютере следующих действий:

- ☐ ввод исходных данных;
- ☐ визуализация исходных данных;
- ☐ выбор функции интерполяции;
- ☐ образование системы уравнений;
- ☐ решение системы уравнений;
- ☐ проверка правильности решения задачи;
- ☐ определение погрешности интерполяции.

В настоящем разделе приводятся сведения о системе Matlab лишь с позиции решения задач интерполяции и умения выполнять перечисленные выше действия.

Ввод исходных данных

Диалог с системой Matlab происходит посредством командного окна, которое становится доступным пользователю сразу же после запуска программы. Окно имеет меню, панель инструментов, полосы прокрутки, а также зону редактирования и просмотра (рис. 6.1).

Здесь же можно увидеть и строку ввода со знаком приглашения >>. Попробуем выполнить простейшие действия. Введем в строку ввода выражение.

```
>> x = 2 + 3
```

Для выполнения действия нажмем клавишу <Enter>. Результат виден на рис. 6.1.

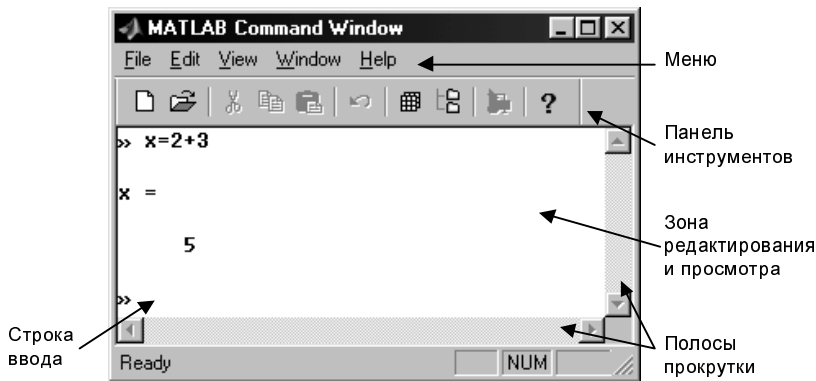


Рис. 6.1

Невозможность редактирования ранее введенной команды простой установкой курсора в нужную строку является одной из особенностей системы Matlab. Для того чтобы повторить ранее введенную команду, необходимо установить курсор в строку ввода и воспользоваться клавишами <↑> (стрелка вверх) и <↓> (стрелка вниз). Эти клавиши позволяют пролистать стек введенных ранее команд и оставить в строке именно ту команду, которая необходима. Команду можно выполнить сразу (нажав клавишу <Enter>) или после редактирования.

Пусть функция задана в виде таблицы (табл. 6.1).

Таблица 6.1

	Значения переменных			
x	1	2	3	4
y	6.2	4.1	1.9	0.6

Создадим и введем два вектора-строки x и y:

```
x = [1 2 3 4]
y = [6.2, 4.1, 1.9, 0.6 ]
```

На рис. 6.2 показаны векторы и отклики, полученные при нажатии клавиши <Enter>.

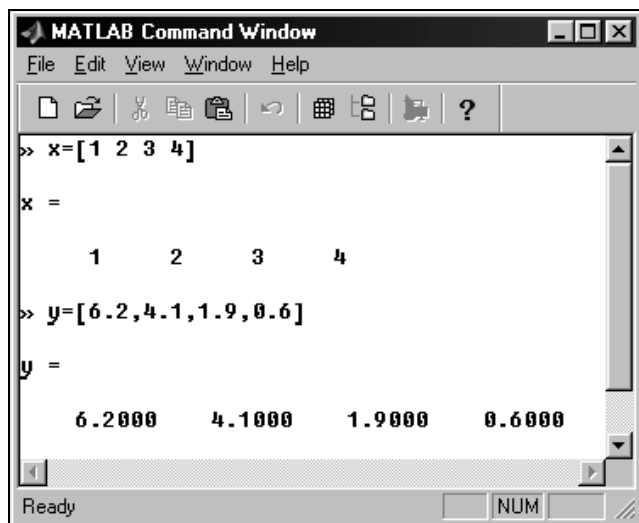


Рис. 6.2

При формировании вектора-строки в качестве разделителя можно использовать пробел или запятую (см. пример). При формировании вектора-столбца в качестве разделителя используется точка с запятой (;). При необходимости сформировать матрицу пользуются теми же правилами — строки отделяются друг от друга точкой с запятой.

При необходимости формирования вектора-строки постоянного шага используется конструкция:

$z = 0 : 2 : 10$

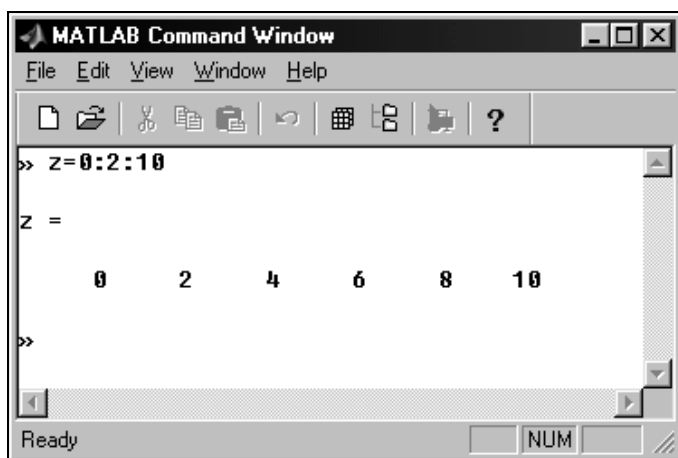


Рис. 6.3

В данном примере формируется вектор-строка от 0 до 10 с шагом 2 (рис. 6.3).

Подавить вывод результата выполнения команды можно, завершив ввод точкой с запятой. Это не повлияет на результат, но зато уменьшит поток лишней информации на экране.

Итак, мы имеем две вектор-строки (x и y), которые содержат интересующие нас данные. Убедиться в этом можно, набрав в строке ввода имя переменной (например, x) и нажав клавишу <Enter> (рис. 6.4).

Визуализация исходных данных

Система Matlab имеет большие возможности графического представления информации. Познакомимся только с теми из них, которые нам необходимы. Основной является функция `plot`, которая имеет вид:

`plot( $x_1, y_1, s_1, x_2, y_2, s_2, \dots, x_n, y_n, s_n$ ).`

Здесь:

- ☐ x_i — i -тый массив аргументов, заданный в виде вектора;
- ☐ y_i — массив значений функции для заданного массива аргументов;
- ☐ s_i — стиль графика для i -той функции.

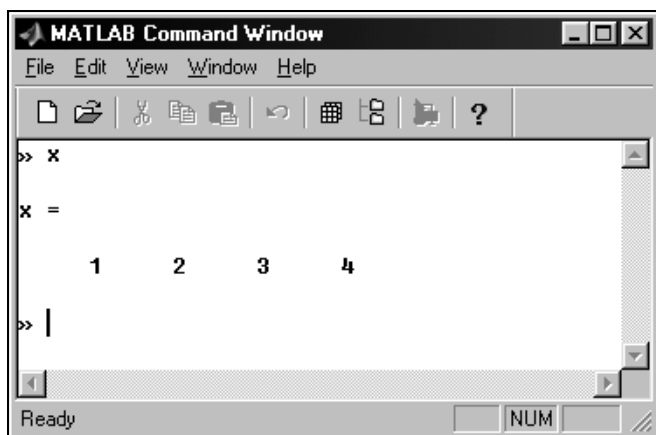


Рис. 6.4

Стиль можно не задавать. В этом случае проблему выбора стиля система Matlab решает самостоятельно.

При желании поработать со стилями их необходимо задавать в виде набора трех символов, заключенных в одиночные кавычки.

Например, `'-q'` (непрерывная зеленая линия с маркерами в виде точек).

Символы выбираются из таблицы стилей [15].

В качестве примера построим график функции, заданной табл. 6.1. Последовательность команд будет иметь вид:

```
x=[1 2 3 4];  
y=[6.2, 4.1, 1.9, 0.6];  
plot(x, y, '-q')
```

После нажатия клавиши <Enter> получим график функции.

При необходимости можно поместить в окне монитора заголовки и надписи, а также нанести координатную сетку, выполнив следующие команды (рис. 6.5):

```
» title('Наш график')  
» xlabel('Ось X')  
» ylabel('Ось Y')  
» grid on
```

В результате выполнения приведенной последовательности команд для функции, представленной табл. 6.1, получим график (рис. 6.6).

Функция $y(x)$ может быть задана в виде формулы, например $y = \sin(x)$ или $y = x^2 + 2 * x + 1$. В последнем случае перед знаком возведения в степень необходимо ставить точку, означающую, что данная операция (возведение в степень) является не матричной и функция $y(x)$ вычисляется для всех значений вектора x .

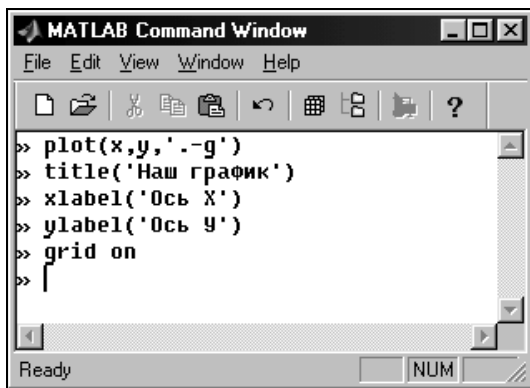


Рис. 6.5

Если на одном графике необходимо поместить две функции $y(x)$ и $z(x)$, то функция plot будет иметь вид:

plot(x, y, x, z).

Например:

```
x = [1, 2, 3, 4, 5];
```

```

y = cos(x);
z = exp(-x);
plot(x, y, x, z).

```

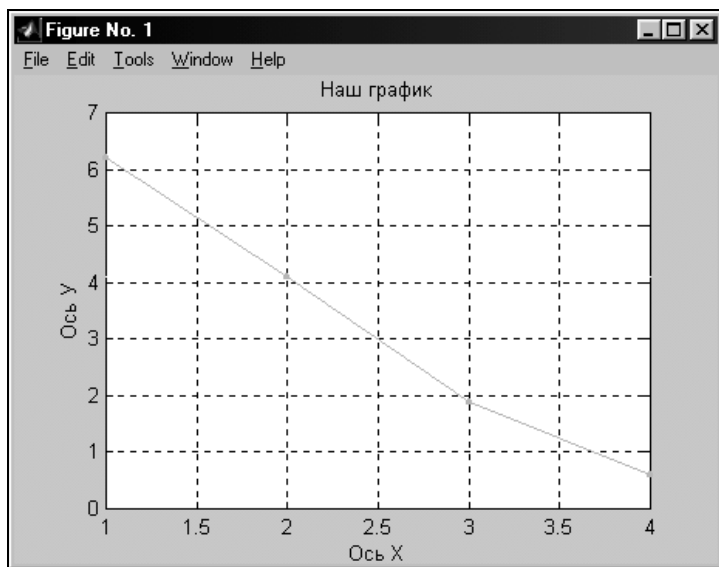


Рис. 6.6

Если функция представляется в символьном виде, то для построения ее графика используется функция `ezplot`:

```
f = '2 * x^2 + 3 * x + 1'
```

`ezplot(f, xn, xk)`, где x_n, x_k — диапазон изменения аргумента.

Откликом здесь является гладкая кривая с представлением на экране вида функции.

Вывести две кривые в одно окно можно также командой `hold on`. Пример приведен в разделе 6.2.

Выбор вида функции интерполяции

Для выбора вида функции интерполяции необходимо график, полученный по исходным данным, сравнить с графиками типичных функций. Это можно сделать либо пользуясь математическими справочниками, либо встроенными функциями системы Matlab.

Например, если необходимо построить график встроенной функции $y = \cos(x)$, то достаточно создать векторы-строки аргумента, представить функцию и воспользоваться функцией `plot`. Сделаем это следующим образом:

```
» x=0:0.1:4;  
» y=cos(x);  
» plot(x, y)
```

В данном случае в первых двух строках применен знак точка с запятой для подавления вывода информации. Обратите внимание, что в результате выполнения этой команды получим значение функции $\cos$ для каждого из аргументов.

По иному обстоит дело в случае применения не встроенной (пользовательской) функции. Тут не обойтись без программирования.

Читателю не следует пугаться слова "программирование". При элементарном усилии над собой этот "недостаток" Matlab можно превратить в его достоинство.

Работая с математическими пакетами, читатель привык к тому, что результаты работы в среде пакета можно сохранить в виде документа. При этом вся информация, введенная пользователем, а также результаты вычислений в текстовом и графическом виде будут сохранены. В Matlab при работе с командным окном такой возможности нет (можно лишь сохранить стек команд). Зато имеется намного более мощная технология, основанная на использовании М-файлов. М-файл — это текстовый файл (с расширением *.m), в котором помещен текст программы на языке программирования системы Matlab. Этот язык имеет очень много общего с другими языками программирования (например, с Бейсиком).

Рассмотрим небольшой пример.

Пусть необходимо сравнить график, полученный на основе экспериментальных данных, с графиками типичных функций. Среди встроенных такие функции отсутствуют, поэтому мы создадим их самостоятельно.

Начнем с дробно-линейной функции $y = x / (a + bx)$.

Дадим этой функции имя `drob_lin`.

Выполним в строке ввода команду `edit`. Перед нами возникнет окно встроенного редактора Matlab.

Наберем в окне следующий текст (рис. 6.7):

```
function ret = drob_lin(x)  
a = 1;  
b = 1;  
ret = x / (a + b * x);
```

Это не что иное, как программа вычисления нашей функции. Заметьте, что параметрам a и b присвоены значения, равные единице. Следующим обязательным действием является сохранение файла с именем, присвоенным функции (т. е. — `drob_lin`.)

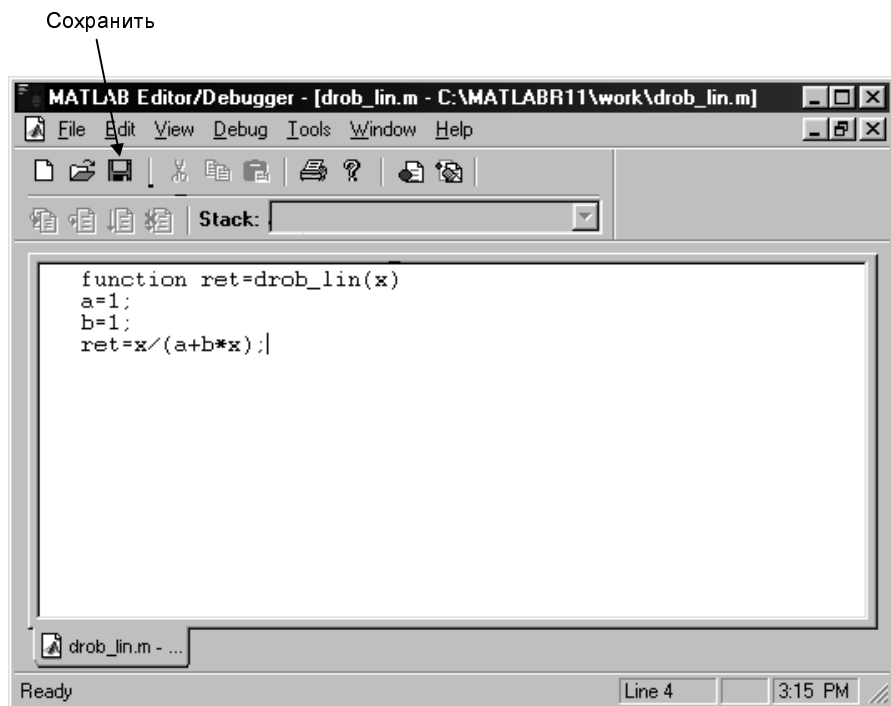


Рис. 6.7

Воспользуемся пунктом **Save** из меню **File** или нажмем соответствующую кнопку на панели инструментов. Посредством стандартной диалоговой панели укажем место сохранения (папку) и имя файла (drob_lin). Папка обязательно должна быть известна системе Matlab (например, Work).

Если вы желаете использовать свою папку, необходимо сообщить системе ее местонахождение. Это, например, можно сделать из командного окна с помощью меню **File | Path | Add to path**.

После сохранения файла окна редактора необходимо закрыть. Теперь можно обращаться к сохраненной функции непосредственно из командного окна. Пример приведен на рис. 6.8.

Для того чтобы изменить параметры a и b функции drob_lin, необходимо внести изменения непосредственно в тело функции, которую можно вызывать для редактирования командой:

```
edit drob_lin.
```

После появления редактора изменяем значения параметров и обязательно сохраняем измененный текст программы.

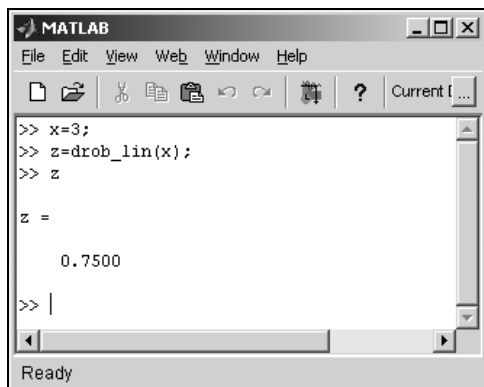


Рис. 6.8

Описанный способ изменения параметров функции приведен исключительно в учебных целях. Более рациональным является написание функции с входными параметрами (аргументами). В этом случае текст программы будет выглядеть следующим образом:

```
function ret = drob_lin(x, a, b)  
ret = x / (a + b * x)
```

Обращение к ней из командного окна будет выглядеть так:

```
>> x = 3;  
>> drob_lin(x, 2, 3)
```

или:

```
>> x = 3;  
>> a = 2;  
>> b = 3;  
>> drob_lin(x, a, b)
```

Ответом будет:

```
ans = 0.2727.
```

При необходимости обработки массива значений аргумента x функция `drob_lin` представляется в следующем виде:

```
function ret = drob_lin(x, a, b)  
ret = x. / (a + b. * x);
```

Точки перед знаками деления и умножения позволяют выполнить эти операции поэлементно, т. е. с каждым элементом массива чисел (вектора-строки). Теперь функция может принимать в качестве параметра как единственное число, так и массив чисел. Например:

```
>> x = [1 2 3 4 5];  
>> drob_lin(x, 2, 3)
```

После нажатия клавиши <Enter> последует ответ:

```
ans = 0.2000    0.2500    0.2727    0.2857    0.2941
```

На основе изложенного первые три этапа исследования экспериментальных данных (ввод, визуализация и выбор функции интерполяции) можно выполнить в командном окне так:

```
>> x = [1 2 3 4]
x =
     1     2     3     4
>> y = [6.2 4.1 1.9 0.6]
y =
     6.2000     4.1000     1.9000     0.6000
>> a = 1;
>> b = 1;
>> z = drob_lin(x, a, b)
z =
     0.5000     0.6667     0.7500     0.8000
>> plot(x, y, x, z)
```

Получим графики, представленные на рис. 6.9.

Из графиков видно, что дробно-линейная функция с параметрами $a = 1$ и $b = 1$ не может быть функцией интерполяции для данных, приведенных в табл. 6.1 (графики функций различны).

Запрограммировать другие типичные функции читатель теперь сможет самостоятельно.

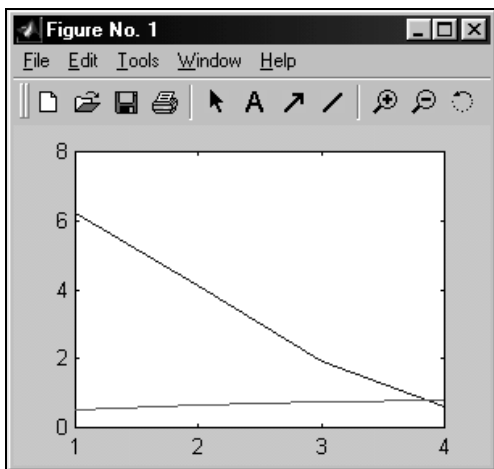


Рис. 6.9

6.2. Интерполяция точная в узлах. Универсальный метод

Универсальный метод требует решения систем алгебраических уравнений. Система Matlab позволяет это сделать несколькими способами.

Матричный способ

Для решения систем линейных уравнений матричным способом в Matlab имеются две следующие операции:

□ / — правое деление матриц;

□ \ — левое деление матриц.

Если названные операции применить к матрицам X и Y , то выражения $\text{inv}(X) * Y$ и $Y * \text{inv}(X)$ эквивалентны выражениям $X \backslash Y$ и X / Y соответственно. Функция обращения inv выполняет деление единичной матрицы на матрицу исходную.

Рассмотрим матричный способ на примере.

Пусть дана следующая система уравнений:

$$2x - 3y + 5z = 1$$

$$-x + y + 2z = 3$$

$$3x + 4y - z = -1$$

В матричном виде систему уравнений можно записать так: $XY = B$.

Здесь:

□ X — матрица коэффициентов;

□ Y — вектор неизвестных;

□ B — вектор правых частей системы уравнений.

Для нашего примера:

$$X = [2 \ -3 \ 5; \ -1 \ 1 \ 2; \ 3 \ 4 \ -1]$$

$$B = [1; \ 3; \ -1]$$

Чтобы найти неизвестный вектор Y , необходимо вычислить выражение $Y = X \backslash B$.

В командном окне Matlab решение выглядит следующим образом:

```
» X = [2 -3 5; -1 1 2; 3 4 -1]
```

```
X =
```

$$\begin{bmatrix} 2 & -3 & 5 \\ -1 & 1 & 2 \\ 3 & 4 & -1 \end{bmatrix}$$

```
» B = [1; 3; -1]
```

```

B =
    1
    3
   -1
» X \ B
ans =
   -0.7794
    0.5441
    0.8382

```

Решение систем уравнений с помощью функции solve

При наличии на вашем компьютере пакета расширения Symbolic Math Toolbox для символьных вычислений решение систем алгебраических уравнений можно получить, используя функцию solve

Функция solve позволяет решать уравнения с одним неизвестным, а также системы уравнений. В случае решения линейных и нелинейных уравнений с одним неизвестным функция записывается в следующем виде:

`solve(f(x), x).`

Здесь:

- x — неизвестные;
- $f(x)$ — алгебраическое или трансцендентное уравнение, записанное в произвольном виде (если знак равенства отсутствует, то считается, что $f(x) = 0$).

Если решается система уравнений, то функция solve представляется в одной из следующих форм:

- `solve('f1', 'f2', , 'fn', x1, x2, , xn);`
- `solve('f1', 'f2', , 'fn').`

В этих функциях приняты обозначения:

- x_i — искомые неизвестные;
- $'f_i'$ — уравнения системы, каждое из которых берется в одинарные кавычки и отделяется от предыдущего запятой;
- $i = 1, 2, , n$.

Прежде чем записать функцию solve в случае решения систем уравнений, необходимо выделить искомые неизвестные в группу символьных объектов. Создается группа символьных объектов с помощью функции `syms`, которая вводится перед функцией solve.

Следует иметь в виду, что решения системы уравнений с помощью функции `solve` не может быть, если система не имеет аналитического решения. Это часто случается при решении системы нелинейных уравнений.

В качестве примера решим систему уравнений матричным способом. Процедуры решения в системе Matlab:

```
sums x y z
```

```
>> Y = solve('2 * x - 3 * y + 5 * z = 1', '-x + y + 2 * z = 3',  
'3 * x + 4 * y - z = -1')
```

После нажатия клавиши <Enter> получим:

$Y =$

$x: [1 \times 1 \text{ sym}]$

$y: [1 \times 1 \text{ sym}]$

$z: [1 \times 1 \text{ sym}]$

Для получения значений неизвестных необходимо воспользоваться конструкцией: $Y.k$ (где k — искомое неизвестное). Для нашего уравнения процесс будет выглядеть следующим образом:

```
>> Y.x
```

$ans =$

$-53 / 68$

```
>> Y.y
```

$ans =$

$37 / 68$

```
>> Y.z
```

$ans =$

$57 / 68$

Можно также воспользоваться функцией `vpa(Y.k, n)`, где n — число значащих цифр ответа.

Функция `vpa` (арифметика произвольной точности) позволяет получить решение с заданной точностью. В нашем случае при $n = 4$ решение будет иметь вид:

```
>> vpa(Y.x, 4)
```

$ans =$

$- .7794$

```
>> vpa(Y.y, 4)
```

$ans =$

$.5441$

```
>> vpa(Y.z, 4)
ans =
    .8382
```

Проверка правильности решения задачи интерполяции

Проверку правильности решения задачи можно осуществить тремя способами:

- методом подстановки;
- табулированием функции аппроксимации и сравнением ее результатов с исходными данными;
- использованием графических образов.

Для проверки правильности решения методом подстановки используется функция `subs`, имеющая вид:

`subs(z, old, new)`.

Здесь:

- `z` — символьное выражение (результат решения системы уравнений);
- `old` — список заменяемых переменных;
- `new` — список новых переменных.

Метод подстановки рассмотрим на примере.

Пусть необходимо решить систему уравнений:

$$x + 3y = 2$$

$$-2x + y = 7$$

Будем решать систему уравнений и проверять правильность ее решения с помощью функций `solve` и `subs`:

```
s=`x + 3 * y = 2, -2 * x + y = 7`;
```

```
s1=`x + 3 * y = 2`;
```

```
s2=`-2 * x + y = 7`;
```

```
[a, b] = solve(s)
```

Нажимаем клавишу <Enter>

```
a =
    -19 / 17
```

```
b =
    11 / 17
```

```
subs(s1, {x, y}, {a, b})
```

Нажимаем клавишу <Enter>

ans =

$$2 = 2$$

`subs(s2, {x, y}, {a, b})`

Нажимаем клавишу <Enter>

ans =

$$7 = 7$$

В выражении функции `subs` вместо фигурных скобок можно использовать квадратные.

Из ответа видно, что система уравнений имеет решение:

$x = -\frac{19}{7}$, $y = \frac{11}{7}$ и что решение верно, т. к. подстановка значений x и y в исходные уравнения приводит к тождеству $\{2 = 2, 7 = 7\}$.

Проверка правильности решения задачи интерполяции может быть осуществлена путем табулирования функции $\varphi(x)$ и сравнения ее с исходными данными. В среде Matlab табулирование осуществляется также с помощью функции `subs`.

Пусть необходимо протабулировать функцию интерполяции:

$$y = 2.3x^2 + 1.35x + 3.6 \quad \text{при } x = 1, 2, 3, 4, 5.$$

Последовательность команд в системе Matlab будет иметь вид:

```
f = 2.3 * x^2 + 1.35 * x + 3.6;
```

```
x1 = [1 2 3 4 5];
```

```
subs(f, x, x1)
```

После ее выполнения имеем:

ans =

7.2500 15.5000 28.3500 45.8000 67.8500

Сравнивая эти данные с исходными, можно судить о правильности решения задачи интерполяции.

Наиболее наглядным может оказаться проверка правильности решения путем использования графических образов. Для этого служит функция `plot`, о которой упоминалось ранее. Рассмотрим ее использование на примере.

Пусть функция $f(x)$ задана в виде таблицы (табл. 6.2).

Таблица 6.2

	Значения переменных					
x	1	2	3	4	5	6
y	2.5	3.7	6.2	4.1	2.9	2

Предположим, что функцией интерполяции является полином:

$$y = -0.45x^2 + 3x + 0.05.$$

Построим графики исходной функции и функции интерполяции.

Последовательность команд в системе Matlab имеет вид:

```
x = [1 2 3 4 5 6];
y = [2.5 3.7 6.2 4.1 2.9 2];
plot(x, y)
```

После исполнения функции plot(x, y) надо нажать клавишу <Enter>.

hold on

```
f = '-0.45 * x ^ 2 + 3 * x + 0.05';
ezplot(f, 1, 6)
```

В результате исполнения функции ezplot на экране монитора получим графики двух функций (рис. 6.10). Видно, что решение ошибочно, т. к. графики сильно отличаются друг от друга.

Наиболее вероятно, что полином второй степени плохо описывает функцию, заданную таблично.

Определение погрешности интерполяции

Погрешность интерполяционной формулы будем оценивать абсолютной среднеквадратической погрешностью, которая имеет вид:

$$\varepsilon = \frac{\sqrt{\sum_{i=1}^n \Delta_i^2}}{n}.$$

Здесь:

- $\Delta_i = y(x_i) - \varphi(x_i)$;
- $y(x_i)$ — функция, заданная в виде таблицы;
- $\varphi(x_i)$ — функция интерполяции;
- n — число узлов исходной функции, заданной таблично.

В большинстве случаев точность оценивается максимальной относительной погрешностью δ :

$$\delta = \frac{\varepsilon}{y_{\min}} \cdot 100\%, \text{ где } y_{\min} \text{ — минимальное значение функции.}$$

Процедуры вычисления погрешностей ε и δ в системе Matlab осуществляются с помощью ряда функций, среди которых еще неизвестные нам: sum, min, length.

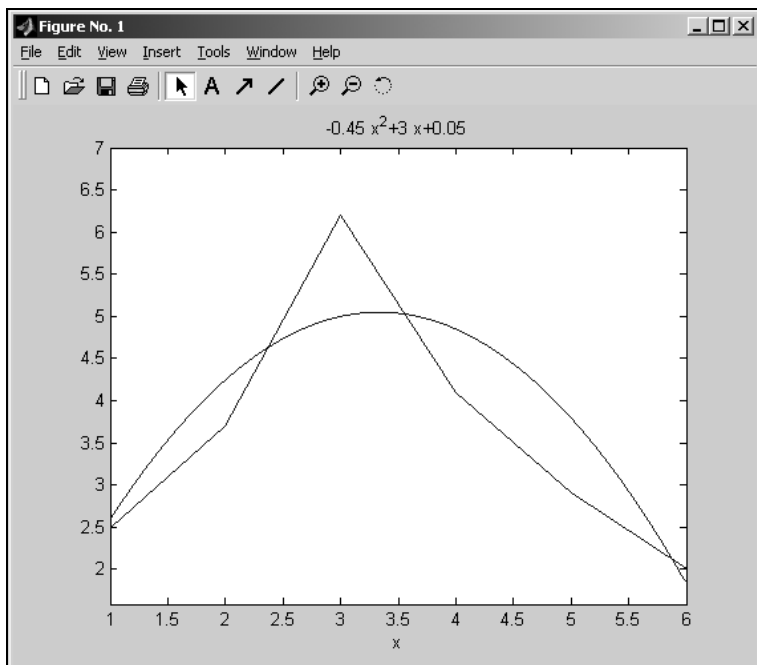


Рис. 6.10

Функция $\text{sum}(A)$ возвращает сумму элементов вектора A . Если же A — матрица, то возвращается вектор-строка, содержащая сумму элементов каждого столбца.

Функция $\text{min}(A)$ возвращает минимальный элемент вектора A . Если же A — матрица, то возвращается вектор, содержащий минимальные элементы каждого столбца.

Функция $\text{length}(A)$ возвращает количество элементов вектора A .

С учетом этих замечаний, процедура вычисления погрешностей интерполяции функции, заданной табл. 6.2, будет иметь вид:

```
x1 = [1 2 3 4 5 6];
f = '-0.45 * x^2 + 3 * x + 0.05';
y1 = [2.5 3.7 6.2 4.1 2.9 2]
```

Нажимаем клавишу <Enter>

```
y1 =
    2.5000    3.7000    6.2000    4.1000    2.9000    2.0000
y2 = subs(f, x, x1)
y2 =
    2.6000    4.2500    5.0000    4.8500    3.8000    1.8500
```

```
e = sum((y1 - y2) . ^ 2)
```

```
e =
```

```
3.1475
```

```
E = sqrt(e) / length(y1)
```

```
E =
```

```
0.2957
```

```
ymin = min(y1)
```

```
ymin =
```

```
2
```

```
↔ = E / ymin * 100
```

```
↔ =
```

```
14.7843
```

Убедиться в правильности решения можно путем сравнения значений ошибок, полученных здесь, и с помощью системы Maple (пример 4.2). Результаты полностью совпадают.

Процедуры решения задачи интерполяции универсальным методом в системе Matlab покажем на примере.

Пример 6.1

Пусть функция $y = f(x)$ задана в виде таблицы (см. табл. 4.2), представляющей собой зависимость давления насыщенного водяного пара P от температуры t .

Необходимо найти зависимость $P = \varphi(t)$ методом интерполяции точной в узлах при ограниченном их числе.

Из табл. 4.3 видно, что табличные разности третьего порядка почти одинаковы. Это позволяет нам предположить, что интерполяционный многочлен является многочленом третьей степени:

$$P(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3.$$

Для определения коэффициентов многочлена составим систему уравнений. Выберем из таблицы координаты:

$\{0, 4.58\}$, $\{10, 9.21\}$, $\{25, 23.76\}$, $\{35, 42.18\}$.

Тогда система уравнений будет иметь вид:

$$a_0 = 4.58$$

$$a_0 + a_1 \cdot 10 + a_2 \cdot 10^2 + a_3 \cdot 10^3 = 9.21$$

$$a_0 + a_1 \cdot 25 + a_2 \cdot 25^2 + a_3 \cdot 25^3 = 23.76$$

$$a_0 + a_1 \cdot 35 + a_2 \cdot 35^2 + a_3 \cdot 35^3 = 42.18$$

Рассмотрим решение системы уравнений в среде Matlab.

Создадим группу символьных объектов:

```
syms a0 a1 a2 a3 b0 b1 b2 b3 t
```

Введем уравнения системы:

```
y1 = 'a0 = 4.58'
```

```
y2 = 'a0 + a1 * 10 + a2 * 10^2 + a3 * 10^3 = 9.21'
```

```
y3 = 'a0 + a1 * 25 + a2 * 25^2 + a3 * 25^3 = 23.76'
```

```
y4 = 'a0 + a1 * 35 + a2 * 35^2 + a3 * 35^3 = 42.18'
```

Команда решения системы:

```
[b0, b1, b2, b3]=solve(y1, y2, y3, y4)
```

После нажатия клавиши <Enter> получим ответ в виде:

$b0 =$

4.58

$b1 =$

0.36448

$b2 =$

0.00568

$b3 =$

0.000417

Проверка решения:

```
subs(y1, {a0}, {b0})
```

```
subs(y2, {a0, a1, a2, a3}, {b0, b1, b2, b3})
```

```
subs(y3, {a0, a1, a2, a3}, {b0, b1, b2, b3})
```

```
subs(y4, {a0, a1, a2, a3}, {b0, b1, b2, b3})
```

После нажатия клавиши <Enter> в каждой строке оператора subs получим ответы:

$4.58 = 4.58$

$9.21 = 9.21$

$23.76 = 23.76$

$42.18 = 42.18$

Решение системы уравнений верно.

Таким образом, функция интерполяции имеет вид:

$\varphi(t) = 0.000417t^3 + 0.00568t^2 + 0.36448t + 4.58.$

Вычислим значения функции в узлах интерполяции, решив задачу табулирования функции $y(x)$:

```
P1 = subs(4.58 + 0.36448 * t + 0.00568 * t^2 + 0.000417 * t^3, t,  
[0, 5, 10, 15, 20, 25, 30, 35])
```

После нажатия клавиши <Enter> получим ответ:

$P_1 =$

4.58 6.5965 9.2098 12.7326 17.4776 23.7576 31.8854 42.1737

Исходные данные функции $y(x)$ представим в виде вектора:

$P = [4.58, 6.54, 9.21, 12.79, 17.54, 23.76, 31.82, 42.18]$

Вычислим значения погрешностей интерполяции:

$e = \text{sum}(p - p_1).^2);$

$E = \text{sqrt}(e) / \text{length}(P)$

$E =$

0.0152.

$y_{\min} = \min(P)$

$\leftrightarrow = E / y_{\min} * 100$

$\leftrightarrow =$

0.3310.

Небольшая абсолютная и относительная погрешности (примерно 0.33 %) свидетельствуют о том, что полученная в результате интерполяции функция хорошо отражает зависимость давления насыщенного водяного пара от температуры. Результаты интерполяции, полученные в среде Maple (см. главу 4), незначительно отличаются от результатов решения задачи в среде Matlab. Отличие результатов можно объяснить различной точностью решения в этих системах.

6.3. Интерполяция нелинейными функциями

Интерполяция точная в узлах нелинейными функциями сводится к решению системы нелинейных уравнений. В системе Matlab решение осуществляется с помощью функции `fsolve`, которая имеет вид:

`fsolve('file',  $x_0$ ).`

Здесь:

□ x_0 — вектор начальных приближений;

□ 'file' — имя М-файла с расширением *.m, содержащего систему уравнений.

Пример 6.2

Пусть функция $f(x)$ представлена в виде таблицы (табл. 6.3).

Таблица 6.3

	Значения переменных					
x	1	2	3	4	5	6
y	6.5	20	53.5	167	473	1470

Необходимо представить функцию в виде формулы. Следуя *разд. 2.3 главы 2*, выберем функцию интерполяции в виде показательной функции:

$$y = ab^x + c.$$

Так как функция содержит три неизвестных, то составим систему трех нелинейных уравнений, выбрав в качестве аргументов узлы интерполяции $x = 1, 4, 6$. Тогда система уравнений будет иметь вид:

$$ab + c = 6.5$$

$$ab^4 + c = 167$$

$$ab^6 + c = 1470.$$

Представим эту систему уравнений в виде функции пользователя с именем файла `myfun.m` и сохраним его с помощью команды **Save** меню **File** или соответствующей кнопки панели инструментов.

