

А. М. Половко

Mathematica

ДЛЯ
СТУДЕНТА

А. М. Половко

Mathematica

ДЛЯ СТУДЕНТА

Санкт-Петербург
«БХВ-Петербург»
2007

УДК 681.3.06(075.8)
ББК 32.973-26-018.2я73
П52

Половко А. М.

П52 Mathematica для студента. — СПб.: БХВ-Петербург, 2007. — 368 с.: ил.

ISBN 978-5-9775-0096-8

Изложены компьютерные технологии решения математических задач в универсальной математической среде Mathematica, приводятся алгоритмы и методы решения. Книга содержит примеры на каждый из методов, оригинальные индивидуальные задания, различные варианты задач, в том числе повышенной сложности, что дает возможность преподавателю осуществлять контроль знаний, а студенту более глубоко изучить систему Mathematica как систему компьютерной алгебры. Небольшой объем, сочетание научности и простоты изложения делают книгу удобной для повседневного пользования.

Для студентов, аспирантов, преподавателей технических вузов и специалистов, применяющих математические вычисления в профессиональной деятельности

УДК 681.3.06(075.8)
ББК 32.973-26-018.2я73

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Татьяна Лапина</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Игоря Цырульниковца</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 24.04.07.

Формат 60×90¹/₁₆. Печать офсетная. Усл. печ. л. 23.

Тираж 1000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.02.953 Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

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

ISBN 978-5-9775-0096-8

© Половко А. М., 2007
© Оформление, издательство "БХВ-Петербург", 2007

Оглавление

Введение	10
Особенности системы Mathematica	11
О книге "Mathematica для студента"	12
Компьютерная математика в науке и образовании	13
Для кого эта книга?	14
 Глава 1. Интерфейс системы Mathematica	16
1.1. Интерфейс системы и его изучение	16
1.2. Главное меню системы	17
1.2.1. Меню <i>File</i>	18
1.2.2. Меню <i>Edit</i>	21
1.2.3. Меню <i>Cell</i>	22
1.2.4. Меню <i>Format</i>	25
1.2.5. Меню <i>Input</i>	26
1.2.6. Меню <i>Kernel</i>	28
1.2.7. Меню <i>Find</i>	30
1.2.8. Меню <i>Window</i>	31
1.2.9. Меню <i>Help</i>	31
Работа с электронной книгой	32
 Глава 2. Основы работы с системой Mathematica в режиме вычислений	34
2.1. Арифметические операторы, функции, константы	34
2.1.1. Арифметические операторы	35
2.1.2. Арифметические функции	35
Функции выполнения арифметических операций	35
Определение делителей целых чисел и наименьшего общего кратного	36

Приведение вещественных чисел к ближайшим целым	37
Вычисление факториалов	37
Получение простых чисел.....	38
2.1.3. Именованные константы	39
2.1.4. Укороченная форма представления арифметических операций	41
2.2. Типы данных.....	42
2.2.1. Арифметические операции с целыми и рациональными числами.....	42
2.2.2. Арифметические операции с вещественными числами	44
2.2.3. Арифметические операции с комплексными числами	45
2.2.4. Переменные	47
2.3. Выражения, их преобразования и вычисления	50
2.3.1. Функции, операторы и символы вычисления выражений.....	50
2.3.2. Подстановки	52
2.3.3. Преобразование выражений.....	55
Функция <i>Simplify</i>	55
Функция <i>FullSimplify</i>	56
Функции <i>Expand</i>	58
Функция <i>Collect</i>	60
Функции <i>Factor</i>	61
Функции преобразования тригонометрических выражений	63
2.3.4. Примеры на преобразование выражений.....	64
2.3.5. Повышение точности вычислений	67
Глава 3. Визуализация вычислений.....	70
3.1. Двумерная графика	71
3.1.1. Графическая функция <i>Plot</i>	71
Определение области изоляции корня.....	72
Проверка достоверности решения задачи	74
Опции функции <i>Plot</i>	75
3.1.2. Построение точечного графика	81
3.1.3. Выбор стиля графика.....	86
Опция <i>PlotStyle</i>	86
3.1.4. Обозначение кривых на графике множества функций	90
3.1.5. Графики специальных типов.....	92
Построение графиков в логарифмическом масштабе	92
Построение графиков в полярной системе координат	94
Построение графиков в виде гистограмм.....	96

3.2. Трехмерная графика.....	101
3.2.1. Создание контурных графиков.....	102
3.2.2. Построение графиков поверхностей.....	102
3.2.3. Построение фигур.....	104

Глава 4. Специальные вычисления 107

4.1. Вычисление сумм.....	107
4.1.1. Вычисление сумм в аналитическом виде.....	107
4.1.2. Вычисление сумм в численном виде.....	110
4.1.3. Использование символа суммирования $\sum_{\oplus}^{\otimes}$	111
4.1.4. Примеры вычисления сумм.....	112
4.2. Вычисление произведений.....	114
4.2.1. Вычисление произведений в аналитическом виде.....	115
4.2.2. Вычисление произведений в численном виде.....	116
4.2.3. Использование символа произведения Π	117
4.2.4. Примеры вычисления произведений.....	118
4.3. Табулирование функции.....	119
4.4. Вычисление пределов.....	124
4.4.1. Технология вычисления пределов системой Mathematica.....	124
4.4.2. Примеры вычисления пределов.....	128
4.5. Разложение функции в степенной ряд.....	129
4.5.1. Технология разложения функции в ряд Тейлора в системе Mathematica.....	129
4.5.2. Погрешности степенных рядов Тейлора.....	134
Случай 1. Ряд знакопеременный, члены ряда быстро убывают, $x < 1$	134
Случай 2. Ряд знакопеременный, $x > 1$	135
Случай 3. Ряд не знакопеременный.....	135
4.5.3. Компьютерные технологии оценки погрешностей рядов.....	135
Способ 1. Табулирование функций.....	135
Способ 2. Визуализация решения.....	136
Способ 3. Вычисление погрешностей.....	137
4.6. Вычисление производных.....	138

Глава 5. Представление данных. Создание векторов и матриц 142

5.1. Типы данных.....	142
5.1.1. Вещественные числа.....	144
5.1.2. Комплексные числа.....	146

5.1.3. Символьные переменные	146
5.1.4. Списки и массивы	146
5.2. Представление и образование векторов и матриц.....	147
5.2.1. Генерация векторов и матриц с помощью функции <i>Range</i>	148
5.2.2. Генерация векторов и матриц с помощью функций <i>Table</i>	149
5.2.3. Выделение и вывод элементов вектора и матрицы.....	150
Использование двойных квадратных скобок	150
Выделение элементов вектора и матрицы с помощью функции <i>Part</i>	151
5.3. Работа со списками. Создание векторов и матриц.....	154
5.3.1. Выявление структуры вектора или матрицы	154
5.3.2. Преобразование и создание векторов и матриц	156
5.3.3. Комбинирование векторов и матриц.....	159
5.3.4. Создание векторов и матриц.....	160
5.4. Математические операции над векторами и матрицами	162
5.4.1. Арифметические операции	162
5.4.2. Матричное умножение	162
5.4.3. Математические операции с векторами и матрицами	163

Глава 6. Математические функции 166

6.1. Элементарные математические функции.....	166
6.1.1. Непосредственное вычисление функции	167
6.1.2. Присвоение аргументу численного значения	168
6.2. Специальные математические функции.....	168
6.2.1. Ортогональные полиномы	168
6.2.2. Интегральные показательные функции	172
6.2.3. Гамма-функция.....	174
6.2.4. Функции Бесселя.....	179
6.2.5. Функции Эйри.....	182
6.2.6. Бета-функция (эйлеров интеграл первого рода).....	185
6.2.7. Функции статистических распределений и функции ошибок	187
6.2.8. Функции генерации случайных чисел.....	191
6.2.9. Эллиптические интегралы и функции.....	194
6.2.10. Специальные числа и полиномы	197
6.3. Интегральные преобразования.....	198
6.3.1. Преобразование Лапласа.....	198
6.3.2. Z-преобразование.....	203
6.3.3. Преобразование Фурье	204

Глава 7. Решение оптимизационных задач.....	207
7.1. Поиск минимального и максимального числа в перечне чисел.....	207
7.2. Классический метод определения экстремума аналитической функции.....	210
7.2.1. Определение координат точек перегиба	213
7.3. Поиск локального минимума аналитической функции с помощью встроенных функций системы Mathematica.....	215
7.4. Отыскание глобального максимума (минимума) аналитической функции.....	220
7.4.1. Математическая формулировка задачи.....	222
7.5. Примеры на решение оптимизационных задач	226
Глава 8. Решение алгебраических и трансцендентных уравнений	229
8.1. Решение уравнений в аналитическом виде	229
8.1.1. Функция <i>Solve</i>	229
8.1.2. Функция <i>Roots</i>	233
8.2. Численные методы решения алгебраических и трансцендентных уравнений.....	237
8.2.1. Функция <i>NSolve</i>	237
8.2.2. Функция <i>NRoots</i>	239
8.2.3. Функция <i>FindRoot</i>	240
8.3. Интервальные методы решения алгебраических и трансцендентных уравнений	242
8.4. Определение корней уравнений с применением интерполяции	245
8.5. Проверка достоверности решения уравнений	247
Глава 9. Решение систем уравнений в среде Mathematica... 252	
9.1. Методы и алгоритмы решения систем алгебраических уравнений.....	252
9.1.1. Решение систем линейных алгебраических уравнений.....	252
Аналитические методы	253
Методы итераций	254
Алгоритмы метода итерации	256
Сравнительная оценка точных и итерационных методов	257
9.1.2. Решение систем нелинейных алгебраических уравнений.....	258
Определение начальных приближений.....	258
Графический способ.....	258
Табличный способ	259
Метод Ньютона.....	259

Итерационные методы	259
Признак окончания итераций	260
9.2. Компьютерные технологии решения уравнений в системе Mathematica	260
9.2.1. Функция $Solve[F, X]$	260
Решение систем линейных алгебраических уравнений.....	261
Решение систем нелинейных уравнений в символьном виде	265
9.2.2. Функция $Solve[F, X, Y]$	266
9.2.3. Функция $NSolve[F, X]$	268
9.2.4. Опции функции $Solve$	269
9.2.5. Функция $FindRoot[F, \{X, x_0\}]$	270
9.2.6. Функция $Eliminate[F, x]$	272
9.2.7. Матричные методы решения систем линейных уравнений ...	274
9.2.8. Особые случаи решения систем уравнений.....	277
9.3. Примеры для самостоятельного решения систем уравнений.....	279
9.3.1. Варианты систем линейных алгебраических уравнений.....	279
9.3.2. Варианты систем нелинейных алгебраических уравнений	279

Глава 10. Решение дифференциальных уравнений..... 284

10.1. Методические замечания.....	284
10.2. Решение дифференциальных уравнений в среде Mathematica	284
10.2.1. Аналитические методы.....	285
Частное решение дифференциального уравнения	286
Решение систем дифференциальных уравнений в аналитическом виде.....	288
Опции функции $DSolve$	292
10.2.2. Численные методы решения дифференциальных уравнений	292
Функция $NDSolve[f, y[x], \{x, x_{min}, x_{max}\}]$	292
Функция $NDSolve[\{f_1, f_2, \dots, y_1(x_0), y_2(x_0), \dots\}, \{y_1[x], y_2[x], \dots\}, \{x, x_{min}, x_{max}\}]$	295

Глава 11. Компьютерные технологии вычисления интегралов 304

11.1. Аналитические методы вычисления интегралов.....	304
11.2. Численные методы вычисления интегралов	307
11.3. Технология вычисления интегралов численными методами	308
11.4. Использование символа интеграла ($\int$)	309
11.5. Вычисление кратных интегралов.....	309
11.6. Вычисление несобственных интегралов	311
11.7. Табличное интегрирование	312

11.8. Проверка правильности вычисления интеграла	317
11.8.1. Вычисление производной первообразной функции	317
11.8.2. Применение различных методов интегрирования	319
11.8.3. Сравнение результатов интегрирования различными системами символьной математики	320
11.8.4. Особенности вычисления интегралов в системе Mathematica.....	321

Глава 12. Компьютерные технологии решения задач интерполяции..... 323

12.1. Виды и этапы компьютерных технологий интерполяции	323
12.1.1. Выбор вида функции интерполяции.....	325
Способ 1. Графоаналитический.....	326
Способ 2. Линеаризация нелинейных функций	327
Способ 3. Анализ табличных разностей.....	328
Способ 4. Использование специальных программ автоматизации интерполяции	329
12.1.2. Определение коэффициентов функции интерполяции.....	329
12.1.3. Определение адекватности функции интерполяции.....	329
12.2. Компьютерные технологии интерполяции в среде Mathematica..	330
12.2.1. Интерполяция, точная в узлах	330
Универсальный метод.....	330
Проверка достоверности решения задачи интерполяции	332
Функция <i>InterpolatingPolynomial</i>	335
Функция <i>Interpolation[data]</i>	338
12.2.2. Интерполяция нелинейными функциями.....	339
Способ 1. Решение системы нелинейных уравнений.....	339
Способ 2. Линеаризация нелинейной функции.....	341
12.3. Интерполяция, приближенная в узлах.....	343
12.4. Паде-аппроксимация.....	348
12.5. Функции аппроксимации в пакетах расширения.....	350
12.5.1. Линейная аппроксимация.....	350
12.5.2. Нелинейная аппроксимация.....	352
12.5.3. Полиномиальная аппроксимация	353
12.5.4. Сплайн-интерполяция.....	355
12.6. Многопараметрическая интерполяция.....	357

Литература 360

Предметный указатель 362

Введение

Книга "Mathematica для студента" пополняет серию книг символьной математики: "Derive для студента", "MATLAB для студента", "Mathcad для студента", выпущенных в свет в 2005—2006 гг. издательством "БХВ-Петербург".

Будем надеяться, что в скором времени издательство выпустит в свет книгу "Maple для студента" — последнюю из серии книг символьной математики. Если это произойдет, то студент получит полную интеллектуальную библиотеку книг компьютерной алгебры. Эти книги будут ему нужны в течение всего периода обучения в вузе.

Ни одна из них в отдельности не может обеспечить учебный процесс по всем предметам данной специальности. Все они нужны студенту при выполнении упражнений и домашних заданий по различным математическим и техническим дисциплинам, при выполнении лабораторных работ, при курсовом и дипломном проектировании.

Не все студенты могут иметь такую библиотеку. Малоимущих должны обеспечить этими книгами библиотеки вузов. Таковы требования компьютеризации обучения, призванного существенно повысить его эффективность.

Появление систем символьной математики — революция в науке и образовании.

Совсем недавно компьютер был лишь мощным калькулятором и не обладал свойством интеллектуального технического средства.

С его помощью можно было решать задачи только в численном виде. При этом пользователь обязан составлять программы на универсальных языках программирования (Фортран, Си, Бейсик и т. п.). Это мог делать только профессиональный программист.

Этим можно объяснить слабое использование вычислительной техники в инженерном деле, науке и образовании.

Появление систем символьной математики, таких как Mathematica, Maple, Derive, Mathcad, MATLAB, существенно изменило ситуацию.

Теперь пользователю нет необходимости составлять программы на универсальном языке программирования. Они уже составлены и в качестве готовых программ находятся в памяти компьютера. Требуется лишь обратиться к ним, а

программирование состоит лишь в том, чтобы объединить некоторые из них в единую программу, соответствующую алгоритму решения задачи.

Обращение к программам осуществляется с помощью встроенных функций или команд. Тогда сущность программирования состоит в том, чтобы создать комбинацию встроенных функций, соответствующую алгоритму. Такое функциональное программирование настолько простое, что им легко овладеет любой пользователь.

Особенности системы Mathematica

- Система Mathematica является очень популярной во всем мире. Так, например, в США официально зарегистрировано свыше миллиона ее пользователей. Издано несколько десятков книг различного назначения.

Считается, и не без основания, что Mathematica — лидер среди систем символьной математики.

- Высокие интеллектуальные возможности системы Mathematica позволяют решать задачи в аналитическом виде.

Преобразования математических выражений осуществляются на таком высоком уровне, что система позволяет получать решения большинства математических задач в аналитическом виде, выводить формулы, доказывать теоремы. Система позволяет:

- определять вещественные и комплексные корни алгебраических и трансцендентных уравнений;
- решать алгебраические и дифференциальные уравнения;
- вычислять неопределенные интегралы;
- осуществлять интегральные преобразования;
- решать задачи оптимизации;
- осуществлять разложение функции в степенной ряд;
- находить пределы, вычислять суммы и произведения математических функций;
- осуществлять упрощение сложных математических выражений до таких уровней, когда выражение становится формулой;
- осуществлять самопроверку результатов решения задачи.

- Mathematica — мощная вычислительная система. Она позволяет без программирования получать численные решения большинства задач прикладной математики. Система поражает объемом вычислений. Например, функции π , e , $n!$ вычисляет практически с любым числом знаков. Она способна выполнять математические действия с абсолютной точностью. При этом количество цифр не ограничено.

Mathematica позволяет также выполнять вычисления с произвольной точностью. Например, встроенная функция $N[F, n]$ вычисляет математическое выражение F с числом знаков после запятой, равным n . При этом n не ограничено.

- ❑ Mathematica — справочная математическая система. Теперь математические справочники не нужны. В считанные секунды пользователь получит таблицы логарифмов, элементарных и специальных функций, таблицы производных, интегралов, сумм и произведений и т. д.
- ❑ Mathematica — справочная система, содержащая несколько электронных книг для пользователя с большим числом примеров.
- ❑ Решение задач осуществляется в режиме диалога и не требует программирования. Язык общения системы — язык функционального программирования высокого уровня. Его можно отнести к классу интерпретаторов, когда система анализирует (интерпретирует) введенное выражение и сразу его исполняет.
- ❑ Оригинальна структура системы Mathematica, состоящая из ядра, пакетов расширения, библиотеки и справочной системы из шести электронных книг. Ядро системы практически не зависит от платформы. Такая структура означает, что система Mathematica является системой самообучающей.

Система связана с Интернетом. Ее главный сайт посвящен системе Mathematica. В нем содержатся различные версии системы, книги, всемирная система вычисления интегралов, галерея графики, дополнительные библиотеки и пакеты расширения.

О книге "Mathematica для студента"

В России имеется только несколько книг по системе Mathematica. Они изданы небольшими тиражами и приведены в списке литературы. По своему объему и содержанию они не могут удовлетворить инженера, студента, преподавателя. Нужна книга для широкого пользователя, в которой бы в доступной для понимания форме излагались компьютерные технологии решения математических и прикладных задач. Такой, кажется, и является настоящая книга. Далее излагаются ее основные особенности.

- ❑ В книге подробно излагаются компьютерные технологии решения математических задач:
 - корректная постановка задачи;
 - выбор алгоритма и метода решения задачи;
 - выбор встроенной функции;
 - решение задачи;
 - проверка достоверности решения;
 - обсуждение результатов.

При решении задач в качестве примеров эта технология соблюдается. Она позволяет: понимать сущность задачи, получать знания, а не только умения и навыки, находить правильное решение за короткое время.

- Целенаправленность книги. В ней излагается все то главное, для чего предназначена Mathematica — выполнять вычисления, получая решения в численном или аналитическом виде.

Многие возможности системы, имеющие лишь косвенное отношение к вычислениям, излагаются поверхностно. К ним относятся: история создания, установка системы, детали интерфейса, графика, не имеющая отношения к визуализации вычислений, система Mathematica, как редактор текстов, справочная и оконная системы, ряд дополнительных функций, опций и директив, имеющих вспомогательное значение.

Это дало возможность в небольшом объеме книги достаточно подробно изложить основное содержание системы Mathematica — компьютерные технологии решения математических задач. При необходимости пользователь может самостоятельно углубить свои знания и умения с помощью справочной системы.

- Изложение компьютерных технологий решения математических задач сопровождается примерами на каждый из методов, что способствует более глубокому их пониманию. С целью самоконтроля знаний книга содержит оригинальные индивидуальные задания на основные математические методы.
- Наличие задач повышенной сложности дает возможность преподавателю осуществлять контроль знаний, а студенту более глубоко изучить систему Mathematica, как систему компьютерной алгебры.

Это в полной мере относится к любому пользователю, изучающему систему Mathematica.

- Книга достаточно научна, отличается простотой и ясностью изложения.

Компьютерная математика в науке и образовании

Большие возможности систем компьютерной математики позволяют во многих случаях получать решения без знания математики. Достаточно изучить встроенные функции и правила диалога.

Тогда возникает естественный вопрос: нужно ли пользователю глубоко изучать системы компьютерной математики? Не достаточно ли только с их помощью получать решения?

Да, нужно. Знания систем компьютерной алгебры существенно облегчают изучение математики, развивают мышление, дают образование, делают человека умнее.

Использование систем компьютерной математики при изучении общинженерных или специальных дисциплин позволяет более глубоко изучать эти дисциплины, придавая им исследовательский характер.

Даже самые мощные компьютерные системы не интеллектуальны. Они не могут ответить на простые вопросы, на которые отвечает даже школьник, например, чему равно $\frac{3}{7} + \frac{7}{11}$, $\sin^2 x + \cos^2 x$, чему равен корень уравнения $ax + b = 1$?

Обучение студентов, основанное на применении компьютерных технологий численных расчетов, больших знаний не дает.

Неумение выпускников вузов применять математику в своей деятельности, особенно инженерной, — большая беда нашего образования.

Студент запоминает математические выражения, формулы, методы лишь на период экзаменационной сессии. А если даже и потом их помнит, то не может ими воспользоваться на практике, т. к. компьютерные технологии требуют программирования на языках высокого уровня, с которыми он только знаком.

Другое дело системы компьютерной алгебры.

Они обладают следующими особенностями:

- ☐ не требуют программирования при решении задач, как в численном, так и в символьном видах;
- ☐ углубляют знания математики, особенно при решении задач в символьном виде;
- ☐ позволяют осваивать компьютер;
- ☐ повышают интерес к образованию;
- ☐ освобождают пользователя от рутинных вычислений и одновременно углубляют знания математики;
- ☐ возможность проведения высококачественных занятий по любому предмету, где необходима математика;
- ☐ системы компьютерной алгебры являются математическими справочниками высокого уровня.

Для кого эта книга?

Не следует думать, что эта книга, судя по ее названию, только для студента.

Да, основным ее читателем, наверное, будет студент. Это объясняется тем, что книга так методически представлена: подробное описание компьютерных технологий решения задач, простота и ясность изложения, большое число примеров, индивидуальные задания, примеры повышенной сложности.

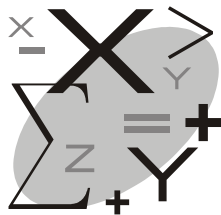
Однако эти особенности вовсе не означают, что в книге поверхностно и неполно излагаются компьютерные технологии решения математических и прикладных задач.

"Mathematica для студента" содержит достаточно полное и глубокое изложение компьютерных технологий решения математических задач с помощью системы Mathematica.

Она необходима инженеру любой специальности, экономисту, бизнесмену и любому специалисту, который по своей деятельности обязан решать прикладные математические задачи.

Успеха вам, читатели, в освоении и применении системы Mathematica в своей профессиональной деятельности.

ГЛАВА 1



Интерфейс системы Mathematica

1.1. Интерфейс системы и его изучение

Интерфейс системы Mathematica реализует отображение окон, палитр, панелей инструментов, знаков, расположение их в различном виде и в разных местах экрана монитора.

Главное окно системы имеет вид, показанный на рис. 1.1.