Содержимое файла может иметь вид:

```
Function F = myfun(x)
```

```
F = [x(1) * x(2) + x(3) - 6.5;      x(1) * x(2)^4 + x(3) - 167;  
x(1) * x(2)^6 + x(3) - 1470];
```

Теперь воспользуемся функцией `fsolve`:

```
x0 = [1; 1; 1];
```

```
x = fsolve('myfun', x0)
```

После нажатия клавиши <Enter> получим решение:

```
x =
```

```
2.1512
```

```
2.9678
```

```
0.1157
```

Тогда функция интерполяции будет иметь вид:

$$\varphi(x) = 2.1512 \cdot 2.9678^x + 0.1157.$$

6.4. Сплайн-интерполяция

Интерполяция кубическими сплайнами в среде Matlab осуществляется с помощью функции `spline`, которая имеет вид:

$$Y_i = \text{spline}(x, y, x_i).$$

Здесь:

- x — вектор узлов интерполяции;
- y — вектор значений функции в узлах интерполяции;
- x_i — вектор аргументов функции $y(x)$ из области ее определения, задаваемый пользователем.

Вместо вектора y функция $y = f(x)$ может быть задана в виде формулы.

Функция `spline` не позволяет получить функцию интерполяции в виде формулы. В этом ее существенный недостаток.

Пример 6.3

Пусть в результате опыта получены данные, приведенные в табл. 6.4.

Таблица 6.4

	Значения переменных							
x	1	4	7	9	13	18	21	27
y	0.4	1.55	2.9	3.9	5.86	8.3	9.8	12.8

Необходимо найти значения функции при $x = 3, 6, 10, 14, 20$.

Процедуры интерполяции будут иметь вид:

```
x = [1 4 7 9 13 18 21 27];
```

```
y = [0.4, 1.55, 2.9, 3.9, 5.86, 8.3, 9.8, 12.8];
```

```
xi = [3 6 10 14 20];
```

```
yi = spline(x, y, xi)
```

После нажатия клавиши <Enter> на экране увидим ответ:

```
y_i = 1.1458    2.4261    4.3980    6.3444    9.2975
```

6.5. Интерполяция точная в узлах (функция `interp1`)

Функция `interp1` наиболее универсальна, она позволяет решать задачи интерполяции несколькими методами. Функция имеет вид:

```
y_i = interp1(x, y, x_i, метод).
```

Здесь:

- x, y — векторы узлов и значений функции интерполяции;
- x_i — вектор значений аргументов, задаваемый пользователем;
- *метод* — аргумент, позволяющий пользователю выбрать метод интерполяции.

Методами интерполяции в Matlab являются:

- 'nearest' — ступенчатая;
- 'linear' — линейная;
- 'cubic' — кубическая;
- 'spline' — кубическими сплайнами.

Если метод не указан, то реализуется линейная интерполяция.

Пример 6.4

Пусть функция $y = f(x)$ задана в виде таблицы (табл. 6.5).

Таблица 6.5

	Значения переменных						
x	2	3	8	12	20	25	36
y	1	2.5	4.6	6.8	5.1	3.9	3

Необходимо определить значения функции при значениях аргументов $x = 4, 6, 15, 30$.

Приведем процедуры решения задачи на ПК.

```
x = [2, 3, 8, 12, 20, 25, ,36];
```

```
y = [1, 2.5, 4.6, 6.8, 5.1, 3.9, 3];
```

```
xi = [4, 6, 15, 30];
```

```
yi = interp1(x, y, xi) — линейная интерполяция
```

```
yi = interp1(x, y, xi, 'linear') — линейная интерполяция
```

```
yi = interp1(x, y, xi, 'nearest') — ступенчатая интерполяция
```

```
yi = interp1(x, y, xi, 'cubic') — кубическая интерполяция
```

```
yi = interp1(x, y, xi, 'spline') — интерполяция кубическими сплайнами.
```

В результате решения задачи получим следующие ответы:

□ линейная интерполяция: 2.92 3.76 6.1625 3.4909;

□ ступенчатая интерполяция: 2.5 4.6 6.8 3.9;

□ кубическая и кубическая сплайнами: 3.4133 4.16 6.9772 3.5263.

Из результатов решения видно, что ответы зависят от метода интерполяции. Какой же из них предпочтительней? Ответ на этот вопрос можно дать лишь после анализа погрешностей интерполяции.

Функция `interp1` не позволяет получить функцию интерполяции $\varphi(x)$ в виде формулы. В этом ее существенный недостаток.

6.6. Интерполяция приближенная в узлах (аппроксимация)

В предыдущей главе обсуждалось, что погрешность, возникающая при интерполяции точной в узлах, происходит за счет неточности исходных данных. В этом случае имеет смысл осуществить сглаживание неточностей, используя функции аппроксимации. Одной из таких функций в Matlab является функция `lsqcurvefit`, которая имеет вид:

`coef = lsqcurvefit(f, x0, a, b).`

Здесь:

- a, b — экспериментальные данные (векторы узлов интерполяции и значений функции);
- x_0 — стартовое значение параметров функции;
- f — функция аппроксимации, задаваемая пользователем.

Решим задачу аппроксимации полиномом третьей степени, используя функцию `lsqcurvefit`. В качестве примера найдем зависимость давления насыщенного водяного пара от температуры, которая уже получена в *разд. 4.1 и 4.2* методами точными в узлах интерполяции.

Приведем процедуры решения задачи аппроксимации и вычисления погрешностей с применением функции `lsqcurvefit` в среде Matlab:

```
a = [0, 5, 10, 15, 20, 25, 30, 35];
b = [4.58, 6.54, 9.21, 12.79, 17.54, 23.76, 31.82, 42.18];
x0 = [1, 1, 1, 1];
f = inline('x(4) * a.^3 + x(3) * a.^2 + x(2) * a + x(1)', 'x', 'a');
coef = lsqcurvefit(f, x0, a, b);
x = [0, 5, 10, 15, 20, 25, 30, 35];
y = subs('4.5693 + 0.3638 * t + 0.006 * t^2 + 0.0004 * t^3', 't', x)
f1 = y;
f2 = b;
e = f1 - f2;
s = sum(e.^2);
E = sqrt(s) / length(f2)
```

Ответ: $E = 0.0753$

$\Leftrightarrow = E / \min(f2) * 100$

Ответ: $\delta = 1.6432$.

6.7. Полиномиальная аппроксимация

Аппроксимация полиномами в среде Matlab осуществляется с помощью функции `polyfit`, которая имеет вид:

`polyfit(x, y, n)`.

Здесь:

- x — вектор узлов интерполяции;
- y — вектор значений функции в узлах интерполяции;
- n — степень полинома.

Откликом при реализации функции `polyfit` является вектор коэффициентов:

$a, b, c, d, \dots$ полинома $ax^n + bx^{n-1} + cx^{n-2} + dx^{n-3} + \dots$

Функция $y(x)$ может быть также представлена в аналитическом виде.

Пример 6.5

Пусть функция задана в виде таблицы (см. табл. 6.4).

Необходимо найти функцию интерполяции, представляющую собой многочлен третьей степени.

Процедуры интерполяции в Matlab имеют вид:

```
x = [1 4 7 9 13 18 21 27];
```

```
y = [0.4, 1.55, 2.9, 3.9, 5.86, 8.3, 9.8, 12.8];
```

```
p = polyfit(x, y, 3)
```

После нажатия клавиши <Enter> получим ответ в следующем виде:

```
-0.0001    0.0067    0.3814   -0.0227
```

Тогда функцией интерполяции будет следующий полином третьей степени:

$\varphi(x) = -0.0001x^3 + 0.0067x^2 + 0.3814x - 0.0227$.

В системе Matlab имеется функция вычисления математического выражения при заданных значениях аргументов. Функция имеет вид:

`polyval(p, x)`.

Здесь:

- p — вычисляемая функция;
- x — вектор аргументов функции.

Воспользуемся этой функцией для проверки достоверности результатов аппроксимации.

Введем функцию: `f = polyval(p, x)` и нажмем клавишу <Enter>.

Откликом будет следующее решение:

```
f = 0.3654    1.6031    2.9356    3.8670    5.8066    8.3148    9.8294   12.7882
```

Сравнивая это решение с вектором y исходных данных, видим, что они отличаются несущественно, а значит интерполяционный полином $\varphi(x)$ третьей степени хорошо отображает исходную функцию.

Вычислим теперь абсолютную среднеквадратическую погрешность аппроксимации по формуле:

$$\varepsilon = \frac{\sqrt{\sum_{i=1}^n \Delta_i^2}}{n}.$$

В нашем случае:

$$\square \Delta_i = y_i - \varphi(x_i) = y_i - f_i;$$

$$\square n = 8.$$

Тогда вычислительные процедуры в системе Matlab будут иметь вид:

$$z = y - f;$$

$$w = z.^* z;$$

$$R = \text{sum}(w);$$

$$E = \text{sqrt}(R)/8$$

После нажатия клавиши <Enter> получим ответ: $E = 0.0128$.

Система Matlab позволяет обоснованно выбрать степень полинома при полиномиальной интерполяции путем вычисления табличных разностей. Для этой цели служит функция `diff`. Эта функция имеет вид:

`diff(v, n).`

Здесь:

$\square v$ — вектор функции $y(x)$;

$\square n$ — порядок конечных разностей.

Полиномиальная аппроксимация имеет смысл лишь тогда, когда n -ая конечная разность функции $y(x)$ при постоянном шаге изменения аргумента x является постоянной. При этом значение n является степенью полинома. Если это условие не выполняется, то многочлен степени n либо не может быть функцией интерполяции, либо является основным источником погрешностей.

Пример 6.6

Пусть данные эксперимента сведены в табл. 6.6.

Таблица 6.6

	Значения переменных									
x	1	2	3	4	5	6	7	8	9	10
y	18	52.2	121.4	239.4	420	677	1024.2	1475.4	2044.4	2745

Необходимо решить задачу полиномиальной аппроксимации

Вначале определим степень полинома, воспользовавшись функцией `diff`. Эти процедуры в системе Matlab будут иметь вид:

```
x = [1 2 3 4 5 6 7 8 9 10];  
y = [18, 52.2, 121.4, 239.4, 420, 677, 1024.2, 1475.4, 2044.4,  
2745];  
diff(y, 1)  
ans =  
34.2 69.2 118 180.6 257 347.2 451.2 569 700.6  
diff(y, 2)  
ans =  
35 48.8 62.6 76.4 90.2 104 117.8 131.6  
diff(y, 3)  
ans =  
13.8 13.8 13.8 13.8 13.8 13.8 13.8
```

Табличные разности третьего порядка одинаковы, поэтому интерполяционный полином будет иметь вид: $\varphi(x) = ax^3 + bx^2 + cx + d$.

Определим теперь значения коэффициентов a , b , c , d , воспользовавшись функцией `polyfit`.

Так как векторы аргумента x и функции y уже введены, то достаточно ввести функцию `polyfit` и нажать клавишу <Enter>:

```
p = polyfit(x, y, 3)  
p =
```

```
2.3 3.7 7 5
```

Подставляя полученные значения в функцию $y(x)$, получим:

$$\varphi(x) = 2.3x^3 + 3.7x^2 + 7x + 5.$$

Проверим правильность решения задачи, воспользовавшись функцией `polyval`:

```
f = polyval(p, x)
```

После нажатия клавиши <Enter> получим:

```
f =  
18 52.2 121.4 239.4 420 677 1024.2 1475.4 2044.4 2745
```

Сравнивая значения f с исходными значениями y , видим, что они совпадают. Задача решена с нулевой погрешностью.

Процедуры определения степени полинома n опущены ввиду их очевидности.

6.8. Интерполяция кубическими полиномами

Интерполяция кубическими полиномами реализуется с помощью функции `icubic`, имеющей вид:

$$y_i = \text{icubic}(x, y, x_i).$$

Здесь:

- x, y — аргумент и функция, заданные в виде векторов (или x — в виде вектора, а y — в виде формулы);
- x_i — вектор значений аргумента, для которых вычисляется значение функции (x_i должно быть в диапазоне значений x).

Пример 6.7

Пусть функция задана в виде таблицы (табл. 6.7).

Таблица 6.7

	Значения переменных						
x	1	3	6	9	12	15	18
y	1	2	5	8	10	14	19

Необходимо найти значение функции при значениях аргумента $x_i = 2, 5, 7, 13, 17$.

Процедуры решения задачи в среде Matlab будут иметь вид:

```
x = [1 3 6 9 12 15 18];
```

```
y = [1 2 5 8 10 14 19];
```

```
xi = [2 5 7 13 17];
```

```
yi = icubic(x, y, xi)
```

После нажатия клавиши <Enter> получаем решение в следующем виде:

$y_i =$

1.1246 3.0928 5.3590 10.7824 17.1211

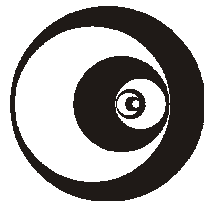
Функция `icubic` отличается от функции `interp1` с опциями 'cubic' и 'spline' методами интерполяции. В этом можно убедиться по результатам решения одних и тех же задач.

При решении предыдущей задачи с помощью функции `interp1` с опцией 'cubic' получаем следующий ответ:

$y_i =$

1.3524 3.8953 6.1111 11.0778 17.3666

Глава 7



Компьютерные технологии многопараметрической интерполяции

7.1. Введение

Функции многих переменных в научных и практических задачах встречаются чаще, чем функции одной переменной. Типичным примером является теория планирования и статистической обработки результатов эксперимента. В теории планирования эксперимента объект исследования представляется в виде "черного ящика" (рис. 7.1).



Рис. 7.1

На рис. 7.1 приняты следующие обозначения:

- Ч.Я. — "черный ящик" — объект, свойства которого необходимо изучить;
- x_i — возмущающие воздействия, называемые в теории планирования эксперимента факторами ($i = 1, 2, \dots, n$);
- y_i — выходные характеристики объекта, называемые откликами ($i = 1, 2, \dots, m$).

Свойства объекта при такой модели изучаются по значениям откликов при воздействии на него n факторов. Перед проведением эксперимента составляется план испытания, представляющий собой таблицу значений факторов

и соответствующих им откликов. Значения откликов определяются в процессе эксперимента. В большинстве случаев отклики $y_1, y_2, \dots, y_m$ являются независимыми, поэтому в результате эксперимента получают функции:

$$y_i = f_i(x_1, x_2, \dots, x_n).$$

Такие функции не являются математическими моделями объекта исследования. Для получения модели необходимо иметь аналитические выражения функций отклика. Получается типичная задача интерполяции функций многих переменных. Ее особенностями являются:

- большие трудности выбора вида функции интерполяции, связанные с невозможностью графического представления ее на плоскости и даже в пространстве;
- трудности доказательства адекватности математической модели.

Интерполяцию функций многих переменных часто называют аппроксимацией, т. к. при определении математической модели обычно используется интерполяция приближенная в узлах.

Получение функции интерполяции называют регрессионным анализом или просто регрессией.

Регрессионный анализ в случае многих переменных можно выполнить только с помощью компьютера, вручную такие задачи решить невозможно.

Задачи регрессионного анализа возникают не только при проведении многофакторного эксперимента. Такие задачи часто встречаются в экономике, технике, образовании, медицине и даже в спорте и гуманитарных науках. В главе 8 сформулированы и решены задачи многопараметрической интерполяции из различных областей знаний.

Универсальные математические программные средства Mathematica, Maple, Derive, Mathcad и Matlab позволяют решать задачи интерполяции многих переменных. Рассмотрим технологии решения этих задач в упомянутых математических системах.

7.2. Интерполяция функций многих переменных в системе Derive

В системе Derive регрессия функции многих переменных осуществляется с помощью функции FIT, имеющей вид:

FIT(M).

Здесь:

- M — матрица размерности $(n+1) \times m$;
- n — число строк матрицы исходных данных;

□ m — число столбцов.

В первой строке матрицы записываются имена аргументов $x_1, x_2, \dots, x_{m-1}$ и вид функции аппроксимации. В последующих n строках — численные значения аргументов и значения функции, полученные опытным путем.

Технологию решения задачи рассмотрим на примере.

Пусть в результате реализации двухфакторного плана испытания получены опытные данные, приведенные в табл. 7.1.

Таблица 7.1

	Значения переменных								
x_1	1	2	3	4	5	6	7	8	9
x_2	0.2	2.4	3.9	5	7.3	8.6	10	11.5	13
y	4.7	10.28	17	24.6	38	50.17	64.6	81.45	100.3

Предположим, что функция регрессии является линейной: $y = b_0 + b_1x_1 + b_2x_2$. Определение коэффициентов b_0, b_1, b_2 в системе Derive осуществляется в следующей последовательности:

- 1. Создадим с помощью пунктов меню **Authort | Matrix** матрицу размером 10×3 , размер матрицы устанавливается в окне **Matrix Setup**.
- 2. Введем в строке пользователя функцию: `FIT#1` (где #1 — строка матрицы на экране монитора).
- 3. Введем матрицу в память ПК нажатием клавиши `<Enter>`.

На экране появится следующая матрица:

FIT

x_1	x_2	$b_0 + b_1x_1 + b_2x_2$
1	0.2	4.7
2	2.4	10.28
3	3.9	17
4	5	24.6
5	7.3	38
6	8.6	50.17
7	10	64.6
8	11.5	81.45
9	13	100.3

После команды Approximate (кнопка  на панели инструментов) откликом является следующая функция: $y = 28.35x_1 - 10.44x_2 - 26.5$.

Значения этой функции $Y_{\text{лин}}$, квадратичной $Y_{\text{кв}}$ и исходной $Y_{\text{исх}}$ приведены в табл. 7.2.

Таблица 7.2

x_1	x_2	$Y_{\text{исх}}$	$Y_{\text{лин}}$	$Y_{\text{кв}}$	$\varphi(x_1, x_2)$
1	0.2	4.7	-0.23	4.7	4.7
2	2.4	10.28	5.15	10.27	10.27
3	3.9	17	17.84	17.02	17.02
4	5	24.6	34.71	24.6	24.6
5	7.3	38	39	38	38
6	8.6	50.17	53.83	50.13	50.13
7	10	64.6	67.56	64.55	64.55
8	11.5	81.45	80.26	81.43	81.43
9	13	100.3	93	100.3	100.3

Квадратичная функция регрессии получена путем редактирования первой строки матрицы. Матрица активизируется, вызывается в строку пользователя нажатием клавиши <F3>, редактируется выражение $b_0 + b_1x_1 + b_2x_2$ путем дополнения его членами $b_{11}x_1^2 + b_{22}x_2^2 + b_{12}x_1x_2$. Дальнейшие процедуры решения очевидны. Они аналогичны описанным ранее.

Решение дает следующую квадратичную функцию регрессии:

$$Y_{\text{кв}} = 3.2 + 0.93x_1 + 0.77x_2 + 0.344x_1^2 + 0.097x_2^2 + 0.29x_1x_2.$$

Из табл. 7.2 очевидно, что линейная функция не может быть математической моделью изучаемого объекта, т. к. значение $Y_{\text{лин}}$ при $x_1 = 1$, $x_2 = 0.2$ отрицательно, что противоречит физическому смыслу задачи. Квадратичная модель адекватна объекту, погрешность формулы практически равна нулю.

Функция FIT позволяет получить функцию регрессии, в состав которой входят: экспоненциальная, логарифмическая, дробно-линейная, показательная и многие другие функции.

В общем случае функция регрессии может иметь следующий вид:

$$y = b_0 + b_1 f(x_1) + b_2 f(x_2) + \dots + b_n f(x_n), \text{ где } f(x_i) \text{ — функция аргумента } x_i.$$

Следует иметь в виду, что функция регрессии должна быть линейной относительно искомых параметров.

В качестве примера найдем для нашей задачи функцию регрессии, если она имеет вид:

$$y = b_0 + b_1x_1 + c_1 \ln(x_1) + b_{11}x_1^2 + b_2x_2 + c_2e^{x_2} + b_{22}x_2^2 + b_{12}x_1x_2$$

Решение задачи с помощью функции FIT дает следующую функцию регрессии:

$$y = 3.427 + 0.329x_1 + 0.618 \ln(x_2) + 0.81x_1^2 + 0.95x_2 - 1.75 \cdot 10^{-7}e^{x_2} + 0.316x_2^2 + 0.33x_1x_2.$$

Результаты табулирования этой функции приведены в табл. 7.2 (функция $\varphi(x_1, x_2)$).

Сравнивая данные функций $\varphi(x_1, x_2)$ и $Y_{\text{кв}}$ видим, что они практически одинаковы и усложнение функции регрессии в данном случае новых результатов не дало. Решение нас убедило лишь в том, что в состав функции регрессии могут входить функции экспоненциальная и логарифмическая.

Остается нерешенным главный вопрос: как выбрать вид функции аппроксимации для получения функции регрессии многих переменных. Этот вопрос в теории интерполяции до конца не решен. Наиболее часто первоначально выбирают линейную или квадратичную функции. После получения решения функцию проверяют на адекватность. В компьютерных технологиях интерполяции проверка осуществляется вычислением погрешностей. По их значениям принимается решение об адекватности модели.

Выбор числа членов функции интерполяции ограничивается размерностью таблицы данных. При линейной интерполяции число строк таблицы должно быть: $n + 1$ (где n — число аргументов функции регрессии).

При квадратичной функции число строк должно быть не менее $\frac{(n+2)(n+1)}{2}$.

В нашем примере квадратичная функция дополнена логарифмическим и экспоненциальным членами, т. е. число неизвестных коэффициентов равно $\frac{(n+2)(n+1)}{2} + 2 = 8$.

Исходная таблица содержит 9 строк. Избыточность опыта равна единице.

7.3. Интерполяция функций многих переменных в системе Maple

В среде Maple регрессия функции многих переменных осуществляется с помощью функции fit, имеющей вид:

`fit[leastsquare]([x, y, ..., z], z =  $\varphi(x, y, \dots)$ , {a, b, ..., h}) ([data]).`

Здесь:

- $x, y, \dots$ — аргументы функции регрессии;
- z — имя функции регрессии;
- $\varphi(x, y, \dots)$ — аналитическое выражение функции регрессии, выбираемое исследователем;
- $a, b, \dots, h$ — коэффициенты функции регрессии, вычисляемые ЭВМ;
- data — табличные данные аргументов $x, y, \dots$ и функции z .

Технологию аппроксимации рассмотрим на примере.

Пусть данные эксперимента приведены в табл. 7.3.

Таблица 7.3

	Значения переменных						
x	1	3	4	6	8	12	15
y	0.2	1	1.4	2	2.8	3.5	4.6
z	6	34	57	115	207	376	608

Необходимо найти функцию регрессии, если функция аппроксимации является квадратичной, имеющей вид:

$$z = b_0 + b_1x + b_2x^2 + b_3y + b_4y^2 + b_5xy$$

Прежде чем решать задачу аппроксимации функции многих переменных, необходимо обратиться к статистическому пакету `stats`. Вызов пакета осуществляется функцией `with(stats)`. Бывает полезным обеспечить вывод коэффициентов регрессии с ограниченным числом знаков. Это можно осуществить командой:

`Digits:=n` (где n — число знаков коэффициентов функции регрессии).

Приведем программу определения функции регрессии для нашего примера.

```
with(stats):
```

```
a:= [1, 3, 4, 6, 8, 12, 15]:
```

```
b:= [0.2, 1, 1.4, 2, 2.8, 3.5, 4.6]:
```

```
c:= [6, 34, 57, 115, 207, 376, 608]:
```

```
f:= fit[leastsquare[[x, y, z], z = b0 + b1x + b2x^2 + b3y + b4y^2 + b5xy, {b0, b1, b2, b3, b4, b5}]] ([a, b, c]);
```

После нажатия клавиши <Enter> получаем решение в следующем виде:

$$\hat{f} = \hat{z} = 2.12 + 3.86x - 0.067x^2 - 2.36y + 1.64y^2 + 8.04xy.$$

Результаты табулирования функции регрессии приведены в табл. 7.4.

Таблица 7.4

	Значения переменных						
z	6	34	57	115	207	376	608
z_f	6.25	34.7	59.55	119.98	215.56	395.47	638.47

В таблице обозначены:

- z — исходные данные;
- z_f — значения функции регрессии.

Поскольку значения функций отличаются друг от друга, то полезно определить погрешности аппроксимации. Абсолютная среднеквадратическая погрешность ε и максимальная относительная $\delta_{\max}$ имеют следующие значения:

□ $\varepsilon = 14.2$;

□ $\delta_{\max} = \frac{\varepsilon}{z_{\min}} \cdot 100 \% = \frac{14.2}{6} \cdot 100 = 236 \%$.

Большая максимальная относительная погрешность может служить основанием для следующего вывода: полученная функция регрессии не является удовлетворительной математической моделью изучаемого объекта. Причиной этого могут быть:

- недостаточный объем экспериментальных данных;
- эксперимент поставлен без научно обоснованного плана испытания;
- неудачно выбран вид функции аппроксимации.

Уточнение функции регрессии требует научных исследований.

7.4. Интерполяция функций многих переменных в системе Mathematica

Технология интерполяции функций многих переменных в системе Mathematica более проста по сравнению с другими математическими системами. Она реализуется функцией Fit, имеющей вид:

Fit[M , X , A].

Здесь:

- M — матрица исходных данных;
- X — перечень базиса функции регрессии;
- A — перечень аргументов функции регрессии.

В перечне базиса функции регрессии могут быть функции самого различного вида, заключенные в фигурные скобки и отделяемые друг от друга запятыми, например:

$\{a, x, x^2, \frac{a}{x}, \frac{x}{1+x}, \frac{b}{2+x}, \text{Exp}[x], \text{Ln}[x]\}.$

Здесь в составе функции регрессии будут члены вида: постоянная, x , x^2 и все другие до логарифмической включительно. Перечень аргументов A заключается в фигурные скобки, аргументы отделяются друг от друга запятыми.

Технологию интерполяции рассмотрим на примере функции двух переменных.

Пусть матрица исходных данных представлена в табл. 7.5.

Таблица 7.5

	Значения переменных											
x	20	30	40	50	60	70	80	90	100	110	120	130
y	18	22	22.5	28.5	30.5	32	33	34	35	35.5	36	36.3
z	36	42	58	71	85	97	112	132	153	187	201	221

В качестве функции регрессии возьмем следующую квадратичную функцию:

$\varphi(x, y) = b_0 + b_1x + b_2x^2 + b_3y + b_4y^2 + b_5xy$

По данным таблицы и виду функции $\varphi(x, y)$ Mathematica найдет коэффициенты b_i (где $i = 0, 1, \dots, 5$). Программа решения этой задачи имеет вид:

```
M = {{20, 18, 36}, {30, 22, 42}, {40, 22.5, 58}, {50, 28.5, 71},
{60, 30.5, 85}, {70, 32, 97}, {80, 33, 112}, {90, 34, 132},
{100, 35, 153}, {110, 35.5, 187}, {120, 36, 201}, {130, 36.3, 221}};
X = {a, x, x^2, y, y^2, x * y};
Fit[M, X, {x, y}]
```

После нажатия клавиш <Shift>+<Enter> получаем ответ в виде следующей функции регрессии:

$z(x, y) = -35.1 - 0.484x + 0.0018x^2 + 6.57y + 0.0655xy - 0.194y^2$

Результаты табулирования функции приведены в табл. 7.6.

Таблица 7.6

	Значения переменных											
z	36	42	58	71	85	97	112	132	153	187	201	221
z(x, y)	34.9	45.8	57	69	82	98	116	135.3	155.8	177.7	200.5	224

Из табл. 7.6 видно, что полученная функция регрессии хорошо аппроксимирует исходные данные. Это подтверждают и расчеты точности интерполяции:

- абсолютная среднеквадратическая погрешность $\varepsilon = 3.67$;
- максимальная относительная погрешность $\delta_{\max} = \frac{\varepsilon}{36} \cdot 100 \% = 10.2 \%$;
- минимальная относительная погрешность $\delta_{\min} = \frac{\varepsilon}{221} \cdot 100 \% = 1.7 \%$.

Приведем еще один достаточно типичный и поучительный пример многопараметрической аппроксимации. Опыт проведен без научно-обоснованного плана испытания. Результаты опыта показаны в табл. 7.7.

Таблица 7.7

	Значения переменных									
x_1	1.2	4.3	7	7.4	10	12	13.5	16	17.8	20
x_2	4.3	6.8	9.3	10.7	11	11.5	8.4	7.3	6.2	4
z	16.4	20.3	22.5	30	32.4	35.8	37	38.6	39.7	41.5

Необходимо:

- найти математическую модель исследуемого объекта путем аппроксимации данных опыта;
- определить погрешность модели;
- научно-обоснованно спланировать эксперимент и обработать его результаты;
- выполнить сравнительный анализ данных исходного опыта и научно-обоснованного.

Предположим, что математической моделью объекта может быть либо линейная, либо квадратичная модель. Тогда по аналогии с предыдущим примером программа решения задачи с помощью функции Fit будет иметь вид:

```
M = {{1.2, 4.3, 16.4}, {4.3, 6.8, 20.3}, {7, 9.3, 22.5},
{7.4, 10.7, 30}, {10, 11, 32.4}, {12, 11.5, 35.8}, {13.5, 8.4, 37},
{16, 7.3, 38.6}, {17.8, 6.2, 39.7}, {20, 4, 41.5}};
```

```
X = {a, x, y}; – если модель линейная,
```

```
X = {a, x, x^2, y, y^2, x * y}; – если модель квадратичная,
```

```
Fit[M, X, {x, y}]
```

После нажатия клавиш <Shift>+<Enter> получаем ответ в виде одной из следующих функций регрессии:

□ линейная модель: $z_1(x, y) = 10.3 + 1.44x + 0.67y$;

□ квадратичная модель:

$$z_2(x, y) = 22.95 + 1.78x - 0.034x^2 - 2.77y + 0.067xy + 0.15y^2$$

Результаты табулирования функций приведены в табл. 7.8.

Таблица 7.8

	Значения переменных									
z	16.4	20.3	22.5	30	32.4	35.8	37	38.6	39.7	41.5
z₁	14.93	21.09	26.67	28.2	32.15	35.37	35.45	38.3	40.17	41.87
z₂	16.28	20.1	25.44	27.27	32.57	36.8	35.8	38.4	39.89	41.65

Погрешности — абсолютная ε и относительная δ функций регрессии, вычисленные по технологии, описанной в главе 4, имеют значения:

□ линейная $\varepsilon = 1.628$, $\delta_{\max} = 9.93 \%$;

□ квадратичная: $\varepsilon = 1.37$, $\delta_{\max} = 8.3 \%$, $\delta_{\min} = 3.3 \%$.

Проведем теперь научно-обоснованный эксперимент и получим математические модели.

В случае линейной модели $z = b_0 + b_1x_1 + b_2x_2$ достаточно лишь трех опытов для определения коэффициентов b_0 , b_1 , b_2 . Но т. к. опыт должен удовлетворять условиям нормировки, симметричности и ортогональности, то ставится полнофакторный двухуровневый эксперимент.

Число опытов N при таком эксперименте определяется соотношением:

$$N = P^n, \text{ где:}$$

□ n — число факторов;

□ P — число уровней в каждом факторе.

В нашем случае: $n = 2$, $P = 2$. Тогда число опытов должно быть: $N = 2^2 = 4$.

Диапазоном изменения значений факторов x_1 , x_2 в рассматриваемом примере является: $1.2 \leq x_1 \leq 20$; $4 \leq x_2 \leq 11.5$.

Тогда планом испытания в случае линейной модели будет табл. 7.9.

В таблице обозначено:

□ $a_1 = 1.2$ — нижний уровень фактора x_1 ;

□ $b_1 = 20$ — верхний уровень фактора x_1 ;

□ $a_2 = 4$ — нижний уровень фактора x_2 ;

□ $b_2 = 11.5$ — верхний уровень фактора x_2 ;

□ z_i (где $i = 1, 2, 3, 4$) — значение отклика.

Таблица 7.9

№ опыта	x_1	x_2	z
1	a_1	a_2	z_1
2	a_1	b_2	z_2
3	b_1	a_2	z_3
4	b_1	b_2	z_4

В табл. 7.9 отсутствуют значения z_i при соответствующих значениях факторов x_1, x_2 . В исходных данных имеется лишь отклик $z_3 = 41.5$.

Примем линейную математическую модель, полученную в результате решения задачи аппроксимации, за объект анализа и вычислим значения z_i на наборе аргументов:

[1.2, 4], [1.2, 11.5], [20, 4], [20, 11.5].

Подставляя эти значения в формулу $z_1(x, y)$ линейной модели, получим:

$z_1 = 14.7$; $z_2 = 19.7$; $z_3 = 41.78$; $z_4 = 46.8$.

Тогда результаты испытания будут иметь вид, представленный в табл. 7.10.

Таблица 7.10

№ опыта	x_1	x_2	z
1	1.2	4	14.7
2	1.2	11.5	19.7
3	20	4	41.78
4	20	11.5	46.8

Линейной моделью в нашем случае является функция: $z = b_0 + b_1x_1 + b_2x_2$. Тогда программа вычисления коэффициентов регрессии b_0, b_1, b_2 будет иметь вид:

```
M = {{1.2, 4, 14.7}, {1.2, 11.5, 19.7}, {20, 4, 41.78},
      {20, 11.5, 46.8}};
```

```
X = {a, x, y};
```

```
Fit[M, X, {x, y}]
```

После нажатия клавиш <Shift>+<Enter> получим следующее решение:

$z(x, y) = 10.29 + 1.44x + 0.668y$

В случае квадратичной модели число неизвестных равно 6. Для ее реализации при полном факторном эксперименте необходимо иметь три уровня в

каждом факторе. Тогда число опытов должно быть: $N = P^n = 3^2 = 9$. Число избыточных опытов равно трем. Следует иметь в виду, что избыточные опыты это не те, которые не нужны. Они необходимы для обеспечения условий нормировки, симметричности, ортогональности.

Планом испытания в данном случае будет табл. 7.11.

Таблица 7.11

№ опыта	x_1	x_2	z
1	a_1	a_2	z_1
2	a_1	$x_2^{(0)}$	z_2
3	a_1	b_2	z_3
4	$x_1^{(0)}$	a_2	z_4
5	$x_1^{(0)}$	$x_2^{(0)}$	z_5
6	$x_1^{(0)}$	b_2	z_6
7	b_1	a_2	z_7
8	b_1	$x_2^{(0)}$	z_8
9	b_1	b_2	z_9

В таблице $x_1^{(0)}$ и $x_2^{(0)}$ являются средними значениями факторов, а именно:

$$\square x_1^{(0)} = \frac{a_1 + b_1}{2};$$

$$\square x_2^{(0)} = \frac{a_2 + b_2}{2}.$$

Вычисляя, как и в случае линейной модели, значения откликов z_i для всех комбинаций факторов, указанных в табл. 7.11, получим все неизвестные ранее отклики. Тогда результатами испытания будет табл. 7.12. Значения откликов z в таблице округлены до двух значащих цифр после запятой. Это необходимо по правилам приближенных вычислений: число значащих цифр результата может быть больше лишь на один знак по сравнению с исходными данными.

Таблица 7.12

№ опыта	x_1	x_2	z
1	1.2	4	16.67
2	1.2	7.75	13.2
3	1.2	11.5	13.94
4	10.6	4	32.16
5	10.6	7.75	31.04
6	10.6	11.5	34.15
7	20	4	41.63
8	20	7.75	42.88
9	20	11.5	48.34

Программа вычисления коэффициентов регрессии имеет вид:

```

M = {{1.2, 4, 16.67}, {1.2, 7.75, 13.2}, {1.2, 11.5, 13.94},
{10.6, 4, 32.16}, {10.6, 7.75, 31.04}, {10.6, 11.5, 34.15},
{20, 4, 41.63}, {20, 7.75, 42.88}, {20, 11.5, 48.34}};
X = {a, x, x^2, y, y^2, x * y};
Fit[M, X, {x, y}]

```

Решением будет следующая функция регрессии:

$$z(x, y) = 22.935 + 1.78x - 0.034x^2 - 2.77y + 0.067xy + 0.15y^2$$

Результаты табулирования линейной и квадратичной функций регрессии приведены в табл. 7.13.

Таблица 7.13

Значения переменных										
z	16.4	20.3	22.5	30	32.4	35.8	37	38.6	39.7	41.5
$z_{1,m}$	14.93	21.09	26.67	28.2	32.15	35.37	35.45	38.3	40.17	41.87
z_1	16.28	20.1	25.44	27.27	32.57	36.8	35.8	38.4	39.89	41.65
$z_{2,m}$	14.9	21	26.6	28.1	32.05	35.27	35.36	38.23	40.08	41.79
z_2	16.24	20.03	25.32	27.1	32.4	36.64	35.7	38.3	39.85	41.63

В таблице обозначены:

□ z — данные исходного научно необоснованного опыта;

- $z_{1,m}$ — данные регрессии научно необоснованного опыта в случае линейной модели;
- z_1 — отклики линейной модели в случае научно обоснованного опыта;
- $z_{2,m}$ — данные регрессии научно необоснованного опыта в случае квадратичной модели;
- z_2 — отклики квадратичной модели в случае научно обоснованного опыта.

Анализ результатов аппроксимации

Математические модели, полученные без научного обоснования плана эксперимента и на основании полно факторного эксперимента, практически совпадают, особенно в случае квадратичной модели. В случае линейной модели число опытов можно сократить с 9 до 4 без существенной потери точности.

7.5. Интерполяция функций многих переменных в системе Mathcad

В Mathcad основной функцией решения задач многопараметрической аппроксимации является функция `interp` [9], имеющая для двухпараметрической интерполяции вид:

`interp( $V_s$ ,  $M_{xy}$ ,  $V_z$ ,  $V$ ).`

Здесь:

- M_{xy} — матрица аргументов x , y ;
- V_z — вектор значений функции;
- V_s — вектор, вычисляемый функцией `regress` или функцией `loess` по данным матрицы M_{xy} и вектора V_z ;
- V — вектор данных, для которых вычисляется значение функции аппроксимации.

Функции `regress` и `loess`:

- функция `regress` обеспечивает полиномиальную аппроксимацию оптимальной функцией в смысле минимума среднеквадратической погрешности (метод наименьших квадратов). Она имеет вид:

`regress( $M_{xy}$ ,  $V_z$ ,  $n$ )` (где n — порядок многочлена, матрица M_{xy} и вектор z определены ранее);

- функция `loess` организует набор полиномов, которые в диапазоне определенных значений аргументов x , y матрицы M_{xy} оптимальным образом оптимизируют функцию V_z . Эта функция имеет вид:

`loess( $M_{xy}$ ,  $V_z$ , span).` Аргумент `span` определяет размер области значений аргументов x , y , которые участвуют в определении полиномов. Если этот

диапазон большой, то функции `loess` и `regress` дают одинаковые результаты. Хорошее решение получается при `span = 0.75`, которое реализуется по умолчанию.

Описанные функции позволяют решать задачи аппроксимации и в случае многочленов более высоких порядков. Однако при $n > 4$ вычисления существенно затягиваются, а число опытов должно быть большим. В общем случае число опытов (значение данных) должно быть:

$$m \geq C_{n+k-1}^k \cdot \frac{n+k}{n}.$$

Здесь:

□ n — число независимых переменных;

□ k — степень полинома.

Так, например, если $n = 4$, $k = 3$, то число опытов должно быть $m \geq 35$. Функция `interp` не позволяет получить функцию регрессии в аналитическом виде. А это означает, что она фактически не позволяет получить по опытным данным математическую модель изучаемого объекта, хотя может быть полезной при определении численных значений функции многопараметрической аппроксимации в диапазоне изменения значений аргументов.

7.6. Интерполяция функций многих переменных в системе Matlab

Рассмотрим вначале технологию интерполяции функции двух переменных. Для этого случая в Matlab используется функция `interp2`, которая имеет вид:

$R = \text{interp2}(X, Y, Z, x_1, y_1).$

Здесь:

□ X, Y, Z — векторы аргументов X, Y и функции $Z = f(X, Y)$;

□ x_1, y_1 — значения аргументов, для которых вычисляются значения функции интерполяции R .

Функция `interp2` позволяет задать метод интерполяции. В этом случае она имеет вид:

$R = \text{interp2}(X, Y, Z, x_i, y_i, \text{method}).$

Опция `method` может иметь следующие значения:

□ 'nearest' — интерполяция сопрягающимися линейными отрезками;

□ 'linear' — линейная интерполяция;

□ 'cubic' — кубическая интерполяция.

Matlab позволяет решать задачи интерполяции функций N переменных. В этом случае функция `interp` имеет вид:

$R = \text{interp}(X1, X2, \dots, XN, W, Y1, Y2, \dots, YN)$.

Здесь:

- $X1, X2, \dots, XN$ — векторы аргументов функции W ;
- W — вектор значений функции;
- $Y1, Y2, \dots, YN$ — векторы аргументов функции регрессии, задаваемые пользователем.

При N -параметрической интерполяции так же, как и при двухпараметрической, можно задать метод интерполяции с помощью опций 'nearest', 'linear', 'cubic'. Функция `interp` не позволяет получить функцию регрессии в виде формулы. Это существенно ограничивает возможности системы Matlab по решению задач многопараметрической интерполяции.

7.7. Многопараметрическая интерполяция точная в узлах

По аналогии с однопараметрической интерполяцией аргументы функции регрессии (факторы) будем называть узлами интерполяции. Тогда задача многопараметрической интерполяции может быть решена методом точным в узлах интерполяции. Технология решения задачи состоит в следующей последовательности действий:

- выбор узлов интерполяции;
- составление системы алгебраических уравнений;
- определение коэффициентов функции регрессии;
- оценка адекватности модели.

Выбор узлов интерполяции

Число узлов интерполяции должно быть равно числу неизвестных математической модели. Значение узлов должно соответствовать значениям факторов научно обоснованного плана испытания. При этом избыточности опытов при планировании и проведении эксперимента не требуется. Если научно обоснованный план испытания реализовать нельзя, то значения узлов следует выбирать так, чтобы они охватывали весь диапазон изменения факторов.

Составление системы алгебраических уравнений

Размерность системы алгебраических уравнений должна быть равна числу неизвестных функции регрессии. При этом уравнения могут быть линейными или нелинейными относительно неизвестных. В этом существенное достоинство этого метода по сравнению с методами, основанными на применении функций интерполяции универсальных математических систем.

Наиболее трудным для пользователя является выбор вида функции регрессии. Достаточно простыми, а поэтому наиболее часто применяемыми, являются линейная и квадратичная модели, представляющие собой многочлены, линейные относительно неизвестных коэффициентов. Однако далеко не всегда эти функции дают желаемые результаты. Проверка на адекватность модели часто дает погрешности, превышающие допустимые. Приходится отказываться от этих моделей. Методики выбора вида функции регрессии, научно обоснованной, доведенной до очевидных математических процедур, не существует. Здесь — большое поле деятельности для пользователя.

Определение коэффициентов функции регрессии

Определение коэффициентов функции регрессии осуществляется путем решения системы уравнений.

В случае линейной модели решение можно получить точными методами с помощью любого универсального программного средства символьной математики. В случае нелинейных систем решение возможно только численными методами Ньютона или итераций. В главах 2—6 изложены компьютерные технологии решения систем уравнений аналитическими и численными методами.

Компьютерные технологии требуют проверки правильности решения систем уравнений. Об этом не следует забывать при решении задач интерполяции. Ошибка может привести в дальнейшем к ненужному поиску нового вида функции регрессии.

Трудности в определении коэффициентов функции регрессии могут возникнуть в случае решения систем нелинейных уравнений. Главная из них для пользователя — это выбор начальных значений неизвестных. Здесь необходимо хорошо понимать сущность задачи, физический смысл коэффициентов и, конечно, иметь терпение при многократных уточнениях начальных значений и попытках получить решение.

Оценка адекватности модели

Итак, решение получено. Однако это еще не означает, что найдена функция регрессии, являющаяся математической моделью изучаемого объекта. Необходимо сравнить данные эксперимента и результаты модели. Для этого следует протабулировать функцию регрессии при аргументах, соответствующих опытным данным, и сравнить данные табуляции с опытными. Если они близки друг другу, то следует вычислить погрешность модели. Максимальная относительная погрешность не должна превышать допустимую. Если это условие выполняется, то задача решена — математическая модель найдена. В противном случае от полученного результата необходимо отказаться и искать причины большой погрешности. Причинами могут быть: неточность данных эксперимента, неудачный выбор вида функции регрессии, ошибка решения по вине пользователя или ПК. Если вы уверены в точно-

сти экспериментальных данных, то повторите решение. В случае идентичности ответов причиной большой погрешности аппроксимации является неудачный выбор вида функции регрессии.

Уточнить вид функции интерполяции можно следующими способами:

- ☐ повысить степень многочлена;
- ☐ выбрать функциональный многочлен;
- ☐ сменить вид функции интерполяции.

Рассмотрим описанную методику на примере.

Пусть зависимость $y = f(x_1, x_2)$ представлена в виде таблицы (табл. 7.14).

Таблица 7.14

	Значения переменных							
x_1	1.2	2.3	4.8	7.3	12.6	18	24.7	31
x_2	2.1	5.2	8.4	11.5	15.5	23.6	31.8	37.6
y	5.6	11.5	18.7	25.8	36.7	54	73	87.7

Необходимо получить математическую модель объекта в виде функции регрессии.

Предположим, что опытные данные получены с высокой точностью и отражают некий физический закон. Тогда функцию регрессии целесообразно получить методом интерполяции точным в узлах. Будем решать задачу в соответствии с предложенной ранее компьютерной технологией.

Предположим, что функция регрессии является линейной вида: $y = b_0 + b_1x_1 + b_2x_2$. Тогда узлами интерполяции могут быть: [1.2, 2.1]; [7.3, 11.5]; [31, 37.6]. Они охватывают весь диапазон изменения аргументов x_1, x_2 , а их число равно числу неизвестных линейной функции регрессии.

Система алгебраических уравнений, соответствующая выбранным узлам интерполяции, будет иметь вид:

$$b_0 + 1.2b_1 + 2.1b_2 = 5.6$$

$$b_0 + 7.3b_1 + 11.5b_2 = 25.8$$

$$b_0 + 31b_1 + 37.6b_2 = 87.7$$

Решим систему уравнений в среде Mathematica. Программа решения имеет вид:

```
M = {b0 + 1.2b1 + 2.1b2 = 5.6,
```

```
b0 + 7.3b1 + 11.5b2 = 25.8,
```

```
b0 + 31b1 + 37.6b2 = 87.7};
```

```
Solve [% , {b0, b1, b2}]
```

После нажатия клавиш <Shift>+<Enter> получим следующие значения коэффициентов линейной функции регрессии: $b_0 = 1.23$, $b_1 = 0.86$, $b_2 = 1.6$. Тогда функция регрессии будет иметь вид: $y = 1.23 + 0.86x_1 + 1.6x_2$.

Программа выдала решение с точностью до шести значащих цифр. Ответ записан с точностью две значащие цифры после запятой.

Результаты табулирования функции регрессии, выполненные по технологии, описанной в *главе 4*, приведены в табл. 7.15.

Таблица 7.15

	Значения переменных							
y	5.6	11.5	18.7	25.8	36.6	54	73	87.7
$y(x_1, x_2)$	5.6	11.47	18.72	25.8	36.72	54.25	73.056	87.7
$\varphi(x_1, x_2)$	5.5785	11.4427	18.6835	25.7584	36.6706	54.1832	72.9723	87.6042

В таблице 7.15 приняты следующие обозначения:

□ y — значения функции, полученные в результате опыта;

□ $y(x_1, x_2)$ — значения функции регрессии, полученной методом интерполяции точным в узлах.

Сравнивая значения функций y и $y(x_1, x_2)$, видим, что они практически совпадают, причем в узлах интерполяции они равны, что подтверждает правильность полученного решения.

Решим теперь эту задачу приближенным методом с помощью функции Fit. Программа в системе Mathematica имеет вид:

```
M = {{1.2, 2.1, 5.6}, {2.3, 5.2, 11.5}, {4.8, 8.4, 18.7},
      {7.3, 11.5, 25.8}, {12.6, 15.5, 36.6}, {18, 23.6, 54},
      {24.7, 31.8, 73}, {31, 37.6, 87.7}};
```

```
X = {a, x1, x2};
```

```
Fit[M, X, {x1, x2}]
```

После нажатия клавиш <Shift>+<Enter> получим решение в виде следующей функции:

$$y = 1.21016 + 0.85976x_1 + 1.58887x_2.$$

Результаты табулирования функции приведены в табл. 7.15 (функция $\varphi(x_1, x_2)$).

Теперь составим научно обоснованный план испытания и по результатам опыта найдем функцию регрессии.

В нашем случае диапазон изменения факторов имеет следующие значения:

□ $1.2 \leq x_1 \leq 31$;

□ $2.1 \leq x_2 \leq 36.6$.

Табл. 7.16 является научно обоснованным планом испытания. В ней также приведены опытные данные (значения y).

Таблица 7.16

x_1	x_2	y
1.2	2.1	5.6
1.2	37.6	62.086
31	2.1	31.214
31	37.6	87.7

Отклики y при значениях аргументов:

$x_1 = 1.2$, $x_2 = 37.6$ и $x_1 = 31$, $x_2 = 2.1$ получены в результате вычислений по формуле функции регрессии, найденной ранее приближенным методом.

Найдем функцию регрессии, соответствующую научно обоснованному плану испытания. Программа определения функции регрессии по данным табл. 7.16 в системе Mathematica будет иметь вид:

```
M = {{1.2, 2.1, 5.6}, {1.2, 37.6, 62.086}, {31, 2.1, 31.214}, {31, 37.6, 87.7}};
```

```
X = {a, x1, x2};
```

```
Fit[M, X, {x1, x2}]
```

После нажатия клавиш <Shift>+<Enter> получим следующее решение:

$$y = 1.23 + 0.86x_1 + 1.6x_2.$$

Данное решение совпадает с тем, которое получено методом точным в узлах интерполяции.

Научно обоснованный план испытания должен удовлетворять условиям нормировки, симметричности и ортогональности. Во многих практических случаях при проведении экспериментов невозможно реализовать все эти условия. Эти случаи имеют место, когда эксперимент проводится на реальном объекте, а значения входных воздействий (факторов) в нужном диапазоне создать невозможно (опасно для жизни людей, слишком дорого, невозможно физически). В подобных случаях опыт проводится без его научного плана испытания. Результатами такого опыта могут быть следующие моменты:

- ☐ математическая модель, полученная в результате решения задачи интерполяции, не адекватна изучаемому объекту;
- ☐ трудность выбора вида функции аппроксимации;
- ☐ большая избыточность опыта.

В указанных случаях облегчить решение задачи многопараметрической интерполяции может следующая методика:

1. По данным эксперимента находится математическая модель любым из известных методов.
2. Составляется научно обоснованный план испытания.
3. Решается задача интерполяции методом точным в узлах интерполяции.
4. Определяются недостающие значения функции научно обоснованного плана испытания по модели, полученной методом точным в узлах интерполяции.
5. По данным научно обоснованного эксперимента находится математическая модель объекта.
6. Вычисляются погрешности модели.

Рассмотрим методику на примере.

Пусть данные научно необоснованного двухфакторного эксперимента приведены в табл. 7.17.

Таблица 7.17

	Значения переменных								
x_1	1	1.6	2.8	4	5.7	8.4	10.2	12.5	14.7
x_2	3.2	7.8	12.5	17	15.3	11.5	6.3	3.9	1.3
y	10.3	12.6	17.4	21.5	23.2	25.5	27.8	29.1	31.6

Необходимо найти математическую модель объекта и определить значения y при:

$$\square x_1 = 1, x_2 = 1.3;$$

$$\square x_1 = 1, x_2 = 17;$$

$$\square x_1 = 14.7, x_2 = 1.3;$$

$$\square x_1 = 14.7, x_2 = 17.$$

Будем решать задачу по описанной ранее методике.

Определение математической модели объекта по данным таблицы

Воспользуемся функцией Fit системы Mathematica. Программа двухпараметрической аппроксимации имеет вид:

```
M = {{1, 3.2, 10.3}, {1.6, 7.8, 12.6}, {2.8, 12.5, 17.4},
{4, 17, 21.5}, {5.7, 15.3, 23.2}, {8.4, 11.5, 25.5},
{10.2, 6.3, 27.8}, {12.5, 3.9, 29.1}, {14.7, 1.3, 31.6}};
```

```
X = {a, x1, x2};
```

```
Fit[M, X, {x1, x2}]
```

После нажатия клавиш <Shift>+<Enter> получим следующий ответ:

$$y = 6.99183 + 1.65035x_1 + 0.451361x_2$$

Разработка научно обоснованного плана испытания

Предположим, что математической моделью объекта является линейная функция вида $y = b_0 + b_1x_1 + b_2x_2$.

Тогда план испытания будет иметь вид, представленный в табл. 7.9.

В нашем случае диапазоном изменения факторов x_1, x_2 являются:

$$1 \leq x_1 \leq 14.7;$$

$$1.3 \leq x_2 \leq 17;$$

т. е. $a_1 = 1$; $a_2 = 1.3$; $b_1 = 14.7$; $b_2 = 17$.

Значения откликов y_1, y_2, y_4 из исходной табл. 7.17 неизвестны, $y_3 = 31.6$. Неизвестные отклики можно определить с помощью полученной ранее модели. Однако это опасно, т. к. функция аппроксимации получена из опыта, не удовлетворяющего условиям нормировки, симметричности и ортогональности, и может иметь значительные погрешности.

Поэтому решим задачу интерполяции методом точным в узлах. В линейной модели три неизвестных: b_0, b_1, b_2 . Для их определения составим систему из трех уравнений. Выберем следующие три пары факторов из таблицы 7.17:

[1.6; 7.8], [5.7; 15.3], [12.5; 3.9].

Тогда система уравнений будет иметь вид:

$$b_0 + 1.6b_1 + 7.8b_2 = 12.6$$

$$b_0 + 5.7b_1 + 15.3b_2 = 23.2$$

$$b_0 + 12.5b_1 + 3.9b_2 = 29.1$$

Решим эту систему уравнений с помощью функции Solve системы Mathematica. Программа решения задачи имеет вид:

```
y = {b0 + 1.6 * b1 + 7.8 * b2 = 12.6
```

```
b0 + 5.7 * b1 + 15.3 * b2 = 23.2
```

```
b0 + 12.5 * b1 + 3.9 * b2 = 29.1} ;
```

```
Solve [% , {b0, b1, b2}]
```

После нажатия клавиш <Shift>+<Enter> получим следующий ответ:

$$b_0 = 6.07569, b_1 = 1.68907, b_2 = 0.489973.$$

Тогда функция аппроксимации будет иметь вид:

$$y = 6.07569 + 1.68907x_1 + 0.489973x_2.$$

Поскольку исходные данные представлены с точностью до одного знака после запятой, то после округления значений коэффициентов решение можно представить в виде следующей функции аппроксимации:

$$y = 6 + 1.7x_1 + 0.5x_2.$$

Определение значений откликов линейной научно обоснованной модели

Вычислим значения откликов при следующих наборах значений x_1, x_2 :

[1; 1.3], [1; 17], [14.7; 1.3], [14.7; 17].

Подставляя эти значения в выражение функции аппроксимации, полученной методом точным в узлах интерполяции, увидим следующие значения откликов:

$$y_1 = 8.40172, y_2 = 16.0943, y_3 = 31.6, y_4 = 39.2346.$$

В системе Mathematica эти расчеты выполняем следующим образом:

1. Вводим выражение функции аппроксимации.
2. Выражение функции аппроксимации редактируем:
 x_1 и x_2 заменяем числами 1 и 1.3.
3. Нажимаем клавиши <Shift>+<Enter>, получаем значения отклика y_1 .
4. Вычисляем значения y_2, y_3 и y_4 путем редактирования исходной функции. Аргументы x_1 и x_2 последовательно заменяем числами:
 $x_1 = 1, x_2 = 17; x_1 = 14.7, x_2 = 1.3; x_1 = 14.7, x_2 = 17$.
5. Нажимаем клавиши <Shift>+<Enter>. Следует при этом иметь в виду, что функция интерполяции, полученная методом точным в узлах, является приближенной. Она может содержать ошибки, источником которых являются исходные данные, взятые для ее получения из научно необоснованной модели.

Полученные значения откликов примем за данные эксперимента. Результаты такого "эксперимента" представлены в виде табл. 7.18.

Таблица 7.18

x_1	x_2	y
1	1.3	3.40172
1	17	16.0943
14.7	1.3	31.6
14.7	17	39.2346

Определение математической модели по данным научно обоснованного плана испытаний

Программа определения функции аппроксимации в системе Mathematica имеет вид:

```
у = {{1, 1.3, 8.40172}, {1, 17, 16.0943}, {14.7, 1.3, 31.6},
{14.7, 17, 39.2346}};
```

```
Х = {a, x1, x2};
```

`Fit[Y, X, {x1, x2}]`

После нажатия клавиш <Shift>+<Enter> откликом будет следующая функция:

$$y = 6.09046 + 1.69119x_1 + 0.488127x_2.$$

Вычисление погрешностей модели

Результаты табулирования функций приведены в табл. 7.19, где приняты следующие обозначения:

- ☐ y — результаты исходного научно необоснованного эксперимента;
- ☐ y_1 — результаты аппроксимации научно необоснованной модели;
- ☐ y_2 — результаты научно обоснованной линейной модели.

Таблица 7.19

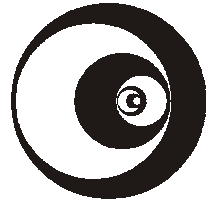
	Значения переменных								
y	10.3	12.6	17.4	21.5	23.2	25.5	27.8	29.1	31.6
y_1	10.0865	11.0767	13.0572	25.0376	17.8532	22.2991	25.2697	29.0656	32.6963
y_2	9.34366	12.6038	16.9274	21.1534	23.1986	25.9099	26.4158	29.134	31.5855

Погрешности моделей, вычисленные по данным таблицы, имеют следующие значения:

- ☐ для модели научно необоснованного эксперимента:
 - $\varepsilon_1 = 3.49$ (абсолютная);
 - $\delta_1 = 33.85$ % (максимальная относительная);
- ☐ для линейной научно обоснованной модели:
 - $\varepsilon_2 = 0.61$ (абсолютная);
 - $\delta_2 = 5.9$ % (максимальная относительная).

Табулирование функций и вычисление погрешностей осуществлено в системе Mathematica по методике, описанной в главе 4.

Из анализа данных табл. 7.19 и значений погрешностей видно, что предложенная методика дала положительные результаты — полученная математическая модель адекватна объекту.



Глава 8

Интерполяция в нашей профессии

8.1. Интерполяция — научная основа моделирования

Научное исследование — это процесс получения знаний о закономерностях функционирования объектов или явлений. Методы исследования могут быть двух видов: натурный эксперимент и моделирование. Натурный эксперимент — это исследование на реальном объекте. Моделирование — исследование процессов на модели, адекватно описывающей реальный объект.

Моделирование не только полезно, но и необходимо в следующих случаях:

- ☐ исследуемый объект слишком большой или слишком малый;
- ☐ процессы протекают очень медленно или очень быстро;
- ☐ невозможно создать реальные условия функционирования изучаемой системы (температуру, влажность, давление и т. п.);
- ☐ оригинал слишком удален от исследователя или вовсе недоступен;
- ☐ оригинал недоступен во времени (был в прошлом или будет в будущем);
- ☐ натурный эксперимент слишком дорогой;
- ☐ натурный эксперимент слишком опасный (излучение, вредные газы и т. п.).

Понятия модель и моделирование встречаются в трех видах:

- ☐ модель — объект, внешне подобный оригиналу (корабль, самолет, гидро-сооружение и т. п.). В этом случае моделирование заключается в копировании оригинала (физическая характеристика оригинала и модели не имеют значения).

- модель — объект, воспроизводящий физические свойства оригинала. Моделирование включает в себя познание свойств оригинала по характеристикам процессов, протекающих в модели.
- модель — искусственный объект, подобный оригиналу только математически. В таком случае моделирование — это процесс познания свойств объекта путем решения математических задач.

Моделирование возможно лишь в том случае, если оригинал и модель обладают свойством подобия. Подобными они могут быть только тогда, когда физические величины в оригинале и модели имеют одинаковые размерности или являются безразмерными и отличаются лишь масштабами.

Данные эксперимента, проводимого на оригинале, часто представляются в цифровой форме. Для получения математической модели приходится решать задачу интерполяции.

Из сказанного очевидно, что научными основами моделирования являются:

- теория размерностей;
- теория подобия;
- интерполяция.

Исследования на реальном объекте затруднительны, т. к. трудно и даже невозможно воспроизвести необходимые для эксперимента условия. Не менее трудно создать физическую модель реального объекта или физического явления. Поэтому наиболее эффективным является математическое моделирование. При этом в большинстве случаев приходится решать задачу синтеза модели: построение математического описания оригинала по данным эксперимента. Эта задача решается методами интерполяции.

В последующих разделах этой главы приводится описание задач, алгоритмов и способов их реализации методами интерполяции.

Применение методов интерполяции в жизни столь обширно, что приведенные примеры лишь капля в море задач моделирования (хотя они иллюстрируют важность интерполяции в нашей жизни).

8.2. Интерполяция и компьютерные технологии в задачах физики

Решать задачи синтеза моделей целесообразно в такой последовательности:

1. Выбор вида функции интерполяции.
2. Определение степени многочлена, если функция интерполяции является многочленом n -ой степени.
3. Выбор вида и метода интерполяции.
4. Определение коэффициентов функции интерполяции.

5. Вычисление погрешностей математической модели.
6. Исследование свойств оригинала на полученной модели.

Интерполяция, как научная основа моделирования, в задачах синтеза моделей играет первостепенную роль.

Будем иллюстрировать технологию синтеза моделей на примерах решения физических задач.

Уточнение закона расширения Вселенной. Постановка задачи

Астроном Хаббл в 1929 году обнаружил, что галактики удаляются от Земли тем быстрее, чем дальше они расположены. Данные опыта Хаббла приведены в табл. 8.1.

Таблица 8.1

Название галактики	R	V
Дева	22	7,5
Пегас	68	24
Персей	108	32
Волосы Вероники	137	47
Большая Медведица 1	255	93
Лев	315	120
Северная Корона	390	134
Близнецы	405	144
Волопас	685	245
Большая Медведица 2	700	260
Гидра	1100	380

В табл. 8.1 приняты обозначения:

- R — удаленность галактики от Земли в миллионах световых лет;
- V — скорость удаления галактики в сотнях миль в секунду.

Хаббл определил, что зависимость $V = f(R)$ линейна и имеет вид: $V = KR$, где K — постоянная, названная в его честь постоянной Хаббла. Необходимо найти закон расширения Вселенной, пользуясь методами интерполяции.

Будем решать задачу определения математической модели расширения Вселенной в установленной нами последовательности.

Выбор вида функции интерполяции

Хаббл утверждает, что функция интерполяции является линейной. Проверим это предположение. На рис. 8.1 приведена зависимость $V = f(R)$, которая действительно близка к линейной. Убедимся в этом, выполнив корреляционный анализ данных.

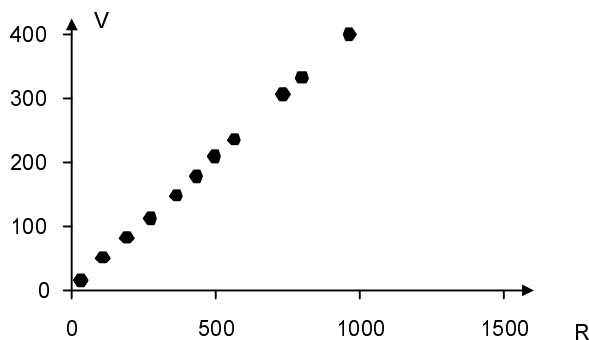


Рис.8.1

Коэффициент корреляции r является критерием формы функции $V = f(R)$.

Если $r \approx 1$, то можно считать, что функция линейна.

Коэффициент корреляции вычисляется по формуле:

$$r = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{\sqrt{(n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2)(n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2)}} \quad (8.1)$$

В формуле приняты следующие обозначения:

- n — число опытов;
- x_i, y_i (где $i = 1, 2, \dots, n$) — независимая и зависимая переменные соответственно.

В нашем случае $n = 11$, $x_i = R_i$, $y_i = V_i$. Расчеты по формуле 8.1 дают следующий результат:

$r = 0.998$, что подтверждает линейный характер зависимости $V = f(R)$.

Таким образом визуализация исходных данных и корреляционный анализ подтверждают предположение Хаббла о линейной функции расширения Вселенной:

$$V = a + bR.$$

Выбор метода интерполяции

Наша задача состоит в отыскании закона физического явления. В этом случае целесообразно применить метод точный в узлах интерполяции. Воспользуемся методом представления и решения систем линейных алгебраических уравнений. Для выбора узлов интерполяции применим метод выбранных точек (натянутой нити). Приложив нить к графику функции $V = f(R)$, выберем две наиболее подходящие точки. Такими могут быть точки с координатами: [68, 24] и [685, 245].

В целях большей наглядности будем искать математическую модель расширения Вселенной в виде функции $R = f(V)$. Тогда система уравнений запишется в следующем виде:

$$\begin{aligned} 68 &= a + 24b \\ 685 &= a + 245b \end{aligned} \quad (8.2)$$

Решить эту систему уравнений можно с помощью любой математической системы по методике, описанной в главах 2—6. В результате решения получим следующие значения коэффициентов a и b :

$$a = 0.995475, \quad b = 2.791855.$$

Тогда математической моделью расширения Вселенной будет:

$$R = 2.79V + 1. \quad (8.3)$$

Сравним данные интерполяции с исходными, для чего вычислим значения R при всех значениях V . С помощью универсальных математических программных средств это наиболее целесообразно выполнить табулированием функции (8.3). Функциями табуляции являются:

- в системе Derive 5: VECTOR([V, R(V)], V, [V(R)]);
- в системе Mathematica 4.2: Table[{V, R(V)}, {V, V(R)}];
- в системе Maple 6: seq(R(V), V = {V(R)});
- в Matlab табулирование функции осуществляется с помощью следующих команд:

```
f = a + bV,
V1 = [R(V)],
subs(f, V, V1);
```

- в системе Mathcad функции табулирования нет (для получения данных расчеты выполняются в режиме калькулятора).

Табулирование функции (8.3) нами осуществлено в системе Maple 6. Результаты табулирования приведены в табл. 8.2 (графа $\Phi_2(V)$).

Из табл. 8.2 видно, что функция интерполяции $R = f(V)$ найдена верно: в узлах интерполяции 24 и 245 значения R совпадают. Однако погрешности в остальных узлах интерполяции значительны.

Таблица 8.2

<i>R</i>	<i>V</i>	$\varphi_2(V)$	$\varphi_3(V)$	$\varphi_4(V)$
22	7.5	21.934	28.798	20.12
68	24	68	68	66.7
108	32	90.335	87.35	89.287
137	47	132.21	124.236	131.634
255	93	260.64	242.268	261.498
315	120	336	315	337.723
390	134	375	353.72	377.246
405	144	403	381.8	405.478
685	245	685	685	690.61
700	260	726.88	733.1	732.96
1100	380	1062	1146	1071.74

Вычислим погрешность интерполяции. За критерий погрешности примем абсолютную ε и относительную δ среднеквадратические погрешности. Вычисления выполним по формулам:

$$\varepsilon = \sqrt{\frac{\sum_{i=1}^n \Delta_i^2}{n}}, \quad \delta = \frac{\varepsilon}{V_{\min}} \cdot 100 \%, \text{ где } n = 11, \Delta_i = f(V_i) - \varphi_2(V_i).$$

Обозначения в формулах очевидны.

Вычислим погрешности, воспользовавшись системой Derive 5. С помощью команды **Declare | Vector** образуем векторы $f(r)$ и $\varphi(R)$:

```
[22, 68, 108, 137, 255, 315, 390, 405, 685, 700, 1100],  
[21.934, 68, 90.335, 132.21, 260.64, 336, 375, 403, 685, 726.88, 1062]
```

Предположим, что на экране монитора они находятся в строках #1 и #2. Тогда абсолютная среднеквадратическая погрешность может быть вычислена с помощью только одной команды: $\varepsilon = \text{sqrt}((\#1 - \#2)^2 / n)$

Нажмем клавишу <Enter>, затем на панели инструментов — кнопку  (Approximate), после чего получим следующий ответ: $\varepsilon = 17.06$. Тогда максимальная относительная погрешность будет:

$$\delta = \frac{\varepsilon}{R_{\min}} \cdot 100 \% = \frac{17.06}{22} \cdot 100 \approx 77.5 \%.$$

Анализ погрешностей интерполяции позволяет сделать следующий вывод: математическая модель (8.3) расширения Вселенной, полученная методом интерполяции точной в узлах, сомнительна. Ее максимальная относительная погрешность слишком велика. Причинами низкой точности интерполяции могут быть: ошибки опытных данных или неудачно выбранная функция интерполяции. Так как уточнить опытные данные мы не можем физически, то уточним функцию интерполяции. Выберем в качестве интерполяционного полином второй степени:

$$R = a + bV + cV^2.$$

Воспользуемся вновь методом интерполяции точной в узлах, выбрав дополнительно к предыдущим узел [315, 120].

Тогда система уравнений будет иметь вид:

$$\begin{aligned} 68 &= a + 24b + 24^2c \\ 315 &= a + 120b + 120^2c \\ 685 &= a + 245b + 245^2c \end{aligned} \quad (8.4)$$

Решение системы уравнений получено с помощью математической системы Maple 6, которая выдала следующие значения коэффициентов:

$$a = 11.29, b = 2.32, c = 0.00175.$$

Тогда математической моделью расширения Вселенной будет:

$$R = 0.00175V^2 + 2.32V + 11.29 \quad (8.5)$$

Результаты табулирования функции приведены в табл. 8.2 (графа $\varphi_3(V)$). Результаты табулирования показывают, что решение выполнено верно: значения функции в узлах интерполирования $V = 24, 120, 245$ совпадают с исходными. Однако отклонения R от исходных значений в остальных узлах велики. При этом погрешности, рассчитанные по описанной выше методике, составляют:

$$\square \varepsilon = 23.1;$$

$$\square \delta_{\max} = 105 \, \%.$$

Относительная погрешность стала еще больше.

В связи с большими погрешностями, превышающими допустимые, которые обычно составляют единицы процентов, полученная математическая модель в виде полинома второй степени сомнительна.

График функции $V = f(R)$ и значение коэффициента корреляции r являются убедительными факторами в пользу линейной модели. Большие погрешности модели можно объяснить неточностью исходных экспериментальных данных.

Применим теперь метод полиномиальной интерполяции, приближенный в узлах, предполагая, что функция интерполяции линейна. Такая интерполя-

ция, называемая часто аппроксимацией, приводит к сглаживанию неточностей исходных данных.

Аппроксимацию по методу наименьших квадратов можно выполнить с помощью универсальных программных средств символьной математики Mathematica, Maple, Derive. Решим эту задачу с помощью системы Derive 5.

Представим исходные данные в виде матрицы размером 12x2, пользуясь кнопкой **Matrix** на панели инструментов.

V	$a + bV$
7.5	22
24	68
32	108
47	137
93	255
120	315
134	390
144	405
245	685
260	700
380	1100

Первая строка матрицы содержит аргумент V функции аппроксимации и линейную функцию $a + bV$, являющуюся моделью расширения Вселенной. Числовые данные матрицы являются исходными опытными данными Хаббла.

Пусть матрица исходных данных на экране монитора находится в первой строке. Тогда функцией аппроксимации в системе Derive 5 является функция: FIT#1. После ввода этой команды матрица на экране будет представлена в следующем виде:

$$\text{FIT} \left| \begin{array}{cc} V & a + bR \\ 22 & 7.5 \\ 68 & 24 \\ \vdots & \vdots \\ 1100 & 380 \end{array} \right|$$

Решение задачи аппроксимации завершается нажатием кнопки  (Approximate). После округления чисел до двух знаков после запятой получим решение в следующем виде:

$$R = 2.82V - 1.05 \quad (8.6)$$

Задача аппроксимации легко решается с помощью математических систем Mathematica и Maple. В этих системах функции аппроксимации имеют вид:

□ Mathematica: $\text{Fit}[\{\{A\}\}, \{X\}, x]$, где:

- A — матрица исходных данных;
- X — перечень базисных переменных (в нашем случае это R);
- x — аргумент функции (т. е. V).

□ Maple 6: $\text{fit}[\text{leastsquare}[[V, R], R = f(V)] ([a, b])$, где:

- V, R — символы аргумента и функции аппроксимации;
- $R = f(V)$ — линейная функция аппроксимации $R = a + bV$;
- a, b — векторы узлов аппроксимации и значений функции (т. е. векторы V и R соответственно).

Результаты табулирования функции (8.6) приведены в табл. 8.2 (графа $\Phi_4(V)$). Значения функции, как и в случае интерполяции точной в узлах, существенно отличаются от исходных.

Погрешности модели (8.6), вычисленные по описанной выше методике, имеют значения:

□ $\varepsilon = 16.58$;

□ $\delta_{\max} = 75.3 \%$;

□ $\delta_{\min} = 1.5 \%$.

Результаты аппроксимации позволяют сделать следующие важные выводы:

- математическая модель (8.6) в результате сглаживания неточностей исходных данных позволила уменьшить погрешность аппроксимации;
- математической моделью расширения Вселенной является линейная модель (8.6), уточняющая модель Хаббла;
- значительные погрешности модели (8.6) объясняются неточностью исходных данных.

Выполненные нами исследования, в основе которых лежит интерполяция, оставляют все же сомнение в правильности выбора вида функции интерполяции.

Обратимся за помощью к программе Formula, содержащей более 1200 функций интерполяции. Программа по нашим исходным данным выдает на экран несколько наилучших функций со значениями погрешностей интер-

поляции. Возможно, среди них найдется такая, которая позволит уточнить модель Хаббла или даже ее опровергнуть.

Решение этой задачи с помощью программы Formula позволило получить, кроме линейной и квадратичной, следующие функции:

$$\square R = \exp(1.0161 \cdot 10^{-6} V^2 + 917.08 \cdot 10^{-3} \ln(V) + 1.4072);$$

$$\square R = \frac{1}{-1.1511 \cdot 10^{-9} V^2 + \frac{332.18 \cdot 10^{-3}}{V} + 201.41 \cdot 10^{-6}};$$

$$\square R = \frac{1}{-295.14 \cdot 10^{-12} V^2 + \frac{362.98 \cdot 10^{-3}}{V}}.$$

При этом точность аппроксимации по расчетам программы Formula выше, чем в случае линейной или квадратичной моделей.

У читателя может возникнуть следующий вопрос: если модели, приведенные выше, имеют малые погрешности, то, возможно, следует отказаться от полученной нами модели (8.6) и считать законом расширения Вселенной одну из приведенных моделей программы Formula, а автором модели считать саму программу Formula? Конечно, это не так. Формулы являются лишь математическими выражениями и не всегда выражают физические законы. В нашей задаче полином одиннадцатой степени даст нулевую погрешность. Однако это не значит, что он является законом расширения Вселенной.