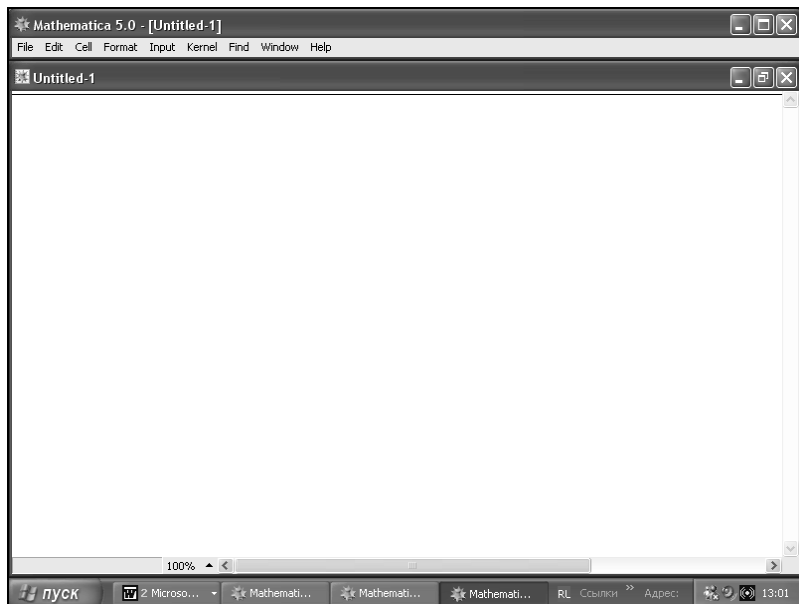


Рис. 1.1. Главное окно системы

Главное окно системы содержит: строку заголовка, главное меню и большой экран редактирования. Справа и внизу окна расположены полосы прокрутки с ползунками, управляемыми мышью. Положение ползунка показывает его место в тексте (номер страницы от начала текста). Внизу возле полосы прокрутки расположена строка состояний, в которой содержится информация о текущем режиме работы.

Интерфейс системы содержит несколько сот наименований пунктов меню, подменю, команд, функций. Изучить их при первом чтении невозможно: из краткого описания нельзя понять содержание. Содержание пунктов меню, подменю, команд можно понять только в процессе решения задач. Глубокие знания интерфейса системы приобретаются только с опытом работы с системой. Поэтому изучение данной главы при первом чтении не требуется. Прочтите ее и приступайте к решению задач. В процессе решения обращайтесь к командам главного меню системы.

Не игнорируйте справочную систему. Лучшим способом добиться успеха в процессе решения задачи является обращение к справочной системе. Это во многих случаях более эффективно, чем обращение к книге.

Однако справочная система не может заменить книгу — никакая справочная система не может быть учебником.

1.2. Главное меню системы

Главное меню системы состоит из следующих пунктов:

- ☐ **File** — действия с файлами: создание файла, открытие из каталога и его закрытие, сохранение файла с прежним или новым именем, печать текста и завершение работы;
- ☐ **Edit** — операции редактирования: редактирование текста с сохранением в специальных форматах, перенос выделенных участков текста, копирование из буфера, удаление;
- ☐ **Cell** — работа с ячейками: удаление и восстановление, объединение и разъединение, установление статуса;
- ☐ **Format** — установка стилей, изменение формата текста на экране и при печати, вывод опций, обеспечивающих желаемый вид текста, управление окном редактирования;
- ☐ **Input** — управление вводом: вставка в текст содержимого ячеек ввода и вывода, создание графиков, таблиц, палитр, гиперссылок, загрузка файла в нужное место текста, создание кнопок различного назначения, изменение цвета рисунков и заливки, запись звуковых сигналов, определение координат точек графика;
- ☐ **Kernel** — управление ядром системы: выбор ядра системы, управление процессом вычислений, удаление текста;

- ☐ **Find** — поиск и замена фрагментов текста, использование гиперссылок, обеспечение работы с этикетками;
- ☐ **Window** — операции с окнами: расположение окон и управление ими;
- ☐ **Help** — управление справочной системой.

Рассмотрим назначение пунктов главного меню более подробно.

1.2.1. Меню *File*

Меню **File** предназначено для работы с файлами. Оно содержит много команд. Рассмотрим назначение и содержание основных из них.

- ☐ Команда **New (Ctrl+N)**.

Команда служит для работы с новым текстом (документом). Она удаляет предыдущий текст, сделав запрос о его сохранении. Окно нового экрана имеет имя **Untitled-N**, где *N* — номер нового окна. Исполнение этой команды не отменяет ранее загруженные файлы пакетов расширения и определенных предыдущих вычислений.

- ☐ Команда **Open (Ctrl+O)**.

Команда осуществляет загрузку необходимых пользователю файлов. Она выводит диалоговое окно **Открыть** (рис. 1.2), в котором осуществляется поиск необходимого файла и его открытие в новом окне системы.

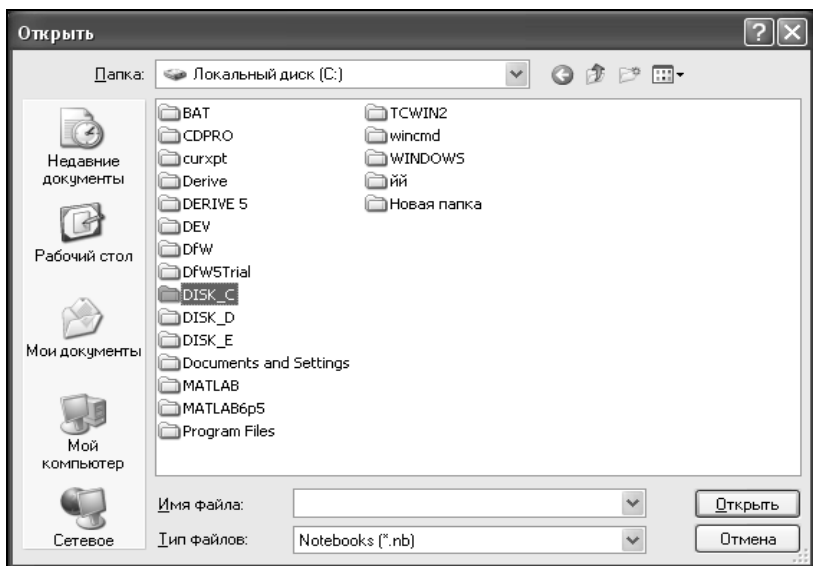


Рис. 1.2. Диалоговое окно загрузки файлов

При необходимости ввода содержимого файла в текущий текст следует использовать команду **Import**.

☐ Команда **Close (Ctrl+F4)**.

Осуществляет закрытие текущего окна.

☐ Команда **Save (Ctrl+S)**.

Команда предназначена для сохранения отредактированного и вновь созданного текста. Запись на магнитный диск осуществляется без изменения имени файла.

☐ Команда **Save As (Shift+Ctrl+S)**.

Эта команда предназначена для сохранения файла в любом месте каталога диска с изменением имени файла.

При обращении к команде открывается диалоговое окно, представленное на рис. 1.3. Выбирается место сохранения документа, записывается в соответствующем поле имя нового файла или подтверждается старое. Расширение ставить не обязательно — система выбирает его сама.

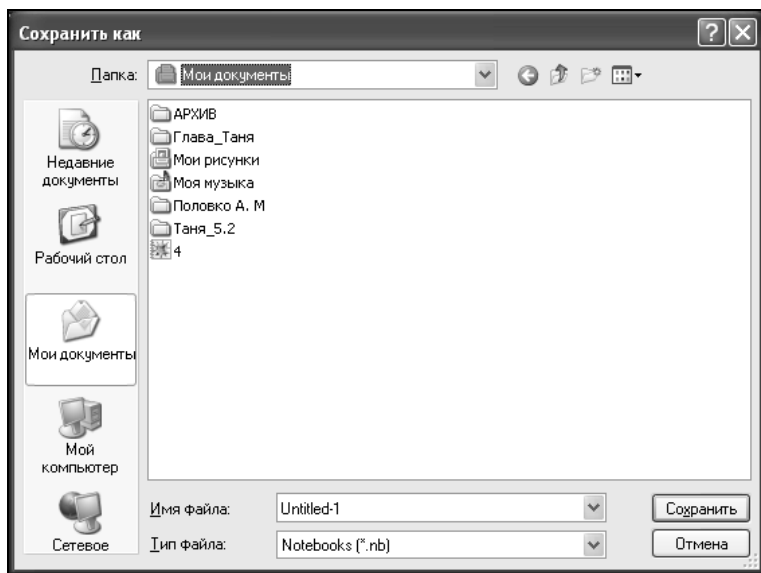


Рис. 1.3. Диалоговое окно сохранения файла

☐ Команда **Save As Special**.

Команда дает возможность записывать и считывать содержимое файлов в специальных форматах.

Такими форматами являются:

- **Version 2 Notebook** — формат старой версии системы (ma);
- **Text** — текстовый формат (txt);
- **Notebook Expression** — формат выражений в блокнотах;
- **Package Format** — формат пакетов расширения (m);
- **Tex** — формат редакторов Tex (tex).

☐ Команда **Open Special**.

Служит для загрузки файлов в специальных форматах. Команда открывает окно, с помощью которого загружаются файлы.

☐ Команда **Import**.

Открывает окно файлов для вставки содержимого нужного файла в текст документа.

☐ Команды **Send To, Send Selection**.

Зарезервированные команды.

☐ Команда **Palettes**.

Выводит палитры математических знаков, операторов и функций.

☐ Команда **Generate Palette from Selection**.

Команда служит для помещения любой части текста в уменьшенное окно (палитру). Палитру можно сохранить на диске.

☐ Команда **Generate Notebook from Palette**.

Команда преобразует палитру в ее содержимое.

☐ Команда **Printing Settings**.

Команда предназначена для установки параметров печати. Она выводит подменю с тремя операциями: **Page Setup, Printing Options, Headers and Footers**. Операция **Page Setup** служит для установки параметров страницы. Она выводит окно, содержащее типичные установки параметров страницы: размер бумаги, способ подачи бумаги, размеры полей, ориентацию.

Операция **Printing Options** предназначена для установки опций печати. Она выводит окно, в котором и устанавливаются опции.

Операция **Headers and Footers** предназначена для установки колонтитулов с помощью выводимого окна.

☐ Команда **Print**.

Команда предназначена для печати всего текста. Она выводит окно печати, в котором устанавливаются номера страниц, число копий, задания разборки копий. В окне также устанавливается тип принтера. Окно показано на рис. 1.4.

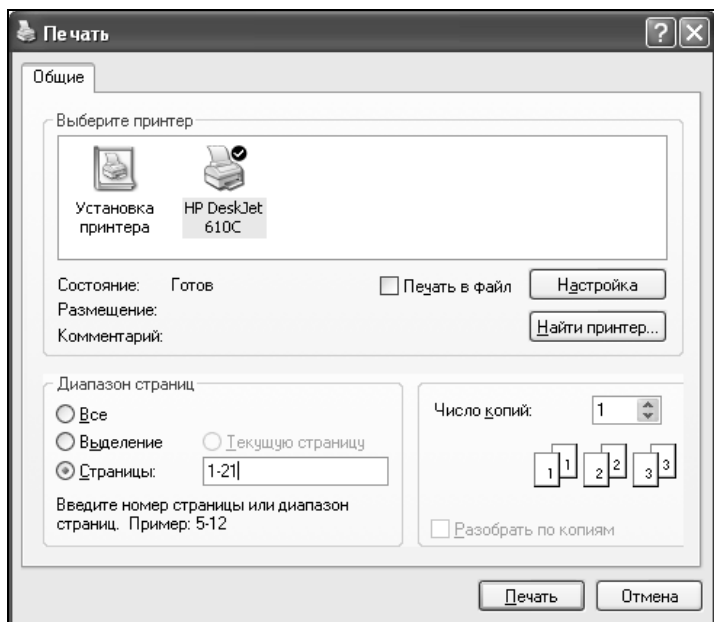


Рис. 1.4. Окно печати текста

В процессе печати текста может появиться окно принтера, в котором отображается ход печати и аварийные ситуации (принтер не включен, отсутствует бумага, кончились чернила).

☐ Команда **Print Selection**.

Команда предназначена для печати набора выделенных ячеек с помощью окна печати.

☐ Команда **Exit**.

Используется для окончания работы с системой. При этом запрашивается необходимость сохранения текста на диске.

1.2.2. Меню **Edit**

Меню **Edit** содержит команды редактирования текста. Меню содержит 16 команд, большинство из которых дублируются клавишами. Команды также осуществляют операции с буфером обмена (помещают, сохраняют, редактируют, считывают информацию). Рассмотрим кратко назначение команд меню **Edit**.

☐ Команда **Undo** — отмена операции.

☐ Команда **Cut (Ctrl+X)** — отмена содержимого выделенной ячейки и помещение его в буфер обмена.

- ☐ Команда **Copy (Ctrl+C)** — копирует содержимое ячейки в буфер без его удаления из ячейки.
- ☐ Команда **Paste (Ctrl+V)** — копирует содержимое буфера обмена в то место, которое указано курсором. Содержимое буфера при этом не меняется.
- ☐ Команда **Clear (Del)** — уничтожает содержимое выделенной ячейки без его сохранения в буфере.
- ☐ Команда **Copy As** — копирует содержимое ячейки в заданном формате.
- ☐ Команда **Paste As** — переносит содержимое буфера в то место текста, которое указано курсором. Содержимое буфера очищается. Эти процедуры выполняются с помощью команды **Paste and Discard**.
- ☐ Команда **Save Selection As** — служит для сохранения в специальных форматах текстов выделенных ячеек. Перечень форматов указан в подменю.
- ☐ Команда **Select All (Ctrl+A)** — выделение всех ячеек.
- ☐ Команда **Insert Object** — открывает стандартное окно с перечнем приложений. Приложениями являются тексты, рисунки и другие объекты. Они внедряются в систему Mathematica и могут редактироваться.
- ☐ Подменю **Motion** — редактирование больших англоязычных текстов стандартным способом.
- ☐ Команда **Expression Input** — ввод выражений в различных форматах и задание вида ячеек с помощью команд подменю.
- ☐ Команда **Make 2D (Shift+Ctrl+Y)** — представление и редактирование содержимого ячеек ввода в двумерном формате.
- ☐ Команда **Check Balance (Shift+Ctrl+B)** — определение места расположения текстового курсора (поиск скобок).
- ☐ Команда **Check Spelling (Alt+;)** — проверка орфографии.
- ☐ Команда **Preferences** — вызов окна настроек системы.

1.2.3. Меню *Cell*

Меню **Cell** служит для работы с ячейками. Оно содержит несколько подменю и большое число команд. Приведем эти команды и опишем их назначение.

- ☐ Команда **Convert To**.

Команда предназначена для преобразования формата содержимого ячеек. Она открывает подменю с перечнем всех применяемых форматов. Текущий формат помечен. Для задания нового формата необходимо его отметить галочкой, щелкнув мышью по определенному пункту подменю.

Форматами ячеек могут быть следующие:

- **InputForm (Shift+Ctrl+I)** — формат ввода;
- **OutputForm** — формат вывода;

- **StandardForm (Shift+Ctrl+N)** — стандартный формат;
- **TradicionalForm (Shift+Ctrl+T)** — традиционный формат;
- **PostScript** — векторный графический формат;
- **Bitmap** — растровый формат изображений;
- **Metafile** — векторный графический формат Windows Metafile.

При работе с текстами с большим числом математических знаков целесообразно использовать стандартный формат.

- ☐ Команда **Display As** — установка формата отображения ячеек.
- ☐ Команда **Default Input Format Type** — установка формата для ячеек ввода.
- ☐ Команда **Default Output Format Type** — установка формата для ячеек вывода.
- ☐ Подменю **Cell Properties**.

Подменю **Cell Properties** устанавливает свойства ячеек. Оно содержит следующие команды:

- **Cell Open** — устанавливает ячейку открытой или закрытой;
- **Cell Editable** — устанавливает ячейку редактируемой или не редактируемой;
- **Cell Evaluatable** — устанавливает ячейку оцениваемой или не оцениваемой;
- **Cell Edit Duplicate** — делает ячейку вновь созданной в случае редактирования;
- **Cell Active** — делает ячейку активной или неактивной;
- **Initialization Cell** — делает ячейку инициализационной или неинициализационной.

Установка перечисленных свойств выполняется через подменю **Cell Properties**. При этом можно установить одновременно несколько непротиворечивых свойств. Для снятия свойства необходимо повторить команду.

Ячейка ввода и соответствующая ей ячейка вывода обрамляются справа скобками: одиночными и общей. Активизируя скобку двойным щелчком, можно скрывать и снова выводить на экран выходную ячейку. Это полезно в том случае, если результат в ней слишком громоздкий.

Редактировать можно содержимое как входной, так и выходной ячеек. Для этого выходную ячейку необходимо сделать редактируемой, установив свойство **Cell Editable**. Редактируемая ячейка имеет символ "?" у своей обрамляющей скобки.

С помощью команды **Cell Evaluatable** исполняются и выдают результаты только оцениваемые ячейки. Неоцениваемые ячейки помечаются знаком "-"

и обрамляющей их скобкой. Оценивание ячеек можно выполнить в любом их сочетании.

Изменение активности ячеек достигается командой **Cell Active**. Активная ячейка помечается символом "A" и управляется кнопкой.

Инициализированная ячейка (устанавливается командой **Initialization Cell**) помечается в скобке знаком "/" и автоматически исполняется при загрузке текста.

Для объединения ряда ячеек в одну группу все ячейки выделяются, и исполняется команда **Group Cells**. При этом они обрамляются одной общей скобкой. Теперь можно объединенные в блок ячейки открывать и закрывать.

Команда **Ungroup Cells** разъединяет объединенные ячейки.

☐ Подменю **Cell Grouping**.

Назначение подменю — группировка ячеек. По умолчанию выбран режим **Automatic Grouping**, по которому ячейки объединяются в соответствии с их стилями. Чтобы воспользоваться командами объединения и разъединения ячеек, необходимо выбрать команду **Manual Grouping**.

- Команда **Open All Subgroups** открывает все выделенные группы и подгруппы ячеек.
- Команда **Close All Subgroups** закрывает все выделенные группы и подгруппы ячеек.
- Команда **Open/Close Group** сокращает число ячеек в группе так, что видимой остается только первая ячейка.

Остальные команды меню **Cell** имеют следующие назначения:

- **Divide Cell (Shift+Ctrl+D)** — разделение сгруппированных ячеек;
- **Merge Cells (Shift+Ctrl+M)** — объединение выделенных ячеек;
- **Animate Selected Graphics (Ctrl+Y)** — анимация с графиком выделенной ячейки;
- **Play Sound** — воспроизведение синтезированного звука;
- **Rerender Graphics** — построение заново графиков;
- **Rerender and Safe Graphics** — построение заново графиков с последующей записью;
- **Make Standard Size** — установка стандартного размера ячейки;
- **Align Selected Graphics** — выравнивание графиков;
- **Cell Size Statistics** — вывод статистики о размерах ячеек.

1.2.4. Меню **Format**

Меню **Format** содержит следующие команды.

☐ Команда **Style**.

Служит для установки параметров текста (шрифт, размер символов, виды выделений и т. д.). Команда открывает подменю стандартных стилей, которые выбираются пользователем и реализуются системой.

☐ Команда **Screen Style Environment**.

Эта команда предназначена для изменения формата текста на экране и имеет следующие установки:

- **Working** — стиль типичный;
- **Presentation** — презентационный стиль с увеличением размера символов;
- **Condensed** — уменьшенный размер символов (сжатый);
- **Printout** — стиль оптимальный для печати (принтерный).

☐ Команда **Print Style Environment**.

Стиль предназначен для изменения формата текста при печати. Имеет те же установки, что и предыдущая команда.

☐ Команда **Show Expression**.

Команда служит для показа выражений в стандартном и развернутом видах. При этом развернутый формат представляет собой внутренний формат типа программы на языке системы Mathematica.

☐ Команда **Option Inspector**.

Команда выводит окно опций, дающее пользователю перечень всех опций программы и возможность их использования для обеспечения желаемого вида информации.

☐ Команда **Remove Options**.

Команда убирает все опции, введенные пользователем, и восстанавливает начальное состояние системы.

☐ Команды третьей, четвертой и пятой групп меню **Format**.

Эти команды предназначены для управления стилем документа. Они позволяют осуществить выбор стилей, устанавливать тип шрифта, начертание символов и их размер, устанавливать цвет текста и фона, осуществлять выравнивание текста по ширине и разбивку на строки.

☐ Команды шестой группы команд меню **Format**.

Эти команды являются командами управления окном текста. К ним часто обращается пользователь.

Эти команды таковы:

- **Show Ruler** — отображение мерной линейки;
- **Show Toolbar** — вывод на экран панели инструментов;
- **Show Page Breaks** — показ линий разрыва страниц;
- **Magnification** — установка масштаба отображения текста экрана.

Для пользователя особо полезной является последняя команда. К ней часто приходится обращаться при работе с текстом, например, при сохранении текста, печати текста и т. д.

Нередко приходится обращаться также к команде вывода на экран панели инструментов.

1.2.5. Меню *Input*

Перечислим команды, входящие в меню **Input**.

☐ Команда **Get Graphics Coordinate**.

Команда позволяет определять координаты любой точки двумерного графика. При исполнении данной команды появляется окно, объясняющее процедуры получения координат. Для этого необходимо: выделить двумерный график, нажать клавишу <Ctrl> и, удерживая ее, поместить в нужное место графика курсор. В строке состояний появятся координаты графика.

Таким способом можно получить семейство координат точек и с помощью команды **Copy** поместить их в буфер обмена. Теперь при необходимости точки можно вставить с помощью команды **Paste** в любое место текста документа.

☐ Команда **3D View Point**.

Эта команда предназначена для вывода селектора точки обзора трехмерного графика.

☐ Команда **Color Selector**.

Команда выводит стандартное окно изменения цвета. С помощью этого окна создаются цвета рисунков и заливки.

☐ Команда **Record Sound**.

Команда выводит стандартное окно звукозаписи операционной системы Window. С помощью этой команды записываются звуковые сигналы речи и музыки в виде файлов.

☐ Команда **Get File Path**.

Команда служит для вставки в документ имени файла с указанием пути к нему. При обращении к команде открывается окно загрузки файлов. После выбора нужного файла и щелчка мыши по кнопке **Open** файл будет вставлен в нужное место текста, отмеченное курсором.

❑ Команда **Create Table/Matrix/Palette**.

Команда служит для задания таблиц, матриц и палитр через главное меню.

При обращении к команде появляется окно задания таблиц, матриц и палитр (рис. 1.5). Командой целесообразно пользоваться в случае больших размеров таблиц и матриц. Команда фактически дублирует задание таблиц и матриц методами, описанными в *главе 5*.

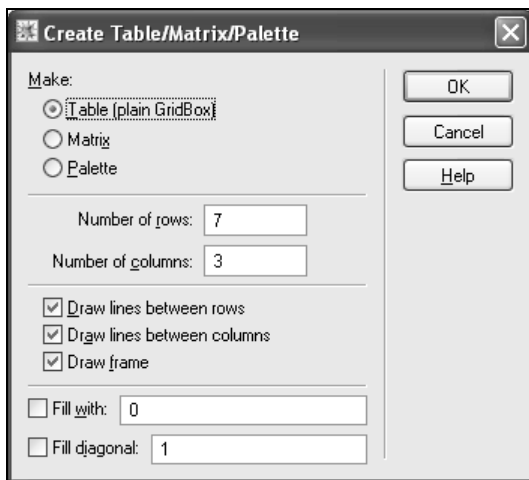


Рис. 1.5. Окно задания таблиц, матриц и палитр

❑ Команда **Create Button**.

Команда служит для создания кнопок различного назначения. При обращении к команде появляется подменю, содержащее такие кнопки.

❑ Команда **Edit Button**.

Команда служит для редактирования кнопок подменю, созданных командой **Create Button**. Команда выводит окно редактирования кнопок. В нем содержится перечень кнопок и новое окно с программой, создающей необходимую кнопку.

Процедуры создания и редактирования кнопок требуют знания программирования на языке системы Mathematica.

❑ Команда **Create Hyperlink**.

Команда предназначена для создания гиперссылок, которые реализуются следующим образом.

В строке ввода пишется короткая фраза. Слово или вся фраза выделяются, и исполняется команда **Create Hyperlink**. При этом открывается окно **От-**

крытие файла. Имя файла устанавливается в соответствующем поле окна или выбирается из окна поиска файлов, выводимого кнопкой обзора файлов. После нажатия кнопки **ОК** будет создана гиперссылка.

☐ Команда **Create Automatic Numbering Object**.

Команда открывает окно создания объекта. Окно позволяет вывести перечень нумерованных объектов, которыми и можно воспользоваться.

☐ Команда **Create Value Display Object**.

Команда выводит окно создания свойств объектов. Оно дает возможность осуществить вставку объектов с использованием опций.

☐ Команды вставки содержимого ячеек.

В четвертом разделе меню **Input** имеются следующие три команды:

- **Copy Input from Above;**
- **Copy Output from Above;**
- **Start New Cell Below.**

Эти команды предназначены для вставки в текст содержимого предшествующих ячеек ввода и вывода или пустых ячеек.

☐ Команда **Complete Selection**.

Команда предназначена для вывода списка имен всех функций ядра системы. Для этого необходимо ввести ключевое слово или его часть и исполнить данную команду.

☐ Команда **Make Template**.

Команда выдает список параметров функции, в имени которой находится текстовый курсор. Например, если введено слово **Plot** и курсор находится в его области, то по команде **Make Template** получим ответ: `Plot[f, {x, xmin, xmax}]`. Из примера видно, что это превосходная справочная система по синтаксису функций ядра системы.

1.2.6. Меню **Kernel**

Меню **Kernel** предназначено для управления ядром системы Mathematica и, как следствие, процессом решения задач. Оно управляет действиями ячеек, вернее, их содержимым. Рассмотрим команды этого меню.

☐ Подменю **Evaluation**.

Подменю **Evaluation** управляет процессом вычислений. Оно содержит следующие команды:

- **Evaluate Cells (Shift+Enter)** — вычисление выделенных ячеек;
- **Evaluate in Place (Shift+Ctrl+Enter)** — вычисление выделенных выражений в строке ввода;

- **Evaluate Next Input (Shift+Enter+ -)** — вычисление следующей строки ввода, расположенной под выделенной ячейкой;
- **Evaluate Subsection** — вычисление всех выделенных ячеек;
- **Evaluate Notebook** — вычисление всех выделенных ячеек сверху вниз;
- **Evaluate Initialization** — вычисление инициализированных ячеек без их выделения;
- **Enter Subsection** — запуск диалога работы с ядром;
- **Exit Subsection** — завершение работы с ядром.

Из краткого описания команд подменю **Evaluation** видно, что они существенно расширяют процедуры вычислений, выполняемых путем нажатия комбинации клавиш <Shift>+<Enter>.

☐ Команда **Interrupt (Alt+.)**.

Команда предназначена для прерывания текущих вычислений. При ее исполнении задается вопрос: как прервать вычисления и сколько шагов вычисления еще нужно сделать. Команду можно отменить.

☐ Команда **Abort (Alt+.)**.

Команда прекращения вычислений. Ее целесообразно использовать при зацикливании. Она выводит процесс вычислений из этого режима с сохранением данных и текущей программы.

☐ Команды выбора ядра системы.

Команды выбора ядра системы находятся во втором пункте меню **Kernel**. Приведем их названия и содержание:

- **Start Kernel** — запуск выбранного ядра;
- **Quit Kernel** — завершение работы ядра;
- **Default Kernel** — выбор ядра, используемого по умолчанию;
- **Notebook Kernel Notebook** — выбор ядра для данного документа;
- **Kernel Configuration Options** — вывод окна установки свойств ядер.

Эти команды позволяют работать не только с установленным ядром системы, но также с другими ядрами, в том числе и подключаться через сеть.

Команды очевидны по их названию и вряд ли требуют подробных объяснений.

☐ Команда **Show/In/Out Names**.

Эта команда предназначена для показа номеров строк ввода и вывода текста документа. Если возле команды стоит галочка, то номера строк будут показаны, в противном случае — нет.

Номера строк бывают полезными при вычислениях: система допускает математические операции с выражениями через номера строк.

☐ Команда **All Output**.

Команда предназначена для удаления текста всех ячеек данного документа.

1.2.7. Меню *Find*

В меню **Find** находятся команды поиска и замены текстов. Они относятся к операциям редактирования.

Меню **Find** содержит три группы команд. Рассмотрим их назначение и содержание.

☐ Первая группа команд.

В первую группу входят следующие команды поиска и замены фрагментов текста:

- **Find (Ctrl+F)** — поиск строк без их замены;
- **Enter Selection (Ctrl+E)** — ввод строки в окно поиска;
- **Find Next (F3)** — поиск по тексту вперед;
- **Find Previous (Shift+F3)** — поиск по тексту назад;
- **Find in Cell Tags** — поиск ячейки с заданной этикеткой (Tag);
- **Replace (Ctrl+R)** — замена строк;
- **Replace and Find Again (Shift+Ctrl+R)** — замена строк с продолжением поиска;
- **Replace All** — замена по всему тексту.

Все перечисленные операции выполняются с помощью специального окна. Технология поиска и замены достаточно проста и очевидна.

☐ Вторая группа команд.

Во вторую группу меню **Find** входят следующие команды:

- **Open Selected** — открытие объектов, содержащих выделенные строки;
- **Scroll to Selection** — прокрутка текста до выделенной строки;
- **GoBack** — возврат назад после использования гиперссылки.

Действия этих команд очевидны.

☐ Третья группа команд.

В третью группу меню **Find** входят следующие команды:

- **Add/Remove Cell Tags (Ctrl+J)**;
- **Cell Tags**;
- **Show Cell Tags**;

- **Cell Tags from Names**;
- **Mace Index**.

Эти команды обеспечивают работу с этикетками.

С помощью команды **Cell Tags** вызывается список этикеток. При выборе имени этикетки выделяются все ячейки, которые помечены этой этикеткой. При необходимости вставить этикетку в строку ввода следует воспользоваться командой **Add/Remove Cell Tags**.

С помощью этой команды можно также удалить этикетку, если она имеется в строке. Команда вызывает окно редактирования этикеток. Кнопка **Add** добавляет этикетку, а кнопка **Remove** — удаляет.

Команда **Cell Tags from Names** позволяет создавать этикетку по номеру ячейки.

Команда **Mace Index** сохраняет все этикетки текста в буфере.

1.2.8. Меню *Window*

Система *Mathematica* позволяет работать одновременно с несколькими окнами. Это бывает удобно, если обрабатываются или создаются одновременно несколько текстов.

Меню **Window** имеет следующие команды для работы с многооконной системой:

- ☐ **Stacks Windows** — каскадное расположение окон;
- ☐ **The Windows Wide** — мозаичное расположение окон по высоте;
- ☐ **The Windows Tall** — мозаичное расположение окон по ширине;
- ☐ **Messages** — вывод окна сообщений об ошибках.

С помощью этих команд можно управлять окнами: местом расположения, размером, сохранять их содержимое.

1.2.9. Меню *Help*

Обращение со справочной системой достаточно просто и очевидно. Управление системой осуществляется командами меню **Help**. Браузер справки вызывается командой **Help Browser (Shift+F1)**.

Справочная система позволяет:

- ☐ получить сведения обо всех командах главного меню;
- ☐ изучить правила записи и набора математических выражений;
- ☐ уточнить назначение любой функции или оператора;

- ☐ ознакомиться с примерами и приспособить их к интересам пользователя (примеры имеются по каждой функции);
- ☐ получить доступ к пакетам расширения;
- ☐ получить справку о системе Mathematica и фирме ее разработавшей;
- ☐ воспользоваться электронной книгой разработчика системы Стивена Вольфрама (раздел справки **The Mathematica Book**).

Справку можно получить достаточно просто по имени функции или алфавитному указателю.

Работа с электронной книгой

Электронная книга — превосходное учебное пособие по изучению системы Mathematica, особенно по ее применению в математических расчетах. Она является руководством для начинающих.

Книга содержит много формул, графиков, использует анимацию. Позволяет работать с примерами: редактировать и сразу получать новые результаты, переносить примеры в буфер обмена, внедрять результаты из буфера обмена в текст пользователя.

Особенностями книги являются: полнота, наглядность, простота поиска.

Однако не следует думать, что электронная книга является единственным квалифицированным учебным пособием по изучению системы Mathematica.

Любая электронная книга имеет ряд недостатков. Основными из них являются:

- ☐ для изучения системы требуется компьютер;
- ☐ вредна для здоровья при ее длительном использовании;
- ☐ содержит большое число страниц, поэтому необходимую справку трудно отыскать;
- ☐ методически, как правило, отработана хуже, чем обычный учебник.

По этим и ряду других причин электронная книга не может заменить обычный учебник и быть единственной обучающей системой.

Основное назначение электронной книги — получение справки в процессе решения задачи. Здесь она имеет приоритет по сравнению с обычной книгой, когда в процессе решения задачи крайне неудобно обращаться к книге.

В целом же обычная книга и справочная система (в том числе и электронная книга) в совокупности являются идеальной системой обучения.

Из изложенного видно, что изучение системы Mathematica не следует начинать с подробного изучения справочной системы и, в частности, с электронной книги.

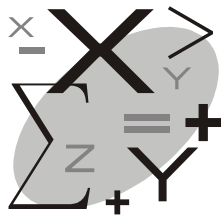
Не следует также подробно изучать интерфейс системы, в частности, огромный объем меню и его команд. Нужно лишь бегло прочитать нашу первую главу и

приступать к решению задач. В процессе решения будут, по мере необходимости, изучаться элементы диалога, уточняться функции и команды, изучаться компьютерные технологии решения задач.

Только так можно изучить систему Mathematica в полном ее объеме. При этом следует иметь в виду, что для этого потребуется много времени.

На практике инженеру, да и не только ему, приходится решать лишь определенный класс задач. В таких случаях освоение компьютерных технологий займет немного времени и окупится стократ полученными в короткое время результатами.

ГЛАВА 2



Основы работы с системой Mathematica в режиме вычислений

Для получения решения с высокой достоверностью необходимо знать:

- ☐ правила представления и ввода данных;
- ☐ функции математических операций;
- ☐ встроенные функции вычислений;
- ☐ компьютерные технологии решения математических задач.

При решении практических задач необходимо соблюдать следующие основные правила:

- ☐ не нужно решать задачу по принципу угадывания, нажимая без разбора клавиши или щелкая по кнопкам панели инструментов. Решая математическую задачу, необходимо первоначально думать, затем принимать решение и только потом нажимать клавиши;
- ☐ необходимо проверять достоверность полученных откликов в процессе решения задачи;
- ☐ стремиться понять методы и алгоритмы решения задачи;
- ☐ всегда анализировать полученный результат;
- ☐ если эти правила не соблюдать, то решение может и не последовать, более того, оно может существовать, но быть абсолютно неверным.

Следует также помнить, что, решая задачу методом угадывания, мы теряем время на получение ненужных вариантов.

2.1. Арифметические операторы, функции, константы

Система Mathematica имеет полный набор арифметических операторов, функций и констант. Рассмотрим их и приведем примеры.

2.1.1. Арифметические операторы

Арифметические операторы системы Mathematica приведены в табл. 2.1.

Таблица 2.1. Арифметические операторы системы Mathematica

Символ/ операция	Содержание	Примеры
+	Сложение	$3+5$, $a+b$, $\sin x + \cos x$
-	Вычитание	$5-3$, $a-b$, $\sin x - \cos x$
*	Умножение	$3*5$, $a*b$, $\sin x * \cos x$
/	Деление	$3/5$, a/b , $\sin x / \cos x$
^	Возведение в степень	5^2 , a^3 , a^k , $(\sin x)^2$
Sqrt[x]	Извлечение квадратного корня	$\text{Sqrt}[2]$, $\text{Sqrt}[a+b]$, $\text{Sqrt}[e^x]$

Арифметические операции в системе Mathematica осуществляются над вещественными и комплексными числами, над списками типа вектор или матрица, над выражениями и функциями.

2.1.2. Арифметические функции

К арифметическим обычно относят следующие функции:

- ☐ выполнение арифметических операций над числами;
- ☐ определение делителей целых чисел;
- ☐ определение наименьшего общего кратного;
- ☐ вычисление факториалов;
- ☐ приведение вещественных чисел к ближайшим целым;
- ☐ получение простых чисел.

Рассмотрим основные из них.

Функции выполнения арифметических операций

Система Mathematica имеет следующие функции выполнения арифметических операций:

- ☐ Plus[x, y, ...] — выполняет суммирование чисел x , y , ...;
- ☐ Times[x, y, ...] — вычисляет произведение чисел x , y , ...;

- `Divide[x, y]` — осуществляет деление x на y ;
- `Mod[x, y]` — возвращает остаток от деления x на y .

Примеры выполнения арифметических операций приведены на рис. 2.1.

```

Plus[1, 2, 3, 4, 5, 6, 7, a]
28 + a
Plus[a, b, E, Pi, 3, 5, Sin[x]]
8 + a + b + e +  $\pi$  + Sin[8]
Times[1, 2, 3, 4, 5, 6, 7, a]
5040 a
Times[z, Exp[2], 3]
3 e2 z
Divide[2., 3]
0.666667
Divide[a + b, a - b]
 $\frac{a + b}{a - b}$ 
Mod[327, 20]
7
Mod[327, -20]
-13
Mod[-327, 20]
13
Mod[-327, -20]
-7

```

Рис. 2.1. Примеры выполнения арифметических операций

Определение делителей целых чисел и наименьшего общего кратного

Функциями определения делителей целых чисел и наименьшего общего кратного являются:

- `Divisors[n]` — возвращает целочисленные делители числа n ;
- `DivisorSigma[k, n]` — возвращает сумму возведенных в степень k положительных делителей числа n ;
- `ExtendedGCD[n, m]` — возвращает наибольший общий делитель чисел n и m ;
- `GCD[n1, n2, ...]` — возвращает наибольший общий делитель целых чисел $n1, n2, \dots$;

- `LCM[n1, n2, ...]` — возвращает наименьшее общее кратное целых чисел `n1, n2, ...`

Примеры определения делителей целых чисел и общего кратного приведены на рис. 2.2.

```
Divisors[112102]  
{1, 2, 23, 46, 2437, 4874, 56051, 112102}  
DivisorSigma[6,3]  
730  
ExtendedGCD[125, 50]  
{25, {1, -2}}  
GCD[1200, 125, 75, 50]  
25  
LCM[24, 36, 12, 8]  
72
```

Рис. 2.2. Определение делителей целых чисел и общего кратного

Приведение вещественных чисел к ближайшим целым

Приведение вещественных чисел к ближайшим целым осуществляется с помощью следующих функций:

- `Round[x]` — округляет `x` до ближайшего целого;
- `Floor[x]` — возвращает наибольшее целое число, не превышающее `x`;
- `Ceiling[x]` — возвращает наименьшее целое число, большее или равное `x`;
- `Quotient[n, m]` — возвращает округленное целое число `n/m`, не превышающее значения `n/m`.

Значение `x` в этих функциях может быть не только числом, но и вектором чисел и даже вектором вычисляемых функций.

Примеры приведения вещественных чисел к ближайшим целым представлены на рис. 2.3.

Вычисление факториалов

Факториалы числа `n` в системе *Mathematica* вычисляются с помощью следующих функций: `Factorial` и `Factorial2`, которые имеют вид:

- `Factorial[n]` — возвращает значение факториала числа `n`;
- `Factorial2[n]` — возвращает значение двойного факториала числа `n`.

Приведенные функции вычисляют факториалы чисел: вещественных, целых и дробных, положительных и отрицательных. Они позволяют определять факториалы очень больших чисел.

```

f=15.4
Round[f]
15.4
15
Floor[f]
15
Ceiling[f]
16
Quotient[7,3]
2
f1={3.5, 5, 7.8, -6.4, -7.6}
Round[f1]
{3.5, 5, 7.8, -6.4, -7.6}
{4, 5, 8, -6, -8}
Floor[f1]
{3, 5, 7, -7, -8}
Ceiling[f1]
{4, 5, 8, -6, -7}
f2={Sin[1.], Cos[1.], Exp[-1.2], 2 Log[3.]}
Round[f2]
{0.841471, 0.540302, 0.301194, 2.19722}
{1, 1, 0, 2}
Floor[f2]
{0, 0, 0, 2}
Ceiling[f2]
{1, 1, 1, 3}
Quotient[Sin[1], Cos[1]]
1

```

Рис. 2.3. Приведение вещественных чисел к ближайшим целым

В случае целых значений n факториалы соответствуют следующим выражениям:

$$n! = 1 \times 2 \times 3 \times \dots \times n,$$

$$n!! = n \times (n-2) \times (n-4) \times \dots$$

При этом $0! = 1$, $0!! = 1$.

Примеры вычисления факториалов приведены на рис. 2.4.

Получение простых чисел

Получение простых чисел осуществляется с помощью следующих функций:

- ☐ **Prime[n]** — возвращает n -е простое число;
- ☐ **PrimePi[x]** — возвращает количество простых чисел, не превышающих x .

Примеры получения простых чисел приведены на рис. 2.5.

```

Factorial[6]
720
6!
720
Factorial[100]
93326215443944152681699238856266700490715968264381621
468592963895217599993229915608941463976156518286253697
920827223758251185210916864000000000000000000000000
Factorial[5.6]
344.702
-3.5!
-11.6317
Factorial2[{3,5,7, 9, 11}]
{3, 15, 105, 945, 10395}
8.3!!
511.045

```

Рис. 2.4. Примеры вычисления факториалов

```

Prime[11]
31
PrimePi[11]
5
Prime[123456789]
2543568463
PrimePi[123456789]
7027260

```

Рис. 2.5. Примеры получения простых чисел

2.1.3. Именованные константы

Именованная константа является объектом, имеющим уникальное имя, начинающееся с прописной буквы. Константа имеет заранее определенное значение, не изменяющееся по ходу вычислений.

В системе Mathematica имеются следующие именованные константы:

- ☐ **E** — основание натурального логарифма;
- ☐ **Pi** — отношение длины окружности к диаметру;
- ☐ **I** — мнимая единица ($\sqrt{-1}$);
- ☐ **Infinity** — положительная бесконечность; при отрицательной бесконечности ставится знак "-" (минус);

- ☐ Degree — число радиан в градусе, имеющее значение $\pi/180$;
- ☐ EulerGamma — постоянная Эйлера, имеющая значение 0,577216;
- ☐ GoldenRatio — константа, имеющая значение $\frac{1+\sqrt{5}}{2}$ (деление отрезка по правилу золотого сечения);
- ☐ Catalan — константа Каталана, имеющая значение 0,915966.

Для численного значения константы необходимо перед ее именем поставить символ N с целью получения результата в виде вещественного числа.

Над именованными константами можно выполнять любые математические операции. При этом только необходимо указать, что вычисления осуществляются над вещественными числами. Для этого существует много способов, например, ставится точка в конце целого числа, используется функция N[%], которая вводится в конце полученного в символьном виде ответа.

Это более подробно излагается далее в главе.

Примеры получения констант и математические действия над ними показаны на рис. 2.6.

```
{N[E],N[Pi],N[Degree]}
{2.71828,3.14159,0.0174533}
{N[EulerGamma],N[GoldenRatio],N[Catalan]}
{0.577216,1.61803,0.915966}
2*N[Pi]+N[E]^2-Sqrt[N[Catalan]]
12.7152
2.*Catalan
1.83193
3.*Pi*EulerGamma-GoldenRatio/Catalan
3.67365
Catalan/EulerGamma
Catalan
EulerGamma
N[%]
1.58687
Pi+E*Catalan-EulerGamma/GoldenRatio
Catalan e -  $\frac{\text{EulerGamma}}{\text{GoldenRatio}} + \pi$ 
N[%]
5.27471
```

Рис. 2.6. Примеры получения констант и математические операции с ними

2.1.4. Укороченная форма представления арифметических операций

В системе Mathematica имеется специфическая форма записи математических операций, называемая *укороченной*. Каждой такой записи (оператору) соответствует своя встроенная функция. Такими функциями являются:

- `AddTo[x, dx]` — прибавляет dx к x и возвращает новое значение x ; оператором функции является: $x+=dx$;
- `SubtractFrom[x, dx]` — вычитает dx из x и возвращает новое значение x , равное $x-dx$; оператором функции является выражение $x-=dx$;
- `TimesBy[x, k]` — умножает x на k и возвращает новое значение x , равное kx ; оператором функции является выражение: $x*=k$;

```
x=3
AddTo[x, 5]
3
8
TimesBy[x, 10]
80
Divide[x, 2]
40

f=Sin[z]
AddTo[f, Cos[z]]
Sin[z]
Cos[z]+Sin[z]
TimesBy[f, f]
(Cos[z] + Sin[z])2
TimesBy[f, (a+b)]
(a+b) (Cos[z] + Sin[z])2
Divide[f, (a-b)]
(a+b) (Cos[z] + Sin[z])2
a-b
```

Рис. 2.7. Укороченная форма записи арифметических операций с помощью функций

```
x = 3
x += 5
3
8
x *= 10
80
x /= 2
40
x -= 25
15
f = Sin[z]
f += Cos[z]
Sin[z]
Cos[z] + Sin[z]
f *= f
(Cos[z] + Sin[z])2
f *= (a+b)
(a+b) (Cos[z] + Sin[z])2
f /= (a-b)
(a+b) (Cos[z] + Sin[z])2
a-b
```

Рис. 2.8. Укороченная форма записи арифметических операций с помощью операторов

□ `Divide[x, k]` — делит x на k и возвращает новое значение x , равное x/k ; оператором функции является: $x/=k$.

Полезность этой формы записи математических операций не только в более короткой форме представления, но также в том, что одновременно с выполнением операций осуществляется присвоение x нового значения.

Укороченные формы записи арифметических операций можно использовать как при численных, так и при символьных вычислениях.

Примеры вычислений в случае укороченной формы записи операций для случая численных и символьных вычислений приведены на рис. 2.7 и 2.8.

На рис. 2.7 представлены вычисления с использованием функций, а на рис. 2.8 — с использованием соответствующих операторов.

2.2. Типы данных

Данные, с которыми работает система Mathematica, весьма обширны. Это числа, переменные, выражения, функции. Типы численных данных приведены в табл. 2.2.

Таблица 2.2. Типы численных данных

Обозначения	Типы чисел	Примеры
Integer	Целочисленные	364, -35
Rational	Рациональные	24/37, -15/87
Real	Вещественные	364.6, -25.16, 16.8 10^{-5}
Complex	Комплексные	2+3*I

Символьные переменные могут образовывать сколь угодно сложные выражения, которые система способна упрощать. Одни символьные переменные могут быть заменены другими, образуя новые выражения и даже формулы.

Система имеет большое количество функций, с которыми пользователь может работать как с обычными данными, выполняя любые математические операции.

В этом разделе приводятся все основные типы данных, а также различные технологии их математических преобразований и вычислений.

2.2.1. Арифметические операции с целыми и рациональными числами

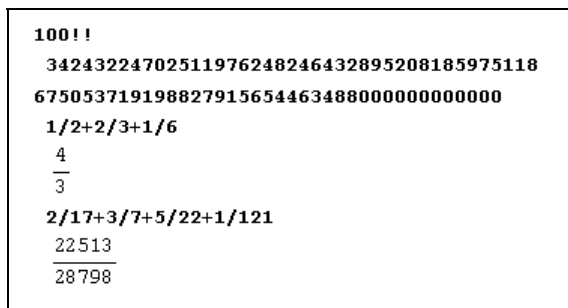
Система Mathematica выполняет вычисления с числами: целыми и рациональными без погрешностей. При этом не имеется ограничений на разрядность чисел. Приведем примеры подобных вычислений.