Программа Formula, как и другие подобные программы, может быть лишь хорошим советчиком по выбору функции интерполяции. Находит математические модели и открывает физические законы человек.

Зависимость температуры кипения воды от давления

Температура кипения воды t повышается с ростом давления P . Эта зависимость приведена в табл. 8.3.

Таблица 8.3

P , атм.	t , °C	P , атм.	t , °C
1	99	31.5	236
2	120	39	248
3	133	50	263
4	143	100	310
5	151	120	324

Таблица 8.3 (продолжение)

P , атм.	t , °C	P , атм.	t , °C
6	158	140	335
7	164	150	341
8	170	170	351
9	174	200	364
10	179	220	372
16	200	225.65	374.15
20	211	-	-

Необходимо найти физический закон кипения воды при различных давлениях.

Закон будет найден, если функция $t = f(P)$ будет представлена в виде математического выражения. Для этого необходимо решить задачу интерполяции.

Методика решения задачи синтеза модели изложена ранее при определении закона расширения Вселенной. Воспользуемся этой методикой.

Выбор вида функции интерполяции

Зависимость $t = f(P)$ приведена на рис. 8.2. Функция достаточно гладкая, особенностей не имеет. Она может быть представлена большим числом различных функций. Какую же выбрать?

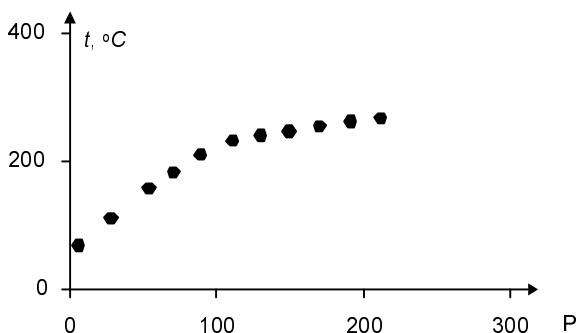


Рис. 8.2

Воспользуемся функцией ALL_SEVEN системы Derive, которая по данным таблицы решает задачу одновременно семью функциями интерполяции:

$$\square a + bx;$$

$$\square ae^{bx};$$

$$\square ax^b;$$

$$\square a + b\ln(x);$$

$$\square a + \frac{b}{x};$$

$$\square \frac{a}{b+x};$$

$$\square \frac{ax}{b+cx}.$$

Технология решения задачи с помощью функции ALL_SEVEN изложена в главе 2. Приведем результаты решения:

$$\#1: 1.09371 \cdot P + 162.562$$

$$\#2: 160.845 \cdot e^{0.00456164 \cdot P}$$

$$\#3: 101.977 \cdot P^{0.241315}$$

$$\#4: 53.6911 \ln(P) + 66.4381$$

$$\#5: 273.749 - \frac{276 \cdot 292}{P}$$

$$\#6: \frac{7.55887 \cdot 10^9}{4.41325 \cdot 10^7 - 1.2735 \cdot 10^5 \cdot P}$$

$$\#7: \frac{1.78582 \cdot 10^8 P}{5.87581 \cdot 10^5 P + 1.79386 \cdot 10^6}$$

Решим теперь задачу табуляции всех функций и по ее результатам выберем наилучшую. Табулирование осуществим функцией VECTOR, которая в нашем случае имеет вид:

VECTOR([P, #1, #2, #3, #4, #5, #6, #7], P, [#8]), где:

$\square$ #1— #7 — номера строк, в которых находятся приведенные выше функции;

$\square$ #8 — строка, в которой находится вектор P, имеющий вид:

$$[1, 2, 3, \dots, 200, 220, 225.65].$$

После ввода функции VECTOR и щелчка по кнопке  (Approximate) получим решение в виде таблицы (табл. 8.4).

Таблица 8.4

1	163.655	161.580	101.977	66.4381	−2.543	171.772	72.4695
2	164.749	162.319	120.543	103.653	135.603	172.270	116.255
3	165.843	163.061	132.934	125.423	181.651	172.772	145.573
4	166.936	163.806	142.491	140.869	204.676	173.276	166.577
5	168.030	164.555	150.374	152.850	218.490	173.783	182.365
6	169.124	165.308	157.138	162.639	227.700	174.294	194.665
7	170.217	166.063	163.093	170.916	234.278	174.807	204.518
8	171.311	166.823	168.434	178.085	239.212	175.323	212.588
9	172.405	167.585	173.290	184.409	243.049	175.842	219.319
10	173.499	168.352	177.753	190.066	246.119	176.365	225.018
16	180.061	173.023	199.101	215.301	256.480	179.566	246.652
20	184.436	176.209	210.116	227.282	259.934	181.765	254.818
31.5	197.013	185.700	234.459	251.671	264.977	188.399	267.764
30	205.216	192.163	246.859	263.138	266.664	192.993	272.392
50	217.247	202.051	262.113	276.478	268.223	200.151	276.814
100	271.933	253.814	309.836	313.694	270.986	240.736	285.014
120	293.807	278.060	323.772	323.483	271.446	261.985	286.428
140	315.681	304.621	336.043	331.760	271.775	287.348	287.447
150	326.618	318.839	341.684	335.464	271.907	301.965	287.857
170	348.492	349.296	352.162	342.184	272.123	336.165	288.534
200	381.304	400.522	366.248	350.910	272.367	404.964	289.299
220	403.178	438.781	374.769	356.027	272.493	468.947	289.695
225.65	409.357	450.237	377.069	357.389	272.524	490.856	289.795

Сравнивая данные таблицы с исходными, убеждаемся, что наилучшей функцией интерполяции является степенная вида ax^b . Ее результаты находятся в третьем слева столбце табл. 8.4, а сама функция находится в строке #3 и имеет вид:

$$t = 102 \cdot P^{0.24} \quad (8.7)$$

Погрешность интерполяции мала:

- абсолютная среднеквадратичная погрешность $\varepsilon = 1.42$;
- максимальная относительная погрешность $\delta = 1.4 \%$.

Решение задачи интерполяции позволяет нам сделать следующие важные выводы:

- математической моделью кипения воды является степенная функция вида:

$$t = aP^b;$$

- функция интерполяции (8.7) является приближенной математической моделью, позволяющей найти температуру кипения воды при любом давлении из диапазона 1—225.65 атмосфер. Функция представлена с точностью до двух знаков после запятой.
- приближенно можно считать, что температура кипения воды пропорциональна корню четвертой степени от давления, т. е.:

$$t = K \sqrt[4]{P} \quad (8.8)$$

Здесь K — коэффициент пропорциональности, равный температуре кипения воды при $P = 1$ атм.

В исходной таблице при $P = 1$ температура кипения воды равна 99°C .

Тогда $t = 99 \sqrt[4]{P}$.

Погрешности этой формулы имеют значения:

- $\varepsilon = 4.455$;
- $\delta_{\max} = 4.5 \%$.

Погрешности незначительны и объясняются неточностью исходных данных и погрешностью вычислений. Закон кипения воды (8.8) найден. Теперь его нужно доказать теоретически.

8.3. Интерполяция в экономических задачах

В экономике важную роль играют статистические данные, которые в большинстве случаев представляются в табличном виде, что позволяет лишь качественно изучать закономерности экономических явлений. Для численных оценок, установления экономических законов, прогнозирования или оптимизации необходимо представление данных в виде математических выражений и формул. Их получение возможно во многих случаях только методами интерполяции.

Приведем примеры, которые обобщают ряд задач в экономике.

Задача о цене за услуги

С целью получения хорошей выгоды бизнесмен желает востребовать за услуги как можно больше. Но этому препятствуют потребители, которые не хотят или не могут платить и отказываются от услуг. Возникает естественный вопрос: какую цену следует назначить, чтобы получить максимальную прибыль? Если запросить много, то будет мало потребителей, а если мало, то потребителей будет много, но суммарный капитал будет мал.

Возникает задача оптимизации, требующая знания потребительского спроса на услугу в зависимости от ее цены.

Эту задачу можно решить с помощью методов интерполяции, если будут известны статистические данные о спросе на услуги. Такие данные можно получить путем опроса населения.

В табл. 8.5 приведены данные опроса населения об оплате посещения лесов в зимнее время.

Таблица 8.5

	Значения переменных				
N, чел.	2625	984	150	13	269
Z, руб.	0	1–30	31–60	61–90	91 и более

В таблице обозначено:

- *N* — количество людей, согласных заплатить за посещение леса *Z* рублей;
- *Z* — цена в рублях за разовое посещение леса.

Требуется найти оптимальную цену за разовое посещение леса.

Задачу можно решить, если найти функцию $N = f(Z)$. Сложность решения этой задачи заключается в выборе вида функции интерполяции. К счастью, в математических системах имеются функции, позволяющие решить задачу интерполяции одновременно несколькими функциями. Такой функцией является функция ALL_SEVEN математической системы Derive 5.

Эта функция имеет вид: ALL_SEVEN($f(x)$), где $f(x)$ — функция, заданная в виде матрицы.

Откликом функции ALL_SEVEN является совокупность следующих семи функций:

$a + bx$, ae^{bx} , ax^b , $a + b \ln(x)$, $a + \frac{b}{x}$, $\frac{a}{b + x}$, $\frac{ax}{b + cx}$.

Технология решения задачи интерполяции подробно изложена в *разд. 2.5 главы 2*. Результатом решения задачи в системе Derive 5 являются следующие три ответа (функции интерполяции):

2008.96 – 315841*Z*

2841.2 *e*^{-0.0701325*Z*}

$\frac{3.15785 \cdot 10^4}{1.1029Z + 26.9129}$

Остальные четыре функции Derive 5 не реализовала и выдала их на экран в исходном виде.

Результаты табулирования этих функций приведены в табл. 8.6.

Таблица 8.6

Функция					
<i>a + bz</i>		<i>ae^{bz}</i>		$\frac{a}{b + z}$	
<i>Z</i>	<i>N</i>	<i>Z</i>	<i>N</i>	<i>Z</i>	<i>N</i>
0	2008.96	0	2841	0	1173.35
15	1535.19	15	992.269	15	726.67
45	587.675	45	121.027	45	412.556
75	–359.847	75	14.7618	75	288.045

В графе *Z* табл. 8.6 значение *Z* является средней ценой за посещение леса. Сравнивая данные таблиц 8.5 и 8.6, видим, что наиболее подходящей является экспоненциальная функция. Представим ее в следующем виде:

$$N = 2841e^{-0.07Z} \tag{8.9}$$

Теперь определим оптимальную цену за посещение леса. Сумма денег, которые будут получены от граждан за каждое посещение леса в зимнее время, будет равна произведению *NZ*, где:

- *N* — количество людей, посещающих лес;
- *Z* — цена услуг.

Умножая выражение (8.9) на *Z*, получим следующую функцию цены услуг:

$$Z_N = 2841Ze^{-0,07Z} \tag{8.10}$$

График этой функции представлен на рис. 8.3.

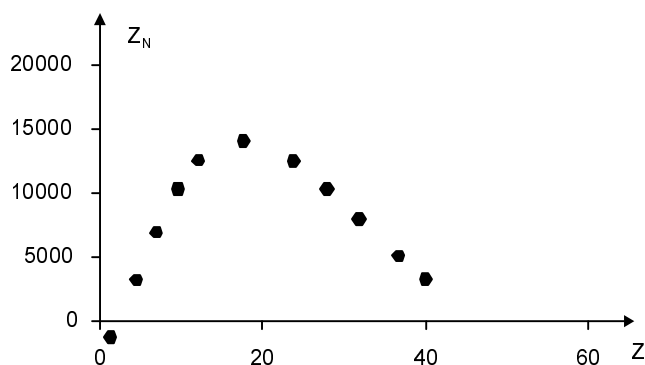


Рис.8.3

Определим координаты точки максимума. Для этого вычислим производную функции (8.10), приравняем ее к нулю и определим корень полученного уравнения.

Производная функции (8.10) имеет вид:

$$\frac{d Z_N}{d Z} = 2841 e^{-0,07Z} (1 - 0.07Z).$$

Приравнивая производную нулю и находя корень уравнения, получим оптимальную цену за разовое посещение леса. Она будет равна 14.29 руб. Подставляя это значение (вместо Z) в выражение (8.10), получим денежные средства, вырученные от граждан за разовое посещение леса в зимнее время. Выручка будет равна:

$$Z_N = 2841 \cdot e^{-0.07 \cdot 14.29} = 17776.13 \text{ руб.}$$

Расчеты выполнены в системе Derive 5. Табулирование функций, определение производной, вычисление корня уравнения осуществлено с помощью функций соответственно VECTOR, DIFFERENTIATE, SOLVE.

В *приложении П3.3* приведены данные опроса о посещении лесов для всех времен года. Они могут быть вариантами задач для студентов и других лиц, интересующихся методами интерполяции в экономике.

Задача о стоимости транспортных услуг

Стоимость транспортных услуг зависит от вида услуг и расстояния перевозки. В табл. 8.7 приведены данные стоимости перевозки леса автопоездом в зависимости от расстояния S .

Возникает естественный вопрос: какова стоимость перевозки леса автопоездом на расстояния 20, 75, 130 км или на любое другое расстояние, отсутствующее в таблице.

Таблица 8.7

	Значения переменных									
S, км	30	40	50	60	70	80	90	100	110	120
G, тыс. руб.	6.36	6.85	7.34	7.84	8.08	8.32	8.57	8.70	8.82	8.94
P, тыс. руб.	6.37	6.88	7.34	7.74	8.076	8.36	8.58	8.74	8.846	8.89

Решение этой задачи возможно путем интерполяции функции, заданной в виде таблицы.

На рис. 8.4 отображены точки функции $G(S)$, заданной табл. 8.7

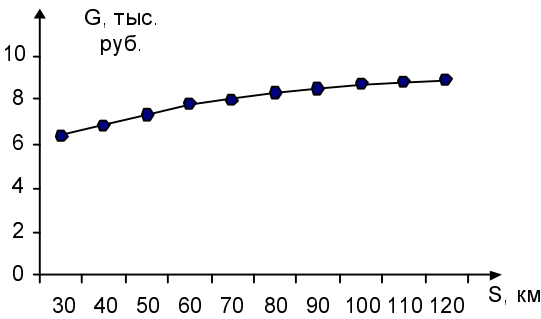


Рис.8.4

Из рисунка видно, что функция является гладкой, почти линейной. Такую функцию можно аппроксимировать полиномом n -ой степени. При этом следует ожидать, что полином будет невысокой степени.

Определим степень полинома n , воспользовавшись методом табличных разностей. Напомним, что если n -ные табличные разности равны, то степень полинома не превышает n . Легче всего вычислить табличные разности с помощью математической системы Matlab.

Эти процедуры реализуются функцией `diff(y, n)`, где:

- y — вектор функции $y(x)$;
- n — номер табличной разности.

В нашем примере вычислительные процедуры будут иметь вид:

```
G = [6.36, 6.85, 7.34, 7.84, 8.08, 8.32, 8.57, 8.70, 8.82, 8.94];  
diff(G, 1)
```

Нажимаем клавишу <Enter>

```
Ответ: 0.49 0.49 0.50 0.24 0.24 0.25 0.13 0.12 0.012  
diff(P, 2)
```

Нажимаем клавишу <Enter>

Ответ: 0.00 0.01 -0.26 0.00 0.01 -0.12 -0.01 0.108

Из полученных откликов видно, что при $n = 2$ табличные разности почти одинаковы. Это значит, что интерполяционной функцией может быть многочлен степени $n = 2$, т. е. $G = a + bs + cs^2$. Решая теперь задачу полиномиальной интерполяции с помощью любого математического средства (главы 2—6), получим:

$$G = 4.475 + 0.072S - 0.0003S^2.$$

Результаты табулирования функции $G(S)$ приведены в табл. 8.7 (графа **P**). Они хорошо согласуются с исходными данными. Доказательством адекватности модели является ее низкая погрешность.

Абсолютная среднеквадратичная погрешность $\varepsilon = 0.04$.

Максимальная относительная погрешность $\delta = 0.66 \%$.

Полученная математическая модель практически идеально моделирует оплату вывозки в зависимости от расстояния.

Задача о себестоимости вывозки леса тепловозами

Себестоимость машино-часов тепловоза серии ТУ-6А на вывозке леса в зависимости от расстояний и руководящего уклона приведена в табл. 8.8.

Таблица 8.8

Руководящий уклон, %	Показатели себестоимости машино-часов в рублях при вывозке леса в зависимости от расстояния, км				
	30	60	90	120	150
10	16.72	13.80	13.33	12.83	12.25
16	12.23	11.13	10.44	10.18	9.79
25	10.70	9.49	8.98	8.64	8.43

Необходимо найти функцию интерполяции для всех вариантов уклона и вычислить значения себестоимости при расстояниях 50, 70, 100 км.

Выбор вида функции интерполяции

На рис. 8.5 приведены графики функции, построенные по данным табл. 8.8. Из графиков видно, что при малых расстояниях функция себестоимости убывает быстрее, чем при больших, когда она почти линейна. В этом случае функцией интерполяции может быть полином n -ой степени. Степень полинома в данном случае установить достаточно трудно, т. к. число данных ограничено.

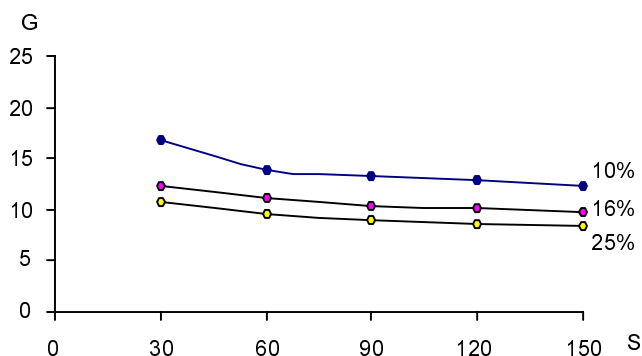


Рис. 8.5

В табл. 8.9 приведены значения себестоимости и табличные разности Δ_1 и Δ_2 для всех случаев уклона.

Таблица 8.9

Уклон	Себестоимость G в рублях и значения табличных разностей				
10	G	16.72	13.8	13.33	12.83
	Δ_1	-2.92	-0.47	-0.5	-0.58
	Δ_2	2.45	-0.03	-0.08	
16	G	12.23	11.13	10.44	10.18
	Δ_1	-1.1	-0.69	-0.26	-0.39
	Δ_2	0.32	0.43	-0.13	
25	G	10.7	9.49	8.98	8.64
	Δ_1	-1.21	-0.51	-0.34	-0.21
	Δ_2	0.7	0.17	0.13	

Из табл. 8.9 видно, что вторые табличные разности далеко не постоянны, поэтому степень полинома больше двух.

Для начала выберем $n = 3$. Тогда интерполяционный полином будет иметь вид:

$$y = a + bx + cx^2 + dx^3.$$

Решая задачу полиномиальной интерполяции, получим следующие многочлены:

$$n = 10 \quad y_{10} = -7.8 \cdot 10^{-6} x^3 + 0.002477x^2 - 0.265x + 22.626$$

$$n = 16 \quad y_{16} = -1.667 \cdot 10^{-6} x^3 + 0.000597x^2 - 0.081x + 14.184$$

$$n = 25 \quad y_{25} = -1.759 \cdot 10^{-6} x^3 + 0.000647x^2 - 0.0863x + 12.748$$

Решение получено с помощью функции FIT системы Derive. Табулирование интерполяционных полиномов осуществлено с помощью функции VECTOR. Результаты приведены в табл. 8.10.

Таблица 8.10

Руководящий уклон	Себестоимость в рублях: исходная G и полученная в результате интерполяции G(x)					
10	G	16.72	13.8	13.33	12.83	12.25
	G(x)	16.69	13.94	13.12	12.97	12.22
16	G	12.23	11.13	10.44	10.18	9.79
	G(x)	12.24	11.097	10.49	10.15	9.798
25	G	10.7	9.49	8.98	8.64	8.43
	G(x)	10.69	9.52	8.938	8.668	8.423

Из сравнения данных себестоимостей исходной и полученной в результате моделирования видно, что они отличаются незначительно (единицы копеек). Полученные нами полиномы могут быть математическими моделями зависимости себестоимости машино-часа тепловоза от расстояния.

В приложении приводятся данные по себестоимости вывозки леса автопоездами различных типов. Читатель может самостоятельно получить математические модели для этих случаев и сравнить с полученными выше результатами. Возможно, полиномиальная модель третьей степени является общей для оценки себестоимости машино-часа вывозки леса автопоездами.

Многопараметрическая задача о себестоимости вывозки леса тепловозами

Себестоимость вывозки леса тепловозами определяется двумя факторами:

- x_1 — расстояние;
- x_2 — уклон.

Откликом является себестоимость $G(x_1, x_2)$. Получим зависимость $G(x_1, x_2)$, решая задачу двухпараметрической интерполяции.

Исходные данные позволяют получить только линейную модель. Для квадратичной модели или более высокой степени у нас мало данных.

Составим план испытания. В нашей задаче расстояние вывозки леса находится в диапазоне 30—150 км, а руководящий уклон в диапазоне 10—25 %.

Тогда $30 \leq x_1 \leq 150$, $10 \leq x_2 \leq 25$. План полного факторного двухуровневого эксперимента имеет вид, представленный в табл. 8.11.

Таблица 8.11

x_1	x_2	y
30	10	16.72
30	25	10.70
150	10	12.25
150	25	8.43

Моделью такого эксперимента является линейная функция вида:

$y = b_0 + b_1x_1 + b_2x_2.$

Получим значения коэффициентов b_0, b_1, b_2 по методике, изложенной в главе 7. Можно пользоваться любым математическим универсальным программным средством. Мы решили эту задачу с помощью системы Derive 5. Ответ получен в виде следующей функции отклика:

$y = -0.028x_1 - 0.328x_2 + 20.29$ (8.11)

Результаты табулирования функции отклика приведены в табл.8.12. Для сравнения в таблице также приведены исходные данные себестоимости.

Таблица 8.12

Руководящий уклон	Себестоимость в рублях: исходная G и полученная в результате двухпараметрической линейной аппроксимации $G(x_1, x_2)$					
10	S, км	30	60	90	120	150
	G, руб.	16.72	13.8	13.33	12.83	12.25
	$G(x_1, x_2)$, руб.	16.17	15.33	14.49	13.64	12.8
16	S, км	30	60	90	120	150
	G, руб.	12.23	11.13	10.44	10.18	9.79
	$G(x_1, x_2)$, руб.	14.2	13.36	12.52	11.67	10.83
25	S, км	30	60	90	120	150
	G, руб.	10.7	9.49	8.98	8.64	8.43
	$G(x_1, x_2)$, руб.	11.25	10.4	9.57	8.72	7.88

Адекватность моделей осуществим оценкой погрешностей интерполяции. Абсолютная максимальная и относительная среднеквадратические погрешности, вычисленные по методике, описанной в главе 2, имеют значения:

$$\varepsilon_{10} = 0.99 \quad \delta_{10} = 8 \%$$

$$\varepsilon_{16} = 1.82 \quad \delta_{16} = 18.5 \%$$

$$\varepsilon_{25} = 0.6 \quad \delta_{25} = 7.6 \%$$

Погрешности достаточно велики. Полученная линейная модель отвергается. Естественно желание исследователя применить модель квадратичную:

$$y = b_0 + b_1x_1 + b_2x_2 + b_{11}x_1^2 + b_{22}x_2^2 + b_{12}x_1x_2 \quad (8.12)$$

Для ее реализации при полном факторном эксперименте необходимо иметь в каждом факторе три уровня. Тогда число опытов $N = 3^2 = 9$, а план испытания будет иметь вид, представленный в табл. 8.13.

Таблица 8.13

x_1	x_2	y
30	10	16.72
30	$x_2^{(0)}$	y_2
30	25	10.4
$x_1^{(0)}$	10	y_4
$x_1^{(0)}$	$x_2^{(0)}$	35
$x_1^{(0)}$	25	y_6
150	10	12.25
150	$x_2^{(0)}$	y_8
150	25	8.43

В теории планирования эксперимента $x_1^{(0)}$ и $x_2^{(0)}$ принимаются равными средним значениям x_1 и x_2 , т. е.:

$$x_1^{(0)} = \frac{20+150}{2} = 90, \quad x_2^{(0)} = \frac{10+25}{2} = 17.5.$$

Теперь нужно определить значения откликов y_2, y_4, y_5, y_6, y_8 . Для этого необходимо знать себестоимость машино-часа вывозки леса на расстояние 90 км при всех значениях уклона, а также себестоимость при значениях уклона 17.5 на всех значениях расстояния. Где их взять? В исходной таблице таких значений нет. Но у нас имеется линейная модель, которая позволяет вычислить себестоимость при любых значениях x_1 и x_2 из всего диапазона их изменения. Подставляя в (8.11) значения x_1 и x_2 , вычислим все необходимые нам значения y . Тогда план эксперимента будет иметь вид, представленный в табл. 8.14.

Таблица 8.14

x_1	x_2	y
30	10	16.72
30	17.5	13.71
30	25	10.7
90	10	13.33
90	17.5	12.025
90	25	8.98
150	10	12.25
150	17.5	10.34
150	25	8.43

Решая теперь задачу двухпараметрической интерполяции с помощью системы Derive 5 по технологии, описанной в главе 7, получим следующую функцию:

$y = 0.00016x^2 + 0.00122xy - 0.0785x - 0.00516y^2 - 0.2449y + 21.335$

(8.13)

Результаты табулирования функции представлены в табл. 8.15.

Таблица 8.15

Руководящий уклон	Себестоимость в рублях: исходная G и полученная в результате двухпараметрической квадратичной аппроксимации $G(x_1, x_2)$					
	S, км	30	60	90	120	150
10	G, руб.	16.72	13.8	13.33	12.83	12.25
	$G(x_1, x_2)$, руб.	16.53	14.98	13.71	12.74	12.06
16	S, км	30	60	90	120	150
	G, руб.	12.23	11.13	10.44	10.18	9.79
	$G(x_1, x_2)$, руб.	14.74	13.13	12.1	11.3	10.85
25	S, км	30	60	90	120	150
	G, руб.	10.7	9.49	8.98	8.64	8.43
	$G(x_1, x_2)$, руб.	10.7	9.7	8.98	8.56	8.43

Сравнивая результаты линейной (табл. 8.12) и квадратичной (табл.8.15) аппроксимаций видим, что погрешности квадратичной аппроксимации существенно ниже, а в случае уклона 25 % их практически нет. Значительные погрешности наблюдаются лишь при уклоне 16 %. Они возникли потому, что недостающие данные нашего "эксперимента" получены путем их расчета по линейной модели, отвергнутой нами по условиям точности.

8.4. Интерполяция в задачах таксации

Существует общегосударственная система нормативов таксации леса. Этими нормативами обязаны руководствоваться все организации и предприятия лесного комплекса. В таксационных справочниках приводятся таблицы хода роста древостоев различного сорта и плотности древесины, сортиментные и товарные таблицы, таблицы выхода фанерного и другого сырья, а также многое другое.

Таблицы получены в результате многолетних обширных исследований десятков тысяч деревьев, произрастающих на многих тысячах площадей леса. Выполнен анализ статистических данных и получены математические модели среднего диаметра древостоев по группам насаждений, относительных их полнот и запасов, хода роста древостоев, сортиментных и товарных таблиц. Математические модели получены методами интерполяции, статистического и корреляционного анализа.

Задачи таксации столь обширны, что полученные математические модели не могут охватить все закономерности жизни леса. Существует много еще нерешенных задач. Интерполяция, как научная основа моделирования, играет не последнюю роль в их решении.

Рассмотрим на примерах применение методов интерполяции в задачах таксации. Примеры возьмем из таблиц хода роста древостоев.

Пример 8.1

Пусть зависимость высоты и диаметра березы бонитета 1 от возраста приведена в табл. 8.16.

Таблица 8.16

	Зависимость высоты и диаметра березы от возраста								
L, лет	20	30	40	50	60	70	80	90	100
H, м	9.8	12.8	15.8	18.6	21.3	23.5	25.3	26.5	27.4
D, см	6.9	9.9	13.5	17.8	21.6	24.6	27.2	29	29.5

В таблице приняты обозначения:

- L — возраст березы;
- H — средняя высота древостоя;
- D — средний диаметр древостоя.

Необходимо найти математическую модель зависимости возраста березы от ее высоты и диаметра: $L = f(H, D)$.

Выбор вида функции регрессии

Основанием для выбора вида функции регрессии является технический справочник [25], в котором ход роста древостоев лесных массивов описывается линейными или квадратичными функциями. Последуем советам справочника и выберем линейную и квадратичную функции двухпараметрической интерполяции:

$$L = b_0 + b_1 H + b_2 D,$$

$$L = b_0 + b_1 H + b_2 D + b_{11} H^2 + b_{22} D^2 + b_{12} HD$$

Окончательный вид функции выберем на основании анализа погрешностей аппроксимации.

Сравнительный анализ универсальных математических программных средств показывает, что задачи многопараметрической интерполяции наиболее просто решаются с помощью систем Mathematica и Derive. Выберем систему Derive, в которой переход от линейной функции аппроксимации к квадратичной осуществляется лишь путем редактирования первой строки матрицы функции FIT.

После ввода матрицы исходных данных и использования команды FIT (см. главу 7) на экране монитора получим следующую матрицу:

FIT	H	D	$b_0 + b_1 H + b_2 D$
	9.8	6.9	20
	12.9	9.9	30
	15.8	13.5	40
	18.6	17.8	50
	21.3	21.6	60
	23.5	24.6	70
	25.3	27.2	80
	26.5	29	90
	27.4	29.5	100

Щелкнем мышью по кнопке  (Approximate). На экране увидим ответ в виде следующей линейной функции:

$$L = 1.52d + 2.265h - 15.98.$$

Результаты табулирования этой функции при значениях h и d приведены в табл. 8.17 с обозначением $\varphi_{L_{\text{лин}}}$.

Таблица 8.17

	Результаты табулирования функций интерполяции								
L	20	30	40	50	60	70	80	90	100
$\varphi_{L_{\text{лин}}}$	16.7	28.07	40.35	53.23	65.13	74.67	82.7	88.16	91
$\varphi_{L_{\text{кв}}}$	20.07	30.2	40.4	48.15	59.8	71.65	81.69	88.5	99.5

Из таблицы видно, что линейная модель имеет значительные погрешности (например, при $L = 20$ и $L = 100$). Вычислим погрешности аппроксимации по формулам:

$$\square \text{ абсолютная среднеквадратичная } \varepsilon = \sqrt{\frac{\sum_{i=1}^N (L_i - \varphi_i)^2}{N}};$$

$\square$ относительная максимальная и минимальная погрешности:

$$\delta = \frac{\varepsilon}{L_{\min}}, \quad \delta = \frac{\varepsilon}{L_{\max}}.$$

В формулах приняты обозначения:

$\square N = 9$ (число опытов);

$\square L_{\min} = 20$;

$\square L_{\max} = 100$.

Вычислительные процедуры здесь очевидны, они подробно описаны в главе 2.

В результате вычислений получим:

$\square \varepsilon = 4.28$;

$\square \delta_{\max} = 21.4 \%$;

$\square \delta_{\min} = 4.3 \%$.

Большая максимальная относительная погрешность линейной модели является основанием для поиска иной функции интерполяции. Такой функцией может быть квадратичная. Получить квадратичную модель в системе Derive исключительно просто. В первой строке функции FIT необходимо отредактировать выражение:

$b_0 + b_1H + b_2D$, заменив его выражением:

$b_0 + b_1H + b_2D + b_{11}H^2 + b_{22}D^2 + b_{12}HD,$

затем щелкнуть мышью по кнопке  (Approximate).

В результате получим следующую квадратичную функцию:

$L = -5.506769D^2 + 1.266DH - 10.988D - 0.611H^2 + 11.74H - 22.$

Результаты табулирования функции $\varphi_{L_{KB}}$ приведены в табл. 8.17.

Погрешности квадратичной модели имеют значения:

- $\varepsilon = 1.14;$
- $\delta_{\max} = 5.7 \ %;$
- $\delta_{\min} = 1.14 \ %.$

Из анализа данных табл. 8.17 и значений погрешностей можно сделать вывод, что квадратичная модель хорошо описывает зависимость возраста березы от ее высоты и диаметра.

Теперь можно по замеренным значениям высоты и диаметра березы определить ее возраст.

Пусть, например, $H = 14$ м, $D = 12$ см. Подставляя эти данные в выражения математических моделей, получим: $L = 34$ года в случае линейной модели и $L = 30.5$ лет в случае модели квадратичной.

При $H = 22$ м, $D = 23$ см — соответственно $L = 68.8$ и $L = 60.6$.

При $H = 26$ м., $D = 28$ см — соответственно $L = 85.5$ и $L = 87.3$.

Из этих расчетов видно, что возраст березы, рассчитанный по линейной и квадратичной моделям, находится в диапазоне возрастов, приведенных в исходной таблице. Отличаются они лишь численными значениями. При этом более точным является возраст березы, рассчитанный по квадратичной модели.

Пример 8.2

Пусть зависимость среднего прироста G древостоя ели от его возраста L , высоты H и диаметра D приведена в табл. 8.18.

Таблица 8.18

	Зависимость прироста древостоя от возраста, высоты и диаметра												
G, м³	3.7	4.2	4.6	4.9	5.1	5.2	5.1	5.1	5	4.9	4.8	4.7	4.4
L, лет	20	30	40	50	60	70	80	90	100	110	120	130	140
H, м	6.4	9.8	13.4	16.8	19.5	21.9	23.8	25.6	27.4	29	29.9	30.8	31.4
D, см	6.7	9.3	12.9	16.5	19.6	22.9	26.4	28.4	30.7	31.6	32.9	34.3	35.1

Из таблицы видно, что средний прирост ели с возрастом дерева растет, а затем, начиная с $L = 100$ лет, убывает. Это происходит за счет уменьшения с возрастом количества деревьев в лесу. Зависимость $G = f(L, H, D)$ в таксационных справочниках отсутствует. Мы получим ее впервые. Здесь имеет место случай трехпараметрической интерполяции. Получим линейную и квадратичную модели:

$$\square G_{\text{л}} = b_0 + b_1 L + b_2 H + b_3 D;$$

$$\square G_{\text{кв}} = b_0 + b_1 L + b_2 H + b_3 D + b_{11} L^2 + b_{22} H^2 + b_{33} D^2 + b_{12} LH + b_{13} LD + b_{23} HD$$

Методика решения задачи аналогична предыдущему примеру. Изменится лишь матрица команды FIT. Теперь она будет иметь следующий вид:

	L	H	D	$b_0 + b_1 L + b_2 H + b_3 D$
	20	6.4	6.7	3.7
	30	9.8	9.3	4.2
	40	13.4	12.9	4.6
	50	16.8	16.5	4.9
	60	19.5	19.6	5.1
	70	21.9	22.9	5.2
FIT	80	23.8	26.4	5.1
	90	25.6	28.4	5.1
	100	27.4	30.7	5
	110	29	31.6	4.9
	120	29.9	32.9	4.8
	130	30.8	34.3	4.7
	140	31.4	35.1	4.4

После нажатия кнопки  (Approximate) на панели инструментов Derive получим следующую модель:

$$G_{\text{л}} = -0.084D + 0.3H - 0.037L + 3.15.$$

Редактируем первую строку матрицы функции FIT, заменяя линейную функцию на следующую квадратичную:

$$b_0 + b_1 L + b_2 H + b_3 D + b_{11} L^2 + b_{22} H^2 + b_{33} D^2 + b_{12} LH + b_{13} LD + b_{23} HD.$$

Вновь нажимаем кнопку  (Approximate). Получим следующее решение:

$$G_{\text{кв}} = -0.15D^2 + 0.223DH + 0.029DL + 0.0444D - 0.0876H^2 - 0.022HL + 0.117H - 0.00145L^2 + 0.05L + 1.98$$

В табл. 8.19 приведены исходные значения прироста G , полученные в результате табулирования функции линейной $G_{\text{лин}}$ и квадратичной $G_{\text{кв}}$ моделей.

Таблица 8.19

	Значения прироста древостоя по данным линейной и квадратичной моделей												
G, м³	3.7	4.2	4.6	4.9	5.1	5.2	5.1	5.1	5	4.9	4.8	4.7	4.4
Гл, м³	3.75	4.18	4.58	4.92	5.09	5.16	5.06	5.05	5.03	5.06	4.85	4.63	4.37
Гкв, м³	3.7	4.19	4.62	4.91	5.08	5.21	5.08	5.12	5.0	4.9	4.81	4.68	4.41

Из сравнения данных таблицы видно, что линейная и квадратичная модели хорошо описывают зависимость прироста древостоя от возраста, средней высоты и среднего диаметра ели. Данные квадратичной модели практически совпадают с исходными данными.

Абсолютная среднеквадратическая погрешность ε , максимальная относительная $\delta_{\max}$ и минимальная относительная $\delta_{\min}$ имеют следующие значения:

□ для линейной модели:

- $\varepsilon = 0.058$;
- $\delta_{\max} = 1.58 \%$;
- $\delta_{\min} = 1.12 \%$;

□ для квадратичной модели:

- $\varepsilon = 0.014$;
- $\delta_{\max} = 0.37 \%$;
- $\delta_{\min} = 0.27 \%$.

В *приложении П4* приведены статистические данные хода роста древостоев сосновых насаждений, сформулированы задачи однопараметрической и двухпараметрической интерполяций. По данным таблицы можно также сформулировать задачи трехпараметрической интерполяции. Решение этих задач полезно при изучении проблем таксации и методов интерполяции. При этом решение можно выполнить не только с помощью Derive 5, но также использовать любую математическую систему, описанную в *главе 7*.

8.5. Интерполяция в задачах массового обслуживания

Решение практических задач часто требует интерполяции математических функций иными функциями, более удобными для решения задачи. Наиболее часто такая задача встречается в следующих случаях:

□ необходимость упрощения расчетов;

- ☐ необходимость физической реализуемости функции в инженерной практике;
- ☐ установление закономерностей изучаемых явлений;
- ☐ создание математических моделей объектов.

Рассмотрим одну из таких задач, встречающихся в теории массового обслуживания.

Зависимость числа обслуживающих бригад от интенсивности потока задач и интенсивности их обслуживания

Многоканальная система массового обслуживания с отказами является типичной в нашей жизни. Примерами таких систем являются: больница с ограниченным числом коек, платная стоянка для автомобилей с ограниченным числом мест, системы обслуживания пассажиров с ограниченным числом мест в транспортном средстве, телефонные узлы связи с ограниченным числом линий связи и т. п.

Функционирование такой системы массового обслуживания с отказами описывается графом состояний (рис. 8.6). Узлам этого графа приписаны состояния системы, а ветвям — интенсивности перехода из состояния в состояние.

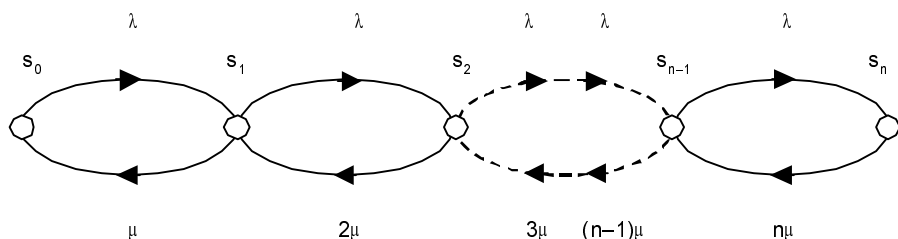


Рис.8.6

Система в любой момент времени может находиться в одном из следующих состояний:

- S_0 — в системе заявок нет;
- S_1 — в системе одна заявка (один канал занят обслуживанием, остальные свободны);
- S_{n-1} — в системе $n - 1$ заявка ($n - 1$ канал занят обслуживанием и только один канал свободен);
- S_n — в системе n заявок (все каналы заняты обслуживанием, очередной заявке будет отказано в обслуживании).

Интенсивности переходов λ и μ имеют следующий смысл:

- λ — есть интенсивность появления заявки, равная количеству заявок в единицу времени;
- μ — интенсивность обслуживания заявки, равная количеству заявок обслуженных в единицу времени.

Так как время появления заявки является случайным, то состояния системы массового обслуживания характеризуются вероятностями состояний.

Критериями эффективности таких систем являются:

- P — вероятность отказа заявке в обслуживании;
- Q — пропускная способность системы;
- K — среднее число каналов, занятых обслуживанием заявок.

Показатели эффективности, соответствующие этим критериям, вычисляются по формулам [28-30]:

$$P = \frac{\rho^n}{n!} P_0 \quad (8.14)$$

$$Q = \lambda(1 - P) \quad (8.15)$$

$$K = \rho(1 - P) \quad (8.16)$$

В этих показателях приняты следующие обозначения:

- λ — интенсивность потока заявок (ед/час);
- $P_0 = \frac{1}{\sum_{i=0}^n \frac{\rho^i}{i!}}$ — вероятность того, что в системе заявок нет;
- $\rho = \frac{\lambda}{\mu}$ — коэффициент передачи ветви;

□ μ — интенсивность обслуживания заявки (ед/час);

□ n — число каналов обслуживания.

Подставляя в (8.14) значение P_0 , получим:

$$P = \frac{\rho^n}{n! \sum_{i=0}^n \frac{\rho^i}{i!}} \quad (8.17)$$

Из формулы (8.17) видно, что с увеличением числа обслуживающих каналов n вероятность отказа заявке в обслуживании убывает.

Физически также понятно, что чем выше интенсивность потока заявок λ и чем ниже интенсивность обслуживания заявки, тем выше вероятность отка-

за очередной заявке в обслуживании. Однако получить зависимости в явном виде между переменными n и λ или n и μ невозможно в связи со сложностью формулы (8.17).

Возникают проблемы практического характера. Например, на сколько необходимо увеличить число обслуживающих каналов n , чтобы при увеличении интенсивности потока заявок λ вероятность отказа в обслуживании P не изменилась. Во сколько раз надо увеличить интенсивность обслуживания заявок, чтобы при прочих равных условиях уменьшить число обслуживающих каналов в m раз.

Эти и многие другие задачи, имеющие важное практическое значение, требуют для их решения знания зависимости между аргументами функции в явном виде. Эти зависимости могут быть получены методами интерполяции. Методика состоит в следующем.

Необходимые зависимости получают путем численного решения уравнения (8.17). В результате решения искомую функцию представляют в виде таблицы. Дальнейшее очевидно. Находим функцию и метод интерполяции, затем математическую модель в виде аналитического выражения.

Пользуясь этой методикой, решим следующую важную для практики задачу.

Допустим, что необходимо найти зависимость $n = f(\rho)$ в явном виде, которая обеспечит постоянную вероятность отказа в обслуживании при изменении интенсивности потока заявок или интенсивности обслуживания.

Уточним формулировку задачи. Пусть вероятность отказа заявке $P = 0.01$, т. е. очередная заявка на обслуживание будет принята в любое случайное время t с вероятностью 0.99. Интенсивность обслуживания заявки $\mu = 2$ ед/час. Необходимо найти зависимость $n = f(\rho)$ при изменении λ в диапазоне $\lambda = 1—5$ ед/час с шагом 0.5. Функцию $n = f(\rho)$ получим, решая следующее уравнение:

$$0.01 = \frac{\rho^n}{n! \sum_{i=0}^n \frac{\rho^i}{i!}} \quad (8.18)$$

Решение уравнения выполним численным методом с помощью функции SOLVE системы Derive 5, которая имеет вид: SOLVE($f(x)$).

Технология решения уравнения в системе Derive:

□ ввод уравнения в символьном виде:

$$p = \frac{\left(\frac{\lambda}{\mu}\right)^n}{n! \sum_{i=0}^n \frac{(\lambda/\mu)^i}{i!}};$$

- подстановка в формулу численных значений P , λ , μ с помощью кнопки **Sub** панели инструментов;
- определение корня уравнения с помощью функции $\text{SOLVE}(\#k)$, где $\#k$ — номер уравнения, находящегося в строке k монитора.

Повторяя эти процедуры столько раз, сколько значений λ , получим функцию $n = f(\rho)$ в виде таблицы.

В табл. 8.20 приведены значения n при изменении ρ в диапазоне 0.5—5. Следует иметь в виду, что количество обслуживающих бригад является числом целым. Однако на конкретном этапе исследования это не имеет значения, поскольку данные нам необходимы только для получения функции интерполяции.

Таблица 8.20

	Значения переменных									
λ	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
μ	1	1	1	1	1	1	1	1	1	1
ρ	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
n	3.12	4.28	5.26	6.16	7	7.8	8.5	9.3	10	10.7
$\varphi(\rho)$	3.21	4.23	5.19	6.12	6.99	7.81	8.59	9.32	10	10.644

На рис. 8.7 приведена зависимость $n = f(\rho)$ в виде точек табл. 8.20.

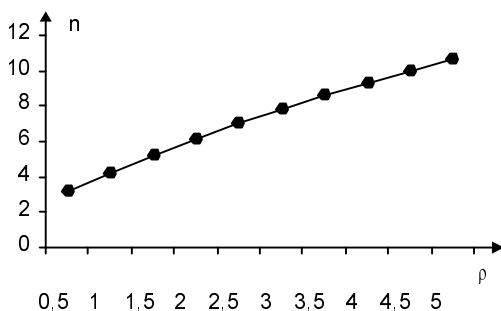


Рис. 8.7

Из рисунка видно, что функция достаточно гладкая, без особенностей. Она несколько отличается от прямой. Выберем в качестве интерполяционного полином второй степени: $n = a + b\rho + c\rho^2$. Определение значений n получено приближенным методом, поэтому возможны погрешности.

Решая задачу интерполяции приближенную в узлах с помощью функции FIT по известной уже нам методике, получим следующую функцию:

$$n = -0.095\rho^2 + 2.17\rho + 2.148 \quad (8.19)$$

Табулируя эту функцию в диапазоне всех значений ρ и сравнивая с табличными значениями (табл. 8.20) убеждаемся, что они отличаются незначительно. На рис. 8.8 приведены зависимости $n = f(\rho)$, построенные по данным таблицы и с помощью формулы (8.19). Из рисунка видно, что они практически совпадают.

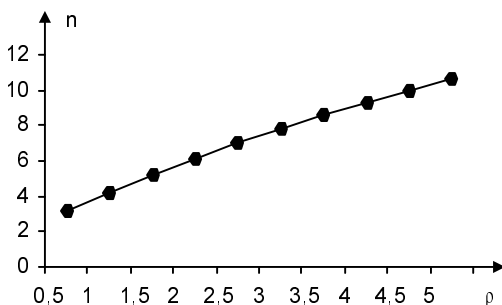


Рис.8.8

Определим погрешности формулы (8.19) по методике, изложенной в главе 2. Вычисления дают следующие погрешности функции $\varphi(\rho)$:

- абсолютная среднеквадратическая погрешность $\varepsilon=0.054$
- максимальная относительная погрешность $\delta_{\max} = 1.7 \%$.

Значение полученного решения методом интерполяции трудно переоценить. Теперь легко определить необходимое количество коек в больнице, мест на платной стоянке, самолетовылетов и т. п., обеспечивающее высокую вероятность (99 %) того, что заявка будет обслужена: больной попадет в больницу, автомобилист поставит машину на платную стоянку, пассажир вовремя прибудет на пункт назначения.

Освоить компьютерные технологии интерполяции можно только путем решения задач. В приложениях приведены практические задачи теории массового обслуживания, требующие решения методами интерполяции. Решая эти задачи, можно научиться многому: решать на ЭВМ уравнения, табулировать функции, выполнять действия с векторами и матрицами, определять погрешности расчетов.

Многопараметрическая интерполяция в задачах массового обслуживания

Функционирование системы управления воздушным движением аэропорта можно представить как многоканальную систему массового обслуживания

(СМО) с отказами. В качестве заявок в этой системе можно рассматривать поток воздушных судов (ВС). Исследования показывают, что интенсивность потока ВС — величина постоянная. Обслуживающим органом в этой системе является диспетчерский пункт, который характеризуется интенсивностью обслуживания μ и числом самолетов n , одновременно обслуживаемых диспетчером. Функционирование такой системы можно описать графом состояний (рис. 8.6). Узлы графа имеют в данном случае следующий смысл:

- S_0 — диспетчер свободен от обслуживания ВС;
- S_i — диспетчер обслуживает i воздушных судов ($i = 1, 2, \dots, n$);
- n — максимальное число воздушных судов, обслуживаемых диспетчером.

Пусть $P_i(t)$ — вероятность того, что в произвольный момент времени t система находится в i -ом состоянии.

Тогда $P_n(t)$ — вероятность того, что диспетчер обслуживает максимально допустимое число ВС. Если при этом появляется дополнительное воздушное судно, то ему будет отказано в обслуживании.

Функционирование данной системы можно описать следующей системой дифференциальных уравнений [28–30]:

$$\frac{dP_0(t)}{dt} = -\lambda P_0(t) + \mu P_1(t)$$

$$\frac{dP_i(t)}{dt} = \lambda P_{i-1}(t) - (\lambda + i\mu)P_i(t) + (i+1)\mu P_{i+1}(t), \quad i = 1, 2, \dots, n-1.$$

$$\frac{dP_n(t)}{dt} = \lambda P_{n-1}(t) - n\mu P_n(t)$$

Вероятности всех состояний, получаемые в результате решения системы уравнений, полностью характеризуют ее эффективность. При этом возникает следующая важная задача: какова длительность τ переходного процесса системы и как она зависит от параметров λ , μ , n ?

Значение τ необходимо знать для оценки загруженности диспетчера, пропускной способности системы, оценки вероятности отказа в обслуживании воздушного судна и ряда других показателей.

Длительность переходного процесса зависит от интенсивности воздушного движения λ , интенсивности обслуживания воздушного судна μ , от числа самолетов n , одновременно обслуживаемых диспетчером. Определение функции $\tau = f(\lambda, \mu, n)$ является задачей многопараметрической интерполяции.

Анализ функционирования систем управления воздушным движением показывает, что диапазон изменения факторов наших аэропортов достаточно широк и имеет следующие значения:

$$0.3 \leq \lambda \leq 13.5 \text{ (ед/час)}, \quad 3 \leq \mu \leq 15 \text{ (ед/час)}, \quad 3 \leq n \leq 10.$$

Составим план испытания. В нашем случае $n = 3$, число уровней в каждом факторе $P=2$. Тогда число опытов при полнофакторном эксперименте будет $N = P^n = 8$. Такой план испытания позволяет получить линейную модель вида: $y = b_0 + b_1\lambda + b_2\mu + b_3n$.

Для квадратичной модели нужно число опытов:

$$C_{n+2}^2 = \frac{(n+2)(n+1)}{2} = 10.$$

Квадратичную модель можно реализовать, увеличивая число уровней в каждом факторе или используя дробно-факторный эксперимент. Ограничимся в дальнейшем линейной моделью. Тогда план испытания будет иметь вид:

Таблица 8.21

№ опыта	λ ед/час	μ ед/час	n	τ , час
1	0.3	3	3	0.3
2	0.3	3	10	0.84
3	0.3	15	3	0.04
4	0.3	15	10	0.56
5	13.5	3	3	1.3
6	13.5	3	10	1.8
7	13.5	15	3	1
8	13.5	15	10	1.5

Значение τ получено нами путем решения системы дифференциальных уравнений методом Рунге–Кутты. Установившееся значение τ определялось по значениям вероятностей $P_0(t)$.

Признаком установившегося процесса принималось следующее условие:

$$|P_0(t + \Delta t) - P_0(t)| \leq E, \text{ где } E \text{ — оценка погрешности в определении } \tau.$$

Выполнение этого условия означает, что наступил установившийся процесс и $\tau = t$. Значения τ для всех диапазонов изменения переменных λ , μ , n приведены в таблице 8.21.

Теперь решим задачу многопараметрической аппроксимации. Решение получим с помощью универсального математического программного средства Mathematica.

Программа на языке Mathematica будет иметь вид:

$$M = \{\{0.3, 3, 3, 0.3\}, \{0.3, 3, 10, 0.84\}, \{0.3, 15, 3, 0.04\}\},$$

$$\{0.3, 15, 10, 0.56\}, \{13.5, 3, 3, 1.3\}, \{13.5, 3, 10, 1.8\},$$

$$\{13.5, 15, 3, 1\}, \{13.5, 15, 10, 1.5\};$$

$$X = \{a, x, y, z\};$$

$$\text{Fit}[M, X, \{x, y, z\}]$$

После одновременного нажатия клавиш <Shift>+<Enter> получим следующую линейную функцию регрессии:

$$\tau = 0.149 + 0.0731x - 0.024y + 0.0736z \quad (8.19)$$

Значения длительности переходных процессов, полученные путем решения дифференциальных уравнений $\tau_{\text{ду}}$ и табулирования функции регрессии $\tau_{\text{рег}}$, приведены в табл. 8.22.

Таблица 8.22

	Значения длительности переходных процессов							
$\tau_{\text{ду}}, \text{ час}$	0.3	0.84	0.04	0.56	1.3	1.8	1	1.5
$\tau_{\text{рег}}, \text{ час}$	0.32	0.83	0.033	0.55	1.28	1.8	1	1.52

Из таблицы видно, что функция регрессии хорошо описывает искомую зависимость:

$$\tau = f(\lambda, \mu, n).$$

Полученные результаты трудно переоценить. Теперь определить длительность переходных процессов в системе управления воздушным движением можно с помощью калькулятора и даже без него. Интерполяция облегчила наш труд в сотни раз.

8.6. Интерполяция в задачах надежности

Основными критериями надежности технических систем являются:

- $P(t)$ — вероятность безотказной работы системы в течение времени t ;
- T_I — среднее время безотказной работы;
- $\lambda(t)$ — интенсивность отказа в момент времени t ;
- $f(t)$ — частота отказа (плотность распределения времени до отказа).

Эти показатели связаны между собой зависимостями:

$$\left. \begin{aligned} P(t) &= e^{-\int_0^t \lambda(t) dt} \\ T &= \int_0^{\infty} P(t) dt \\ f(t) &= \lambda(t) P(t) \end{aligned} \right\} \quad (8.20)$$

Из выражений (8.20) видно, что все показатели надежности могут быть определены, если известна интенсивность отказа системы $\lambda(t)$.

Интенсивность отказа системы определяется по формуле:

$$\lambda(t) = \sum_{i=1}^n \lambda_i(t) \quad (8.21)$$

Здесь:

□ $\lambda_i(t)$ — интенсивность отказа i -го элемента;

□ n — число элементов системы.

Из (8.20) и (8.21) видно, что если известны интенсивности отказов элементов и время непрерывной работы системы, то определить ее показатели надежности нетрудно.

Интенсивности отказов элементов получают либо по данным определительных испытаний, либо по данным об их отказах в процессе эксплуатации системы. При этом результаты представляются в виде таблиц, определяющих зависимость $\lambda_i = f(t)$. По этим данным по формулам (8.20) невозможно вычислить показатели надежности системы. Необходимо иметь функции $\lambda_i(t)$ в виде аналитических выражений. Интерполяция опытных данных и является тем методом, который позволяет получить математические функции, определяющие зависимость интенсивности отказов элементов от времени.

8.6.1. Оценка надежности техники по опытными данным

Рассмотрим методику на примере.

На испытание поставлено 200 образцов техники. Опыт проводился в течение времени $t = 2000$ час. За это время зафиксировано 36 отказов. Их распределение во времени приведено в табл. 8.23.

Таблица 8.23

	Зависимость числа отказов от времени									
Δt , час	200	200	200	200	200	200	200	200	200	200
$n(\Delta t)$	1	2	2	4	3	4	4	5	5	6

В таблице приняты следующие обозначения:

□ Δt — промежуток времени, в течение которого фиксировались отказы;

□ $n(\Delta t)$ — число отказов в промежутке Δt .

Необходимо по данным таблицы определить интенсивность отказов техники и по формулам (8.20) вычислить показатели ее надежности.

Решение задачи можно выполнить с помощью любой из систем, описанных в главах 2—6. Мы рассматриваем решения, полученные посредством системы Derive 5.

Получим методом интерполяции функцию $\lambda(t)$ в виде формулы. Для этого определим значения интенсивности отказов на всех промежутках Δt и представим их в табличной форме.

Интенсивность отказов $\lambda(\Delta t)$ вычисляется по формуле:

$$\lambda(\Delta t) = \frac{n(\Delta t)}{N_{\text{ср}} \Delta t} \quad (8.22)$$

Здесь:

$N_{\text{ср}}$ — среднее число исправных образцов техники на промежутке времени Δt , вычисляемое по выражению $N_{\text{ср}} = \frac{N_{\text{н}} + N_{\text{к}}}{2}$, где:

$N_{\text{н}}$ — число исправных образцов в начале промежутка Δt ,

$N_{\text{к}}$ — число исправных образцов в конце промежутка времени Δt .

Значения $\lambda(\Delta t)$, вычисленные по формуле (8.22), приведены в табл. 8.24.

Таблица 8.24

	Зависимость интенсивности отказов от времени									
t , час	100	300	500	700	900	1100	1300	1500	1700	1900
$\lambda \cdot 10^{-5}$, ед/час	2.5	5.05	5.1	10.36	7.9	10.75	11	14.1	14.5	18

Отказы техники являются событиями случайными. Они могут возникать в любое время промежутка Δt . В табл. 8.23 указано время t в середине промежутков Δt .

Выбор вида функции интерполяции

Зависимость $\lambda = f(t)$, соответствующая данным табл. 8.24, приведена на рис. 8.9 в виде точек.

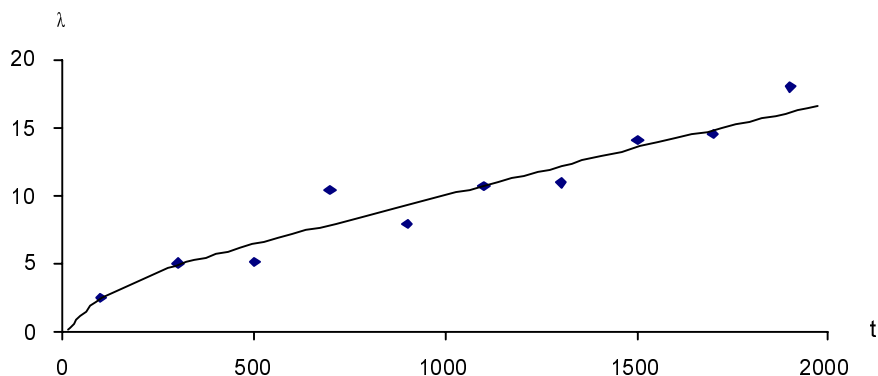


Рис. 8.9

Сложность выбора вида функции интерполяции — одна из особенностей решения задач надежности. Не всякая функция, если она с высокой точностью аппроксимирует табличные данные, может быть функцией интерполяции $\lambda = f(t)$.

Отказы, как явления случайные, подчиняются определенным законам. Одним из них является необходимость обеспечения следующего равенства:

$\int_0^{\infty} f(t)dt = 1$. Из этого равенства следует, что функция $\lambda(t)$ должна быть такой, чтобы соответствующий ей закон распределения времени до отказа $f(t)$ удовлетворял приведенному условию.

Из рис. 8.9 видно, что интенсивность отказов техники является возрастающей функцией времени и может быть представлена полиномом первой или второй степени. Однако полиномиальные функции принципиально не могут быть в данной задаче функциями интерполяции, т. к. при этом не может быть обеспечено приведенное выше интегральное условие.

Какую же функцию следует выбрать в качестве интерполяционной? Интерполяционными в задачах надежности могут быть функции, соответствующие законам распределения времени до отказа, применяемым в теории надежности.

Таковыми в нашем случае могут быть функции Гаусса, Гамма, Вейбулла. Выберем в качестве аппроксимирующего закон Вейбулла. Плотность распределения $f(t)$ закона Вейбулла имеет вид:

$$f(t) = akt^{k-1}e^{-\lambda t^k} \quad (8.23)$$

Интенсивность отказов, соответствующая этому закону, выражается формулой:

$$\lambda(t) = akt^{k-1} \quad (8.24)$$

При значении $k > 1$ функция $\lambda(t)$ является возрастающей, что соответствует данным табл. 8.24. Это является основанием считать, что функция (8.24) может быть интерполяционной.

Выбор метода интерполяции

Исходные данные в силу ограниченности объема и времени испытаний являются приближенными. Поэтому целесообразно использовать метод интерполяции приближенный в узлах. К сожалению, функции и команды интерполяции универсальных математических программных средств символьной математики не позволяют определить по табличным данным коэффициенты a и k функции (8.24). Это объясняется нелинейностью функции относительно неизвестной переменной k . Поэтому воспользуемся методом интерполяции точным в узлах.

Определение коэффициентов a и k функции $\lambda(t)$

Выберем в качестве узлов интерполяции значения $t = 300$ и $t = 1500$.

Тогда система уравнений на основании данных табл. 8.24 будет иметь вид:

$$\left. \begin{aligned} ak300^{k-1} &= 5.05 \cdot 10^{-5} \\ ak1500^{k-1} &= 14.1 \cdot 10^{-5} \end{aligned} \right\} \quad (8.25)$$

В результате решения этой системы нелинейных уравнений, получим:

$$\square a = 0.81 \cdot 10^{-6};$$

$$\square k = 1.64.$$

Подставляя эти значения в (8.24), получим искомую функцию интерполяции интенсивности отказов техники:

$$\lambda(t) = 1.328 \cdot 10^{-6} t^{0.64} \quad (8.26)$$

График функции $\lambda(t)$ и табличные данные интенсивности отказов приведены на рис. 8.9.

Исходные данные интенсивности отказов λ и результаты табулирования функции (8.26) $\lambda(t)$ приведены в табл. 8.25.

Из сравнения данных видно, что они значительно отличаются друг от друга, погрешность интерполяционной формулы велика:

$$\square \text{ абсолютная среднеквадратическая погрешность } \varepsilon = 1.43 \cdot 10^{-5};$$

$$\square \text{ максимальная относительная погрешность } \delta_{\max} = 57 \%;$$

$$\square \text{ минимальная относительная погрешность } \delta_{\min} = 8 \%.$$

Таблица 8.25

	Зависимость интенсивностей отказов от времени									
t , час	100	300	500	700	900	1100	1300	1500	1700	1900
$\lambda \cdot 10^{-5}$, ед/час	2.5	5.05	5.1	10.36	7.9	10.75	11	14.1	14.5	18
$\lambda(t) \cdot 10^{-5}$, ед/час	2.53	5.11	7.09	8.79	10.3	11.7	13.1	14.3	15.5	16.7

Вычислим показатели надежности исследуемой техники по формулам (8.20):

$$P(t) = e^{-\int_0^t \lambda(t) dt} = e^{-1.328 \cdot 10^{-6} \int_0^t t^{0.64} dt} = e^{-0.81 \cdot 10^{-6} t^{1.64}}$$

$$T_1 = \int_0^{\infty} P(t) dt = \int_0^{\infty} e^{-0.81 \cdot 10^{-6} t^{1.64}} dt = 122.8 \text{ час.}$$

$$f(t) = \lambda(t) P(t) = 1.328 \cdot 10^{-6} t^{0.64} e^{-0.81 \cdot 10^{-6} t^{1.64}}$$

Сформулированную нами задачу можно также решить, используя статистические универсальные математические средства Statistica, Statgraphics и др., позволяющие по статистическим данным определить плотность распределения случайной величины — времени до отказа.

Интенсивность отказов элементов мала, она находится в пределах 10^{-5} — 10^{-7} ед/час. Для получения данных о надежности таких элементов с точностью, необходимой для практики (по меньшей мере два знака после запятой), требуется большой объем и достаточное время испытания. При этом в процессе испытания трудно реализовать реальные условия работы элементов. По этим причинам данные об отказах элементов, полученные из опыта, имеют большие погрешности. Вычисленные показатели надежности сложной системы по этим данным часто содержат недопустимо большие ошибки. Это мы видели и в нашем примере.

Единственным способом получения характеристик надежности элементов с необходимой точностью является сбор и обработка данных об их отказах в процессе эксплуатации техники. В следующем разделе приводится методика и пример применения методов интерполяции при оценке надежности техники по данным о ее отказах в процессе эксплуатации.

8.6.2. Определение показателей надежности по данным эксплуатации восстанавливаемой техники

В настоящем разделе описана методика оценки надежности техники по данным ее эксплуатации. При этом предполагается, что при возникновении

отказа в процессе эксплуатации техника восстанавливается. Такой случай является типичным для большинства технических систем.

Проблема здесь состоит в следующем. Для определения общепринятых показателей надежности (таких, как вероятность и среднее время безотказной работы) необходимо знать интенсивность отказов элементов. Без этих данных рассчитать надежность в процессе проектирования сложной технической системы невозможно. Интенсивность отказов является характеристикой надежности невосстанавливаемой техники до первого отказа. Для ее получения необходимо проводить опыт с отбрасыванием отказавших элементов, при этом число элементов в опыте убывает с течением времени.

При эксплуатации восстанавливаемых систем отказавшие элементы заменяются новыми или отремонтированными. Число элементов остается постоянным. Здесь опыт проводится с заменой элементов вместо отказавших.

Далее приводится разработанная нами методика и компьютерная технология определения интенсивности отказов элементов и других показателей надежности по данным эксплуатации восстанавливаемых систем.

Зависимость между показателями надежности невосстанавливаемой и восстанавливаемой техники существует и определяется следующим уравнением Вольтерра второго рода:

$$\omega(t) = f(t) + \int_0^t \omega(\tau) f(t - \tau) d\tau. \quad (8.27)$$

Функция $f(t)$ является плотностью распределения времени до отказа невосстанавливаемой техники.

Функция $\omega(t)$ — это параметр потока отказов восстанавливаемой техники. Она определяется по формуле:

$$\omega(t) = \frac{n(\Delta t)}{N_0 \Delta t} \quad (8.28)$$

Здесь:

- $n(\Delta t)$ — число отказов восстанавливаемого устройства за промежуток времени Δt ;
- N_0 — число устройств, находящихся в эксплуатации.

Из интегрального уравнения (8.27) видно, что для определения плотности распределения времени до отказа $f(t)$, а значит и для всех остальных показателей надежности невосстанавливаемой техники, необходимо иметь аналитическое выражение функции $\omega(t)$. Его можно получить по данным об отказах восстанавливаемой техники в процессе ее эксплуатации, применяя методы интерполяции.

Методика получения показателей надежности по данным об отказах восстанавливаемой техники в процессе ее эксплуатации состоит в следующем.

1. Интегральное уравнение (8.27) представляется в преобразованиях Лапласа:

$$\omega(s) = f(s) + \omega(s) f(s) \text{ или } f(s) = \frac{\omega(s)}{1 + \omega(s)}. \quad (8.29)$$

2. По данным об отказах восстанавливаемой техники определяется по формуле (8.28) параметр потока отказов $\omega(t)$. Решение представляется в табличной форме.
3. Решается задача интерполяции, в результате которой находится аналитическое выражение функции $\omega(t)$.
4. Находится преобразование Лапласа функции $\omega(t)$.
5. Определяется по формуле (8.29) преобразование Лапласа плотности распределения времени до отказа $f(s)$.
6. Находится обратное преобразование Лапласа функции $f(s)$. В результате получаем плотность распределения времени до отказа $f(t)$.
7. По $f(t)$ вычисляются все показатели надежности невосстанавливаемой системы:

$$\left. \begin{aligned} P(t) &= 1 - \int_0^t f(t) dt \\ \lambda(t) &= \frac{f(t)}{P(t)} \\ T_1 &= \int_0^{\infty} P(t) dt \end{aligned} \right\} \quad (8.30)$$

Рассмотрим предложенную методику на примере оценки надежности авиационной техники.

В табл. 8.26 приведены данные об отказах восьми самолетов ТУ-154 М в течение двух лет их эксплуатации.

Таблица 8.26

	Данные об отказах самолета ТУ–154 М							
Δt , час	2351	4066	4596	3381	2630	3665	4585	3158
$n(\Delta t)$	260	431	421	306	351	369	546	363
$\omega \cdot 10^{-3}$, ед/час	13.8	13.2	11.4	11.3	16.6	12.5	14.9	14.4

В таблице также приведены значения параметра потока отказов $\omega(t)$, вычисленные по формуле (8.28). Обозначения в таблице имеют следующий смысл:

- Δt — часы налета воздушных судов в течение каждого квартала двухлетней эксплуатации самолетов;
- $n(\Delta t)$ — число отказов восьми воздушных судов в течение квартала;
- ω — параметр потока отказов в диапазоне Δt , т. е. в промежутке времени налета ВС в каждом квартале.

Для получения функции интерполяции табл. 8.26 должна быть уточнена.

Функция $\omega(t)$ при $t = 0$ также равна нулю, т. к. эксплуатация самолета начинается с момента времени, когда он исправный, т. е. $\omega(0) = 0$. Количество отказов фиксировалось в течение квартала. Отказы возникали с начала эксплуатации и до конца в данном квартале. В связи с этим целесообразно считать, что число отказов относится к середине данного квартала. Так, например, 260 отказов в первом квартале относится ко времени $\frac{2351}{2} = 1175.5$ час., 431 отказ относится ко времени $2351 + \frac{4066}{2} = 4384$ час и т. д. Тогда значения можно представить в табл. 8.27.

Таблица 8.27

	Значения числа отказов и параметра потока отказов самолета во времени								
Δt , час	0	1175	4384	8714	12702	15708	18855	22980	26858
$n(\Delta t)$	0	260	431	421	306	351	369	546	363
$\omega \cdot 10^{-3}$, ед/час	0	13.8	13.2	11.4	11.3	16.6	12.5	14.9	14.4

Зависимость $\omega = f(t)$ может быть аппроксимирована полиномом второй степени:

$$\omega = a + bt + ct^2.$$

Поскольку отказы являются явлениями случайными, то наиболее целесообразно для получения функции $\omega(t)$ использовать интерполяцию приближенную в узлах. Эту задачу можно решить с помощью любой математической системы, описанной в главах 2—6. Мы представляем задачу, решенную с помощью системы Mathematica. Получена следующая функция интерполяции:

$$\omega(t) = 0.00687 + 7.95 \cdot 10^{-7} t - 2.004 \cdot 10^{-11} t^2.$$

По приведенной выше методике и известной функции $\omega(t)$ легко получить показатели надежности самолета ТУ-154 М. Приведем лишь конечный результат, представляющий для нас наибольший интерес.

Вероятность безотказной работы самолета ТУ-154 М определяется следующей формулой:

$$P(t) = 1.018e^{-0.0068t} - 0.019e^{-0.000156t} + 0.001e^{-0.000038t}$$

(8.31)

Результаты табулирования функции $P(t)$ приведены в табл. 8.28.

Таблица 8.28

	Значения вероятности безотказной работы самолета от времени											
t , час	0	4	10	20	30	40	50	60	70	80	90	100
$P(t)$	1	0.97	0.93	0.87	0.81	0.76	0.71	0.66	0.61	0.57	0.53	0.5

Из таблицы видно, что анализируемые самолеты ТУ-154 М недостаточно надежны. Так, например, вероятность безотказной работы самолета в течение времени $t = 4$ час (время полета на среднее расстояние) составляет 0.97, а вероятность безотказной работы в течение 100 часов эксплуатации равна 0.5. Следует при этом иметь в виду, что не все отказы ведут к тяжелым последствиям типа аварий или катастроф. Такие отказы редкие. Большинство отказов не опасны для пассажиров. Однако они требуют ремонта и снижают готовность самолета к вылету, удорожая при этом его эксплуатацию. Такие отказы для пассажира означают увеличение цены на билеты и повышают вероятность задержки рейса.

В *приложении 5* приведены статистические данные об отказах систем самолета ТУ-154 М. Специалисту по эксплуатации самолетов, инженеру, проектирующему самолетные системы, весьма полезно самостоятельно решить задачу интерполяции по определению функции $\omega(t)$ и определить показатели надежности системы.

Решение задачи интерполяции и определение надежности систем самолета ТУ-154 М позволит специалисту изучить методику оценки надежности восстанавливаемых систем по данным их эксплуатации, которая является достаточно универсальной. Кроме того, расчеты позволят оценить надежность самолета по показателям работоспособности его систем и дадут возможность выявить причины, влияющие на эти факторы.

Серьезные исследования ожидают специалиста по расчету надежности воздушных судов гражданской авиации.

8.7. Интерполяция в обучении

Широкое использование компьютерных технологий в образовании объясняется многими фактами, основными из которых являются массовость обучения и высокая информативность.

В наше время компьютерные технологии являются хорошим стимулом для повышения интереса к приобретению учащимися знаний, поскольку позволяют существенно уменьшить непроизводительный труд студента (создание графиков, отчетов, выполнение сложных расчетов и многое другое). Имеется достаточное число компьютерных обучающих программ — практически по всем предметам учебных планов инженерных специальностей. Они могут значительно увеличить эффективность учебного процесса.

Интерполяция в этих условиях является одной из научных основ, позволяющих оценить качество обучения в ВУЗе.

Обучение вообще и компьютерное в частности невозможно без контроля знаний. При этом традиционные методы контроля уже не пригодны для оценки знаний обучаемых. Четырехбалльная система (неудовлетворительно, удовлетворительно, хорошо, отлично) изжила себя. Тестирование знаний в процессе обучения стало для преподавателя затруднительным, ему на помощь должен прийти компьютер.

В процессе тестирования знаний ЭВМ задает большое число вопросов, при этом фиксируется количество верных и неверных ответов. При такой системе практически невозможно оценить знания по четырехбалльной системе. Нужна иная, например, стобалльная система, которая позволяет не только оценить знания учащегося, но также установить его рейтинг.

Интерполяция данных дает возможность преподавателю научно обосновано оценить знания студента, установить его рейтинг, найти взаимосвязь между традиционными и рейтинговыми оценками.

Большое число лабораторных работ студентов связано с получением табличных данных и построением графиков. Это работы по физике, химии, электротехнике, таксации и многим другим общетехническим и специальным дисциплинам. Здесь без интерполяции не обойтись.

Закономерность явлений природы становится открытием лишь тогда, когда она представлена в виде математической модели. И здесь не обойтись без интерполяции, позволяющей найти по данным эксперимента математическую модель в виде формул.

Эффективность обучения можно оценить при помощи компьютерных технологий. При этом весьма полезной, а при функционировании рейтинговой системы в ВУЗе — даже необходимой, является оценка знаний группы студентов, целого потока, факультета или всего учебного заведения. Такая система позволяет оценить состояние учебной работы, методику преподавания, качество подготовки студентов и многое другое. Закономерность этих процессов опять же позволяет установить интерполяция.

Рассмотрим задачи интерполяции, позволяющие научно обосновано оценить знания учащихся по результатам их тестирования.

Оценка знаний баллами рейтинг-системы. Постановка задачи

Дано:

n — число вопросов контроля знаний;

x_i , ($i = 1, 2, \dots, n$) — балл, получаемый учащимся по i -тому вопросу в традиционной системе оценки знаний.

Надо определить оценку знаний баллами рейтинг-системы.