Пример 2.1

Необходимо выполнить следующие вычисления:

$$100!! , 1/2+2/3+1/6, 2/17+5/22+1/121.$$

Решение приведено на рис. 2.9.



```

100!!
34243224702511976248246432895208185975118
675053719198827915654463488000000000000
1/2+2/3+1/6
4
3
2/17+3/7+5/22+1/121
22513
28798

```

Рис. 2.9. Решение примера 2.1

Из рис. 2.9 видно, что система вычислила двойной факториал числа 100 с абсолютной точностью, причем за доли секунды. Суммирование рациональных чисел выполнено весьма эффектно. Результатом в обоих случаях также является число рациональное. При этом программа осуществила сокращение дроби в конечном результате.

Следует иметь в виду, что любые вычисления в случае целых и рациональных чисел имеют результат лишь при условии наличия точного решения. Если точное решение не существует, то программа приближенного решения не выдает. Она лишь повторяет исходное выражение. Покажем это на примере.

Пример 2.2

Необходимо выполнить вычисление следующих функций:

$$\frac{\pi}{2} \sin \frac{\pi}{2}, e^{-1}, \ln 5, \ln_2 8, \ln e^2, \sin^2 1 + \cos^2 1, e^{-x} \cdot e^x.$$

Решение приведено на рис. 2.10.

Из рис. 2.10 видно, что программа выдала точное значение функций $\sin \frac{\pi}{2}$, $\ln_2 8$, $\ln e^2$, $e^{-x} \cdot e^x$ и не вычислила выражений e^{-1} , $\ln 5$, т. к. точное их значение не существует. К сожалению, система "не знает", что $\sin^2 1 + \cos^2 1 = 1$. Это свидетельствует о ее недостаточной интеллектуальности. Правда, этот ответ можно получить, если обратиться к командам упрощения выражений.

```

Pi/2*Sin[Pi/2]

$$\frac{\pi}{2}$$

E^(-1)

$$\frac{1}{e}$$

Log[5]
Log[5]
Log[2,8]
3
Log[E^2]
2
Sin[1]^2+Cos[1]^2
Cos[1]^2+Sin[1]^2
E^(-x)*E^x
1

```

Рис. 2.10. Решение задач примера 2.2

2.2.2. Арифметические операции с вещественными числами

Вещественные числа в системе Mathematica представляются в *естественной* или *нормальной форме*.

При представлении числа в естественной форме целая часть отделяется от дробной точкой, например: 1.35, -16.264, 0.25. При этом 0 целых можно опустить и вместо 0.25 вводить .25. Такие числа в информатике часто называют числами с фиксированной точкой (запятой).

Точка в конце числа является признаком того, что число вещественное. Например, число 321 является целочисленным, а 321. — вещественным, число 2/7 является рациональным, а 2./7 — вещественным.

При представлении числа в нормальной форме число записывается в виде мантиссы с целой и дробной частями и порядка в виде степени числа 10, отделяемые знаком умножения, например: $2 \cdot 10^2$, $-5 \cdot 10^3$, $0.231 \cdot 10^{-3}$, $.32 \cdot 10^{-7}$. Вместо знака умножения можно использовать пробел. Такие числа называют числами с плавающей точкой (запятой), т. к. их представление не однозначно:

$$5.25 \cdot 10^2 = 52.5 \cdot 10^1 = 52500 \cdot 10^{-2} = .525 \cdot 10^3$$

Арифметические операции над вещественными числами дают приближенный результат. Система Mathematica оперирует приближенными числами весьма широкого диапазона. Максимальные и минимальные числа имеют значения: $1.79769 \cdot 10^{308}$, $2.22507 \cdot 10^{-308}$.

Эти числа можно вывести на экран с помощью системных переменных `$MaxMachineNumber` и `$MinMachineNumber`.

Вещественные числа в системе Mathematica могут представляться с помощью следующих встроенных функций:

- `N[k]`, где k — число любого типа, например, целочисленное или рациональное;
- `N[k, n]`, где k — число любого типа, n — число цифр результата.

На рис. 2.11 приведены результаты представления чисел 5 и $2/7$ в виде вещественных с помощью встроенных функций. Обратите внимание на то, что при представлении вещественного числа $2./7$ с помощью функции `N[k, n]` результатом является число лишь с шестью знаками, в то время как $n=27$.

```
N[5]
5.
N[5,20]
5.00000000000000000000
N[2/7]
0.285714
N[2/7,27]
0.285714285714285714285714286
N[2./7,27]
0.285714
```

Рис. 2.11. Образование вещественных чисел с помощью встроенных функций

2.2.3. Арифметические операции с комплексными числами

Комплексное число представляется в следующем виде:

$$z = \text{Re}[z] + I * \text{Im}[z] \quad \text{или} \quad z = \text{Re}[z] + \text{Im}[z] * I,$$

где:

- `Re[z]` — вещественная часть комплексного числа z ;
- `Im[z]` — мнимая часть комплексного числа z ;
- I — мнимая единица: $I = \sqrt{-1}$, представляется прописной буквой латинского алфавита.

В выражении комплексного числа знак умножения можно заменять пробелом.

Вещественная и мнимая части комплексного числа могут быть числами любого вида, а также символьными переменными, выражениями и функциями, например: `2+3*I`, `0.5-6.3 I`, `a+I*b`, `Log[6.7] + I*Sin[x]`.

Система Mathematica имеет следующие функции выполнения операций над комплексными числами:

- $\text{Abs}[z]$ — вычисляет модуль комплексного числа z ;
- $\text{Arg}[z]$ — возвращает модуль комплексного числа z ;
- $\text{Conjugate}[z]$ — возвращает комплексное число сопряженное z ;
- $\text{Re}[z]$ — возвращает действительную часть комплексного числа z ;
- $\text{Im}[z]$ — возвращает мнимую часть комплексного числа z .

На рис. 2.12 приведены примеры использования этих функций.

$$\begin{aligned}
 & z = 5 - 3i \\
 & 5 - 3i \\
 & \text{Abs}[z] \\
 & \sqrt{34} \\
 & \text{Arg}[z] \\
 & -\text{ArcTan}\left[\frac{3}{5}\right] \\
 & \text{Conjugate}[z] \\
 & 5 + 3i \\
 & \text{Re}[z] \\
 & 5 \\
 & \text{Im}[z] \\
 & -3
 \end{aligned}$$

Рис. 2.12. Операции над комплексными числами

$$\begin{aligned}
 & z1 = 2 + 3i \\
 & z2 = 3 - 2i \\
 & z1 + z2 \\
 & 2 + 3i \\
 & 3 - 2i \\
 & 5 + i \\
 & z1 \cdot z2 \\
 & 12 + 5i \\
 & z1 / z2 \\
 & i \\
 & \text{Sqrt}[z1 - z2] \\
 & \sqrt{-1 + 5i} \\
 & \text{N}[\%] \\
 & 1.67415 - 0.895977i
 \end{aligned}$$

Рис. 2.13. Математические операции над комплексными числами

Система позволяет выполнять любые математические вычисления над комплексными числами, которые доступны для случая вещественных чисел.

Пример 2.3

Даны два комплексных числа: $z1 = 2 + 3i$, $z2 = 3 - 2i$. Необходимо выполнить над числами операции сложения, умножения, деления, извлечение квадратного корня из разности чисел.

Решение приведено на рис. 2.13.

2.2.4. Переменные

Переменные — объекты, принимающие различные значения как численные, так и символьные. При этом переменной может быть любой объект, даже такой, как графический или звуковой. Имя переменной называется *идентификатором*, оно всегда начинается с буквы и может содержать любые символы.

Переменные в системе Mathematica являются глобальными, т. е. такими, которые можно изменять в процессе вычислений. Например, присвоим переменной x значение $x=1.5$ и вычислим $\sin[x]$, затем присвоим $x=0.35$ и вновь вычислим $\sin[x]$. Получим два разных ответа. Система допускает переприсваивание переменной. Решение имеет вид:

```
x=1.5
Sin[x]
1.5
0.997495
x=0.35
Sin[x]
0.35
0.342898
```

Присваивание в системе Mathematica реализуется двумя способами: с помощью функций и операторов.

Основные функции присваивания имеют вид:

- ❑ `Set[y, f]` — вычисляет значение функции f и присваивает результат значению y ; присвоение является глобальным;
- ❑ `Set[{y1, y2, ...}, {f1, f2, ...}]` — вычисляет значения функций $f1, f2, \dots$ и присваивает их, соответственно, значениям $y1, y2, \dots$; присвоение глобальное;
- ❑ `SetDelayed[y, f]` — присваивает переменной y невычисленное f (функция задержанного присваивания);
- ❑ `SetDelayed[{y1, y2, ...}, {f1, f2, ...}]` — присваивает переменным $y1, y2, \dots$ невычисленные значения $f1, f2, \dots$ (задержанное присваивание).

Рассмотрим эти функции более подробно и приведем примеры.

Аналогом функции `Set[y, f]` является оператор $y=f$.

На рис. 2.14 приведены вычисления с помощью функции `Set[y, f]` и оператора $y=f$.

Из рис. 2.14 видно, что присвоение переменной y значения f с помощью функции и оператора идентичны.

На рис. 2.15 показан пример присвоения значений трех функций, соответственно, переменным $y1, y2, y3$.


```

Set [y, Sin[1.]]
0.841471
y
0.841471
Set [y, Sin[1.5]]
0.997495
y
0.997495
x=1.
Sin[x]
1.
0.841471
x=1.5
Sin[x]
1.5
0.997495

```

Рис. 2.14. Присвоение переменной с помощью функции и оператора присваивания

```

Set [{y1,y2,y3}, {E^2., Cos[1.], 2./3+7/4}]
{7.38906, 0.540302, 2.41667}
y1
7.38906
y2
0.540302
y3
2.41667

```

Рис. 2.15. Присвоение значений функций большому числу переменных

Из рис. 2.15 видно, что функция `set` выдала решение без задержки, сразу же после нажатия комбинации клавиш `<Shift>+<Enter>`. Получить отдельно значения переменных `y1`, `y2`, `y3` можно всегда в процессе любых вычислений.

Рассмотрим теперь функции задержанного присваивания.

Аналогом функции задержанного присваивания `SetDelayed[y, f]` является оператор `y:=f`.

Эти функция и оператор вычисляют значение `f`, присваивают его `y`, но решение (при нажатии комбинации клавиш `<Shift>+<Enter>`) на экран не выдают. Только

при обращении к y получаем результат присваивания (задержанное присваивание).

Эти процедуры хорошо видны на рис. 2.16.

```
SetDelayed[y,Sin[1.]]  
y  
0.841471  
(y-1)/(y+1)  
-0.0860883  
y:=Sin[1.]  
y  
0.841471  
y^3  
0.595823
```

Рис. 2.16. Процедуры задержанного присваивания

Отличия операторов присваивания "=" и ":=" покажем на примере.

На рис. 2.17 приведены вычисления значений $\sin x$ при $x=1$ и $x=2$ с обоими видами присваивания.

```
x=Sin[1.] ;  
y=x  
0.841471  
x=Sin[2]  
y  
Sin[2]  
0.841471  
z:=Sin[1.]  
w=z  
0.841471  
z:=Sin[2.]  
w=z  
0.909297
```

Рис. 2.17. Вычисления с двумя видами присваивания

Из рис. 2.17 видно, что при использовании оператора "=", когда вычисленные значения функции присваиваются переменной немедленно, функция $\sin 2$ не

вычислена. Результатом вычисления является значение функции `Sin1`, которое было присвоено переменной `y` ранее.

При использовании оператора задержанного присваивания `:=` функция `Sin2` вычислена, т. к. переменной `z` ранее не было присвоено значение `Sin1`.

Переменные с одним и тем же именем могут быть в разных местах программы. Это приводит к конфликтам. Решение может оказаться ошибочным. Поэтому в системе Mathematica предусмотрена возможность снятия присвоенных значений. Это осуществляется с помощью функции `Clear[y]` или символа `"=."`.

На рис. 2.18 приведены процедуры вычисления `Sinx` при `x=1` и `x=2` с использованием функции `Clear[x]`.

```
x =Sin[1.] ;  
y=x  
0.841471  
Clear [x]  
x=Sin[2]  
y  
Sin[2]  
0.909297
```

Рис. 2.18. Вычисления функции `Sinx` с использованием функции `Clear[x]`

2.3. Выражения, их преобразования и вычисления

2.3.1. Функции, операторы и символы вычисления выражений

При вводе математических выражений и выполнении вычислений нужно соблюдать следующие правила синтаксиса:

- ☐ знак умножения можно заменять пробелом;
- ☐ любая встроенная функция начинается с заглавной буквы, а аргументы помещаются в квадратных скобках;
- ☐ круглые скобки `"( )"` применяются для определения приоритета вычислений и выделения самостоятельных частей выражения;
- ☐ квадратные скобки `"[ ]"` используются для параметров функций;
- ☐ фигурные скобки `"{ }"` служат для записи векторов, матриц и других данных в виде списков.

При вычислении выражений используются следующие функции, операторы, символы:

- ☐ $N[f]$ — вычисляет в виде вещественного результата значение функции f ;
- ☐ $N[f, n]$ — вычисляет в виде вещественного результата значение функции f с числом цифр, равным n ;
- ☐ f/N — вычисляет выражение f в численном виде;
- ☐ $N[\%]$ — вычисление в численном виде функции, предшествующей операции;
- ☐ $\%$ — возвращает результат последней операции;
- ☐ $\%\%$ — возвращает результат предпоследней операции;
- ☐ $\% \dots \%$ — возвращает результат операции, выполненной в строке, отстоящей от конца на число символов $\%$;
- ☐ $\%n$ — возвращает результат операции в строке n .

Пример использования перечисленных функций, операторов и символов приведен на рис. 2.19 при вычислении функции $z=3 \operatorname{Log}[5]+3/7$.

```
z=3*Log[5]+3/7
%

$$\frac{3}{7} + 3 \operatorname{Log}[5]$$


$$\frac{3}{7} + 3 \operatorname{Log}[5]$$

N[%]
5.25689
N[z]
5.25689
N[z,20]
5.2568851658737296952
z/N
5.25689
%%
5.2568851658737296952
```

Рис. 2.19. Вычисление функции с помощью различных функций, операторов и символов

Если результатом вычислений являются вещественные числа, то программа может выдать список, содержащий мантиссу и порядок каждого числа. Для этого предназначена функция `MantissaExponent[x]`. Так, например, веществен-

ные числа 23.325 , $125 \cdot 10^5$, $0.0234 \cdot 10^{-3}$ с помощью этой функции преобразуются в вектор, образованный из мантисс и векторов чисел:

```
MantissaExponent[{23.325, 125.10^5, 0.0234 10^(-3)}]
{{0.23325, 2}, {0.125, 8}, {0.234, -4}}
```

2.3.2. Подстановки

Подстановки являются математическим аппаратом вычисления функций при численных значениях переменных. Они позволяют табулировать функции при широком изменении аргумента. В случае символьных переменных подстановки позволяют создавать новые математические выражения, получать формулы.

В системе Mathematica подстановки осуществляются с помощью символа `/.`. Комбинации этих символов, функций и переменных имеют вид:

```
f(x) /. x -> a
f(x, y, ...) /. {x->a, y->b, ...}
{f1(x), f2(x), ...} /. x->a
{f1(x, y, ...), f2(x, y, ...), ...} /. {x->a, y->b, ...}
f(x) /. x->{x0, x1, ...}
```

Выражения подстановок имеют следующий смысл:

- $f(x) /. x \rightarrow a$ — осуществляет подстановку в выражение $f(x)$ значения $x=a$. При этом если a является численным значением переменной, то осуществляется не только подстановка, но также вычисление выражения $f(x)$. Аргумент x может быть числом вещественным и комплексным;
- $f(x, y, ...) /. \{x \rightarrow a, y \rightarrow b, \dots\}$ — осуществляет подстановки в выражение $f(x, y, \dots)$ переменных $a, b, \dots$. Откликом является численное значение функции, если все переменные — числа, и функция, если хотя бы одна из переменных является символьной;
- $\{f_1(x), f_2(x), \dots\} /. x \rightarrow a$ — осуществляет подстановки во все выражения значения $x=a$. Откликом является вектор чисел, если a — число, и вектор математических выражений, если a — символьное значение переменной;
- $\{f_1(x, y, \dots), f_2(x, y, \dots), \dots\} /. \{x \rightarrow a, y \rightarrow b, \dots\}$ — подстановки переменных $a, b, \dots$ во все функции и вычисление их значений, если переменные — числа;
- $f(x) /. x \rightarrow \{x_0, x_1, \dots\}$ — табулирование функции $f(x)$.

Приведем примеры подстановок.

Пример 2.4

Необходимо осуществить подстановки аргументов $x=1$, $y=2$; $x=a$, $y=b$ в выражения:

$$z_1 = x E^x + \ln x - 1, \quad z_2 = x^2 + y^2 + \sin^2(x) + \cos^2(x) + 1.5.$$

Определить также значение z_2 при $x=y=a$.

В данном случае выражения подстановок будут иметь вид:

$$z_1 /. x \rightarrow 1;$$

$$z_2 /. \{x \rightarrow 1, y \rightarrow 2\}$$

Реализация этих выражений приведена на рис. 2.20.

```

z1=x E^x+Log[x]-1.;
z1/.x→1.
1.71828
z2=x^2+y^2+Sin[x]^2+Cos[x]^2+1.5;
z2/.{x→1,y→2}
7.5
z2/.y→2.
5.5+x^2+Cos[x]^2+Sin[x]^2
z1/.x→a
-1.+a e^a+Log[a]
z2/.{x→a,y→b}
1.5+a^2+b^2+Cos[a]^2+Sin[a]^2
z2/.{x→a,y→a}
1.5+2 a^2+Cos[a]^2+Sin[a]^2

```

Рис. 2.20. Результаты решения примера 2.4

Из рис. 2.20 видно, что система не только осуществила подстановку данных в выражения, но также определила их численные значения. К сожалению, она не смогла упростить выражение $\sin^2(x) + \cos^2(x)$, которое при любом значении x равно 1, что известно даже школьнику.

Пример 2.5

Необходимо подставить значения $x=1$ и $y=2$, а также $x=1$, $y=a$ одновременно в следующие функции:

$$z_3 = \sin x + \cos x, e^{-x}, (x-1)/(x+1), \ln x;$$

$$z_4 = \sin(x+y) + \cos(x/y), E^{-|x+y|}, (xy-1)/(xy+1), \ln(x^2+y^2).$$

В данном случае выражения подстановки будут иметь вид:

$$z_3 /. x \rightarrow 1.$$

$$z_4 /. \{x \rightarrow 1, y \rightarrow 2\}$$

Решение приведено на рис. 2.21.

```
z3={Sin[x]+Cos[x],E^(-x),(x-1)/(x+1),Log[x]}
z3/.x->1.
{Cos[x]+Sin[x],e^-x,-1+x/(1+x),Log[x]}
{1.38177,0.367879,0.,0.}
z4={Sin[x+y]+Cos[x/y],E^(-x-y),(x y-1)/(x y+1),Log[x^2+y^2]};
z4/.{x->1.,y->2.}
{1.0187,0.0497871,0.333333,1.60944}
z4/.{x->1.,y->a}
{Cos[1/a]+Sin[1.+a],e^-1.-a,-1+1.a/(1+1.a),Log[1.+a^2]}
```

Рис. 2.21. Решение задачи примера 2.5

Пример 2.6

Выполнить табулирование функций $x e^{-x}$, x^2 , $\sin x$, $2/(x^2-1)$ при $x=0.3, 0.6, 0.9, 1.2, 1.5, 1.8$. Решение представить в форме таблицы с помощью встроенной функции `TableForm[%]`.

Решение примера приведено на рис. 2.22.

```
{x,x E^(-x),x^2,Sin[x],2/(x^2-1)}/.x->{0.3,0.6,0.9,1.2,1.5,1.8}
{{0.3,0.6,0.9,1.2,1.5,1.8},{0.222245,0.329287,0.365913,0.361433,0.334695,0.297538},
{0.09,0.36,0.81,1.44,2.25,3.24},{0.29552,0.564642,0.783327,0.932039,0.997495,0.973848},
{-2.1978,-3.125,-10.5263,4.54545,1.6,0.892857}}
```

0.3	0.6	0.9	1.2	1.5	1.8
0.222245	0.329287	0.365913	0.361433	0.334695	0.297538
0.09	0.36	0.81	1.44	2.25	3.24
0.29552	0.564642	0.783327	0.932039	0.997495	0.973848
-2.1978	-3.125	-10.5263	4.54545	1.6	0.892857

Рис. 2.22. Решение примера 2.6

2.3.3. Преобразование выражений

При выводе формул, получении математических моделей, в процессе любых символьных вычислений исследователю приходится преобразовывать математические выражения с целью их упрощения, понимания физической сущности полученных выражений, высокой наглядности.

Эти процедуры, как правило, не формализованы. Их выполнение — одна из наиболее интеллектуальных задач.

Система *Mathematica* имеет несколько десятков встроенных функций преобразования и упрощения выражений, приведения их к желаемому виду. Эти функции позволяют выполнять следующие операции: раскрытие скобок, приведение к общему знаменателю, приведение подобных членов, сокращение аналитических выражений, упрощение тригонометрических выражений и многое другое.

Интеллектуальность системы компьютерной алгебры во многом зависит от числа таких функций и качества программ их использующих. В этом смысле система *Mathematica* — одна из наиболее интеллектуальных систем компьютерной алгебры.

Рассмотрим следующие основные функции символьных преобразований системы *Mathematica*:

- ☐ `Simplify[f]` — упрощает выражение `f`;
- ☐ `FullSimplify[f]` — упрощает выражение `f`, имеющее в своем составе специальные функции;
- ☐ `Expand[f]` — раскрывает и расширяет выражение `f`;
- ☐ `Collect[f, x]` — представляет выражение `f` по степеням `x`;
- ☐ `TrigExpand[f]` — преобразовывает тригонометрические выражения;
- ☐ `Factor[f]` — раскладывает математические выражения на множители.

Каждая из этих функций имеет несколько форм и видов представления.

Рассмотрим подробно эти функции и приведем примеры.

Функция *Simplify*

Основное назначение функции `Simplify[f]` — упрощение математических выражений. При этом ограничения на сложность выражений, тип функций и переменных в них отсутствуют.

Функция упрощает выражения путем приведения членов к общему знаменателю, упрощения тождеств, приведения подобных, разложения полиномов на множители и понижения их степени путем сокращения, преобразования функций, в том числе и тригонометрических.

Технология преобразования элементарна и состоит в следующем:

1. Ввод функции, требующей преобразования, с присвоением ей уникального имени, например, `f`.

2. Ввод функции `Simplify[f]`.
3. Получение решения путем нажатия клавиш решения `<Shift>+<Enter>`.

Функцию `f` можно отдельно не вводить, если она не нужна для дальнейших вычислений.

Приведем примеры упрощения выражений с помощью функции `Simplify[f]`.

Пример 2.7

Необходимо упростить следующие выражения:

$$957828/3831312,$$

$$(315+297x-62x^2-42x^3+3x^4+x^5)/(105-41x-x^2+x^3),$$

$$(\sin x + \cos x)^2 - 1,$$

$$a^4 + a^3x + a^2x^2 + ax^3 + x^4 + a(-1 + a^4)/(-a + x),$$

$$\frac{2+a+b-c}{a+b-c} - \frac{b+ab+c+ac}{b+c} + a$$

$$7.59375 + 25.3125x + 33.75x^2 + 22.5x^3 + 7.5x^4 + x^5,$$

$$(2x + a^{\log(a,x)} + 3)/3$$

Упрощение выражений с помощью функции `Simplify[f]` приведено на рис. 2.23.

Результаты упрощения выражений впечатляют. Сокращение дробей, понижение степени полиномов, преобразование символьных выражений, преобразование функций — таков неполный перечень операций, выполненных функцией `Simplify[f]`. Ответы далеко не тривиальны. Вряд ли студент, да и не только он, смог бы упростить все выражения примера до такой степени, как это получено на рис. 2.23.

Функция *FullSimplify*

Эта функция, как и предыдущая, предназначена для упрощения выражений. Только упрощения она выполняет более "квалифицированно", чем функция `Simplify[f]`, а кроме того, она может оперировать специальными функциями. Покажем это на примерах.

Пример 2.8

Необходимо упростить следующие выражения:

$$x \sin x \cos y + (x^2 + 1)/x \sin y \cos x,$$

$$(\sin x + \cos x)^2,$$

$$(1 - \cos 2x)/2 + \cos x^2 + a.$$

Simplify[957828/3831312]

$$\frac{1}{4}$$

Simplify $\left[\frac{315 + 297 x - 62 x^2 - 42 x^3 + 3 x^4 + x^5}{105 - 41 x - x^2 + x^3}\right]$

$$3 + 4 x + x^2$$

Simplify[(Sin[x]+Cos[x])^2-1]

$$\sin[2 x]$$

Simplify $\left[a^4 + a^3 x + a^2 x^2 + a x^3 + x^4 + \frac{a(-1+a^4)}{-a+x}\right]$

$$\frac{a-x^5}{a-x}$$

$\frac{2+a+b-c}{a+b-c} - \frac{b+ab+c+ac}{b+c} + a$

$$a + \frac{2+a+b-c}{a+b-c} - \frac{b+ab+c+ac}{b+c}$$

Simplify[%]

$$\frac{2}{a+b-c}$$

Simplify $[7.59375 + 25.3125 x + 33.75 x^2 + 22.5 x^3 + 7.5 x^4 + x^5]$

$$1. (1.5 + x)^5$$

Simplify[(2 x+a^Log[a,x]+3)/3]

$$1+x$$

Рис. 2.23. Упрощение выражений с помощью функции **Simplify[f]**

Simplify[x *Sin[x] * Cos[y]+(x^2+1)/x*SIN[y] Cos[x]]

$$x \cos[y] \sin[x] + \frac{(1+x^2) \cos[x] \sin[y]}{x}$$

FullSimplify[x*SIN[x]*Cos[y]+(x^2+1)/x*SIN[y]*Cos[x]]

$$\frac{\cos[x] \sin[y]}{x} + x \sin[x+y]$$

Simplify[(Sin[x]+Cos[x])^2]

$$(\cos[x] + \sin[x])^2$$

FullSimplify[(Sin[x]+Cos[x])^2]

$$1+\sin[2 x]$$

FullSimplify[(1-Cos[2 x])/2+Cos[x]^2+a]

$$1+a$$

Рис. 2.24. Упрощение выражений с помощью двух функций

Показать преимущества функции `FullSimplify`, выполнив упрощение выражений двумя способами.

Решение задачи приведено на рис. 2.24.

Из рис. 2.24 видны преимущества функции `FullSimplify` перед функцией `simplify`. Последняя не смогла упростить даже такое выражение, как $(\sin x + \cos x)^2$.

Функции `Simplify[f]` и `FullSimplify[f]` упрощают выражения с вещественными и комплексными переменными. На рис. 2.25 приведены примеры упрощения таких выражений.

```

FullSimplify[(a+b I)/(a-b I)+(a-b I)/(a+b I)]
-2 +  $\frac{4 a^2}{a^2 + b^2}$ 
FullSimplify[1/(2 I) (E^(I x)-E^(-I x))]
Sin[x]
FullSimplify[Cos[x]+I Sin[x]-E^(I x)+(a-I b)/(a+I b)]
 $\frac{a - i b}{a + i b}$ 

```

Рис. 2.25. Упрощение выражений с комплексными переменными

Функции *Expand*

Функции `Expand[f]` выполняют преобразования вида раскрытия и расширения выражений. Функций этих много. Рассмотрим основные из них:

- ☐ `Expand[f]` — раскрывает скобки произведений и возводит выражения в целую положительную степень;
- ☐ `ExpandAll[f]` — раскрывает скобки произведений и возводит выражения в целую положительную степень в любом месте выражения f ;
- ☐ `ExpandNumerator[f]` — раскрывает скобки произведений и возводит выражения в целую положительную степень в числителе выражения f ;
- ☐ `ExpandDenominator[f]` — раскрывает скобки произведений и возводит выражения в целую положительную степень в знаменателе выражения f ;
- ☐ `PowerExpand[f]` — раскрывает скобки и возводит в целую положительную степень вложенные выражения функции f ;
- ☐ `ComplexExpand[f]` — раскрывает скобки произведений и возводит выражения в целую положительную степень, считая, что переменные выражения f являются вещественными;

- ❑ `ComplexExpand[f, {x1, x2, ...}]` — раскрывает скобки произведений и возводит выражения в целую положительную степень, считая, что все переменные выражения `f` являются комплексными;
- ❑ `FunctionExpand[f]` — представляет упрощенные выражения функции `f`, содержащей специальные функции.

На рис. 2.26 и 2.27 приведены примеры расширения разнообразных выражений с помощью функций `Expand`.

Expand[(a+b) (a-b) (a+c)/(a+1)^3]

$$\frac{a^3}{(1+a)^3} - \frac{ab^2}{(1+a)^3} + \frac{a^2c}{(1+a)^3} - \frac{b^2c}{(1+a)^3}$$

ExpandAll[(a+b) (a-b) (a+c)/(a+1)^3]

$$\frac{a^3}{1+3a+3a^2+a^3} - \frac{ab^2}{1+3a+3a^2+a^3} + \frac{a^2c}{1+3a+3a^2+a^3} - \frac{b^2c}{1+3a+3a^2+a^3}$$

ExpandAll[Sqrt[x+2^3]/((a+b) (a-b) (a-b)+(c+2 I)^2)]

$$-4 + 4i c + c^2 + \frac{\sqrt{8+x}}{a+b}$$

ExpandNumerator[(x^2-1)^5/(x+1)^3]

$$\frac{-1 + 5x^2 - 10x^4 + 10x^6 - 5x^8 + x^{10}}{(1+x)^3}$$

ExpandDenominator[(x^2-1)^5/(x+1)^3]

$$\frac{(-1+x^2)^5}{1+3x+3x^2+x^3}$$

Рис. 2.26. Раскрытие выражений с помощью первых четырех функций `Expand`

Анализ результатов преобразований, приведенных на рис. 2.26 и 2.27, позволяет раскрыть их сущность, увидеть возможности функций `Expand`.

Функции `Expand` позволяют:

- ❑ осуществлять раскрытие и расширение выражений при любых видах переменных (вещественных и комплексных, численных и символьных);
- ❑ упрощать выражения в процессе раскрытия и расширения;
- ❑ осуществлять раскрытие и расширение в любом желаемом месте выражения (числитель, знаменатель, аргумент элементарной или специальной функции и т. д.);
- ❑ осуществлять действия со специальными функциями;
- ❑ осуществлять раскрытие и расширение функций, представляя переменные в виде вещественных или комплексных переменных.

PowerExpand[Sqrt[(x+2)^4 a b]/(x-1)^3]

$$\frac{\sqrt{a} \sqrt{b} (2+x)^2}{(-1+x)^3}$$

PowerExpand[Log[(x+3)^3]/(a+b)^2]

$$\frac{3 \operatorname{Log}[3+x]}{(a+b)^2}$$

ComplexExpand[(a+b)^2+(a-b I)^2]

$$2 a^2 + (2 - 2 i) a b$$

ComplexExpand[(a+b)^2+(a-b I)^2, a]

$$2 b \operatorname{Im}[a] - 2 \operatorname{Im}[a]^2 + 2 b \operatorname{Re}[a] + 2 \operatorname{Re}[a]^2 + i (2 b \operatorname{Im}[a] - 2 b \operatorname{Re}[a] + 4 \operatorname{Im}[a] \operatorname{Re}[a])$$

FunctionExpand[{EulerE[5,x],BernoulliB[5,x],Fibonacci[7,x]}]

$$\left\{ -\frac{1}{2} + \frac{5x^2}{2} - \frac{5x^4}{2} + x^5, -\frac{x}{6} + \frac{5x^2}{3} - \frac{5x^4}{2} + x^5, 1 + 6x^2 + 5x^4 + x^6 \right\}$$

TableForm[Out[27]]

$$\begin{array}{l} -\frac{1}{2} + \frac{5x^2}{2} - \frac{5x^4}{2} + x^5 \\ -\frac{x}{6} + \frac{5x^2}{3} - \frac{5x^4}{2} + x^5 \\ 1 + 6x^2 + 5x^4 + x^6 \end{array}$$

FunctionExpand[(x-2)^2/EulerE[5,x]+(a-b)/(a-b)^2]

$$\frac{1}{a-b} + \frac{(-2+x)^2}{-\frac{1}{2} + \frac{5x^2}{2} - \frac{5x^4}{2} + x^5}$$

FunctionExpand[EulerE[5,x]+BernoulliB[5,x]-x^2-x^3-1/x]

$$-\frac{1}{2} - \frac{1}{x} - \frac{x}{6} + \frac{3x^2}{2} + \frac{2x^3}{3} - 5x^4 + 2x^5$$

f=(x+1) (x+2) (x+5) (3+y)

$$(1+x) (2+x) (5+x) (3+y)$$

Expand[f,x]

$$10 (3+y) + 17 x (3+y) + 8 x^2 (3+y) + x^3 (3+y)$$

Expand[f,y]

$$3 (1+x) (2+x) (5+x) + (1+x) (2+x) (5+x) y$$

Рис. 2.27. Раскрытие выражений с помощью последних четырех в списке функций Expand

Функция Collect

Функция Collect представляется в следующих видах:

- ☐ `Collect[f, x]` — осуществляет приведение общих членов функции `f` по переменной `x`;

□ `Collect[f, {x1, x2, ...}]` — осуществляет приведение общих членов функции `f` по переменным `x1, x2, ...`

Действия функции покажем на следующем примере.

Пример 2.9

Необходимо осуществить приведение подобных в следующих выражениях:

$(x+a)^3 - (x-a)^2,$
 $(x+1)^5 - \text{EulerE}(5, x) - (x+a)^3,$
 $(x-y)(x+y)^2 \sqrt{(x+y)^3},$
 $(x+y)^2 + (x-1)^3 - (y-1)^3,$
 $1/(x+y) + 1/(x-y) + x-y, \{x, y\}.$

Решение приведено на рис. 2.28.

```
Collect[(x+a)^3-(x-a)^2,x]
-a^2+a^3+(2 a+3 a^2) x+(-1+3 a) x^2+x^3
Collect[(x+1)^5-EulerE[5,x]-(x+a)^3,x]
3/2 -a^3+(5-3 a^2) x+(15/2-3 a) x^2+9 x^3+15 x^4/2
Collect[(x-y)(x+y)^2 Sqrt[(x+y)^3],x]
x^3 Sqrt[(x+y)^3]+x^2 y Sqrt[(x+y)^3]-x y^2 Sqrt[(x+y)^3]-y^3 Sqrt[(x+y)^3]
Collect[(x+y)^2+(x-1)^3-(y-1)^3,{x,y}]
-2 x^2+x^3-3 y+4 y^2-y^3+x(3+2 y)
Collect[1/(x+y)+1/(x-y)+x-y,{x,y}]
x+1/(x-y)-y+1/(x+y)
```

Рис. 2.28. Приведение подобных с помощью функции `Collect`

Из рис. 2.26 видно, что программа не привела подобных в последнем выражении, она лишь его повторила.

Функции *Factor*

Основное назначение функций `Factor[f]` — разложение функции `f` на множители. Эти функции также позволяют осуществлять операции выноса за скобки, приведения к общему знаменателю и некоторые другие действия, которые увидим при решении примеров.

Жестких требований на функцию `f` и ее аргументы не накладывается. Она может быть полиномом неограниченной степени, тригонометрической функцией,

может иметь в своем составе элементарные и специальные функции. Ее аргументами могут быть вещественные и комплексные числа, а также символьные переменные.

Система Mathematica имеет большое число встроенных функций `Factor[f]` и опций к ним. Далее приведены основные из них:

- ❑ `Factor[f]` — разложение функции f на простые множители;
- ❑ `FactorList[f]` — возвращает перечень множителей и число степеней каждого из них, например, $\{\{1, 1\}, \{x-3, 1\}, \{x+2, 2\}, \{x+a, 1\}\}$, что означает: $f=(x-3)(x+2)^2(x+a)$;
- ❑ `FactorTerms[f]` — выносит за скобки общий множитель, содержащийся в f ;
- ❑ `FactorTermsList[f]` — выделяет общий множитель, содержащийся в f , представляя функцию в ином виде по сравнению с предыдущей;
- ❑ `FactorInteger[n]` — возвращает простые множители целого числа n ;

```
Factor[18 x^3+45 x^2+ x-14]
(-1+2 x) (2+3 x) (7+3 x)
Factor[x^3+x^2 (3+a-b)+x (3 a-3 b-a b)-3 a b]
-(b-x) (3+x) (a+x)
Factor[a^2+a-1]
-1+a+a^2
Factor[6 x^3+x^2 (13 I-6)-6 x (1+13/6 I)+6]
(-1+x) (3 I+2 x) (2 I+3 x)
Factor[Sin[4 x],Trig->True]
4 Cos[x] (Cos[x]-Sin[x]) Sin[x] (Cos[x]+Sin[x])
Factor[1/16 (Cos[5 x]+5 Cos[3 x]+10 Cos[x]), Trig->True]
Cos[x]^5
FactorList[x^5+x^4-6 x^3-x^2-x+6]
{{1, 1}, {-2+x, 1}, {-1+x, 1}, {3+x, 1}, {1+x+x^2, 1}}
FactorTerms[3 x^3-3 x^2-51 x-45]
3 (-15-17 x-x^2+x^3)
FactorTermsList[3 x^3-3 x^2-51 x-45]
{3, -15-17 x-x^2+x^3}
FactorInteger[1235]
{{5, 1}, {13, 1}, {19, 1}}
```

Рис. 2.29. Примеры применения функций `Factor[f]`

- ❑ `Factor[f, Trig -> True]` — разложение тригонометрической функции на простые множители.

На рис. 2.29 приведены примеры применения перечисленных функций. Из рисунка видно, что не все предложенные функции удается факторизовать. Так, например, программа не смогла разложить функцию $f=a^3+a-1$ на множители, хотя решение достаточно простое и имеет вид:

$$f = \left(a + \frac{\sqrt{5}}{2} + \frac{1}{2} \right) \left(a - \frac{\sqrt{5}}{2} + \frac{1}{2} \right).$$

Обратим также внимание на то, что функция `Factor[f]` выполняет разложение на множители тригонометрических выражений. Для этого только нужно применить опцию `Trig >= True`.

Функции преобразования тригонометрических выражений

Преобразование тригонометрических выражений (упрощение, раскрытие, расширение) может осуществляться с помощью функций `Simplify` и `Expand`. Однако имеют место случаи, когда эти функции не могут выполнить преобразования и возвращают исходное выражение. В таких случаях полезно использовать специальные функции преобразования тригонометрических выражений.

Система *Mathematica* имеет большое число функций упрощения выражений, содержащих тригонометрические функции. Приведем основные из них:

- ❑ `TrigReduce[f]` — упрощает выражение, содержащее тригонометрические функции;
- ❑ `TrigExpand[f]` — осуществляет расширение тригонометрических функций;
- ❑ `TrigFactor[f]` — раскладывает на множители тригонометрическое выражение;
- ❑ `TrigToExp[f]` — представляет тригонометрическое выражение в экспоненциальной форме;
- ❑ `ExpToTrig[f]` — представляет экспоненциальное выражение в тригонометрической форме.

Примеры преобразования тригонометрических выражений с помощью этих функций приведены на рис. 2.30.

Нами описаны основные функции упрощения, расширения, разложения выражений на множители. Система *Mathematica* имеет около пятидесяти других функций работы с выражениями. Эти функции при практических инженерных расчетах большого значения не имеют. Их изучение понадобится только отдельным редким пользователям. Для широкого круга пользователей, тем более для студентов, они вряд ли нужны, поэтому нами они не изучаются — нельзя объять необъятное.


```

TrigReduce[(Sin[x]+Cos[x])^2]
1+Sin[2 x]
TrigReduce[2 Sin[1/2 (x+y)] Sin[1/2 (y-x)]]
Cos[x]-Cos[y]
TrigExpand[Sin[4 x]]
4 Cos[x]^3 Sin[x] - 4 Cos[x] Sin[x]^3
TrigExpand[Cosh[Cos[x y]]]
Cosh[1/2 Cos[x y] - 1/2 i Sin[x y]] Cosh[1/2 Cos[x y] + 1/2 i Sin[x y]] +
Sinh[1/2 Cos[x y] - 1/2 i Sin[x y]] Sinh[1/2 Cos[x y] + 1/2 i Sin[x y]]
TrigFactor[Cos[x]-Cos[y]]
-2 Sin[x/2 - y/2] Sin[x/2 + y/2]
TrigToExp[Cos[x]+sin[x]]
e^{-ix}/2 + e^{ix}/2 + sin[x]
TrigToExp[1/I Sin[I x]]
-e^{-x}/2 + e^x/2
ExpToTrig[E^x+E^(-x)]
2 Cosh[x]

```

Рис. 2.30. Примеры преобразования выражений с помощью тригонометрических функций

2.3.4. Примеры на преобразование выражений

В табл. 2.3 приведены выражения, требующие преобразования. Они не систематизированы по признаку упрощения, расширения, разложения. Пользователь должен самостоятельно выбрать функцию преобразования. Его основная задача — получить ответ, совпадающий с приведенным.

Не следует огорчаться, если ответы не совпадают. Может оказаться, что несколько встроенных функций способны преобразовать выражение, но ответы получаются разные. Ответы также могут не совпадать с полученными с помощью иных систем компьютерной алгебры.

Все это объясняется тем, что многие процедуры упрощения в математике не формализованы. Этот процесс настолько интеллектуален, что допускает множество ответов. Нужно только показать, что результат преобразования является верным.

Успеха вам, читатель!

Таблица 2.3. Задания

№ задачи	Выражение	Ответ
1	$\frac{x^4 - 6x^3 - 4x^2 - 18x - 21}{x^3 - 7x^2 + 3x - 21}$	$x - 3$
2	$1/8(\cos 4x + 4 \cos 2x + 3)$	$\cos^4 x$
3	$\cos x(4 \sin x - 8 \sin^3 x)$	$\sin 4x$
4	$\frac{2^{2x-1}}{\sqrt{\pi \Gamma(x) \Gamma(x+1/2)}}$	$\Gamma(2x)$
5	$\frac{x^4 + x^3(a+b) + x^2(ab-1) - x(a+b) - ab}{(x+1)(x^2 + x(a+b) + a+ab)}$	$x - 3$
6	$1/16(\cos 5x + 5 \cos 3x + 10 \cos x)$	$\cos^5 x$
7	$\cos x + i \sin x$	e^{ix}
8	$\frac{2x + a^{\log a^x} + 3}{3}$	$x + 1$
9	$1/64(-\sin 7x + 7 \sin 5x - 21 \sin 3x + 35 \sin x)$	$\sin^7 x$
10	$7 \sin x - 56 \sin^3 x + 112 \sin^5 x - 64 \sin^7 x$	$\sin 7x$
11	$1/i \ln(iz + \sqrt{1-z^2})$	$\arcsin z$

Таблица 2.3 (окончание)

№ задачи	Выражение	Ответ
12	$x^4 + 7.5x^3 - 7.57x^2 - 57.099x + 64.386$	$(x-1.2)(x+3.5)(x+7.3)(x-2.1)$
13	$1/2(1+1/i) \cdot e^{ix} + (1-1/i) \cdot e^{-ix}$	$\sin x + \cos x$
14	$x^4 + 17x^3 + 13x^2 - 233x - 204$	$(x-4)(x+17)(x+3)(x+1)$
15	$\arcsin \left(x\sqrt{1-y^2} - y\sqrt{1-x^2} \right)$	$\arcsin x - \arcsin y$
16	$2 \sin(1/2(x+y)) \cos(1/2(x-y))$	$\sin x + \sin y$
17	$x^6 + 4x^4 + 41x^2 + 36$	$(x-1)(x+1)(x-2)(x+2)(x-3)(x+3)$
18	$(\cos x + i \sin x)''$	$\cos nx + i \sin nx$
19	$x^5 - ax^4 + 5x^4 - 4ax^3 + 6x^3 - a^2x^2 - ax^2 - 5a^2x + 6ax - 6a^2$	$(x+2)(x+3)(x-a)(x^2+a)$
20	$\arcsin x + \arccos x$	$\pi/2$
21	$1.5e^{ix} - i \sin x + 0.5e^{-ix}$	$2 \cos x$
22	$1/64 (\cos 7x + 7 \cos 5x + 21 \cos 3x + 35 \cos x)$	$\cos^7 x$
23	$3x^3 + 7x + i(10x^2 + 30)$	$(x+2i)(x+3i)(3x-5i)$

2.3.5. Повышение точности вычислений

Одним из способов повышения точности вычислений является представление с высокой точностью вещественных чисел в виде рациональных с последующими арифметическими действиями только над рациональными числами.

Преобразование чисел в рациональные числа осуществляется с помощью следующих функций:

- ☐ `Rationalize[z]` — число z преобразуется в рациональное;
- ☐ `Rationalize[z, dz]` — число z преобразуется в рациональное с заданной точностью dz .

Приведем примеры преобразования вещественных чисел в рациональные.

Пример 2.10

Необходимо преобразовать следующие числа, выражения и функции в рациональную форму:

$62.364, 3/0.18, 2.3 \cdot 10^{-5}, 3.2^{14}, \ln 0.625, (x+a)/5.63 \cos 1., \tan 1.$

В данном случае следует воспользоваться функцией `Rationalize[z]`, которая будет иметь вид:

`Rationalize[{62.364, 3/0.18, 2.3 10-5, 3.214, Log[0.625], (x+a)/5.63 * Cos[1.], Tan[1]}]`

Решение приведено на рис. 2.31.

Из рис. 2.31 видно, что функция `Rationalize[x]` не только преобразовала вещественные числа в рациональную форму, но также вычислила выражения и функции, представив их в форме рациональных чисел. Она не смогла вычислить только $\tan 1$, т. к. число 1 не представлено в вещественной форме.

С помощью функции `N[%]` получено решение с представлением чисел в вещественной форме.

```
Rationalize[{62.364, 3/0.18, 2.3 10-5,
Log[0.625], 3.214, (x+a)/5.63 * Cos[1.], Tan[1]}]
{
  15591/250, 2345624805922133/140737488355328, 23/1000000, -54735074/116456705,
  234666196450/19877, 529505159 (a+x)/5517492731, Tan[1]}
N[%]
{62.364, 16.666666666666664, 0.000023, -0.4700036292457356,
  1.180591620717412 * 10-5, 0.0959684379872362 (a+x), 1.5574077246549023}
```

Рис. 2.31. Преобразование чисел, выражений, функций в рациональную форму

Пример 2.11

Необходимо вычислить и представить результаты в рациональной форме выражения: $\ln 5 + 2/3 - \cos 1.2$ с точностью 10^{-7} , 10^{-12} , 10^{-20} . Сравнить полученные результаты.