Решим задачу при следующих допущениях:

- ☐ обучаемый не аттестуется при оценке неудовлетворительно;
- ☐ традиционными оценками знаний являются: удовлетворительно, хорошо, отлично, кодируемые соответственно числами 3, 4, 5.

Решение задачи

Обозначим:

☐ x — балл в традиционной системе оценки знаний;

☐ z — балл в рейтинг-системе, соответствующий баллу x .

Тогда зависимость $z(x)$ может быть представлена в табл. 8.29.

Таблица 8.29

	Соответствие баллов традиционной и рейтинговой систем оценки знаний		
x	3	4	5
z	z_3	z_4	z_5

Теперь задача состоит в определении математической модели в виде функции $z = f(x)$, и решается она методами интерполяции. Особенность данной задачи состоит в том, что таблица соответствия в строке z не содержит числовых значений.

Решение будем производить в последовательности, принятой нами в предыдущих главах.

Выбор вида функции интерполяции

К оценке "отлично" предъявляются более высокие требования, чем к оценке "хорошо", а тем более "удовлетворительно". Это является основанием предполагать, что функция $z = f(x)$ не должна быть линейной. Так как таблица соответствия представлена только наполовину в числовом виде, то все методы выбора вида функции интерполяции, рассмотренные в предыдущих главах, здесь не могут быть применены. Исследователь вправе выбрать любую из функций. Предположим, что функция $z = f(x)$ представлена полиномом n -ой степени. Выбор степени полинома здесь очевиден — в таблице

соответствия содержатся только три значения узлов интерполяции. Тогда степень интерполяционного полинома $n = 2$ будет иметь вид:

$$\varphi(x) = a + bx + cx^2.$$

Выбор метода интерполяции

Наши данные не являются ни опытными, ни расчетными, поэтому они не содержат ошибок. Это дает нам право использовать точные методы интерполяции. Наиболее удобным из них является классический метод сведения задачи интерполяции к решению системы линейных алгебраических уравнений. В этом случае система будет иметь вид:

$$\left. \begin{aligned} z_3 &= a + 3b + 9c \\ z_4 &= a + 4b + 16c \\ z_5 &= a + 5b + 25c \end{aligned} \right\} \quad (8.32)$$

Здесь также можно применить полиномиальную интерполяцию, приближенную в узлах. В случае, когда число неизвестных интерполяционного полинома (a, b, c) равно числу узлов (x_1, x_2, x_3), можно получить точное решение.

Выбор универсального программного средства для решения задачи

Исследователь может выбрать любое программное средство из рассмотренных нами в предыдущих главах, если он применит точный метод, связанный с решением системы (8.32). Не рекомендуем только использовать систему Matlab, т. к. могут возникнуть серьезные трудности решения задачи интерполяции с символьными переменными.

При решении задачи полиномиальной интерполяции приближенным методом наименьших квадратов можно использовать одну из следующих математических систем: Mathematica, Maple, Derive, которые относятся к средствам символьной математики.

Решение задачи

Решим задачу с помощью математической системы Derive 5 методом наименьших квадратов, воспользовавшись функцией FIT(4). Рассмотрим последовательность решения задачи.

1. Введем матрицу A , соответствующую таблице 8.29 размером 4×2 . После нажатия клавиши <Enter> на экране появится матрица в следующем виде:

$$\#1 : \left| \begin{array}{cc} x & a + bx + cx^2 \\ 3 & z_3 \\ 4 & z_4 \\ 5 & z_5 \end{array} \right|$$

2. Введем функцию: $\text{FIT}(\#1)$.

На экране отобразится выражение:

$$\#2 : \text{FIT} \left[\begin{array}{cc} x & a + bx + cx^2 \\ 3 & z_3 \\ 4 & z_4 \\ 5 & z_5 \end{array} \right]$$

3. Выполним команду Approximate (кнопка  на панели инструментов).

На экране отобразится ответ:

$$\varphi(x) = \frac{z_3(x^2 - 9x + 20)}{2} - z_4(x^2 - 8x + 15) + \frac{z_5(x^2 - 7x + 12)}{2} \quad (8.33)$$

В системе Mathematica задача решается с помощью функции $\text{Fit}[\{\{A\}\}, \{X\}, x]$, где:

□ A — матрица;

□ X — перечень базисных переменных (в нашем случае x, x^2);

□ x — аргумент функции.

В системе Maple интерполяция приближенная в узлах также решается с помощью функции fit , имеющей вид:

$$\text{fit}[\text{leastsquare}[[x, y], y = \varphi(x)]([a, b])$$

Здесь:

□ x, y — символы аргумента и функции аппроксимации (в нашем случае x и z);

□ $\varphi(x)$ — функция аппроксимации $a + bx + cx^2$;

□ a, b — векторы узлов аппроксимации и значений функции. В нашем случае $a := [3, 4, 5]$; $b := [z_3, z_4, z_5]$.

При традиционных методах контроля знаний оценка обычно выставляется по значению среднего балла, полученного обучаемым при ответе на n вопросов. Количество средних значений при ответе на n вопросов будет $2n+1$. При этом среднее значение балла традиционной системы будет иметь одно из значений:

$$x_{\text{ср}} = \frac{3n+k}{n} = 3 + \frac{k}{n}, \quad k = 0, 1, 2, \dots, 2n \quad (8.34)$$

Здесь:

□ n — число заданных вопросов,

□ k — сумма баллов, каждый из которых является дополнением к оценке 3 по любому из вопросов.

Подставляя в (8.33) вместо x его среднее значение, получим следующее выражение для оценки знаний баллами рейтинг-системы:

$$\varphi(x) = z_3(1 - 1.5\gamma + 0.5\gamma^2) + z_4\gamma(2 - \gamma) + 0.5z_5\gamma(\gamma - 1) \quad (8.35)$$

Здесь $\gamma = \frac{k}{n}$.

Важнейшей особенностью полученной функции интерполяции $\varphi(x)$ является ее независимость от диапазона баллов рейтинг-системы. Это может быть стобалльная, двенадцатибалльная или любая другая система. Достаточно только знать таблицу соответствия баллов рейтинг-системы и традиционной системы оценки знаний. Нами предлагается табл. 8.30, полученная методами экспертных оценок и итераций.

Таблица 8.30

	Соответствие баллов рейтинговой и традиционной систем оценки знаний		
x	3	4	5
z	51–73	74–90	91–100

На основании этой таблицы экзаменатор может установить рейтинг студента в достаточно широком диапазоне баллов. Формула (8.35), полученная методами интерполяции, устраняет этот произвол и позволяет объективно оценить знания учащегося баллами рейтинг-системы по стобалльной шкале. При этом существует множество вариантов выбора значений z_3 , z_4 , z_5 , из которого целесообразно выделить следующие три наиболее предпочтительные:

- ☐ пессимистический — значениям x , соответствующим оценкам 3, 4, 5, приписываются нижние значения z , отвечающие баллам 51, 75, 100. Этот вариант соответствует высоким требованиям экзаменатора.
- ☐ оптимистический — значениям x , соответствующим оценкам 3, 4, 5, приписываются верхние значения z , отвечающие баллам 75, 85, 100.
- ☐ усредненный — значения баллов соответствуют средним значениям диапазона z , т. е. $z_3 = 63$, $z_4 = 80$, $z_5 = 100$.

Во всех предлагаемых вариантах принято $z_3 = 100$, что необходимо для реализации стабильной системы в любом из вариантов.

Подставляя в (8.35) значения z_3 , z_4 , z_5 , получим основные формулы оценки знаний баллами рейтинг системы:

- ☐ вариант пессимистический:

$$z = 51 + 25.5\gamma - 0.5\gamma^2 \quad (8.36)$$

□ вариант усредненный:

$$z = 63 + 16.5\gamma + \gamma^2 \quad (8.37)$$

□ вариант оптимистический:

$$z = 75 + 7.5\gamma + 2.5\gamma^2 \quad (8.38)$$

Интерполяционный полином (8.35) не учитывает того, что учащийся может по одному из вопросов получить неудовлетворительную оценку, но, тем не менее, быть аттестованным, т. к. по остальным вопросам был дан положительный ответ. В этом случае зависимость $z = f(x)$ представляется в табл. 8.31.

Таблица 8.31

	Зависимость традиционной и рейтинговой систем оценки знаний			
x	2	3	4	5
z	z_2	z_3	z_4	z_5

Читателю предлагается самостоятельно решить задачу интерполяции для этого случая. Ответом будет следующая функция:

$$\varphi(x) = -z_2(0.17x^3 - 2x^2 + 7.8x - 10) + z_3(0.5x^3 - 5.5x^2 + 19x - 20) - z_4(0.5x^3 - 5x^2 + 15.5x - 15) + z_5(0.17x^3 - 1.5x^2 + 4.33x - 4)$$

Компьютерные технологии оценки знаний

При оценке знаний наиболее часто используется выборочный метод контроля, когда обучаемому задается вопрос и предоставляется m возможных ответов, из которых только один (реже несколько) верный. Учащийся должен указать номер верного ответа. При таком методе возможны только два исхода: верно (оценка 5) и неверно (оценка 0). В этом случае воспользоваться полученными выше формулами невозможно.

Здесь интерполяционным полиномом может быть многочлен третьей степени:

$$Z = a_0 + a_1x + a_2x^2 + a_3x \quad (8.39)$$

Читателю предлагается вновь решить задачу интерполяции описанным выше методом.

Ответом будет:

$$Z = \frac{z_3x(x^2 - 9x + 20)}{6} - \frac{z_4x(x^2 - 8x + 15)}{4} + \frac{z_5x(x^2 - 7x + 12)}{10} \quad (8.40)$$

Подставляя в (8.37) значения x из таблицы 8.31, получим следующие формулы оценки знаний баллами рейтинг-системы:

□ вариант пессимистический:

$$Z = -0.5x^3 + 5.5x^2 + 5x \quad (8.41)$$

□ вариант усредненный:

$$Z = 0.375x^3 - 3.5x^2 + 28.125x \quad (8.42)$$

□ вариант оптимистический:

$$Z = 1.25x^3 - 12.5x^2 + 51.25x \quad (8.43)$$

Пусть:

n — число заданных вопросов,

k — число вопросов, на которые получен верный ответ.

Тогда:

средний балл будет: $x_{\text{ср}} = \frac{5k}{n}$, $k = 0, 1, 2, \dots, n$,

число средних будет равно $n+1$.

Подставляя в формулы оценки значений $x_{\text{ср}}$, получим:

□ пессимистический вариант:

$$Z = -62.5\gamma^3 + 137.5\gamma^2 + 25\gamma \quad (8.44)$$

□ вариант усредненный:

$$Z = 46.875\gamma^3 - 87.5\gamma^2 + 140.625\gamma \quad (8.45)$$

□ оптимистический вариант:

$$Z = 156.25\gamma^3 - 312.5\gamma^2 + 256.25\gamma \quad (8.46)$$

Полученные здесь с помощью методов интерполяции математические модели могут использоваться как при традиционных, так и при компьютерных технологиях оценки знаний учащихся. При этом может использоваться любая балльная система оценок.

Закономерность получения знаний по закону Парето

Приведем еще один пример из области образования. Многие социальные, информационные, биологические и другие процессы описываются гиперболическим законом Парето:

$$n(x, N) = \frac{A}{x^{1+\alpha}} \quad (8.47)$$

В общем случае параметры этого закона имеют следующий смысл:

□ N — множество элементов процесса;

□ x — некоторый параметр множества;

□ $n(x, N)$ — число элементов множества, обладающих параметром x ;

□ A, α — параметры закона Парето.

Графически функция (8.47) имеет вид, показанный на рис. 8.10.

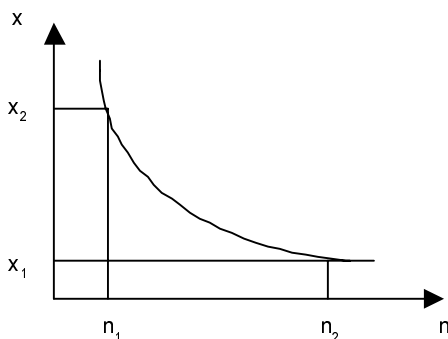


Рис. 8.10

Примерами явлений, подчиняющихся этому закону, являются: число городов n с населением x , число богатых людей с богатством x , число ученых n и их научная продукция x и т. п.

Во всех таких случаях при $n_2 > n_1$ справедливо: $x_2 > x_1$

Этот закон хорошо описывает закономерность получения знаний студентами. Оценку "отлично" получает малое число студентов, "хорошо" — большее число, а оценку "удовлетворительно" — наибольшее.

Такое положение можно оспаривать. В ряде случаев студенты имеют высокие оценки и большинство из них получают оценки "хорошо" и "отлично". Бывают случаи и противоположные: большинство студентов получают оценки "удовлетворительно" и даже "неудовлетворительно". Такие случаи означают, что учебный процесс идет плохо.

В первом случае, возможно, преподаватель чересчур либеральный или слишком мало учебного материала преподнесено студентам. Во втором случае — либо преподаватель излишне строгий, либо слабо поставлена методическая работа (отсутствие литературы, необеспеченность лаборатории, слабая посещаемость занятий). Но во всех случаях, когда нарушается закон Парето, обучение имеет изъяны.

Закон Парето может быть индикатором процесса обучения. Следует при этом иметь в виду, что оценка на экзамене еще не является оценкой знаний. Если студент получил отличную оценку, то это означает только то, что он усвоил учебный материал на отлично, но не означает, что у него отличные знания предмета.

Оценки успеваемости студентов представляют собой функцию $x = f(n)$, заданную таблицей. Получить ее аналитическое выражение можно только методами интерполяции. Рассмотрим пример из нашей практики.

Статистика знаний 101 студента приведена в табл. 8.32.

Таблица 8.32

x	3			4			5		
Z	51—58	59—66	67—74	75—80	81—86	87—90	91—94	95—98	99—100
$z_{\text{ср}}$	54.5	62.5	70.5	77.5	83.5	88.5	92.5	96.5	99.5
n	26	19	14	13	9	5	8	4	3
$\varphi(n)$	59.74	64.93	70.4	71.8	79.2	92.6	81.7	98.2	106

В таблице приняты обозначения:

- x — успеваемость студента при традиционной системе оценки знаний (удовлетворительно, хорошо, отлично);
- z — диапазон баллов в рейтинг-системе оценки знаний, соответствующей баллам традиционной системы;
- $z_{\text{ср}}$ — среднее число баллов диапазона z ;
- n — число студентов, получивших соответствующий балл из диапазона z ;
- $\varphi(n)$ — баллы рейтинг-системы, вычисленные по функции интерполяции, которая будет получена далее.

Получим зависимость $z_{\text{ср}} = f(n)$. Оценки знаний не объективны. Причин здесь много: ограниченное число вопросов контроля, случайность выбора экзаменационного билета, субъективность оценки преподавателем и т. п. В связи с этим применение точных методов интерполяции нецелесообразно. Воспользуемся интерполяцией приближенной в узлах.

На рис. 8.11 приведена зависимость $z_{\text{ср}} = f(n)$. Из рисунка видно, что функция близка к параболической и это дает нам основание ее аппроксимировать функцией Парето.

Случай здесь сложный, т. к. параболическая функция Парето является функцией нелинейной относительно определяемого параметра α . В связи с этим функциями интерполяции, имеющимися в универсальных программных средствах и оптимизирующих искомые функции по критерию наименьших квадратов воспользоваться нельзя.

К счастью, функция Парето может быть линеаризована путем логарифмирования.

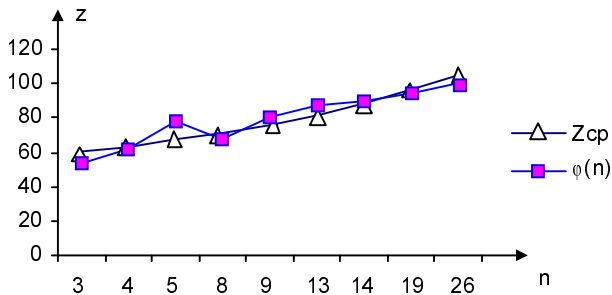


Рис. 8.11

В нашем случае аргументом x функции (8.47) является число студентов из множества $N = 101$, получивших соответствующий балл ($n(x, N)$ — балл z_{cp} , полученный группой студентов из множества баллов от 51 до 100).

Тогда функция (8.47) может быть записана в виде: $z_{cp} = \frac{A}{n^k}$, где $k = 1 + \alpha$.

Логарифмируя эту функцию получим:

$$\ln(z_{cp}) = \ln(A) - k \ln(n)$$

или, обозначая:

$$\ln(z_{cp}) = Z, \quad \ln(n) = N,$$

окончательно получим:

$$Z = \ln(A) - kN. \tag{8.48}$$

Теперь можно воспользоваться линейной аппроксимацией, преобразовав исходные данные z_{cp} и n в логарифмический масштаб.

Данные сведены в табл. 8.33.

Таблица 8.33

	Значения переменных в логарифмическом масштабе								
Z	4	4.135	4.256	4.35	4.425	4.483	4.527	4.57	4.6
N	3.258	2.944	2.64	2.565	2.197	1.61	2.079	1.386	1.097

Воспользуемся теперь функцией FIT системы Derive. Исходная матрица будет иметь на экране дисплея вид:

FIT

N

B - kN

3.258

4

2.944

4.135

1.097

4.6

После команды Approximate (кнопка  на панели инструментов) получаем решение в виде следующей линейной функции:

$$X = 4.955 - 0.2655N \quad (8.49)$$

Из решения получим параметры закона Парето: k и A .

Так как $\ln(A) = 4.955$, то $A = 141.88$, $k = 0.2655$.

Тогда функция интерполяции будет иметь вид:

$$Z_{cp} = \frac{141.88}{n^{0.2655}} \quad (8.50)$$

Результаты табулирования функции приведены в табл. 8.32 в строке $\varphi(n)$. Сравнение значений функций $\varphi(n)$ и z_{cp} показывает, что функция (8.50) может быть интерполяционной. Таким образом, закон Парето может быть математической моделью оценки знаний студентов баллами рейтинг-системы. Оценим адекватность модели, для чего вычислим абсолютную (ε) и относительную (δ) погрешности аппроксимации по формулам:

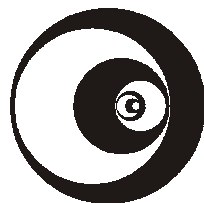
$$\varepsilon = \sqrt{\frac{\sum_{i=1}^N (Z_{cpi} - \varphi_{n,i})^2}{N}}, \quad \delta = \frac{\varepsilon}{Z_{cp}}$$

Вычисления дают следующие результаты:

$$\varepsilon = 5.4, \quad \delta_{\max} = \frac{5.4}{54.5} 100 = 9.9 \%, \quad \delta_{\min} = \frac{5.4}{99.5} 100 = 5.43 \%$$

Адекватность доказана: погрешность вполне допустима для такой модели как оценка знаний. В этом также можно убедиться по виду графиков на рис. 8.11. Исходные данные обозначены точками, сплошной линией обозначена функция интерполяции.

Глава 9



Автоматизация решения задач интерполяции

Компьютерные технологии решения задач с помощью универсальных программных средств символьной математики, описанные в *главах 2—6*, не освобождают пользователя от необходимости выбора метода и функции интерполяции, доказательства адекватности полученной модели.

Эти этапы являются наиболее творческими и трудными. Естественно желание авторов универсальных математических программных средств создать такие системы, которые бы полностью автоматизировали процессы решения задач интерполяции. Такими системами можно считать программные средства TableCurve, CurveExpert, SIMPLE FORMULA.

Получение математических моделей в этих программах реализуется следующим образом. Исследователь вводит числовые исходные данные и получает ответ в виде большого количества (сотни и тысячи) функций интерполяции, которые являются математическими моделями изучаемого объекта. Из множества решений программа предлагает несколько функций, погрешность которых мала. Исследователю остается только выбрать из них наилучшую. Он полностью освобождается от выбора метода и вида функции интерполяции, доказательства адекватности решения.

На первый взгляд кажется, что теперь решение задачи полностью автоматизировано и нет необходимости обращаться к таким системам как Mathematica, Maple, Derive, Mathcad, Matlab. В действительности это не так. Подобная автоматизация имеет следующие недостатки:

- критерий минимальной погрешности является неудовлетворительным для выбора математической модели (функции интерполяции);
- в программах реализуется единственный метод интерполяции: приближенный в узлах (метод аппроксимации по критерию минимума среднеквадратической ошибки).

Эти недостатки могут привести к ошибочным решениям: математическая модель, выбранная из множества решений, может оказаться не самой лучшей или может вообще не являться моделью объекта. Об этом подробно излагается в примерах решения задач интерполяции.

Рассмотрим весьма кратко программные средства автоматизации и технологию решения задач с помощью этих средств.

9.1. Программа TableCurve 2D

Ввод данных

Для ввода данных необходимо вызвать пункт **Edit** главного меню, а затем в появившемся окне — пункт **TableCurve Editor**. На экране появится пустая таблица со столбцами **X**, **Y**, **Weights**, курсор займет положение в первой ячейке столбца **X**.

Ввод данных осуществляется последовательным набором чисел в ячейки **X**, **Y**, **Weights**. После записи каждого числа нажимается клавиша <Enter> и курсор перемещается в очередную ячейку. Кнопки **AutoEntry X**, **AutoEntry Y**, **AutoEntry W** служат для обеспечения автоматического ввода чисел X , Y , $Weights$, если шаг переменной постоянный. Автоматический ввод обеспечивается активизацией соответствующей кнопки (щелчок мыши по ней, после чего появляется значок $\surd$). Если все числа X , Y равнозначны, то в ячейки **Weights**, после ее активизации, будут автоматически записываться единицы.

Перемещение курсора по ячейкам возможно несколькими способами:

- ☐ нажатием клавиши <Enter>;
- ☐ щелчком мыши по соответствующей ячейке таблицы или кнопке, расположенной в столбце справа от таблицы: $\uparrow$ (вверх), $\downarrow$ (вниз), $\leftarrow$ (влево), $\rightarrow$ (вправо).

Ячейки ввода нумеруются (столбец **XY#**) и могут быть помечены знаком $\surd$ на кнопках столбца **E_x**. Все это необходимо при работе с большими массивами данных.

Справа от таблицы расположена полоса прокрутки и кнопки следующих основных функций и команд:

- ☐ **OK** — окончание ввода и переход к очередному окну, например, окну сохранения данных **TableCurve Data is Not Saved**;
- ☐ **Cancel** — переход к окну **TableCurve 2D**;
- ☐ **Help** — помощь пользователю;
- ☐ **Calculation** — вызов окна **Calculations**;
- ☐ **Save** — сохранение данных;
- ☐ **Graph** — графическое отображение данных в виде точек;

- ☐ **Titles** — вызов окна записи имени таблицы и осей **X,Y**;
 - ☐ **Sort Table** — сортировка данных по возрастанию или убыванию номеров строк;
 - ☐ **Reverse X,Y** — реверсирование столбцов **X,Y**;
 - ☐ **Clear All** — вызов окна удаления данных **Clear All X-Y Data**.
- Внизу окна **TableCurve Editor** имеются шесть следующих кнопок:
- ☐ **Delete** — удалить строку данных, указанную курсором;
 - ☐ **Insert** — вставить пустую ячейку данных;
 - ☐ **Next** — переместить курсор по ячейкам таблицы;
 - ☐ **Copy** — поместить копию выделенного объекта в буфер обмена;
 - ☐ **Cut** — вырезать выделенный объект и поместить в буфер обмена;
 - ☐ **Paste** — вклеить содержимое буфера обмена.

Сохранение данных

Нажатие кнопки **OK** после ввода исходных данных откроет окно **TableCurve Data**. Пользователю предлагается сохранить исходные данные. При нажатии кнопки **Yes** появится диалоговое окно **Save X Y Data to File** для ввода имени и местоположения файла с расширением .prn. После ввода этих данных и нажатия кнопки **OK** файл будет сохранен на диске. Сохранив данные, можно либо выйти из системы, либо продолжить анализ данных.

Ввод данных из сохраненного файла осуществляется следующим образом. Вызывается пункт главного меню **File | Import**. При отсутствии каких-либо ранее введенных данных появится диалоговое окно для ввода заголовка графика и названия его осей X, Y . После нажатия кнопки **OK** появится рабочее окно со статистической информацией и графиком, построенным по введенным данным. Если до вызова пункта меню **File | Import** в системе уже имелись данные, то появится диалоговое окно **Reading New Data**, позволяющее добавить новые данные к уже имеющимся (кнопка **Yes**) или убрать старые данные перед загрузкой новых (кнопка **No**). Пункт меню **File | Import Digital Filter** позволяет ввести не все данные из файла, а только те, которые соответствуют условиям, определенным в специальном файле. Пункт меню **File | Import Clipboard** позволяет вставить данные из буфера обмена туда, куда они могут быть помещены, например, в систему Matlab (рис. 9.1).

Если нет необходимости сохранять данные, то в окне **TableCurve Data** надо нажать кнопку **No**, после чего появится новое окно **TableCurve XY**, в котором данные представлены на графике в виде точек. В окне также отображаются результаты вычислений следующих числовых характеристик X и Y :

- ☐ минимальное и максимальное значения (Min, Max);
- ☐ ранг (Range);

- ☐ среднее значение (Mean);
- ☐ медиана (Median);
- ☐ стандартное отклонение (Std);
- ☐ значение одной переменной при минимальном или максимальном значении другой переменной (atXmin, atXmax, atYmin, atYmax).

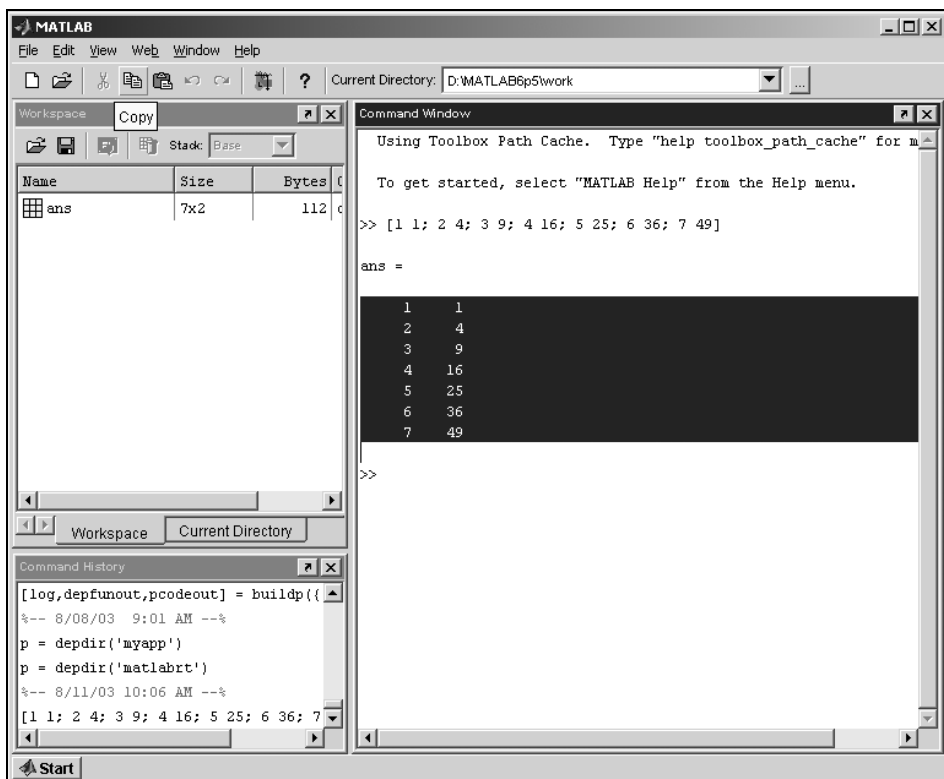


Рис. 9.1

Выбор формулы интерполяции

После ввода данных можно начать решение задачи. Для этого необходимо вызвать пункт **Process** главного меню. Появится окно команд выбора группы выражений. Команды имеют следующие назначения:

- ☐ **Curve-Fit All Equations** — использование всех групп выражений;
- ☐ остальные команды **Curve-Fit** до команды **Curve-Fit WaveForm Functions** предназначены для использования соответствующих групп формул;
- ☐ следующие три команды, расположенные за чертой, предназначены для работы с формулами, которые выбрал пользователь из имеющихся в программе;

- ☐ вслед за ними три команды предназначены для работы с формулами, которые создал сам пользователь;
- ☐ последний пункт меню позволяет задавать параметры вычислений.

Перед выбором команды следует ознакомиться со списком формул, имеющихся в программе TableCurve. Для этого необходимо обратиться через пункт **Help** главного меню к команде **Equation List**.

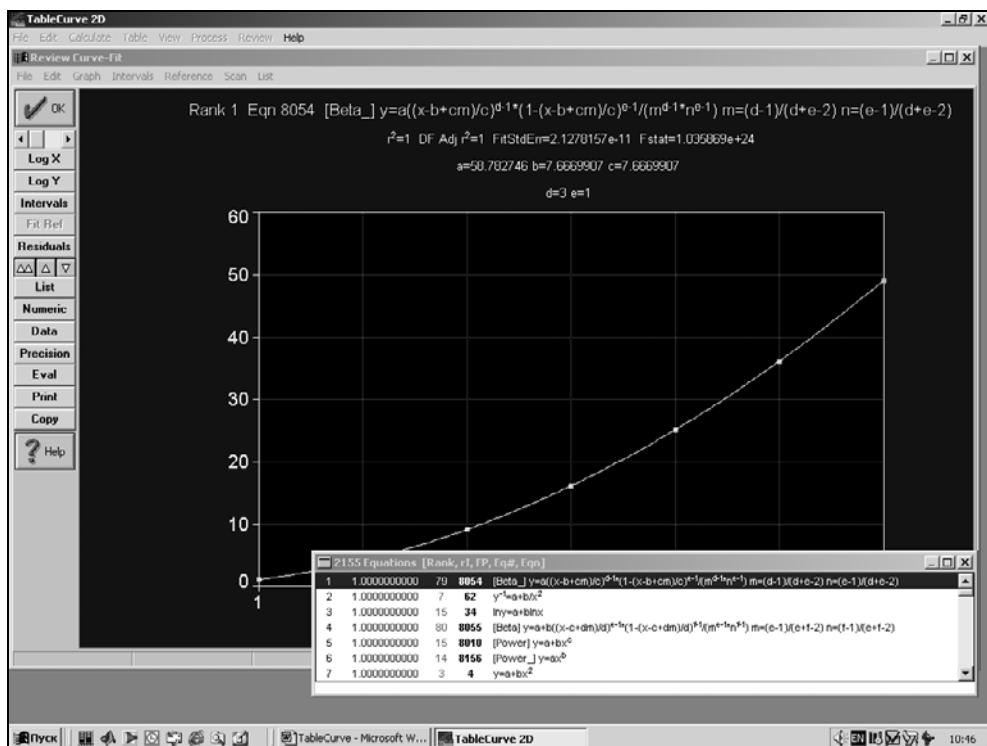


Рис. 9.2

После вызова пункта **Process** главного меню, а затем команды **Curve-Fit All Equations** появляется окно, отображающее ход процесса решения задачи интерполяции. После окончания этого процесса появляется графическое окно и окно с формулами. Графическое окно имеет собственное меню и рабочие кнопки анализа результатов вычислений (рис. 9.2). В графическом окне на черном фоне представляется график функции интерполяции, имеющей минимальную погрешность, а также точки координат исходных данных. По их взаимному расположению можно судить о точности функции интерполяции. В окне с формулами приведены все выражения, полученные программой в процессе решения задачи. Вверху графического окна приводится формула и значение всех ее коэффициентов. Окно с формулами

является интерактивным списком. Пользователь выбирает нужную формулу, щелкая по ней левой кнопкой мыши. При этом изменяется информация в графическом окне: появляется новая формула и соответствующий ей график. Повторяя этот процесс многократно, можно выбрать желаемую функцию аппроксимации.

Кнопки, расположенные слева от графического окна, имеют следующие значения:

- ☐ **OK** — закрывает графическое окно (повторно вызвать его можно с помощью пункта меню **Review | Graf Start**);
- ☐ **logX, LogY** — включают/выключают логарифмический масштаб по осям X, Y ;
- ☐ **Intervals** — показывает доверительные интервалы;
- ☐ **Residuals** — показывает график остатков для текущего выражения;
- ☐ **List** — убирает/активизирует окно с формулами;
- ☐ **Numeric** — показывает числовую информацию относительно текущего выражения;
- ☐ **Data** — показывает числовую информацию в расчетных точках;
- ☐ **Precision** — показывает информацию относительно точности вычислений;
- ☐ **Eval** — предоставляет дополнительные вычислительные процедуры;
- ☐ **Print** — распечатывает график;
- ☐ **Copy** — копирует график в буфер обмена;
- ☐ **Help** — предоставляет контекстную помощь.

Движок прокрутки, расположенный ниже кнопки **OK**, позволяет изменять масштабы по осям X, Y .

Кнопки со стрелками дают возможность перемещаться по списку формул.

Примеры решения задач интерполяции

Пример 9.1

Допустим, что надо найти математическую модель расширения Вселенной по опытным данным астронома Хаббла (см. табл. 8.1).

Технология решения задачи с помощью программы TableCurve состоит в выполнении следующих операций:

1. Ввод исходных данных.
2. Получение решения в виде графика и формул.
3. Выбор оптимального решения.

Из этого перечня видно, что в нем отсутствуют такие пункты, как выбор вида функции интерполяции и оценка адекватности модели. Эти пункты являются основными и наиболее трудными для исследователя при решении

задачи интерполяции с помощью компьютерных технологий, описанных в предыдущих главах книги. Будем решать задачу в указанной последовательности.

Ввод исходных данных

Вызываем пункт **Edit** главного меню, после появления вспомогательного окна активизируется команда **TableCurve Editor**. На экране появляется пустая таблица данных и мигающий курсор в первом ее квадрате. Поскольку данные X и Y равноценны и вес их одинаков, то необходимо активизировать кнопку **Weights** (щелчком мыши). Теперь программа будет писать единицу в каждую ячейку столбца **Weights**. Введем исходные данные опыта Хаббла.

Получение решения

По окончании ввода данных нажмем кнопку **OK**. Появится окно **TableCurve Data is Not Saved**. Программа спрашивает, будем ли мы сохранять данные. Нажмем кнопку **No** (Нет). Появится окно **TableCurve XY Table Status**, в котором программа сообщает статистические данные о введенных значениях расстояния до галактик R и скорости их удаления V . Кроме того, она представляет на плоскости X — Y функцию $R = f(V)$ в виде точек. Статистические данные здесь не нужны, график же весьма полезен — он позволяет обосновать математическую функцию интерполяции. По виду графика можно предположить, что функция $R = f(V)$ является линейной.

Для получения формул интерполяции вызовем пункт **Process** главного меню, а затем команду **Curve-Fit All Eguation** во вспомогательном меню.

Появится новое окно, в котором отобразится процесс решения задачи, а затем через короткое время откроется новое окно, где будет предложено перейти к просмотру результатов вычислений и выбору оптимального решения.

Выбор оптимального решения

Просмотр полученных функций интерполяции осуществляется путем нажатия одной из кнопок **Graph Start** или **List Start**. После нажатия кнопки **Graph Start** появится графическое окно на черном фоне и таблица функций. В окне отображается кривая, проходящая через точки таблицы, а вверху графика функция интерполяции и значения ее коэффициентов. В нашем случае это:

$Y = a + bx + cx^{2.5} + dx^3 + e / x^2$, где:

□ $a = 6.907$;

□ $b = 2.83$;

□ $c = -0.000176$;

□ $d = 9.34$;

□ $e = -345.15$.

С полученным решением согласиться нельзя. Оно противоречит графику функции (функция линейна) и корреляционному анализу (коэффициент корреляции близок к единице). Формула не является моделью расширения Вселенной. Физический закон не открыт. Нужно искать иную функцию, воспользовавшись кнопками прокрутки. Если окно мешает, то его можно убрать, щелкнув мышью по кнопке . Вновь восстановить окно с формулами можно с помощью кнопки **List** (слева от окна решений). Щелкая по формулам мышью, будем получать выбранную формулу и значение коэффициентов сверху графического окна.

Процесс выбора подходящей формулы можно существенно упростить, если обратиться к соответствующей группе выражений. В нашем случае функция интерполяции на основании корреляционного анализа и графика должна быть близка к линейной. Чтобы выбрать такую функцию нужно обратиться к группе линейных выражений. Для этого необходимо вызвать пункт **Process** главного меню и с помощью мыши команду **Curve-Fit Robust Straight Line**. После появления окна функций интерполяции через некоторое время активизируется окно линейных функций, в котором будет представлено четыре варианта линейной функции $y = a + bx$. Щелкая мышью поочередно по всем вариантам, получим следующие значения коэффициентов:

- ☐ $a = -1.0533, b = 2.823;$
- ☐ $a = 0.9826, b = 2.806;$
- ☐ $a = 0.4517, b = 2.894;$
- ☐ $a = 0.5130, b = 2.894.$

Вычислим погрешности линейных функций интерполяции. Для этого протабулируем эти функции для всех значений V таблицы Хаббла. Результаты табулирования приведены в табл. 9.1.