Решение задачи приведено на рис. 2.32.

```
Log[5] + 2/3 - Cos[1.2]
1.91375
Rationalize[Log[5] + 2/3 - Cos[1.2], 10^-7]
17861
-----
9333
Rationalize[Log[5] + 2/3 - Cos[1.2], 10^-12]
1356791
-----
708971
Rationalize[Log[5] + 2/3 - Cos[1.2], 10^-20]
178590203
-----
93319660
```

Рис. 2.32. Решение задачи примера 2.8

```
Rationalize[N[{Pi, E, EulerGamma, Catalan}], 10^-7]
{ 75948 / 24175 , 15062 / 5541 , 2579 / 4468 , 8709 / 9508 }
Rationalize[N[{Pi, E, EulerGamma, Catalan}], 10^-12]
{ 4272943 / 1360120 , 1084483 / 398959 , 612173 / 1060562 , 660555 / 721157 }
Rationalize[N[{Pi, E, EulerGamma, Catalan}], 10^-20]
{ 245850922 / 78256779 , 325368125 / 119696244 , 349378999 / 605283294 , 105640241 / 115332106 }
N[%]
{3.14159, 2.71828, 0.577216, 0.915966}
```

Рис. 2.33. Представление постоянных в виде рациональных чисел

Пример 2.12

Необходимо представить в виде рациональных чисел константы π , e , Эйлера, Каталана. Решение получить с точностью 10^{-7} , 10^{-12} , 10^{-20} .

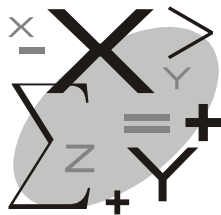
В данном случае функция `Rationalize` будет иметь вид:

```
Rationalize[N[{Pi, E, EulerGamma, Catalan}], 10^-n]
```

Решение приведено на рис. 2.33.

Большие возможности системы *Mathematica* по преобразованию математических выражений — одно из существенных ее преимуществ перед другими системами компьютерной алгебры. Эти возможности позволяют упрощать выражения, выводить формулы, доказывать теоремы.

ГЛАВА 3



Визуализация вычислений

Визуализация вычислений является одним из этапов компьютерных технологий решения математических задач. Она не только полезна, но и необходима во многих случаях. Такими случаями являются:

- ☐ определение области изоляции вещественных корней при решении алгебраических и трансцендентных уравнений;
- ☐ определение вида функции интерполяции при синтезе математической модели объекта или явления;
- ☐ проверка адекватности решения математической задачи;
- ☐ изучение особенностей функции: наличие минимумов и максимумов, точек перегиба, разрывов непрерывностей;
- ☐ изучение особенностей решения прикладных задач: вида полученной функции, длительности переходных процессов, установившегося значения и др.;
- ☐ сравнение различных вариантов решения задачи и выбор оптимального.

В отношении графики система Mathematica является лидером среди систем компьютерной алгебры. Она имеет богатые возможности и позволяет строить практически любые функции. Технология построения графиков достаточно проста и основана на использовании встроенных функций.

Большое число опций позволяет оформлять графические образы практически в любом желаемом виде.

Графики в системе Mathematica являются объектами. Им можно присваивать имена, а затем оперировать как любыми объектами.

Мы не будем подробно изучать графические возможности системы Mathematica. Они значительно шире, чем визуализация вычислений, и могут быть темой отдельного издания. Мы будем изучать графику лишь в области ее применения для визуализации вычислений. Более подробное изложение графических возможностей системы имеется в [10].

3.1. Двумерная графика

3.1.1. Графическая функция *Plot*

Графическая функция `Plot` позволяет строить графики функции $y = f(x)$ в двумерном пространстве в прямоугольной системе координат. На одном графике возможно изображение нескольких функций. По умолчанию на экран выводится координатная сетка.

Встроенная функция `Plot` представляется в следующих видах:

```
Plot[f, {x, xmin, xmax}]
```

```
Plot[{f1, f2, ...}, {x, xmin, xmax}]
```

где:

- ☐ f — функция, график которой строится;
- ☐ f_i — i -я функция, график которой строится, $i = 1, 2, \dots$;
- ☐ x — аргумент функции f ;
- ☐ $x_{\min}, x_{\max}$ — интервал изменения аргумента x .

Приведем примеры построения графиков с помощью функции `Plot`.

Пример 3.1

Необходимо представить в виде графика функцию $y = xe^{-x}$ в диапазоне x от 0 до 5.

В данном случае функция `Plot` представляется в следующем виде:

```
Plot[x Exp[-x], {x, 0, 5}]
```

Реализация функции приведена на рис. 3.1.

Пример 3.2

Необходимо построить на одном графике в диапазоне x от 0 до 1.5 следующие функции:

$$y_1 = xe^{-x}, \quad y_2 = 1/(1+x^4), \quad y_3 = 0.6, \quad y_4 = -1 + e^x.$$

В данном случае функция `Plot` представляется в следующем виде:

```
Plot[{x Exp[-x], 1/(1+x^4), 0.6, -1+Exp[x]}, {x, 0, 1.5}]
```

Ее реализация приведена на рис. 3.2.

Двумерная графика с применением функции `Plot` даже в том простом виде, в каком она представлена примерами, может широко использоваться при ком-

пьютерных технологиях решения математических задач. Покажем это на конкретных примерах.

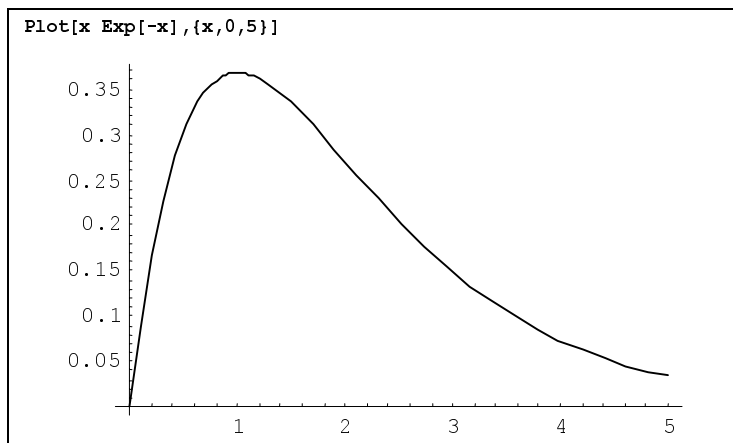


Рис. 3.1. Графическое представление функции $x e^{-x}$

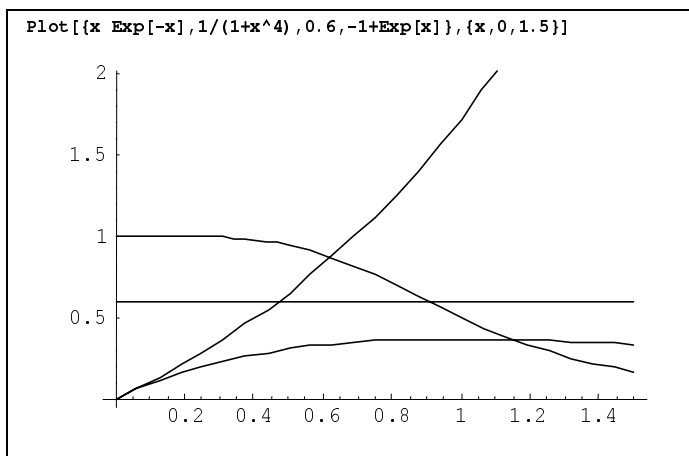


Рис. 3.2. Построение семейства функций на одном графике

Определение области изоляции корня

Большинство нелинейных и трансцендентных уравнений решается только приближенными численными методами. Для этого необходимо знать число корней

уравнения и области их изоляции. Наилучшим способом решения этой задачи является графоаналитический, когда по виду графика находятся число корней и области их изоляции.

Пример 3.3

Необходимо определить число вещественных корней уравнения

$$\ln(4 - 2x) + x^2 - 2 = 0$$

и области их изоляции.

Представим функцию $\ln(4 - 2x) + x^2 - 2 = 0$ в виде графика. Функция `Plot` в данном случае будет иметь вид:

```
Plot[Log[4-2 x]+x^2-2, {x, -2, 2}]
```

Ее реализация показана на рис. 3.3.

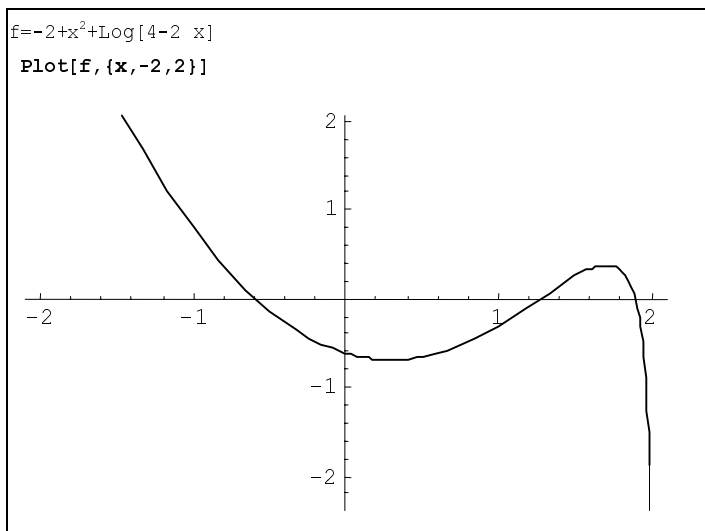


Рис. 3.3. График функции $\ln(4 - 2x) + x^2 - 2$

Из графика рис. 3.3 видно, что уравнение имеет три вещественных корня (три пересечения с осью абсцисс). При этом областями их изоляции могут быть: $-1—0$; $1—1.5$; $1.5—2$. Из графика также видно, что функция имеет три особые точки: минимум, максимум и точку перегиба, координаты которых можно приблизительно определить. Теперь не трудно решить уравнение численными методами, подробно изложенными в главе 8.

Проверка достоверности решения задачи

Проверка достоверности решения задачи с помощью графических образов осуществляется путем сравнения графиков функций истинной и полученной в результате вычислений. Приведем типичный для этого случая пример.

Пример 3.4

Функция разложения e^x в ряд Тейлора вокруг $x = 0$ имеет вид полинома третьей степени: $y = 1 + x + x^2/2 + x^3/6$. Необходимо установить диапазон значений x , для которого такое разложение допустимо.

Решим задачу с помощью визуализации вычислений. Представим функции y и e^x на одном графике и путем их сравнения установим искомый диапазон.

Графики функций $y = 1 + x + x^2/2 + x^3/6$ и e^x , полученные с помощью функции `Plot`, приведены на рис. 3.4.

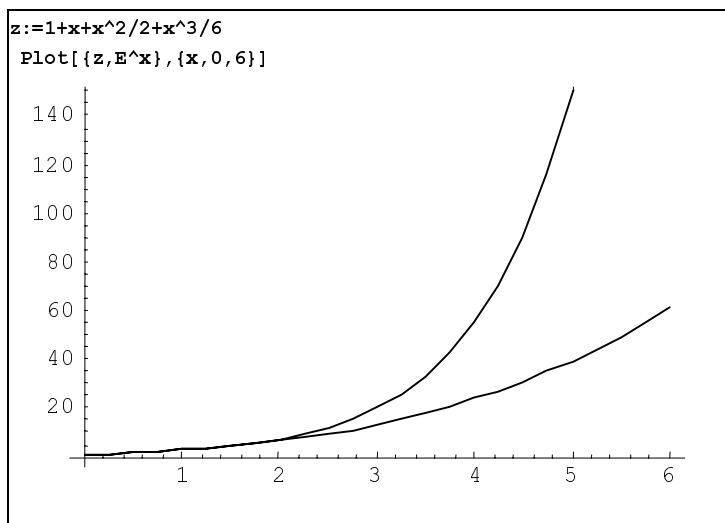


Рис. 3.4. Графики функций $y = 1 + x + x^2/2 + x^3/6$ и e^x

Из графиков видно, что разложение показательной функции в ряд Тейлора третьей степени допустимо в диапазоне x от 0 до 2. С ростом x погрешность функции $y(x)$ существенно возрастает, и такое разложение недопустимо. Необходимо увеличить степень полинома.

Опции функции *Plot*

Рассмотренная выше встроенная функция `Plot` использует опции по умолчанию. Пользователя часто такие графики не устраивают по следующим причинам:

- ☐ плохо выбраны масштабы;
- ☐ отсутствуют обозначения осей;
- ☐ в случае ряда функций кривые не отличаются стилем;
- ☐ отсутствует имя графика.

Для создания графика желаемого вида используются опции функции `Plot`.

Все опции можно получить, если вывести их список командой `Options[Plot]`.

Опции функции `Plot` задаются в следующем виде:

имя опции -> значение опции

Значение опции может быть численным или символьным.

Рассмотрим опции, которые часто необходимы при визуализации вычислений. Такими являются следующие опции:

- ☐ установление масштаба по осям;
- ☐ определение имени осей;
- ☐ определение имени графика;
- ☐ выбор стиля графика.

Для этих целей используются следующие опции:

```
PlotRange -> {ymin, ymax}  
AxesLabel -> {"Tx", "Ty" }  
PlotLabel -> "T"  
Axes -> None
```

В приведенных опциях приняты следующие обозначения:

- ☐ $y_{\min}$, $y_{\max}$ — минимальное и максимальные значения функции по оси y ;
- ☐ T_x , T_y — надписи по осям x и y ;
- ☐ T — текст имени графика.

Опции имеют следующий смысл:

- ☐ `PlotRange -> {ymin, ymax}` — устанавливает масштаб по оси y от $y_{\min}$ до $y_{\max}$ с автоматическим выбором шага;
- ☐ `AxesLabel -> {"Tx", "Ty"}` — устанавливает надписи содержания T_x и T_y по осям x и y ;

□ `PlotLabel -> "T"` — текст названия графика, в том числе и на русском языке;

□ `Axes -> None` — уничтожает оси графика.

Функция `Plot` может иметь одновременно несколько опций, отделяемых друг от друга запятой. Покажем на примерах, как представляется функция `Plot` с опциями:

□ `Plot[f, {x, 1, 5}, PlotRange->{-1, 2}]` — построение графика функции f в диапазоне x от 1 до 5 и масштабе по оси y от -1 до 2;

□ `Plot[f, {x, 0, 7}, PlotLabel->{-1, 2}, AxesLabel->{"x", "y(x)}"]` — построение графика функции f в диапазоне x от 0 до 7 и масштабе y от -1 до 2 с надписями x и $y(x)$ по осям;

□ `Plot[f, {x, 0, 5}, PlotLabel -> "График функции y=sin(x)"]` — построение графика функции $y = \sin x$ с его названием.

Приведем примеры использования функции `Plot` с опциями.

Пример 3.5

Масштаб графика по оси y функции, изображенной на рис. 3.3, выбран неудачно: нужная часть графика занимает только небольшую часть рисунка. Для большего ее выделения уменьшим диапазон y от $y_{\min}=-1$ до $y_{\max}=0.6$ и сузим диапазон x от -1 до 2. Дадим также имя рисунку: "График функции $\text{Log}(4x-2)+x^2-2$ ". При таком виде графика функция `Plot` будет иметь вид:

```
Plot[f, {x, -1, 2}, PlotRange ->{-1, 0.6}, PlotLabel ->
"График функции ln(4x-2)+x^2-2"]
```

Реализация функции показана на рис. 3.5.

Пример 3.6

Необходимо построить график функции

$$f = \frac{e^x}{(1 - e^{-2x})(2 - x)}$$

в диапазоне x от 0 до 5 с именем графика и обозначением осей координат.

В данном случае функция `Plot` будет иметь вид:

```
Plot[f, {x, 0, 5}, AxesLabel->{"x", "y(x)"}, PlotLabel->
"График функции с разрывом непрерывности"]
```

Решение задачи приведено на рис. 3.6.

Приведем еще один пример построения семейства функций на одном графике. Эти графики нам будут необходимы при выборе вида функции интерполяции.

```
f=Log[4-2 x]+x^2-2
-2 + x^2 + Log[4 - 2x]
Plot[f, {x, -1, 2}, PlotRange->{-1, 0.6}, PlotLabel->"График функции
Log(4 x-2)+x^2-2"]
```

График функции $\text{Log}(4x-2)+x^2-2$

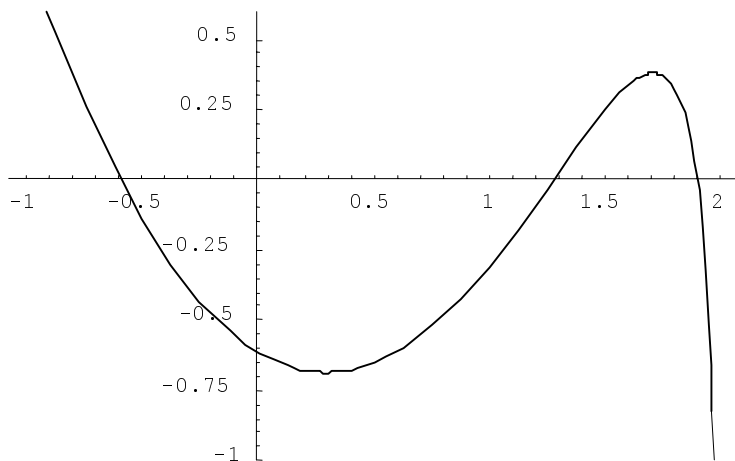


Рис. 3.5. График функции примера 3.5

Пример 3.7

Необходимо представить в виде графиков следующие функции:

- ☐ степенную $y = ax^b$ с параметрами $a = 7$, $b = 2, 0.5, -0.1$;
- ☐ показательную $y = ab^x$ с параметрами $a = 2$, $b = 1, 3, 0.5$;
- ☐ логарифмическую $y = a + b \ln x$ с параметрами $a = 1$, $b = 2, 4, 6$;
- ☐ дробно-линейную $y = x/(a + bx)$ с параметрами $a = 1$, $b = 2, 4, 6$.

В данном случае целесообразно использовать функцию `Plot` для построения на одном графике множества однообразных функций с различными параметрами.

Функции `Plot` в нашем примере будут иметь вид:

- ☐ `Plot[{7 x^2, 7 x^0.5, 7 x^(-0.1)}, {x, 0, 3}, AxesLabel -> {"x", "y(x)"}, PlotLabel->"График степенной функции  $y=a x^b$ ";`

- ❑ `Plot[{2 1^x, 2 3^x, 2 0.5^x},{x, 0, 1}, AxesLabel->{"x", "y(x)"}, PlotLabel->"График показательной функции y=a b^x"];`
- ❑ `Plot[{1+2 Log[x], 1+4 Log[x], 1+6 Log[x]}, {x, 1, 5}, AxesLabel->{"x", "y(x)"}, PlotLabel->"График логарифмической функции y=a+b ln x];`
- ❑ `Plot[{x/(1+2 x), x/(1+4 x), x/(1+6 x)}, {x, 0, 1}, AxesLabel->{"x", "y(x)"}, PlotLabel->"График дробно-линейной функции y=x/(a+b x)].`

Решения примера приведено на рис. 3.7—3.10.

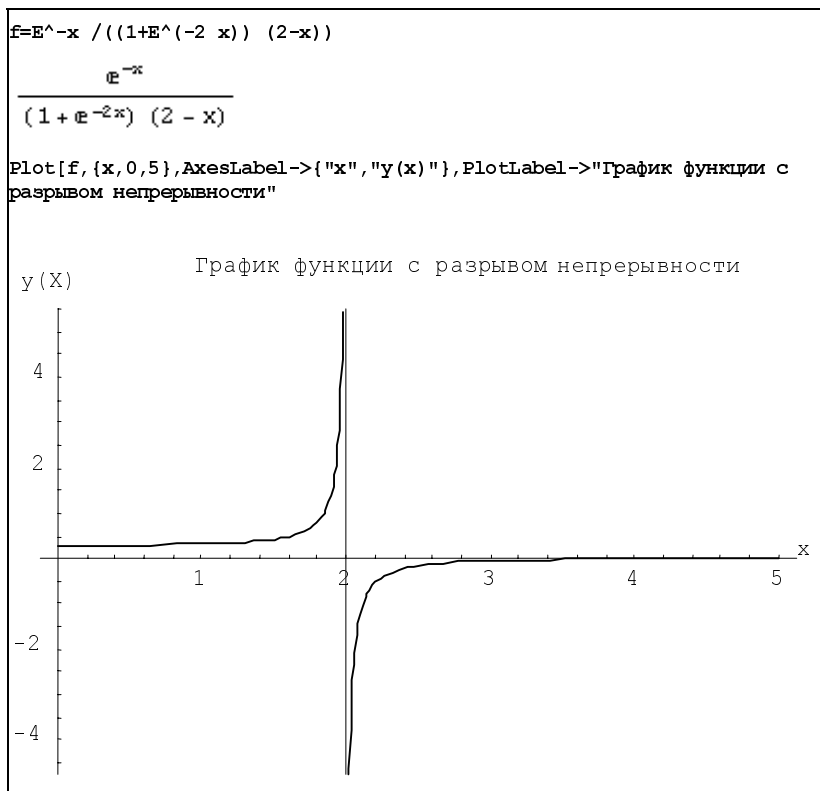


Рис. 3.6. График функции с разрывом непрерывности

```
Plot[{7 x^2, 7 x^0.5, 7 x ^(-0.1)}, {x, 0, 3}, AxesLabel
->{"x", "y(x)"}, PlotLabel-> "График степенной функции y=ax^b"]
```

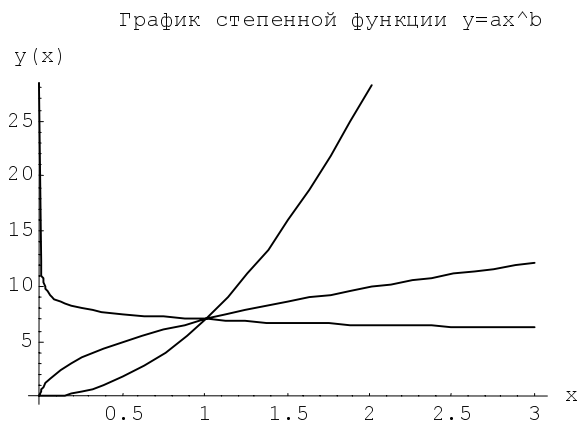


Рис. 3.7. График степенной функции

```
Plot[{2 1^x, 2 3^x, 2 0.5^x}, {x, 0, 1}, AxesLabel-> {"x", "y(x)"},
PlotLabel-> "График показательной функции y=ab^x"]
```

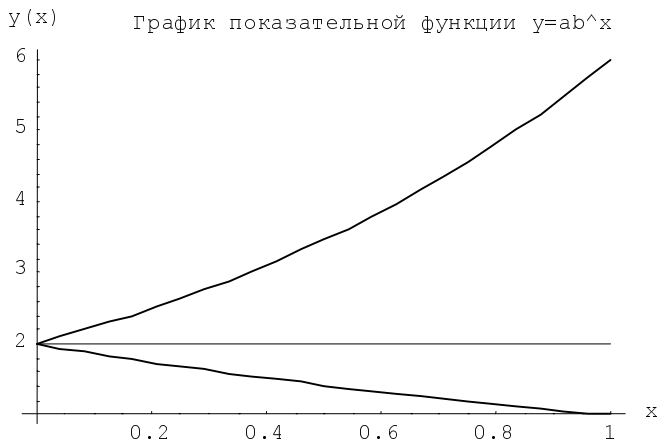


Рис. 3.8. График показательной функции


```
Plot[{1+2 Log[x], 1+4 Log[x], 1+6 Log[x]}, {x, 1, 5}, AxesLabel->
{"x", "y(x)"}, PlotLabel->"График логарифмической функции"]
```

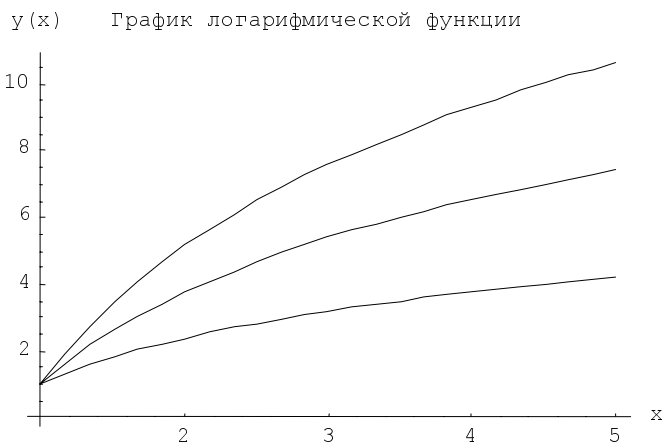


Рис. 3.9. График логарифмической функции

```
Plot[{x/(1+2 x), x/(1+4 x), x/(1+6 x)}, {x, 0, 1}, AxesLabel->
{"x", "y(x)"}, PlotLabel->"График дробно-линейной функции  
y=x/(a+bx)"]
```

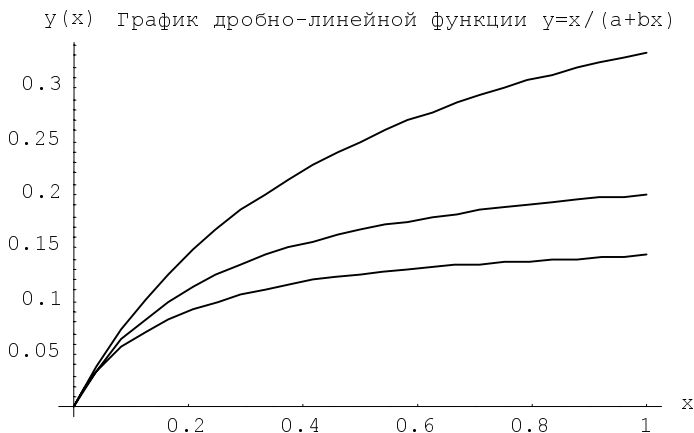


Рис. 3.10. График дробно-линейной функции

3.1.2. Построение точечного графика

Представление графика в виде точек имеет важное практическое значение при решении задачи интерполяции, реализуемой с целью получения математической модели изучаемого объекта или явления. График позволяет выбрать вид функции интерполяции.