Таблица 9.1

	Результаты табулирования линейной функции										
<i>V</i>	7.5	24	32	47	93	120	134	144	245	260	380
<i>R</i>	22	68	108	137	255	315	390	405	685	700	1100
<i>y</i> ₁	20.12	66.70	89.28	131.63	261.49	337.71	377.23	405.46	690.58	732.93	1071.68
<i>y</i> ₂	22.03	68.33	90.77	132.86	261.94	337.70	376.99	405.05	688.45	730.54	1067.26
<i>y</i> ₃	22.16	69.91	93.06	136.47	269.59	347.73	388.25	417.19	709.48	752.89	1100.17
<i>y</i> ₄	22.22	69.97	93.12	136.53	269.65	347.79	388.31	417.25	709.54	752.95	1100.23
<i>y</i> ₅	21.97	73.86	96.42	138.07	262.89	334.86	372.	398.51	672.25	715.02	1099.39

В таблице:

- y_i — значения функций интерполяции, полученных программой TableCurve ($i = 1, 2, 3, 4, \dots$).
- y_5 — функция, полученная программой как наилучшая, имеющая наименьшую погрешность.

Заметим, что функция $y_1 = -1.05 + 2.82x$, выданная данной программой совпадает с той, что получена путем аппроксимации с помощью функции FIT системы Derive 5 (см. главу 8).

В табл. 9.2 приведены абсолютные среднеквадратические ξ и максимальные и минимальные относительные $\delta_{\max}, \delta_{\min}$ погрешности формул. Вычисления получены с помощью системы Maple 6.

Таблица 9.2

	y_1	y_2	y_3	y_4	y_5
ξ	16.58	16.87	21.64	21.67	11.2
$\delta_{\max}, \%$	75.36	76.66	98.38	98.48	50.93
$\delta_{\min}, \%$	1.5	1.53	1.97	1.97	1.02

По данным табл. 9.1 и 9.2 неопытный исследователь может сделать ложный вывод, утверждая, что моделью расширения Вселенной является формула $y = a + bx + cx^{2.5} + dx^3 + e / x^2$, первоначально полученная программой и обеспечивающая минимальную погрешность. Корреляционный анализ, визуализация функций и сравнительный анализ линейных моделей позволяют утверждать, что закон расширения Вселенной является линейной функцией $R = a + bV$. Из выполненных расчетов видно, что программа TableCurve в действительности не освобождает нас от выбора функции интерполяции и доказательства адекватности модели. Она подсказывает исследователю множество решений, из которых он должен выбрать то, которое может быть математической моделью изучаемого объекта.

Пример 9.2

Пусть функция $y = f(x)$ задана таблицей (табл. 9.3).

Таблица 9.3

	Значения функции $y = f(x)$										
x	0	5	10	15	20	25	30	35	40	45	50
y	4	6.9	10.15	15.45	24.17	38.55	62.26	101.35	165.8	272	447.03
$y(x)$	4.39	6.57	9.99	15.46	24.29	38.67	62.28	101.24	165.065	272.04	447.03

Необходимо найти математическую модель исследуемого объекта. Технология решения задачи подробно описана в примере 9.1, поэтому приведем лишь конечные результаты. Функция интерполяции, полученная с помощью программы TableCurve, имеет вид:

$$\ln(y) = (a + cx) / (1 + bx + dx^2) \text{ или:}$$

$$y = e^{((a + cx) / (1 + bx + dx^2))}.$$

Коэффициенты функции имеют следующие значения:

$$\square a = 1.48;$$

$$\square b = -0.00544;$$

$$\square c = 0.0704;$$

$$\square d = 3.669 \cdot 10^{-5}.$$

Результаты табулирования этой функции приведены в табл. 9.3 ($y(x)$). Сравнивая значения функций y и $y(x)$ видим, что они практически одинаковы и незначительно отличаются в начале таблицы. Абсолютная среднеквадратическая погрешность $\xi = 0.187$, максимальная относительная погрешность $\delta_{\max} = 4.67 \%$, $\delta_{\min} = 0.04 \%$.

А теперь проанализируем результаты решения. Табл. 9.3 получена в результате табулирования функции $y = a + b t^{(cx)} + d t^{(fx)}$ при следующих значениях коэффициентов:

$$\square a = 2;$$

$$\square b = 3;$$

$$\square c = 0.1;$$

$$\square d = -1.5;$$

$$\square f = -1.8.$$

Функция, полученная программой TableCurve, совсем не похожа на эту функцию, несмотря на то, что ее погрешность мала. Почему же программа не нашла эту функцию и можно ли ее решение принять за математическую модель изучаемого объекта? Программа не нашла и не могла найти решение потому, что в ней воспроизводятся только функции линейные относительно неизвестных коэффициентов, в то время как наша функция нелинейна. Здесь решение можно получить методами интерполяции точными в узлах, которые не реализованы в программе.

Полученное программой решение нельзя принять за математическую модель изучаемого объекта. Мы не уверены в том, что решение будет верным при значениях x , отсутствующих в таблице. Тем более опасно производить какие-либо преобразования этого выражения (дифференцирование, интегрирование и др.).

Автоматизация решения задач интерполяции безусловно нужна. Программа TableCurve весьма полезна при решении практических задач. Она помогает исследователю выбрать вид функции интерполяции, а в ряде случаев даже найти математическую модель. Однако нужно быть весьма осторожным при выборе типа модели. Может оказаться, что среди множества функций математической моделью является не та, которая имеет минимальную погрешность. Более того, среди множества функций искомой модели может и не быть вовсе.

9.2. Программа CurveExpert 1.3

Программа CurveExpert запускается с помощью файла Cvxpt.exe. После запуска появляется окно **CurveExpert 1.3** (рис. 9.3), состоящее из пунктов главного меню, панели инструментов, таблицы **X, Y** для ввода исходных данных, графического окна, отображающего исходные данные, и области моделей аппроксимации.

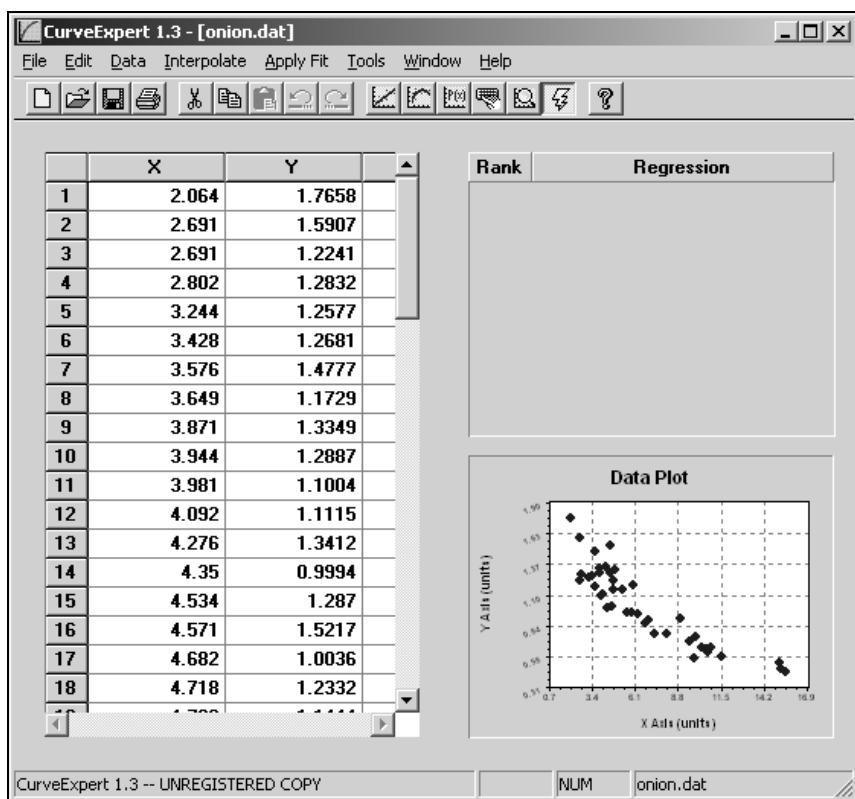


Рис.9.3

При решении задач интерполяции обычно пользуются кнопками панели инструментов, имеющих следующие назначения:

- ☐  — открыть новое окно **CurveExpert** с удалением прежних данных;
- ☐  — открыть сохраненный файл;
- ☐  — сохранить данные;
- ☐  — распечатать результаты;
- ☐  — вырезать выделенный объект и поместить в буфер обмена;
- ☐  — поместить копию выделенного объекта в буфер обмена;
- ☐  — вклеить содержимое буфера обмена;
- ☐  — отменить предыдущее действие;
- ☐  — вернуть отмену;
- ☐  — выполнить линейную аппроксимацию;
- ☐  — выполнить квадратичную аппроксимацию;
- ☐  — выбрать степень полинома функции интерполяции;
- ☐  — выбрать тип математической модели;
- ☐  — построить график функции и активизировать окно **CurveExpert**, в котором отображаются функции интерполяции различного вида;
- ☐  — выполнить начальный поиск модели;
- ☐  — открыть файл помощи.

Ввод данных

Числа X , Y вводятся в таблицу данных последовательно. При этом после набора каждого числа нажимается клавиша <Enter>. Активизировать очередную ячейку также возможно клавишей <Tab>. Перемещение по таблице данных осуществляется клавишами со стрелками: <↑> (вверх), <↓> (вниз), <→> (вправо), <←> (влево). Введенные числа отображаются на графике. Редактирование данных осуществляется стандартными методами. Исходные данные можно сохранить. Для этого следует щелкнуть мышью по кнопке сохранения данных  (Save data file) на панели инструментов, после чего

появится окно **Save As** (рис. 9.4) В поле **File name** надо ввести имя файла с расширением **.dat**, затем нажать кнопку **Save** (Сохранить).

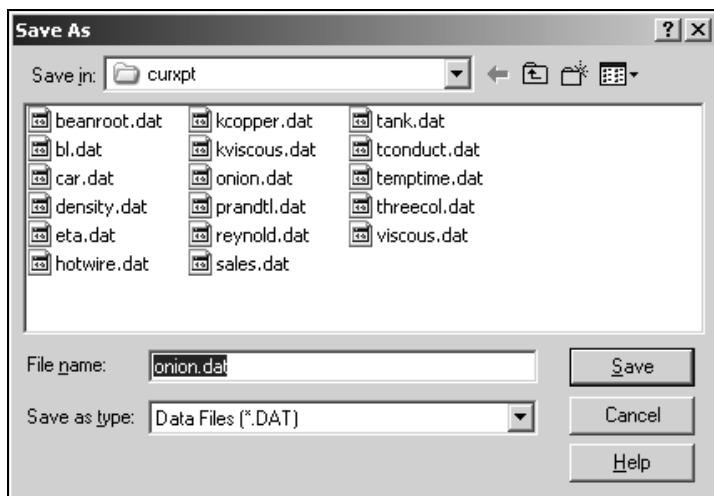


Рис. 9.4

Для открытия файла необходимо щелкнуть по кнопке  (Open a data file) на панели инструментов, после чего появится окно **Open File** (рис. 9.5), где надо выбрать нужный файл при помощи мыши, после чего нажать кнопку **Open** (Открыть). Появится новое окно **File Import Options**. Нажатие кнопки **ОК** приведет к открытию нового файла с исходными данными.

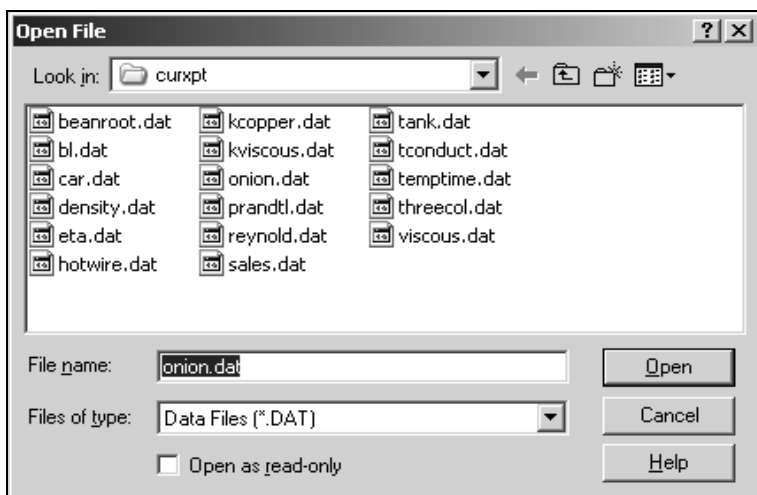


Рис.9.5

Построение графика функции

Построение графика осуществляется нажатием любой из кнопок выбора вида функции интерполяции:



— линейная: $y = a + bx$;



— квадратичная: $y = a + bx + cx^2$;



— полиномиальная: $y = a + bx + cx^2 + dx^3 + \dots$;



— интерполяция нелинейными функциями;



— множественная интерполяция.

В случае линейной или квадратичной интерполяции график функции появляется сразу. При полиномиальной интерполяции появляется окно **Polynomial** и в поле **Please enter the degree of the regression polynomial** устанавливается необходимая степень полинома (рис. 9.6).

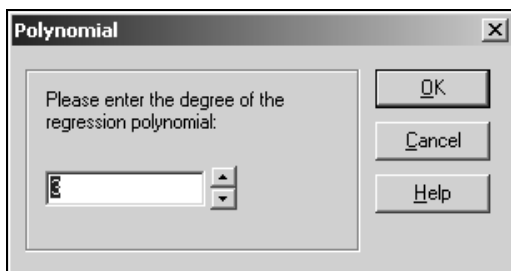


Рис.9.6

После нажатия кнопки **ОК** появляется график функции аппроксимации.

При интерполяции многими функциями надо нажать кнопку  (Run CurveFinder), после чего появится окно **CurveFinder**. В этом окне приводятся имена математических моделей (9 кнопок, отмеченных знаком ✓). Если теперь нажать кнопку **ОК**, то в диалоговом окне **CurveExpert 1.3** появится список функций и одновременно графическое окно, которое закроет окно **CurveExpert 1.3**. В графическом окне увидим график функции, значения среднеквадратического отклонения и коэффициента корреляции. В окне **Curve Finder 1.3** можно снять ненужные имена моделей, чтобы они не появлялись в виде графиков. Закрыв графическое окно, в окне исходных данных (справа от таблицы) увидим имена функций интерполяции, например:

1. MMF Model: $y = (ab + cx^d) / (b + x^d)$
2. Polynomial Fit: $a + bx + cx^2 + dx^3 + \dots$

3. Exp. Association (3): $y = a(b - e^{cx})$
4. Exponential Association: $y = a(1 - e^{-bx})$
5. Quadratic Fit: $y = a + bx + cx^2$
6. Sinusoidal Fit: $y = a + b\cos(cx + d)$
7. Harris Model: $y = 1 / (a + bx^c)$
8. Linear Fit: $y = a + bx$
9. User-Defined Model (появляется окно **Define User Model**, позволяющее дополнительно выбрать большое число типов функций).

Выбор одной из перечисленных функций осуществляется щелчком мыши по соответствующей строке. Один щелчок изменяет график, расположенный ниже перечня имен функций, что позволяет визуально судить о точности интерполяции. Два щелчка приводят к появлению окна **Logarithm Fit** (рис. 9.7), в котором отображается красным цветом график функции и точки исходных данных. В верхнем правом углу находятся значения абсолютной среднеквадратической погрешности S и коэффициента корреляции r .

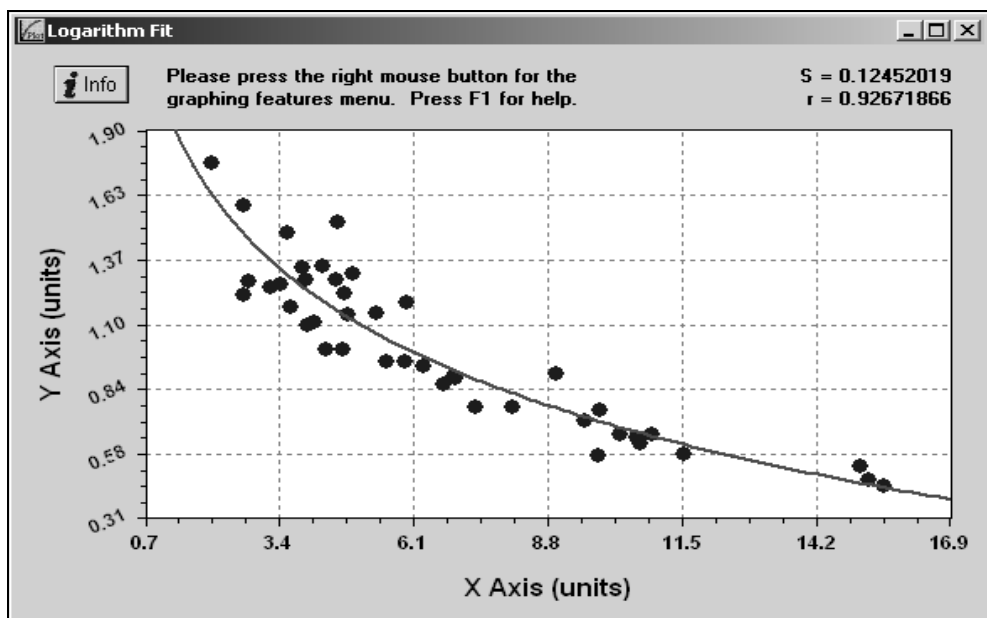


Рис.9.7

Определение функции интерполяции

Для определения функции интерполяции следует нажать кнопку **Info** в графическом окне. Появится новое окно **Model Information** (рис.9.8). В этом

окне приводится формула интерполяции, а на вкладке **Coefficients** значения ее коэффициентов.

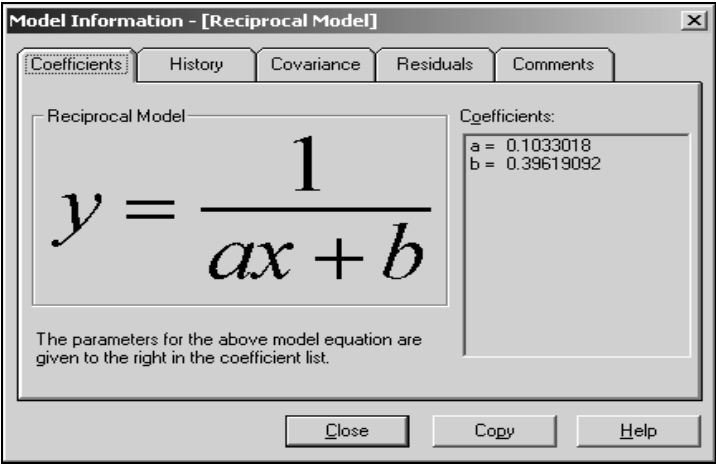


Рис.9.8

Выбор вида функции интерполяции

Из множества функций, предлагаемых программой, исследователь должен выбрать ту, которая является математической моделью исследуемого объекта. Программа CurveExpert ответа на этот вопрос не дает. Она выдает лишь большое число функций и позволяет ранжировать их по величине абсолютной среднеквадратической погрешности. Однако этот критерий не является критерием выбора функции интерполяции. Более того, среди множества функций бывает нельзя найти ни одной, отвечающей требованиям математической модели изучаемого объекта. Эту проблему должен решать исследователь.

Приведем примеры решения задач интерполяции с помощью программы CurveExpert, на которых покажем компьютерные технологии, достоинства и недостатки программы.

Пример 9.3

Температура кипения воды $t_{\text{кип}}$ при различных давлениях (ниже нормального атмосферного) приведена в табл. 9.4 [24].

Таблица 9.4

	Зависимость температуры кипения воды от давления										
P , мм.рт.ст.	4,6	9,2	17,5	31,8	55,3	92,5	233,7	289	403	526	634
$t_{\text{кип}}$, °С	0	10	20	30	40	50	70	75	83	90	95

Таблица 9.4 (продолжение)

	Зависимость температуры кипения воды от давления							
P , мм.рт.ст.	680	700	710	720	730	740	750	760
$t_{\text{кип}}$, °C	96,9	97,7	98,1	98,5	98,9	99,3	99,6	100

Необходимо найти математическую модель $t = f(P)$.

Воспользуемся программой CurveExpert и описанной выше методикой.

Ввод данных

После запуска программы с помощью файла Cvxpt.exe введем данные табл. 9.4. При этом аргументом X будет давление P , а функцией Y — температура t . При вводе данных полезно следить за появляющимися точками на графике. Это дает возможность определить ошибки ввода (значительное отклонение ошибочной точки от остальных). Так как наша задача одновариантная, а исходные данные не могут быть изменены, то не будем их сохранять.

Построение графика

Нажмем на панели инструментов кнопку построения графика функции . Нам предлагается девять видов функции интерполяции. Согласимся с программой и щелкнем по кнопке **ОК**. Появится окно исходных данных и имена девяти функций аппроксимации. Выберем подходящие функции, для чего, нажимая поочередно строку имени функции, наблюдаем за взаимным расположением точек исходных данных и графика функции. По их взаимному расположению визуально подбираем подходящие функции. Щелкая дважды левой кнопкой мыши по строкам с именами выбранных функций, получаем соответствующие графики, а также значения их погрешностей и коэффициенты корреляции.

Получение математической модели

Для получения функции аппроксимации необходимо в графическом окне нажать кнопку **Info**. Появится окно **Model Information**, в котором приводится искомая формула и значение ее коэффициентов. В результате этих процедур нами будут получены следующие наилучшие формулы аппроксимации:

$$\square y_1 = (ab + cx^d) / (b + x^d),$$

$$a = -3.275, b = 40.70, c = 141.77, d = 0.69, S = 2.25, r = 0.998;$$

$$\square y_2 = a(b - e^{-cx}),$$

$$a = 91.69, b = 1.07, c = 0.0058, S = 5.16, r = 0.991;$$

$$\square y_3 = a + bx + cx^2 + dx^3,$$

$$a = 7.98, b = 0.46, c = -0.000917, d = 6.2 * 10^{(-0.007)}, S = 5.46, r = 0.991.$$

В табл.9.5 приведены значения абсолютных ξ среднеквадратических и относительных $\delta_{\max}$, $\delta_{\min}$ погрешностей полученных моделей, вычисленные нами по формулам 2.1, 2.2 (см. главу 2).

Таблица 9.5

	y_1	y_2	y_3
ξ	1.93	4.66	5.08
$\delta_{\max}$	19.3	46.56	50.8
$\delta_{\min}$	1.93	4.66	5.08

Незначительные отличия в погрешностях, вычисленных нами и выданных программой, объясняются тем, что при вычислениях абсолютных погрешностей S мы округляли числа до двух значащих цифр после запятой.

Анализ результатов показывает, что математическая модель кипения воды при пониженных давлениях найдена. Наиболее подходящей является первая формула. Однако физический закон этого явления нами не открыт. Формула $y = (ab + cx^d) / (b + x^d)$ является лишь математическим выражением, не имеющим физического смысла. Программа не смогла найти нужную функцию аппроксимации.

Пример 9.4

Пусть данные эксперимента представлены в табл. 9.6.

Таблица 9.6

	Значения переменных								
x	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8
y	3.725	3.304	2.93	2.599	2.305	2.044	1.813	1.608	1.426

Надо найти математическую модель объекта исследования. Задача считается решенной, если максимальная относительная погрешность не превышает 5 %.

Решение задачи по описанной выше технологии привело нас к следующему единственному результату:

$$y = (a + bx) / (1 + cx + dx^2), \text{ где } a = 0.000269, b = 87.445, c = 13.75, d = 24.17.$$

При этом среднеквадратическая погрешность $S = 0.06$, а коэффициент корреляции $r = 0.999$.

Расчет погрешностей по формулам 2.1, 2.2 дал следующие результаты:

- ☐ абсолютная среднеквадратическая погрешность $\xi = 0.049$;
- ☐ максимальная относительная $\delta_{\max} = 3.46 \%$;
- ☐ минимальная относительная $\delta_{\min} = 1.32 \%$.

Адекватность модели по критерию погрешности доказана. И всё же полученная функция интерполяции не является физической моделью объекта. Табл. 9.6 — это результат табулирования функции $y = 4.2 e^{0.6x}$, которая не сходственна с полученной $y = (a + bx) / (1 + cx + dx^2)$.

Последняя функция "работает" в узком диапазоне аргументов $x = 0.2\text{—}1.8$. Вне этого диапазона могут возникать недопустимо большие ошибки. Например, при $x = 0$ функция имеет значение $y = 4.2$, а полученная модель дает значение $y = a = 0.000269$.

Приведенный пример ещё раз убеждает нас в том, что программы автоматизации решения задач интерполяции не освобождают исследователя от выбора функции интерполяции и проверки адекватности модели.

9.3. Программа SIMPLE FORMULA.

После запуска этой программы появляется окно, содержащее ее имя **SIMPLE FORMULA V 1.5**, объяснения задачи и методов ее решения, а также окно ввода типа меню. Автор называет свою программу программой вычисления коэффициентов для каждой из 1200 формул методом наименьших квадратов. Термин интерполяция или аппроксимация в программе не используется.

Окно ввода данных состоит из следующих трех строк:

- ☐ считывание файла данных с диска;
- ☐ ввод координат точек с клавиатуры;
- ☐ выход из программы.

Содержание этих пунктов очевидно из названий. Рассмотрим технологию их реализации пользователем. Активизируем с помощью клавиш со стрелками или мыши строку **Ввод координат точек с клавиатуры**, нажмем клавишу <Enter>. Появится окно ввода для координаты X . Набираем на клавиатуре соответствующее число, затем нажимаем клавишу <Enter>. На экране появится окно ввода для координаты Y . Вводим необходимое число и нажимаем клавишу <Enter>. Вновь появится окно ввода для координаты X и т. д. Редактирование данных осуществляется способами, общепринятыми в компьютерных редакторах. После набора координат трех точек появляется графическое окно, в котором находится график функции, проходящий через

три точки, и рекомендуемые формулы интерполяции со значениями абсолютных среднеквадратических погрешностей и максимальных отклонений. На экране появляется также окно **Simple Formula** со следующими пунктами меню:

- ☐ **Ввод точек.**
- ☐ **Ввод / вывод файлов.**
- ☐ **Структура формулы.**
- ☐ **Тестирование формул.**
- ☐ **База свойств веществ.**
- ☐ **Выход из программы.**

Программа предлагает или продолжать ввод данных в текстовом режиме, или провести анализ формул, или выйти из программы.

Для продолжения ввода надо выполнить пункт меню **Ввод точек**. Появится новое окно (назовем его окном условий продолжения ввода) со следующими пунктами меню:

- ☐ **Ввод в текстовом режиме.**
- ☐ **Использовать большую картинку.**
- ☐ **Выключить расчет при вводе T.**
- ☐ **Рассчитать формулу.**
- ☐ **Улучшить формулу.**
- ☐ **Упростить формулу.**

Программа предлагает либо продолжить ввод данных в прежнем текстовом режиме, либо выключить расчет при вводе очередной точки, чтобы не показывать графическое окно после набора каждой очередной координаты, либо закончить ввод и преобразовать (если в этом есть необходимость) выданные программой формулы (улучшить их или упростить). При выполнении пункта меню **Ввод в текстовом режиме** появляется окно ввода X , Y и ввод продолжается в прежнем режиме. При выполнении пункта меню **Выключить расчет при вводе T** появится **Окно для объявлений**, в котором программа предупреждает пользователя, что расчет при вводе точек отключен и ввод данных продолжается обычным способом без появления графического окна. По окончании ввода (или в процессе ввода) в окне **Ввод точек** активизируется пункт меню **Рассчитать формулу**. После нажатия клавиши <Enter> появляется графическое окно с графиком и набором наилучших формул аппроксимации. Решение задачи закончено.

Теперь можно представить график на полном экране, выполнив пункт **Использовать большую картинку**, или **Преобразовать формулы** (улучшить, упростить). При выполнении пункта **Улучшить формулу** программа обычно предлагает формулы более сложные, но с меньшими погрешностями. При

выполнении пункта **Упростить формулу** с экрана исчезают сложные выражения и остаются наиболее простые.

В процессе решения задачи часто возникает необходимость возврата к предыдущему окну. Это осуществляется нажатием клавиши <Esc>.

При выходе из программы появляется **Окно для объявлений**. Пользователю предлагается сохранить данные (Y / N). При нажатии клавиши <N> данные не сохраняются. При нажатии клавиши <Y> появляется окно сохранения данных, в котором надо указать имя файла и место его сохранения. После нажатия клавиши <Enter> данные будут сохранены на диске.

Программа SIMPLE FORMULA существенно облегчает процесс выбора функции интерполяции и доказательство адекватности модели, что является наиболее трудным этапом компьютерной технологии интерполяции. Этот этап осуществляется путем анализа полученных решений с применением возможности операций над полученными программой функциями. Программа позволяет:

- ☐ устанавливать число слагаемых и базис функций интерполяции;
- ☐ осуществлять поиск желаемой функции из их множества или из списка наилучших формул;
- ☐ автоматизировать вычисление значений множества функций (тестирование формул);
- ☐ изменять переменные и редактировать коэффициенты полученных формул;
- ☐ выбирать желаемую функцию из всего множества;
- ☐ улучшать, упрощать, сохранять желаемые формулы.

Программа имеет в своем составе базу свойств веществ, таких как теплопроводность, электрическое сопротивление, теплоемкость, вязкость, плотность. База свойств состоит из набора математических моделей в виде формул и графиков.

Нет необходимости детально описывать компьютерную технологию реализации перечисленных выше функций. Диалог, реализованный на русском языке, предельно понятен пользователю.

Рассмотрим примеры решения задач интерполяции с помощью программы SIMPLE FORMULA.

Пример 9.5

Пусть данные эксперимента приведены в таблице (см. табл. 9.6). Необходимо найти математическую модель объекта исследования. В примере 9.4 найденная математическая модель удовлетворяла условиям адекватности по критерию среднеквадратической погрешности. Однако она не являлась физическим законом. Решим теперь эту задачу с помощью программы SIMPLE FORMULA.

Ввод исходных данных

Данные будем вводить с выключенным расчетом, т. к. в этой задаче нет необходимости оценивать влияние отдельных координат на конечный результат.

Получение решения

После окончания ввода данных в окне **Ввод точек** выполним пункт меню **Рассчитать формулу**. Появится графическое окно. Программа решила задачу аппроксимации методом наименьших квадратов. На экран выданы следующие наилучшие формулы:

$$\square y_1 = \sqrt[3]{15.121} e^{-x},$$

$$\varepsilon = 0.039739, \quad L_{\max} = 0.206;$$

$$\square y_2 = e^{(-0.6001 + 1.435)},$$

$$\varepsilon = 0.00004868, \quad L_{\max} = 0.00022;$$

$$\square y_3 = e^{(-0.000026356 / x - 0.6x + 1.435)},$$

$$\varepsilon = 0.000044, \quad L_{\max} = 0.00022;$$

$$\square y_4 = e^{(-0.000095128x - 0.59993x + 1.435)},$$

$$\varepsilon = 0.000045, \quad L_{\max} = 0.00022;$$

$$\square y_5 = e^{(-0.000079865 \ln(x) - 0.00004568 / x - 0.6x + 1.435)}$$

$$\varepsilon = 0.000044, \quad L_{\max} = 0.00021.$$

Выполнив пункт меню **Улучшить формулу**, новых существенных результатов не получим, а после выполнения пункта меню **Упростить формулу** программа рекомендует формулы y_3 и y_4 .

Выбор наилучшего решения

Из пяти, предложенных программой, решений наилучшими по критерию среднеквадратической погрешности являются формулы y_3 и y_5 .

Абсолютная среднеквадратическая погрешность этих формул по сравнению с другими минимальна, хотя и отличается от них незначительно. Однако именно эти формулы мы должны отвергнуть, т. к. при $x = 0$ $y_3 = y_5 = 0$, что противоречит остальным формулам и не соответствует ответу, который мы в данном случае знаем. Первая формула по точности уступает остальным, поэтому нам остается выбрать решение из вариантов 2 или 4. В формуле 4 коэффициент при x^2 мал и его можно без потери точности отбросить. Тогда наш выбор падает на формулу 2, т. к. практически $y_2 = y_4$. Формула 2 дает точное решение. После следующих очевидных преобразований, получим:

$$y_2 = e^{(-0.6001x + 1.435)} = e^{1.435} e^{(-0.6001x)} = 4.2e^{(-0.6)}, \text{ что совпадает с ответом.}$$

Программа SIMPLE FORMULA позволила найти правильное решение. Однако следует иметь в виду, что ответ нам был известен заранее, поэтому мы

сумели из множества решений выбрать желаемое. Следует также иметь в виду, что мы не анализировали другие решения, которых в программе 1200.

Пример 9.6

Пусть функция $y(x)$ представлена в виде табл. 9.7.

Таблица 9.7

	Значения функции $y = f(x)$										
x	0	0.1	0.3	0.5	0.7	0.9	1.1	1.3	1.5	1.7	1.9
y	8.9	7.263	5	3.62	2.786	2.28	1.793	1.787	1.674	1.606	1.564
y₄	8.95	7.205	4.95	3.64	2.83	2.32	1.996	1.773	1.645	1.585	1.589

Необходимо найти математическую модель объекта исследования.

Компьютерная технология решения этой задачи особенностей не имеет. Поэтому приведем лишь конечные результаты. Программа выдала следующие решения:

$$\square y_1 = e^{(2.1817 / e^x)},$$

$$\varepsilon = 0.033, L = 0.178;$$

$$\square y_2 = e^{(2.205 / e^x - 0.021)},$$

$$\varepsilon = 0.0317, L = 0.201;$$

$$\square y_3 = e^{(2.1786 / e^x + 0.00231e^x)},$$

$$\varepsilon = 0.033, L = -0.1779;$$

$$\square y_4 = e^{(2.0825 / e^x + 0.1094e^x - 0.305x)},$$

$$\varepsilon = 0.011, L = 0.0586.$$

Выбор наилучшего решения

По критерию среднеквадратической ошибки наилучшим решением является формула 4. Результаты табулирования этой формулы приведены в табл. 9.7. Из таблицы видно, что данные модели хорошо согласуются с табличными. Максимальная относительная погрешность $\delta = 0,011 / 1,564 * 100 = 0.64 \%$. Математическая модель найдена. Однако полученная формула не соответствует исходной, по которой рассчитана табл. 9.7. Таблица является результатом табулирования функции:

$$y = 1.5 + 7.4e^{(-2.5x)}.$$

Полученная с помощью программы, формула хорошо "работает" лишь в диапазоне значений $0 \leq x \leq 1.9$. Математические операции над ней производить опасно. Вот простой пример.

Производные функций $y(x)$ и $y_4(x)$ имеют вид:

$$y'(x) = -18.5e^{(-2.5x)},$$

$$y_4'(x) = 0.11e^{0.7x} - 0.31e^{(-0.31x)} - 2.08e^{(-1.31x)} * e^{(0.11e^x + 2.08e^{(-x)})}.$$

В табл. 9.8 приведены значения этих функций в диапазоне значений аргументов, соответствующих исходным данным.

Таблица 9.8

	Значения функций интерполяции										
x	0	0.1	0.3	0.5	0.7	0.9	1.1	1.3	1.5	1.7	1.9
y'(x)	-20.4	-14.4	-8.74	-5.3	-3.21	-1.95	-1.18	-0.72	-0.74	-0.26	-0.16
y₄'(x)	-18.5	-14.9	-8.41	-5.05	-3.17	-2.05	-1.33	-0.85	-0.46	-0.14	0.18

Из таблицы видно, что производные этих функций существенно отличаются, более того — при $x \geq 1.8$ у них разные знаки. Почему же программа SIMPLE FORMULA в примере 9.6 нашла точное решение, а в примере 9.7 нет? Все объясняется тем, что в примере 9.6 функция интерполяции $y = a * e^{cx}$ может быть сведена к линейной путем логарифмирования: $\ln(y) = \ln(a) + cx$.

Обозначая: $Y = \ln(y)$, $A = \ln(a)$, получим линейную функцию $Y = A + cx$.

Эти процедуры выполняет программа и находит методом наименьших квадратов значения коэффициентов a и c . Функция $y = a + b * e^{cx}$ таким или другим подобным способом не может быть преобразована в линейную относительно неизвестных коэффициентов a , b , c . Поэтому метод наименьших квадратов здесь не приемлем. В этом случае задача может быть решена методами интерполяции точными в узлах, которые в программе SIMPLE FORMULA, как и в других рассмотренных нами программах, не реализованы.



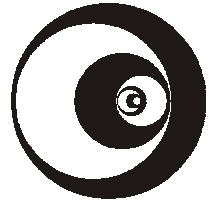
ПРИЛОЖЕНИЯ

Приложения

Наилучшим способом познания метода является доведение решения задачи до числа. В приложениях приведены задачи двух видов: абстрактные математические и конкретные из различных областей знаний.

Первые полезны для решения задач с целью глубокого усвоения теории, вторые нужны специалистам, занимающимся прикладными вопросами, требующими решения задач интерполяции. Это, прежде всего, задачи моделирования, планирования и статистической обработки эксперимента, анализа данных, представленных в табличной форме. Их решение необходимо для студентов, изучающих такие предметы, как компьютерные технологии в науке и образовании, информатика, моделирование, прикладная математика и многие другие специальные дисциплины.

Решение предлагаемых задач в большинстве случаев возможно с помощью любой из математических систем, описанных в *главах 2—6*. Однако в ряде случаев решение наиболее целесообразно проводить с помощью конкретного универсального программного математического средства. В таких случаях даются рекомендации по выбору метода интерполяции и математической системы его реализации.



Приложение 1

Абстрактные математические задачи

Решение задач методами точными в узлах интерполяции

Пусть исходные данные восьми задач представлены в виде табл. П1.1. Необходимо найти для каждой из задач функцию интерполяции. В качестве интерполяционной следует выбрать одну из следующих нелинейных функций:

$$y = a + b \ln(x), \quad y = a + \frac{b}{x}, \quad y = \frac{1}{a + bx}, \quad y = \frac{x}{a + bx}, \quad y = ax^b, \quad y = ax^b + c$$
$$y = ab^x, \quad y = ab^x + c$$

Таблица П1.1

Задача 1		Задача 2		Задача 3		Задача 4	
x	y	x	y	x	y	x	y
1	4.9	1	3.4	1	0.7	1	6
1.4	3.8	2	7	2.1	2.3	2	4.9
2.1	3	2.7	9.5	4.7	8.8	3	4
4.7	2	3.5	13.5	8	21	4	3.4
6.3	1.7	5.2	21	17.5	75	5	3
8.5	1.6	7	31	25.6	145	6	2.5
11	1.54	9.2	44	73	805	7	2.15

Таблица П1.1 (продолжение)

Задача 5		Задача 6		Задача 7		Задача 8	
<i>X</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>
0.1	4.4	0	2.6	0.1	0.31	1	0.2
0.9	19.4	0.2	3.1	0.3	0.28	2.5	0.34
1.7	85	0.45	4	0.5	0.26	3.7	0.4
2.5	368	0.67	5.2	0.7	0.24	5	0.44
3.3	1600	0.8	6.1	0.9	0.22	7.3	0.47
4.1	7000	1	8	1.1	0.2	9.8	0.5
4.9	30500	2.5	62	1.3	0.19	12.3	0.51

Решение задач методом аппроксимации многими функциями

Пусть исходные данные шести задач представлены в виде табл. П1.2.

Таблица П1.2

Задача 1		Задача 2		Задача 3		Задача 4		Задача 5		Задача 6	
<i>X</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>
1	2.3	2.3	15	50	130	5	81	0	6.1	1	0.9
2	2	3.2	23	100	136	10	44	3	1.7	2	1.7
3	1.8	4.7	36	150	138	15	32	6	1	3	2.3
4	1.67	5.4	44	200	140	20	25	9	0.7	4	2.8
5	1.5	7.1	61	250	142	25	22	12	0.6	5	3.2
6	1.4	9.3	86	300	143	30	20	15	0.45	6	3.7
7	1.25	11.5	112	350	144	35	18	18	0.4	7	4
8	1.1	15.2	160	400	145	40	16,5	21	0.33	8	4.4

Необходимо найти функции интерполяции для каждой из задач.

Примечание

Универсальные программные средства символьной математики позволяют решать задачу интерполяции одновременно следующими функциями: $a + bx$, ae^{bx} , ax^b , $a + b\ln(x)$, $a + \frac{b}{x}$, $\frac{a}{b + x}$, $\frac{ax}{b + cx}$. Полученные в результате решения функции табулируются. По результатам табулирования выбирается функция интерполяции, наиболее точно отображающая исходные данные, вычисляется абсолютная и максимальная относительная погрешности. Если погрешности превосходят допустимые, то результаты аппроксимации многими функциями отвергаются и ведется поиск функции интерполяции иными способами, описанными в главе 1.

Решение задач методами полиномиальной аппроксимации

Пусть исходные данные трех задач представлены в табл. П1.3.

Таблица П1.3

		Значения переменных								
Задача 1	x	1	1.5	2	2.5	3	3.5	4	4.5	5
	y	4	5	5.4	5.5	5	4.3	3	1.7	−0.3
Задача 2	x	1	2	3	4	5	6	7	8	9
	y	0.25	0	6.4	21	47	86	140	210	300
Задача 3	x	1.2	2.7	4.1	5.8	7.1	9.6	11.5	13.8	15
	y	31	15	−10	−51	−89	−170	−233	−290	−344

Необходимо найти функции интерполяции для каждой из задач.

Примечание

Степень полинома целесообразно установить методом анализа табличных разностей, которые можно определить с помощью системы Matlab.

Решение задач многопараметрической интерполяции

Пусть исходные данные трех задач многопараметрической интерполяции приведены в табл. П1.4.

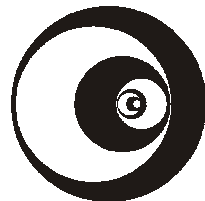
Необходимо найти функции интерполяции для каждой из задач.

Примечание

Задачи многопараметрической интерполяции рекомендуем решать с помощью одной из следующих математических систем: Mathematica, Maple, Derive. Системы Mathcad и Matlab использовать нецелесообразно, т. к. с помощью этих систем получить решение в виде аналитических функций затруднительно.

Таблица П1.4

		Значения переменных						
Задача 1	x	1	2.5	4.7	6.2	8.7	10	12.4
	y	9.8	7.5	5.8	4.1	2.9	1.8	0.5
	z	456	398	395	33.5	331	276	219
Задача 2	x	6.7	5.1	3.8	1.5	0.9	0.5	0.1
	y	20.3	25.6	31.5	42.8	57.4	70.2	96.3
	z	60	128	195	321	442	545	74.8
Задача 3	x	1.3	4.1	6.2	8	9.8	12	14.2
	y	15.4	11.6	9.3	8.1	6.9	5.2	4.3
	z	1	2	3.2	5	6.5	6	4.5
	w	130	113	105.3	105.5	105.8	99.1	99.8



Приложение 2

Задачи интерполяции в физике и химии

Исходные данные приведенных здесь задач интерполяции в физике и химии представляют собой закономерности физических явлений, представленные в табличной форме. Поэтому при их решении целесообразно использовать методы интерполяции точные в узлах. При этом наиболее трудным является выбор вида функции интерполяции. Если таковая найдена, то это равносильно открытию физического закона. Интерполяция дает возможность оценить погрешность экспериментальных данных, уточнить известный ранее физический закон и даже сделать открытие нового закона.

Если математическая модель, полученная в результате интерполяции по методу точному в узлах, не адекватна объекту, то причинами могут быть:

- неудачно выбрана функция интерполяции;
- велика погрешность исходных данных.

В таких случаях следует, не меняя вида функции интерполяции, воспользоваться методами приближенными в узлах. Если при этом погрешность интерполяции уменьшается, то причиной неадекватности модели является неточность исходных данных. В противном случае неадекватность объясняется неудачным выбором функции интерполяции. Далее приводится 10 задач. Задача считается решенной, если относительная максимальная погрешность интерполяции не превышает 5—10 %.

Исходные данные заимствованы из [24].

Задача П2.1

Зависимость температуры кипения жидкостей t от давления P приведена в табл. П2.1.

Таблица П2.1

	Значения переменных							
<i>P</i>, мм. рт. ст.	10	20	40	60	100	200	400	760
<i>t</i>, °С нитробензол	84.6	99.3	115.4	125.8	139.9	161.2	185.8	210.6
<i>t</i>, °С октан	19.2	31.5	45.1	53.8	65.7	83.6	104	125.6
<i>t</i>, °С ксилол	28.3	41.1	55.3	64.4	76.8	95.5	116.7	139.1

Необходимо методом интерполяции найти функцию $t = f(P)$ для каждой из жидкостей и определить температуру их кипения при давлениях 50, 300 и 600 мм. рт. ст.

Задача П2.2

Зависимость плотности жидкостей G от температуры t приведена в табл. П2.2.

Таблица П2.2

<i>t</i>, °С	Зависимость плотности жидкости G (кг/м ³) от температуры				
	Уксусная кислота	Пропиловый спирт	Четырех-хлористый углерод	Метиловый спирт	Бензол
0		819	1634	810	900
20	1049	804	1595	792	879
40	1028	788	2555	774	858
60	1006	770	1517	756	836
80	984	752	1477	736	815
100	960	733	1435	714	793
120	936	711	1391	690	769
140	909	688	1344	664	744
160		660	1297	634	719
180		629	1247	598	691

Необходимо найти функции интерполяции $G = f(t)$ и вычислить значения плотностей при температурах 70, 110 и 150 °С.

Задача П2.3

Зависимость давления насыщенного водяного пара P в мм. рт. ст. от температуры t приведена в табл. П2.3.

Таблица П2.3

	Зависимость температуры воды от давления										
$t, ^\circ\text{C}$	0	5	10	15	20	25	30	35	40	45	50
P	4.58	6.54	9.21	12.79	17.54	23.76	31.82	42.18	55.32	71.88	92.51
$t, ^\circ\text{C}$	55	60	65	70	75	80	85	90	95	100	
P	118	149.4	187.5	233.7	289.1	355.1	433.6	525.8	633.9	760	

Необходимо найти функцию интерполяции $P = f(t)$ и определить значения давления насыщенного водяного пара при температурах 22, 73 и 88 °С.

Задача П2.4

Ускорение свободного падения тела g на различной высоте над Землей h приведено в табл. П2.4.

Таблица П2.4

	Зависимость ускорения свободного падения тела от высоты								
$h, \text{км}$	0	0.05	0.1	0.5	1	2	3	5	10
$g, \text{м/сек}^2$	9.8066	9.8065	9.8063	9.8051	9.8036	9.8005	9.7974	9.7922	9.7759
$h, \text{км}$	20	30	50	100	500	5000	10000	50000	400000
$g, \text{м/сек}^2$	9.7452	9.7147	9.6542	9.505	8.45	3.08	105	0.13	0.0025

Необходимо найти функцию интерполяции $g = f(h)$ и определить значения ускорения на высотах 350, 2000 и 10000 км.

Задача П2.5

Плотность атмосферы Q на различной высоте над Землей h приведена в табл. П2.5.

Таблица П2.5

	Зависимость плотности атмосферы от высоты									
h , км	0	0.05	0.1	0.2	0.3	0.5	1	2	3	5
Q , м/сек ²	1.225	1.219	1.213	1.202	1.190	1.162	1.112	1.007	0.909	0.736
h , км	8	10	12	15	20	30	50	100	120	
Q , м/сек ²	0.026	0.414	0.312	0.195	0.089	0.018	$1.027 \cdot 10^{-3}$	$5.55 \cdot 10^{-7}$	$2.44 \cdot 10^{-8}$	

Необходимо методом интерполяции найти функцию $Q = f(h)$ и определить значения плотности атмосферы на высотах 6, 40 и 110 км.

Примечание

Плотность атмосферы зависит от широты места, времени года, солнечной активности. В таблице приведены усредненные значения плотности атмосферы. При этом на уровне Земли температура принята равной 15°C, давление 760 мм. рт. ст.

Задача П2.6

Скорость звука в воздухе C на различной высоте над Землей h приведена в табл. П2.6.

Таблица П2.6

	Зависимость скорости звука от высоты								
h , км	0	50	100	200	300	400	500	600	700
C , м/сек	340.29	340.10	339.91	339.53	339.14	338.76	338.38	337.98	337.60
h , км	800	900	1000	5000	10000	20000	50000	80000	
C , м/сек	337.21	336.82	336.43	320.54	299.53	295.07	293.8	282.54	

Необходимо методом интерполяции найти зависимость $C = f(h)$ и вычислить скорость звука на высотах 100, 7500 и 70000 м.

Примечание

Скорость звука, приведенная в таблице, получена в предположении, что температура и давление воздуха на поверхности земли равны соответственно 15°C и 760 мм. рт. ст.

Задача П2.7

Давление P и плотность насыщенных водяных паров воды Q при различных температурах t приведены в табл. П2.7.

Таблица П2.7

$t, ^\circ\text{C}$	$P, \text{мм.рт.ст.}$	$Q, \text{г/м}^3$	$t, ^\circ\text{C}$	$P, \text{мм.рт.ст.}$	$Q, \text{г/м}^3$
0	4.58	4.84	17	14.53	14.5
1	4.92	5.22	18	15.48	15.4
2	5.29	5.60	19	16.48	16.3
3	5.68	5.98	20	17.54	17.3
4	6.10	6.40	21	18.6	18.3
5	6.54	6.84	22	19.8	19.4
6	7.01	7.3	23	21.1	20.6
7	7.57	7.8	24	22.4	21.8
8	8.05	8.3	25	23.8	23
9	8.61	8.8	30	31.8	30.3
10	9.21	9.4	40	55.3	51.2
11	9.84	10.0	50	92.5	83
12	10.52	10.7	60	149.4	130
13	11.23	11.4	70	233,7	198
14	11.99	12.1	80	355.1	293
15	12.79	12.8	90	525.8	424
16	13.63	13.6	100	760	598

Данные таблицы позволяют сформулировать следующие две задачи:

- ☐ по данным таблицы найти функцию интерполяции $P = f(t)$;
- ☐ по данным таблицы найти функцию интерполяции $Q = f(t)$.

Необходимо оценить адекватность моделей абсолютной и относительной погрешностями. Максимальная относительная погрешность не должна превышать 5 %.

Задача П2.8

Зависимость температуры кипения воды t от давления ниже нормального атмосферного P приведена в табл. П2.8.

Таблица П2.8

P , мм.рт.ст.	t , °C	P , мм.рт.ст.	t , °C
4,6	0	634	95
9,2	10	680	96,9
17,5	20	700	97,7
31,8	30	710	98,1
55,3	40	720	98,5
92,5	50	730	98,9
233,7	70	740	99,3
289,0	75	750	99,6
403,0	83	760	100
526,0	90		

Требуется найти математическую модель явления кипения воды в зависимости от давления методом интерполяции. Решение необходимо получить с максимальной относительной погрешностью не превышающей 5 %.

Задача П2.9

Удельная теплота парообразования воды r при различной температуре t и нормальном атмосферном давлении приведена в табл. П2.9.

Таблица П2.9

t , °C	r , ккал/кг	t , °C	r , ккал/кг
0	597	80	551
5	594	100	539
10	592	160	497
15	589	200	464
18	587	300	335
20	586	370	105
30	580	374	27
50	569	374,15	0

Необходимо найти методом интерполяции математическую модель явления парообразования. Адекватность модели надо оценить погрешностью интерполяции. Максимальная относительная погрешность функции $r = f(t)$ не должна превышать 5 %.

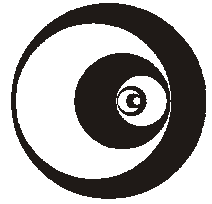
Задача П2.10

Сила света электрических ламп накаливания S в зависимости от их мощности W приведена в табл. П2.10.

Таблица П2.10

	Зависимость силы света от мощности лампы								
W , Вт	15	25	40	60	100	150	300	500	1000
S , кд	10	18	30	51	103	173	388	695	1530

Необходимо решить задачу интерполяции и вычислить силу света электрических ламп мощностью 75 Вт, 200 Вт и 750 Вт. Максимальная относительная погрешность функции интерполяции $S = f(W)$ не должна превышать 5 %.



Приложение 3

Задачи интерполяции в экономике

Задача оценки роста технико-экономических показателей предприятия

Технико-экономические показатели угледобывающей промышленности предприятия "Тула-уголь" за период с 1988 по 1996 годы приведены в табл. ПЗ.1.

Таблица ПЗ.1

	Ед. изм.	Технико-экономические показатели угледобывающей промышленности по годам								
		1988	1989	1990	1991	1992	1993	1994	1995	1996
Y_1	Тыс. тонн	193330	18810	15745	13740	13615	12550	8470	6210	5560
Y_2	Тыс. тонн	17967	15659	13267	10589	9216	7761	5135	3583	2583
Y_3	Тонн в сутки	52968	46630	39436	31862	27572	23368	15596	12020	8801
Y_4	Чел.	21969	19781	17686	17051	18465	18101	16325	11132	8747
Y_5	Тонн в месяц	68.1	66	62.5	51.8	41.6	35.7	26.2	26.8	24.7

В таблице приняты обозначения показателей:

□ Y_1 — производственная мощность;

□ Y_2 — добыча угля;

- Y_3 — среднесуточная добыча;
- Y_4 — численность рабочих;
- Y_5 — производительность труда.

По данным таблицы требуется найти закономерности изменения технико-экономических показателей предприятия "Тула-уголь" и осуществить прогнозирование темпов развития предприятия на ближайшие пять лет (до 2001г.). Для этого необходимо решить задачу интерполяции, получив пять математических функций $y_i = f(T)$ (где $i = 1, 2, 3, 4, 5$) и определить их значения для последующих пяти лет.

Задача оценки себестоимости производства на транспорте

Данные себестоимости машино-часов автопоездов в зависимости от расстояния вывозки леса приведены в табл. П3.2.

Таблица П3.2

Тип авто-поезда	Показатели себестоимости машино-часа вывозки леса в рублях на расстоянии 30—120км									
	30	40	50	60	70	80	90	100	110	120
КрАЗ-255Л	6.36	6.85	7.34	7.84	8.08	8.32	8.57	8.70	8.82	8.94
МАЗ-509	5.09	5.35	5.61	5.87	6.01	6.15	6.29	6.35	6.41	6.47
КАМАЗ-35410	6.60	7.09	7.59	8.09	8.34	8.59	8.83	8.92	9.01	9.10
Урал-375Н	6.05	6.43	6.81	7.18	7.37	7.55	7.74	7.93	8.12	8.31
ЗИЛ-131	5.36	5.55	5.74	5.94	6.02	6.10	6.19	6.29	6.39	6.49

Необходимо найти зависимость себестоимости машино-часа вывозки леса от расстояния для каждого из типов автопоездов. Решение этой задачи может вызвать затруднения, связанные с обеспечением высокой точности интерполяции: задача должна быть решена с точностью два знака после запятой (рубли и копейки). Если возникнут такие трудности, то следует обратиться к программам Formula или TableCurve. Они могут выбрать вид функции интерполяции, обеспечивающей допустимую погрешность модели.

Задача ценообразования за услуги

Данные опроса населения об оплате посещения лесов в разное время года приведены в табл. П3.3.

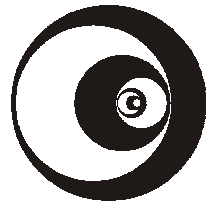
Таблица П3.3

	N — количество людей, согласных заплатить за разовое посещение леса Z рублей, (по данным опроса населения)				
Z	0	1-30	31-60	61-90	91 и выше
Весна	2595	1033	163	16	234
Лето	2103	1182	256	25	475
Осень	2509	1037	189	30	276

Требуется найти оптимальную цену за разовое посещение леса в разное время года. Оптимальной следует считать такую цену, при которой суммарный доход максимальный.

Задача может быть решена, если известна математическая модель за посещение леса в виде функции интерполяции $Z = f(N)$.

Рекомендуем решать задачу по методике, изложенной в *главе 8 (разд. 8.3)*, где приведен оптимальный вариант для случая посещения леса в зимнее время.



Приложение 4

Задачи интерполяции в таксации

Статистические данные хода роста древостоев сосновых насаждений приведены в табл. П4.1.

Таблица П4.1

	Зависимость роста древостоев сосновых насаждений от возраста												
<i>L</i> , лет	20	30	40	50	60	70	80	90	100	110	120	130	140
<i>H</i> , м	7.3	10.7	14	17.1	19.8	21.9	23.8	25.3	26.8	28	29	29.6	29.9
<i>D</i> , см	7.6	10.2	11.9	14.7	19.1	22.3	26.4	28.7	31	33.2	34.8	36	36.9
<i>N</i>	4631	3432	2563	1776	1190	886	678	604	531	475	448	421	402
<i>V</i> , м³	0.018	0.039	0.074	0.138	0.253	0.399	0.584	0.720	0.879	1.05	1.17	1.29	1.37
<i>Q</i> , м³	83	134	190	247	302	352	396	435	471	502	528	543	552
<i>G</i> , м³	4.1	4.8	5.2	5.7	6.1	6.3	6.4	6.4	6.4	6.3	6.1	5.8	5.6

В таблице приняты следующие обозначения:

- *L* — возраст древостоев;
- *H* — средняя высота;
- *D* — средний диаметр;
- *N* — число стволов на 1 га;
- *V* — объем средней модели;
- *Q* — запас древесины на 1 га;
- *G* — средний прирост.

Данные хода роста древостоев заимствованы из [25].

В таблице 7 параметров. Всего возможно сформировать 56 задач: 21 задачу на однопараметрическую интерполяцию и 35 задач на двухпараметрическую. Рекомендуем следующие варианты задач:

□ однопараметрическая интерполяция:

$HL, DL, NL, VL, GH, DH, VH, VD, GD, GV$;

□ двухпараметрическая интерполяция:

$QNV, QNL, GLH, GLD, GNV, DLH$.

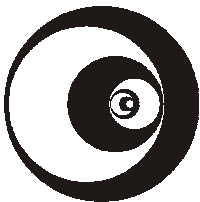
Первый символ в этих кодах задач означает функцию, последующие — аргументы. Например, код задачи VD означает: найти зависимость объема средней модели сосны V от среднего диаметра D .

Примечание

При решении задач таксации следует иметь в виду, что в соответствии с лесотаксационным справочником моделями хода роста древостоев являются модели полиномиальные. Нам остается выбрать лишь степень полинома. Это целесообразно сделать методом анализа табличных разностей с помощью системы Matlab.

Если вам удастся найти не полиномиальную функцию интерполяции, а иную с допустимой погрешностью, то вы близки к открытию закона о ходе роста древостоев.

Приложение 5



Задачи интерполяции в надежности

Задача определения параметра потока отказов

В эксплуатации находится $N = 200$ образцов техники. Данные об их отказах в течении 2000 часов эксплуатации приведены в табл.П5.1.

Таблица П5.1

	Зависимость числа отказов от времени работы									
Δt , час	200	200	200	200	200	200	200	200	200	200
$n(\Delta t)$	2	4	3	6	6	7	6	9	8	11

Надо найти математическую модель параметра потока отказов $w(t)$, пользуясь методами интерполяции.

Примечание

- 1) Табличные значения параметра потока отказов вычисляются по формуле $w(t) = \frac{n(\Delta t)}{N_0 \Delta t}$. При этом $w(0) = 0$, а значения $w(t)$, вычисленные по формуле и данным таблицы, относятся к середине интервала Δt .
- 2) Оцените погрешность математической модели и достаточность экспериментальных данных.

Задача определения интенсивности отказа элементов

В опыте по определению показателей надежности находилось $N = 400$ элементов. В течение 720 часов их испытания произошло 44 отказа. Распределение отказов во времени приведено в табл. П5.2.

Таблица П5.2

	Распределение отказов во времени									
Δt , час	72	72	72	72	72	72	72	72	72	72
$n(\Delta t)$	2	2	3	4	4	6	5	5	6	7

Необходимо найти математическую модель интенсивности отказов элементов $\lambda(t)$, определить их вероятность и среднее время безотказной работы, а также определить погрешность модели.

Примечание

Интенсивность отказов элементов надо вычислить по формуле:

$$\lambda(t) = \frac{n(\Delta t)}{N_0 \Delta t}, \text{ где } N_{cp} = \frac{N_H + N_K}{2},$$

N_H — число исправных элементов в начале промежутка времени Δt ,

N_K — число исправных элементов в конце промежутка времени Δt .

Задача нахождения закона распределения времени до отказа

В эксплуатации находится $N_0 = 500$ образцов техники. За 720 часов ее работы произошло n отказов. Распределение отказов во времени приведено в табл. П5.3.

Таблица П5.3

	Распределение отказов во времени									
Δt , час	72	72	72	72	72	72	72	72	72	72
$n_1(\Delta t)$	4	8	11	15	18	20	22	24	26	26
$n_2(\Delta t)$	2	7	12	17	21	25	27	28	29	29
$n_3(\Delta t)$	40	36	33	31	28	26	24	22	20	18
$n_4(\Delta t)$	1	1	2	2	3	3	4	4	4	5
$n_5(\Delta t)$	43	54	58	56	52	46	39	33	26	21

Необходимо определить вид закона распределения до отказа и его параметры.

Примечание

1. В задаче пять вариантов. Каждый из них соответствует одному из следующих законов: Вейбулла, Рэлея, гамма, экспоненциальному. Необходимо найти закон распределения для данного варианта.

2. Вид закона распределения можно установить по виду графика функции $f(t)$.

Табличные значения $f(t)$ вычисляются по формуле $f(t) = \frac{n(\Delta t)}{N_0 \Delta t}$.

3. Математическую модель функции $f(t)$ следует искать методом интерполяции точным в узлах.

Задача определения показателей надежности самолета ТУ-154М по данным эксплуатации

Данные об отказах пяти систем восьми самолетов ТУ-154М в течение двух лет эксплуатации приведены в табл. П5.4.

Таблица П5.4

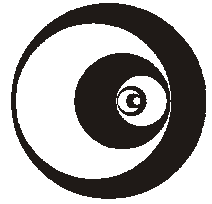
Δt , час	Двигатель			Шасси		Система кондицион.		Система управления		Радио-аппаратура	
	$n(\Delta t)$	$\omega 10^{-3}$ отк/час		$n(\Delta t)$	$\omega 10^{-3}$ отк/час	$n(\Delta t)$	$\omega 10^{-3}$ отк/час	$n(\Delta t)$	$\omega 10^{-3}$ отк/час	$n(\Delta t)$	$\omega 10^{-3}$ отк/час
2351	13	0,7		14	0,7	18	1	11	0,05	36	2
4066	12	0,4		35	1,1	18	0,5	15	0,5	68	2
4596	14	0,4		30	0,8	28	0,8	13	0,35	70	2
3381	8	0,3		27	0,99	16	0,6	11	0,4	40	1,5
2630	13	0,6		34	1,6	30	1,4	13	0,6	41	2
3665	19	0,6		28	0,9	23	0,8	10	0,3	44	1,5
4585	15	0,4		30	0,8	63	2	13	0,35	59	1,6
3158	9	0,3		29	1,1	28	1	8	0,3	43	2

В таблице обозначено:

- Δt — часы налета самолетов в течение каждого квартала двухлетней эксплуатации;
- $n(\Delta t)$ — число отказов за промежуток времени Δt .

Примечание

1. В задаче пять независимых вариантов в соответствии с числом подсистем самолета.
2. Прежде, чем решать задачу, следует ознакомиться с методикой определения параметра потока отказов $\omega(t)$, описанной в *главе 8 (разд. 8.6)*.
3. Специалистам по надежности авиационной или другой техники весьма полезно по полученной методом интерполяции функции $\omega(t)$ определить показатели надежности каждой из систем, пользуясь компьютерной технологией, описанной в *главе 8*.



Приложение 6

Задачи интерполяции в системах массового обслуживания

Задача о числе обслуживающих бригад и вероятности обслуживания

Зависимость вероятности отказа P в обслуживании заявки в многоканаль-

ной системе выражается формулой:
$$P = \frac{\rho^n}{n! \sum_{i=0}^n \frac{\rho^i}{i!}}.$$

Здесь:

□ $\rho = \frac{\lambda}{\mu};$

□ n — число обслуживающих бригад;

□ λ — интенсивность потока заявок;

□ μ — интенсивность обслуживания заявки.

Зависимость $n = f(\rho)$ при различных значениях P , рассчитанная по формуле, приведена в табл. Пб.1.

Необходимо найти функцию интерполяции зависимости числа обслуживающих органов n от ρ , обеспечивающих заданную вероятность P .

Примечание

В задаче четыре варианта. Для их решения воспользуйтесь методикой главы 8, где приведена сформулированная задача для случая $P = 0.01$.

Таблица П6.1

<i>P</i>	Число обслуживающих бригад, <i>n</i>									
	ρ									
	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
0,1	1.9	2.66	3.33	4	4.6	5.15	5.82	6.3	7	7.45
0,05	2.26	3.17	4	4.72	5.4	6.05	6.74	7.35	8	8.58
0,025	2.65	3.68	4.56	5.37	6.13	6.88	7.58	8.26	8.95	9.6
0,003	3.7	5	6.11	7.1	8.03	8.9	9.76	10.57	11.37	12.15

Задача о времени пребывания заявки в очереди

Среднее время $T_{\text{оч}}$ пребывания заявки в очереди многоканальной системы массового обслуживания с неограниченным числом заявок выражается формулой:

$$T_{\text{оч}} = \frac{\rho^{n+1} P_0}{n! n \left(1 - \frac{\rho}{n}\right)^2 \lambda}$$

Здесь:

$$\square \rho = \frac{\lambda}{\mu};$$

$\square n$ — число каналов обслуживания;

$\square \lambda$ — интенсивность потока заявок;

$\square \mu$ — интенсивность обслуживания заявки;

$\square P_0$ — вероятность того, что в системе на обслуживании заявки нет.

Рассчитанная по формуле зависимость $n = f(\rho)$ при постоянных значениях P_0 , $T_{\text{оч}}$, λ приведена в табл. П6.2.

Необходимо найти методом интерполяции зависимость числа обслуживающих бригад n , обеспечивающих заданное среднее время пребывания заявки в очереди от ρ (где $T_{\text{оч}} = 0.25$ час). Интенсивность потока заявок величина постоянная ($\lambda = 32$ ед/час). В задаче семь вариантов для различных значений P_0 .

Таблица П6.2

<i>P</i>	Число обслуживающих бригад, <i>n</i>					
	ρ					
	0.5	1	1.5	2	2.5	3
0,01	0.52	1.04	1.56	2.1	2.35	3.23
0,05	0.47	0.92	1.64	1.78	2.17	3.12
0,1	0.44	0.89	1.63	1.85	2.31	3.24
0,2	0.43	0.85	1.24	1.59	1.92	2.22
0,3	0.42	0.82	1.19	1.51	1.79	2.05
0,4	0.41	0.79	1.14	1.45	1.72	1.81
0,5	0.4	0.77	1.1	1.39	1.64	1.73

Список литературы

1. Тюрин Ю.Н., Макаров А.А. Анализ данных на компьютере. — М.: Финансы и статистика, 1995.
2. Очков В.Ф. Mathcad PLUS 6.0. — М.: КомпьютерПресс, 1996.
3. Аладьев В.З., Гершгорн Н.А. Вычислительные задачи на персональном компьютере. — Киев: Тэхника, 1991.
4. Аладьев В.З., Шишаков М.Л. Введение в среду пакета Mathematica 2.2. — М.: Филинь, 1997.
5. Дьяконов В.П. Справочник по системе символьной математики. — М.: СК Пресс, 1998.
6. Дьяконов В.П. Системы символьной математики Mathematica 2 и Mathematica 3. — М.: СК Пресс, 1998.
7. Манзон Б.М. Maple V Power Edition. — М.: Филинь, 1998.
8. Говорухин В.Н., Цибулин В.Г. Введение в Maple. — М.: Мир, 1997.
9. Mathcad. Руководство пользователя Mathcad 6.0, Mathcad Plus 6.0. Перевод с английского. — М.: Филинь, 1996.
10. Дьяконов В.П. Mathcad 8/2000. Специальный справочник. — СПб.: Питер, 2000.
11. Дьяконов В.П. Справочник по системе символьной математики Derive. — М.: СК Пресс, 1998.
12. Дьяконов В.П. Компьютерная математика. Теория и практика. — М.: Нолидж, 2001.
13. Аладьев В., Шишаков М. Автоматизированное рабочее место математика. — М.: Лаборатория базовых знаний, 2000.
14. Егоренков Д.Л., Фрадков А.Л., Харламов В.Ю. Основы математического моделирования. — СПб.: БГТУ, 1996.

15. Дьяконов В. Matlab. Учебный курс. – СПб.: Питер, 2001.
16. Потемкин В.Г. Система Matlab. Справочное пособие. – М.: Диалог Мифи, 1997.
17. Дьяконов В., Круглов В. Математические пакеты расширения Matlab. Специальный справочник. – СПб.: Питер, 2001.
18. Демидович Б.П., Марон И.А. Основы вычислительной математики. – М.: Наука, 1970.
19. Пулькин С.П. Вычислительная математика. – М.: Просвещение, 1972.
20. Копченкова Н.В., Марон И.А. Вычислительная математика в примерах и задачах. – М.: Наука, 1972.
21. Форсайт Дж., Малькольм М., Моулер К. Машинные методы математических вычислений – М.: Мир, 1980.
22. Воробьева Г.Н., Данилова А.Н. Практикум по вычислительной математике. – М.: Высшая школа, 1990.
23. Гутер Р.С., Резниковский П.Т. Программирование и вычислительная математика, выпуск 2. – М.: Наука, 1971.
24. Енохович А. С. Справочник по физике и технике. – М.: Просвещение, 1989.
25. Лесотехнический справочник по северо-западу СССР, под ред. Мошалева А. Г. – Ленинград.: ЛТА, 1984.
26. Адлер Ю. П., Маркова Е. В., Грановский Ю. В. Планирование эксперимента при поиске оптимальных условий. – М.: Наука, 1976.
27. Налимов В. В., Чернова Н. А. Статистические методы планирования экстремальных экспериментов. – М.: Наука, 1965.
28. Розенберг В. Я., Прохоров А. И. Что такое теория массового обслуживания. – М.: Советское радио, 1962.
29. Ивченко Г. И., Каштанов В. А., Коваленко И. Н. теория массового обслуживания. – М.: Высшая школа, 1982.
30. Гнеденко Б. В., Коваленко И. Н. Введение в теорию массового обслуживания. – М.: Наука, 1987.
31. Половко А. М. Основы теории надежности. – М.: Наука, 1964.
32. Надежность технических систем: справочник. Под ред. Ушакова И. А. – М.: Радио и связь, 1985.
33. Сучак Е. В., Василенко Н. В. и др. Надежность технических систем. – Красноярск.: МГП Раско, 2001.
34. Герман О. В. Введение в теорию экспертных систем и обработку знаний. – Минск.: Дизайн ПРО, 1995.

Предметный указатель

А

Абсолютная погрешность, 95, 100
Абсолютная среднеквадратическая погрешность, 61, 134, 135, 139, 187, 189, 210, 218, 223, 239, 246, 272, 284
Аппроксимация Паде, 48, 49, 72, 96, 120

Д

Дробно-линейная функция, 38, 41, 65, 88

З

Закон Вейбулла, 245
Закон Парето, 259

И

Интерполяционные полиномы, 22
Интерполяция алгебраическими многочленами, 22
Интерполяция приближенная в узлах, 35, 67, 92, 112
Интерполяция точная в узлах, 19, 51, 64, 77, 101, 127, 163, 174

К

Команда **with(numapprox)**, 125
Команда **Digits**, 113, 186
Команда **evalf**, 102
Команда **readlib(mtaylor)**, 125
Команда **with(linalg)**, 107
Команда **with(numapprox)**, 121, 122
Команда **with(orthopoly)**, 122
Команда **with(stats)**, 113

Л

Логарифмическая функция, 38, 41, 65

М

Максимальная относительная погрешность, 62, 95, 100, 108, 168, 187, 189, 210, 218, 223, 239, 246, 272, 285
Метод выбранных точек, 42, 46, 209
Метод итераций, 67
Метод наименьших квадратов, 42, 44, 47, 212, 254
Метод Ньютона, 66, 74
Метод средних, 42, 43, 44, 47
Метод табличных разностей, 222
Минимальная относительная погрешность, 189, 246

О

Определитель Вандермонда, 23
Относительная максимальная погрешность, 139

П

Погрешность интерполяции, 107
Показательная функция, 39, 65
Полиномы Чебышева, 49, 121

Р

Ряд Тейлора, 125

С

Сплайн-интерполяция, 122, 142, 173
Сплаины, 33

Среднеквадратичная погрешность, 67

Степенная функция, 38, 41, 65

Т

Табличные разности, 24, 25

Ф

Формула Бесселя, 33

Формула Гаусса, 31, 32

Формула Лагранжа, 23, 24, 25

Формула Ньютона, 27

Формула Стирлинга, 32

Формулы Ньютона, 29, 74

Функция ALL_SEVEN, 71, 215, 216, 219

Функция allvalues, 117

Функция chebpade, 121

Функция Collect, 87

Функция confracform, 125

Функция diff, 178, 222

Функция evalm, 107

Функция Expand, 85

Функция ezplot, 158, 168

Функция Factor, 85

Функция Find, 130, 131

Функция FindRoot, 80, 89, 99

Функция fit, 112, 185, 255

Функция Fit, 92, 93, 99, 100, 187, 201

Функция FIT, 67, 69, 70, 182, 184, 225, 230, 231, 239, 261

Функция FIXED_POINT, 67

Функция fsolve, 115, 117, 119, 172, 173

Функция genfit, 150

Функция hornerform, 125

Функция icubic, 180

Функция intercept, 145

Функция interp, 111, 142, 144, 146, 147, 194, 195

Функция interp1, 174, 175, 180

Функция interp2, 195

Функция InterpolatePolynomial, 86

Функция InterpolatingPolynomial, 85

Функция Interpolation, 91

Функция isolve, 116

Функция LAGRANGE, 63

Функция laurent, 125

Функция length, 169

Функция linfit, 148, 149

Функция linterp, 141

Функция ListPlot, 97, 99

Функция loess, 147, 148, 194

Функция lsolve, 129, 130

Функция lsqcurvefit, 176

Функция MatrixForm, 84

Функция min, 169

Функция Minerr, 133, 134

Функция minimax, 125

Функция msolve, 116

Функция NEWTONS, 67

Функция NSolve, 78

Функция Pade, 96

Функция PADE, 72

Функция plot, 105, 107, 156, 157, 158, 167

Функция Plot, 97

Функция Plus, 95

Функция POLY_INTERPOLATE, 63

Функция polyfit, 177, 179

Функция polyval, 177, 179

Функция Print, 83

Функция readlib, 123

Функция regress, 146, 194

Функция rsolve, 116

Функция seq, 105

Функция Show, 98

Функция Simplify, 85

Функция slope, 145

Функция solve, 101, 103, 115, 119, 164

Функция Solve, 77, 78, 89

Функция SOLVE, 237

Функция spline, 122, 173

Функция subs, 103, 166, 167

Функция sum, 169

Функция syms, 164

Функция Table, 83, 84, 90

Функция taylor, 125

Функция TOL, 133

Функция VECTOR, 56, 73, 216, 225

Функция vpa, 165

Функция with(stats), 186

Ц

Центральные разности, 26