В системе Mathematica имеются следующие функции построения точечного графика:

```
ListPlot[{y1, y2, ...}]
```

```
ListPlot[{{x1, y1}, {x2, y2}, ...}]
```

В функциях приняты следующие обозначения:

□ y_i — i -е значение функции $y(x)$;

□ x_i — i -е значение аргумента функции $y(x)$.

Назначение точечных функций:

□ `ListPlot[{y1, y2, ...}]` — выдает точки функции $y(x)$ при значениях x от 0 до n с шагом 1, т. е. 0, 1, 2, ..., n ;

□ `ListPlot[{{x1, y1}, {x2, y2}, ...}]` — выдает точки функции $y(x)$, заданные координатами $x_1, y_1; x_2, y_2; \dots$

Приведем пример построения точечного графика.

Пример 3.8

Результатами эксперимента являются данные, приведенные в табл. 3.1.

Таблица 3.1. Экспериментальные данные

x	1	2.3	3.7	5.2	7.3	9.8	12	14.5	18
y	2.3	4.8	7.3	12	15.6	11.8	6.8	5	2.1

Необходимо построить точечный график, пользуясь встроенной функцией `ListPlot`.

В данном случае функцию $y(x)$ можно представить в виде следующей матрицы:

```
f={{1, 2.3}, {2.3, 4.8}, {3.7, 7.3}, {5.2, 12}, {7.3, 15.6}, {9.8, 11.8}, {12, 6.8}, {14.5, 5}, {18, 2.1}}
```

Тогда функция `ListPlot` будет иметь вид:

```
ListPlot[f, AxesLabel->{"x", "y(x)"}, PlotStyle->PointSize[0.02]]
```

Функция имеет две опции: обозначение осей `AxesLabel` и размер точек `PlotStyle`.

Решение задачи приведено на рис. 3.11.

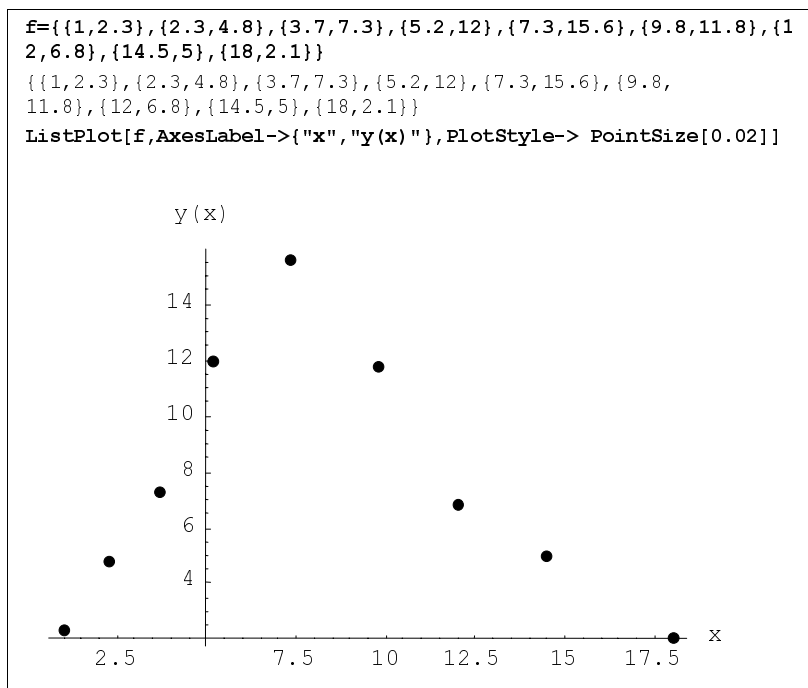


Рис. 3.11. Построение точечного графика

Точечная визуализация позволяет осуществлять проверку правильности решения задачи интерполяции. Сущность этого способа состоит в следующем.

Результатом эксперимента, как правило, является функция, представленная в табличной форме. По данным таблицы путем интерполяции определяется математическая модель изучаемого объекта, представляющая собой математическое выражение типа формулы.

Проверить адекватность модели можно путем визуализации данных эксперимента и математической модели. Если графики этих функций совпадают, то можно быть уверенным в достоверности математической модели.

Так как графики в системе Mathematica являются объектами, то над ними можно производить математические действия, в том числе и объединять в один график, т. е. функции представлять на одном графике.

Объединение графиков можно осуществлять с помощью функции `Show[r1, r2]`, где `r1` — график точечный, полученный по данным таблицы, `r2` — график функции интерполяции (математической модели).

Приведем пример проверки адекватности модели путем визуализации решений.

Пример 3.9

Пусть в результате эксперимента получена функция в виде табл. 3.2, а путем интерполяции — математическая модель в виде следующей формулы $y = 2.5x^{1.3}$.

Таблица 3.2. Экспериментальные данные объекта моделирования

x	1	1.5	2	2.5	3	3.5	4	4.5	5
y	2.5	4.7	5.8	8.7	10	13	15	18	20

Необходимо проверить достоверность модели, построив и объединив графики двух функций: исходную точечную и математическую модель.

Воспользуемся для этой цели функциями `ListPlot`, `Plot` и `Show`. Эти функции для нашего примера имеют вид:

```
f={{1, 2.5},{1.5, 4.7},{2, 5.8},{2.5, 8.7},{3, 10},{3.5,13},{4,
15},{4.5,18},{5,20}}
r1:=ListPlot[f,AxesLabel->{"x","y(x)"},PlotStyle->PointSize[0.02]]
r2:=Plot[2.5 x^1.3,{x,1,5}]
Show[r1,r2,PlotRange->{2,21}]
```

Построение графиков с помощью этих функций показано на рис. 3.12.

Из рис. 3.12 видно, что кривая математической модели хорошо согласуется с экспериментальными данными и может быть моделью изучаемого явления.

Компьютерные технологии интерполяции подробно рассмотрены в *главе 12*.

Система Mathematica позволяет строить точечные графики множества объектов. Для этой цели имеются встроенные функции подпакета `MultipleListPlot`, вызываемого функцией: `<<Graphics`MultipleListPlot``.

Подпакет `MultipleListPlot` содержит много различных функций, одна из которых имеет следующий вид:

```
MultipleListPlot[y1, y2, ...]
```

где `y1`, `y2`, ... — списки, которые могут быть представлены в виде вектора, или матрицы, или в виде табулируемой функции.

```
f={{1, 2.5},{1.5, 4.7},{2, 5.8},{2.5, 8.7}, {3,
10},{3.5,13},{4, 15},{4.5,18},{5,20}}
{{1,2.5},{1.5,4.7},{2,5.8},{2.5,8.7},{3,10},{3.5,13},{4,15},{4.
5,18},{5,20}}
r1:=ListPlot[f,AxesLabel->{"x","y(x)"},PlotStyle->
PointSize[0.02];
r2:=Plot[2.5 x^1.3,{x,1,5}];
Show[r1,r2,PlotRange->{2,21}]
```

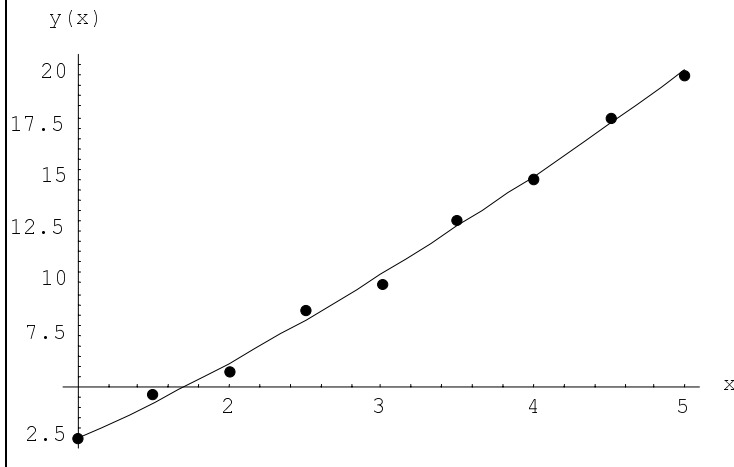


Рис. 3.12. Объединение двух графиков

Приведем пример построения множества графиков при различном представлении данных.

Пример 3.10

Представить на одном графике следующие три функции:

$$y_1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$$

$$y_2 = \{(2, 4), \{3, 6\}, \{4, 8\}, \{5, 10\}, \{6, 12\}, \{7, 14\}, \{8, 16\}, \{9, 18\}\}$$

$$y_3 = \text{Table}[\{x, \text{Exp}[x]\}, \{x, 0, 4, 0.5\}]$$

В данном случае функция `MultipleListPlot` будет иметь вид:

```
MultipleListPlot[y1, y2, y3]
```

Решение примера приведено на рис. 3.13.

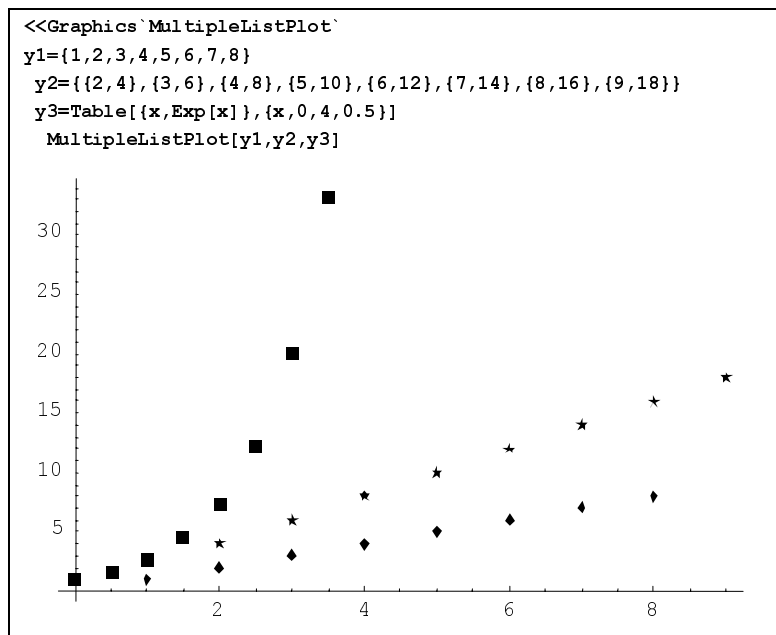


Рис. 3.13. График множества (трех) объектов

Из рис. 3.13 видно, что графики выведены в виде малых различных фигурок, позволяющих различать исходные объекты (функции).

Функция `MultipleListPlot` имеет опцию `PlotJoined`, которая позволяет соединять точки графика линиями. Опция имеет вид:

`PlotJoined -> A`

где `A` — `True` или `False`. `True` означает, что точки нужно соединять линиями, `False` — не нужно.

Пример 3.11

Необходимо изобразить точечные графики примера 3.10, соединив линиями объекты y_1 , y_2 . Функцию y_3 оставить в прежнем виде.

В данном случае функция `MultipleListPlot` будет иметь вид:

```
MultipleListPlot[y1, y2, y3, PlotJoined->{True, True, False}]
```

Решение примера представлено на рис. 3.14.

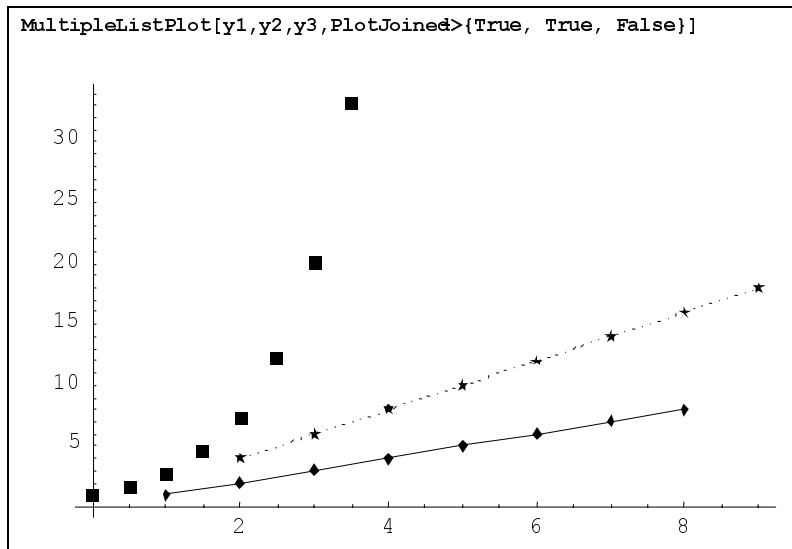


Рис. 3.14. Графики функций, построенные с помощью опции `PlotJoined->A`

3.1.3. Выбор стиля графика

Необходимый стиль графика можно образовать с помощью опций и директив.

Опция *PlotStyle*

Опция `PlotStyle` позволяет выбрать цвет линий и их толщину. Она имеет следующие варианты:

`PlotStyle -> {Hue[c1], Hue[c2], ...}`

`PlotStyle -> {GrayLevel[k1], GrayLevel[k2], ...}`

где:

□ $c_1, c_2, \dots$ — коды цвета линий соответствующих функций;

□ $k_1, k_2, \dots$ — коды толщин линий соответствующих функций.

Коды c_i и k_i выбираются из диапазона 0—1.

Пример 3.12

Необходимо представить функции $y_1 = xe^{-x}$, $y_2 = x^2$ в виде графика в диапазоне x от 0 до 5. Линии графика должны отличаться толщиной.

В данном случае функция Plot имеет вид:

```
Plot[{x Exp[x], x^2}, {x, 0, 5}, PlotStyle -> {GrayLevel[0.2],
GrayLevel[0.8]}]
```

Решение приведено на рис. 3.15.

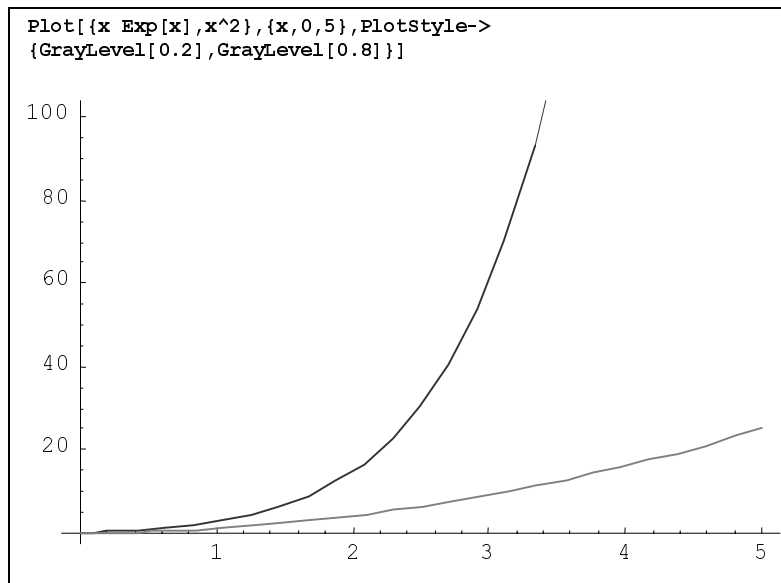


Рис. 3.15. Решение примера 3.12

Толщина линий функций $x e^{-x}$ и x^2 на рис. 3.15 плохо различима. На экране монитора она хорошо видна.

Стиль графика можно установить с помощью директив, которые позволяют формировать линии разных форм.

Система имеет следующие директивы опции PlotStyle:

```
Dashing[{r1, r2, ...}]
```

```
AbsoluteDashing[{d1, d2, ...}]
```

```
PointSize[d]
```

```
AbsolutePointSize[d]
```

```
Thickness[k]
```

```
AbsoluteThickness[k]
```


Директивы позволяют выбрать следующие стили графика:

- ❑ `Dashing[{r1, r2, ...}]` — устанавливает величину последующих, циклически повторяемых, пунктиров пунктирной линии; значение длины r_1 задается в долях ширины линии графика;
- ❑ `AbsoluteDashing[{d1, d2, ...}]` — устанавливает величину последующих, циклически повторяемых, штрихов пунктирной линии; значение длины d_1 задается в пикселах;
- ❑ `PointSize[d]` — задает график в виде кружков диаметром d , измеряемых в долях общей ширины графика;
- ❑ `AbsolutePointSize[d]` — задает график в виде кружков диаметром d , измеряемых в пикселах;
- ❑ `Thickness[r]` — устанавливает толщину r линий графика, как долю его полной ширины;
- ❑ `AbsoluteThickness[r]` — устанавливает толщину r линий графика в пикселах.

Приведем примеры построения графиков с использованием функции `PlotStyle` с директивами.

Пример 3.13

Необходимо построить в различных стилях графики функций:

$$y_1 = \ln(4 - 2x + x^2 - 2), \quad y_2 = 1/(1 + x^4),$$

используя директивы функции `PlotStyle`. Следует реализовать два вида стилей: пунктирные линии и графики с линиями различной толщины.

В случае графиков в виде пунктирных линий функцию `Plot` представить в виде:

```
Plot[{Log[4-2 x]+x^2-2, 1/(1+x^4)}, {x, -2, 2}, PlotStyle ->
{AbsoluteDashing[1,3], AbsoluteDashing[1,10]}]
```

Для случая представления графиков с разной толщиной линий функцию `Plot` представить в виде:

```
Plot[{Log[4-2 x]+x^2-2, 1/(1+x^4)}, {x, -2, 2}, PlotStyle ->
{Thickness[0.007], Thickness[0.001]}]
```

Графики функций для этих случаев приведены на рис. 3.16 и 3.17.

Опции и директивы позволяют создавать линии различного вида, что дает возможность с помощью одной функции получать графики с большим числом функций различного стиля.

```
Plot[{Log[4-2 x]+x^2-2,1/(1+x^4)}, {x,-2,2},PlotStyle->
{AbsoluteDashing[{1,3}],AbsoluteDashing[{1,10}]}]
```

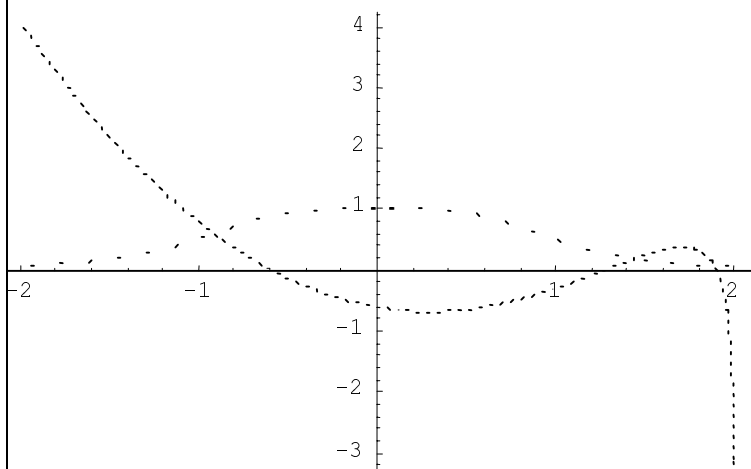


Рис. 3.16. Представление функций в виде пунктирных линий

```
Plot[{Log[4-2 x]+x^2-2,1/(1+4x^4)}, {x,-2,2}, PlotStyle->
{Thickness[0.007],Thickness[0.001]}]
```

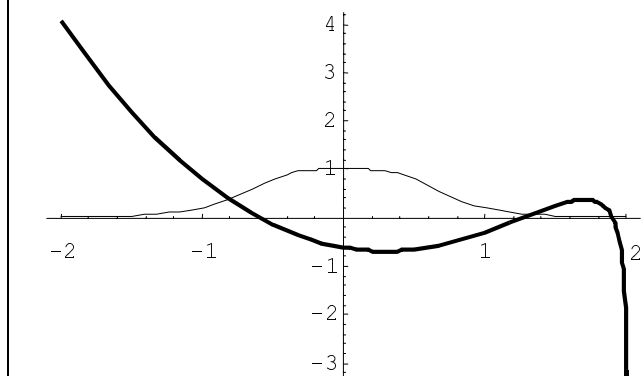


Рис. 3.17. Представление функций в виде линий разной толщины

Пример 3.14

Необходимо построить на одном графике в диапазоне x от 1 до 10 графики функций $y_1 = \sin x$, $y_2 = \ln x$, $y_3 = (x-1)/(x+1)$ со стилями разного вида.

На рис. 3.18 представлены графики функций со стилями: пунктирные линии ($\sin x$), без стилей ($\ln x$), линии разной толщины ($(x-1)/(x+1)$).

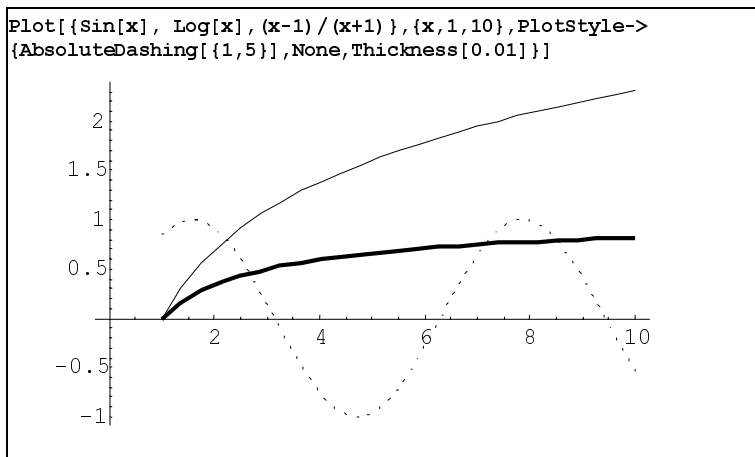


Рис. 3.18. Графики с разными стилями

3.1.4. Обозначение кривых на графике множества функций

Обозначение кривых в поле графика существенно повышает наглядность графического представления функций. Для этого система Mathematica имеет ряд опций. Они находятся в подпакете `Legend` пакета `Graphics`, к которому обращаются так:

```
<< Graphics`Legend`
```

В подпакете имеется следующая опция обозначения кривых:

```
PlotLegend -> {"T1", "T2", ...}
```

где "T₁", "T₂", ... — строки, обозначающие функции $y_1, y_2, \dots$

Для применения этой опции необходимо, чтобы кривые функций $y_1, y_2, \dots$ отличались между собой. Это можно осуществить с помощью опций стилей.

Приведем пример.

Пример 3.15

Пусть необходимо на одном рисунке представить в виде графиков следующие функции: $y_1 = \sin x$, $y_2 = \ln x$ в диапазоне x от $x=1$ до $x=10$. Обозначения кривых выполнить в виде математических выражений. Кривые должны отличаться стилями.

В нашем случае функция `Plot` будет иметь вид:

```
Plot[{Sin[x], Log[x]}, {x, 1, 10}, AxesLabel -> {"x", "y(x)"},  
PlotStyle -> {AbsoluteDashing[{1, 5}], Thickness[0.01]}, PlotLegend ->  
{ "Sin(x)", "Ln(x)" }]
```

Результаты решения приведены на рис. 3.19.

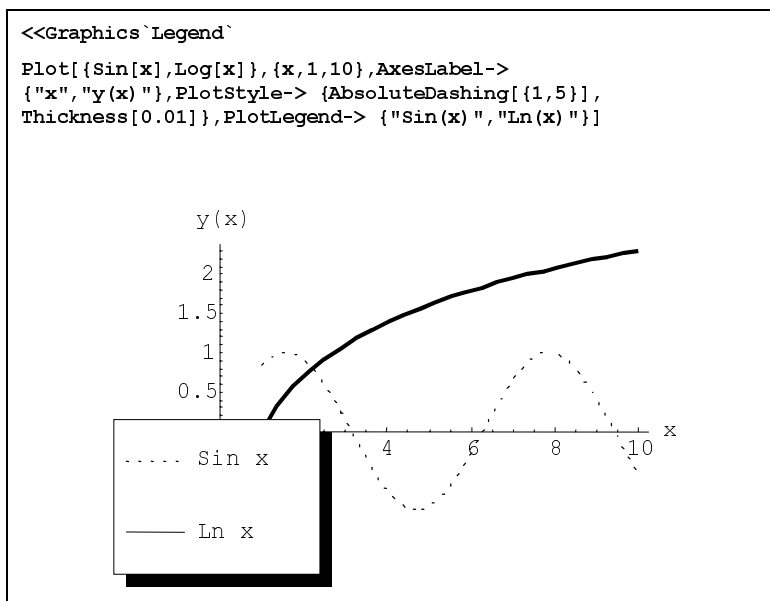


Рис. 3.19. Обозначение кривых математическими функциями

Из рис. 3.19 видно, что функция `Plot` установила имя осей, представила функции в виде различных стилей и обозначила кривые математическими функциями. Теперь ясно, какой функции принадлежит данная кривая.

В подпакете `Legend` имеется много других опций обозначения кривых. Они позволяют задавать цвет и форму легенды, устанавливать ее в нужном месте. Это украшает рисунок, но не меняет его содержания. С ними можно познакомиться в [10] или в справочной системе (меню **Help**).

3.1.5. Графики специальных типов

В данном разделе рассматриваются графики следующих типов:

- ☐ в логарифмическом масштабе;
- ☐ в полярной системе координат;
- ☐ в виде гистограмм.

Функции, позволяющие строить графики специальных типов, находятся в подпакете `Graphics`, обращение к которому имеет вид:

```
<< Graphics`Graphics`
```

Построение графиков в логарифмическом масштабе

Построение графиков в логарифмическом масштабе осуществляется с помощью следующих функций:

```
LogPlot[f, {x, xmin, xmax}]
LogLinearPlot[f, {x, xmin, xmax}]
LogLogPlot[f, {x, xmin, xmax}]
LogListPlot[f, {{x1, y1}, {x2, y2}, ...}]
LogLinearListPlot[f, {{x1, y1}, {x2, y2}, ...}]
LogLogListPlot[f, {{x1, y1}, {x2, y2}, ...}]
```

В функциях приняты следующие обозначения:

- ☐ f — функция, график которой создается;
- ☐ $x_{\min}, x_{\max}$ — диапазон изменения аргумента x функции f ;
- ☐ x_i, y_i — координаты точки i функции f , заданной в виде матрицы (таблицы), $i = 1, 2, \dots$

Назначение функций:

- ☐ функция `LogPlot[f, {x, xmin, xmax}]` — создает линейно-логарифмический график функции f в диапазоне x от $x_{\min}$ до $x_{\max}$ (полулогарифмический масштаб с логарифмическим масштабом по оси y);
- ☐ функция `LogLinearPlot[f, {x, xmin, xmax}]` — создает логарифмический линейный график функции f в диапазоне x от $x_{\min}$ до $x_{\max}$ (логарифмический масштаб по оси x);
- ☐ функция `LogLogPlot[f, {x, xmin, xmax}]` — строит график функции f в логарифмическом масштабе по обоим осям;
- ☐ функции `LogListPlot[f, {{x1, y1}, {x2, y2}, ...}]`, `LogLinearListPlot[f, {{x1, y1}, {x2, y2}, ...}]`, `LogLogListPlot[f, {{x1, y1}, {x2, y2}, ...}]` аналогичны трем первым. Разница лишь в том, что они строят точечные графики.

Приведем примеры построения графиков в логарифмическом масштабе.

Пример 3.16

Необходимо построить в полулогарифмическом масштабе график функции

$$y = x^8 e^x + \sinh x$$

в диапазоне x от $x = 1$ до $x = 5$.

В этом случае функция `LogPlot` будет иметь вид:

```
LogPlot[x^8 Exp[x]+ Sinh[x], {x, 1, 5}]
```

График функции приведен на рис. 3.20.

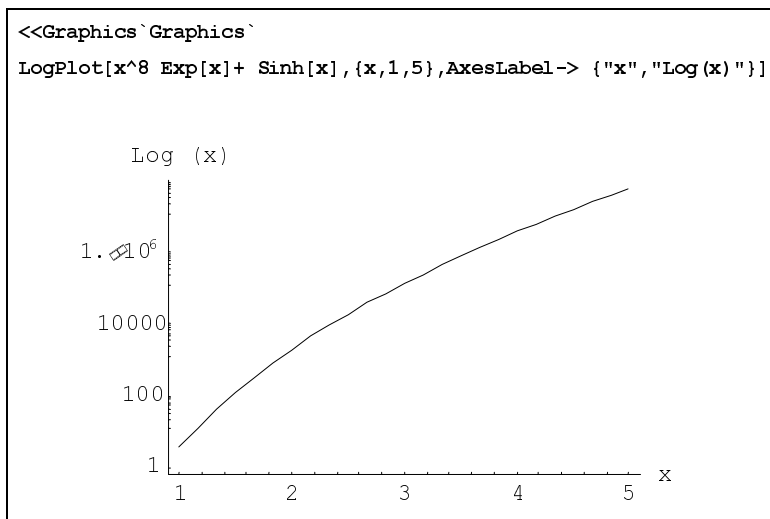


Рис. 3.20. График функции примера 3.16

Из рис. 3.20 видно, что логарифмический масштаб выбран по оси y .

Пример 3.17

Функция задана в виде следующей матрицы:

```
{{1, 1}, {2, 8}, {3, 27}, {4, 64}, {5, 125}, {6, 216}, {7, 343}, {8, 512},  
{9, 729}, {10, 1000}}
```

Необходимо создать точный график функции в логарифмическом масштабе по оси y .

В данном случае функция `LogPlot` будет иметь вид:

```
LogListPlot[f, AxesLabel->{"x", "Log(x)"}]
```

Решение приведено на рис. 3.21.

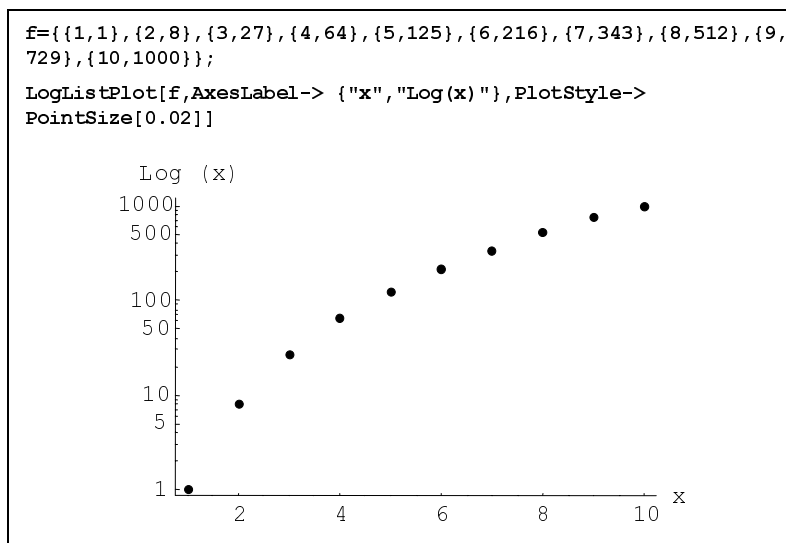


Рис. 3.21. Точечный логарифмический график функции примера 3.17

В целях большей наглядности график точечной функции рис. 3.21 построен в стиле `PointSize`.

Построение графиков в полярной системе координат

Система имеет следующие функции для представления графиков в полярной системе координат:

```
PolarPlot[f, {t, tmin, tmax}]
```

```
PolarPlot[{f1, f2, ... }, {t, tmin, tmax}]
```

```
PolarPlot[{r1, r2, ...}, {t, tmin, tmax}]
```

В этих функциях приняты следующие обозначения:

- f — функция, график которой строится;
- f_i — i -я функция многофункционального графика;
- $t_{\min}, t_{\max}$ — значение угла t полярной системы координат от $t_{\min}$ до $t_{\max}$;
- r_i — i -й радиус списка радиусов полярной системы координат.

Приведенные функции строят следующие графики в полярной системе координат:

- ❑ `PolarPlot[f, {t, tmin, tmax}]` — строит график положения конца вектора f при изменении угла t от $t_{\min}$ до $t_{\max}$;
- ❑ `PolarPlot[{f1, f2, ...}, {t, tmin, tmax}]` — строит графики множества функций при изменении угла t от $t_{\min}$ до $t_{\max}$;
- ❑ `PolarPlot[{r1, r2, ...}, {t, tmin, tmax}]` — строит график списка r , равномерно изменяя угловую координату.

Приведем примеры построения графиков в полярной системе координат.

Пример 3.18

Необходимо представить функцию $(1+2\sin x)/(1+2\cos x)$ в виде графика в полярной системе координат при изменении x в диапазоне от 0 до 2π .

В данном случае функция `PolarPlot` будет иметь вид:

```
PolarPlot[(1+Sin[x])/(1+Cos[x]), {x, 0, 2 Pi}]
```

Решение приведено на рис. 3.22.

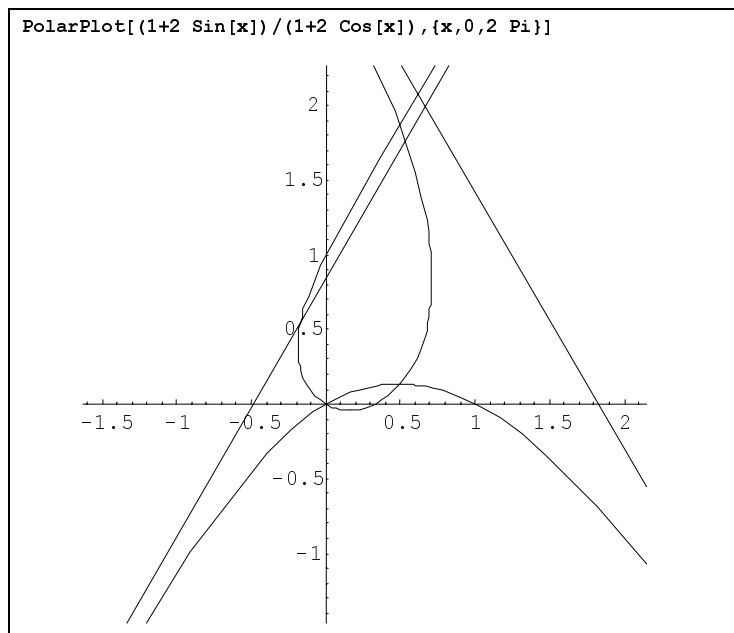


Рис. 3.22. График функции примера 3.18

Пример 3.19

Необходимо построить в полярной системе координат семейство следующих функций:

$$-3\sqrt{\frac{x+3}{3-x}}, \quad 3\sqrt{\frac{x+3}{3-x}}, \quad e^{-2x}, \quad \sin(5x+1),$$

если $x_{\min} = -3$, $x_{\max} = 3$.

Решение приведено на рис. 3.23.

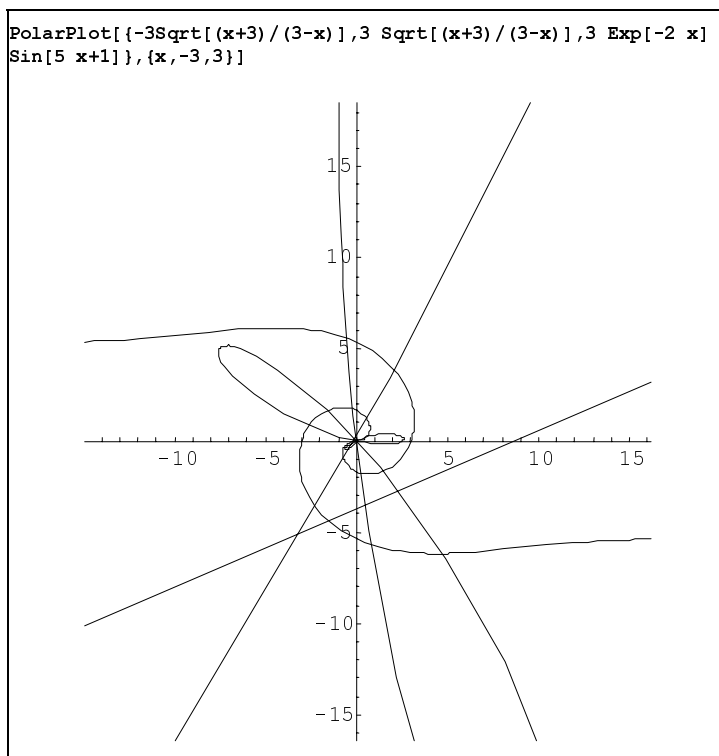


Рис. 3.23. График семейства функций примера 3.19

Построение графиков в виде гистограмм

Столбцовые диаграммы

Построение столбцовых диаграмм осуществляется с помощью следующих функций:

```
BarChart[{c1, c2, ...}]  
StackedBarChart[{c1, c2, ...}]  
PercentileBarChart[{c1, c2, ...}]  
GeneralizedBarChart[{c1, c2, ...}]
```

Приведенные функции позволяют строить следующие столбцовые диаграммы.

- ❑ `BarChart[{c1, c2, ...}]` — строит столбцовую диаграмму по данным спискам $c_1, c_2, \dots$. Столбцы располагаются рядом и отличаются цветом. Списки данных могут иметь разную длину. В этом случае отсутствующие данные считаются нулевыми (пробел на графике). Если в списке имеются отрицательные данные, то они строятся как столбцы, обращенные вниз. Длина столбца соответствует величине данного.
- ❑ `StackedBarChart[{c1, c2, ...}]` — строит столбцовую диаграмму, располагая одни столбцы над другими. Столбцы отличаются цветом. Длина столбцов пропорциональна сумме i -х элементов списков.
- ❑ `PercentileBarChart[{c1, c2, ...}]` — строит столбцовую диаграмму так же, как и функция `StackedBarChart[{c1, c2, ...}]`, но длина столбцов исчисляется в процентах. За 100% принимается длина максимального столбца, расположенного выше горизонтальной оси, за -100% — ниже горизонтальной оси.
- ❑ `GeneralizedBarChart[{c1, c2, ...}]` — строит столбцовую диаграмму с заданной шириной и высотой столбцов.

Рассмотренные функции имеют опции, которые существенно меняют вид диаграмм. Они имеют цветовые эффекты, меняют расположение диаграммы (горизонтальное, вертикальное), осуществляют обвод столбцов и т. д.

Приведем примеры построения столбцовых диаграмм.

Пример 3.20

Пусть необходимо построить столбцовые диаграммы результатов расчета, представленных в виде следующей матрицы: $\{\{-2, 6, 3, 10, -1\}, \{4, 6, -8, 10\}\}$. Представить столбцовые диаграммы в виде расположения столбцов рядом и друг над другом с абсолютным и процентным отсчетами.

В данном случае должны быть построены три столбцовые диаграммы. Функции построения диаграмм будут иметь вид:

- ❑ случай расположения столбцов рядом:
`BarChart[{-2, 6, 3, 10, -1}, {4, 6, -8, 10}]`
- ❑ случай расположения столбцов друг над другом с абсолютным отсчетом:
`StackedBarChart[{2, 6, 3, 10, -1}, {4, 6, -8, 10}]`

□ случай расположения столбцов друг над другом с процентным отсчетом:

`Percentile BarChart[{2, 6, 3, 10, -1}, {4, 6, -8, 10} ]`

Результаты реализации этих функций приведены на рис. 3.24—3.26.

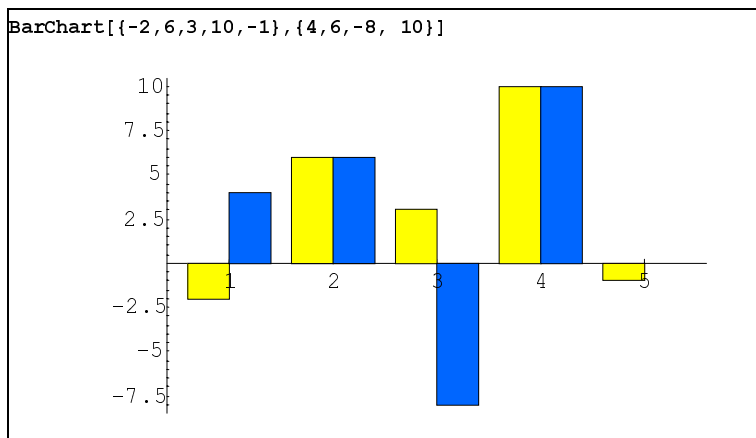


Рис. 3.24. Столбцовая диаграмма расположения столбцов рядом

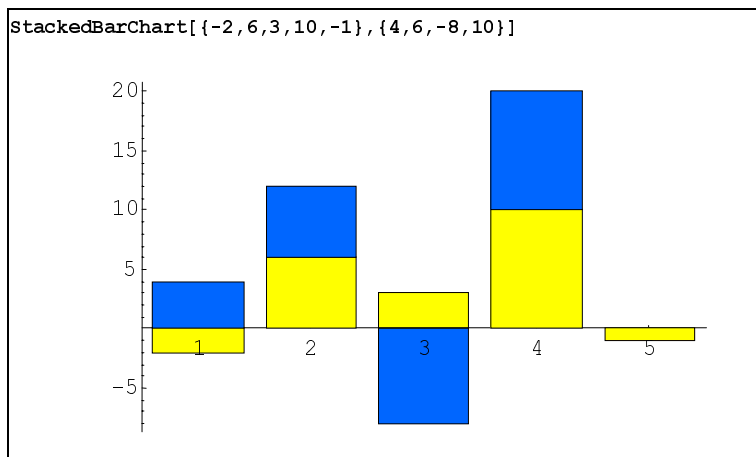


Рис. 3.25. Столбцовая диаграмма с расположением столбцов друг над другом и абсолютным отсчетом

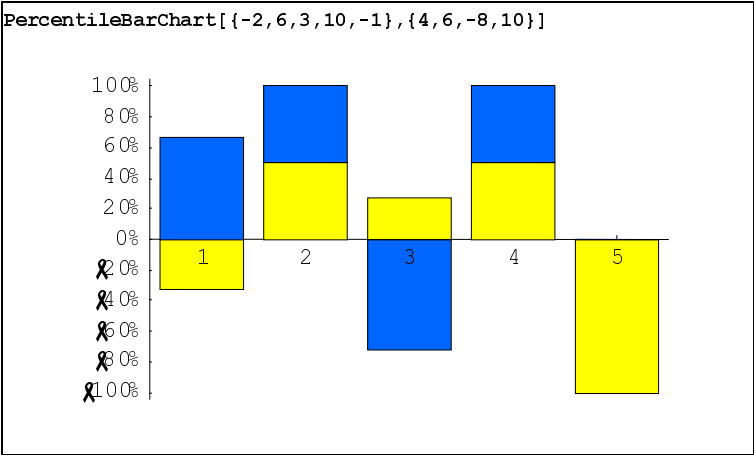


Рис. 3.26. Столбцовая диаграмма с расположением столбцов друг над другом и процентным отсчетом

Круговые диаграммы

Для построения круговых диаграмм система предоставляет ряд функций. Одной из основных является следующая функция:

```
PieChart[D]
```

где D — данные в виде списка, характеризующие долю круговой диаграммы.

Вид диаграммы задается опциями. Весь список опций можно получить командой Options[PieChart]. Приведем только две из них:

- ☐ PieStyle->>{GrayLevel[s₁], GrayLevel[s₂], ...} — построение круговой диаграммы с раскраской чередующихся секторов. Сектора по размеру соответствуют функции PieChart[D];
- ☐ PieLabels->{"I₁", "I₂"}, где I — имя i-го сектора.

Приведем примеры построения круговых диаграмм.

Пример 3.21

Пусть круговая диаграмма состоит из 5 секторов, список данных о которых имеет вид: {1, 3, 7, 9, 18}.

Необходимо представить данные в виде круговой диаграммы с раскраской секторов.

В данном примере функция `PieChart` будет иметь вид:

```
PieChart[{1, 3, 7, 9, 18}, PieStyle->{GrayLevel[0.7], GrayLevel[0.9]}]
```

Реализация этой функции приведена на рис. 3.27.

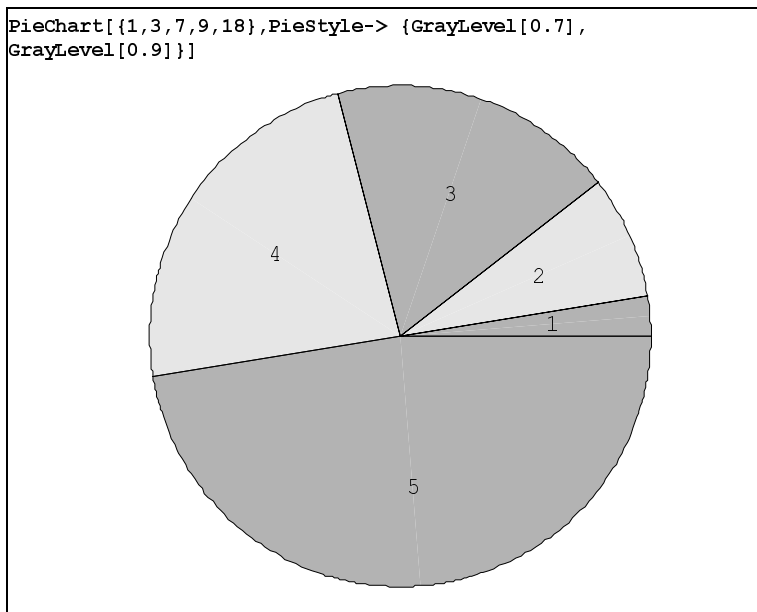


Рис. 3.27. Круговая диаграмма с раскраской чередующихся секторов

Пример 3.22

Пусть круговая диаграмма состоит из четырех секторов, каждый из которых является рейтингом автомобиля: "Москвич", "Жигули", "Волга", "Мерседес".

Необходимо результаты рейтинга представить в виде круговой диаграммы с надписями автомобилей в ее секторах.

В данном случае функция `PieChart` будет иметь вид:

```
PieChart[{3, 7, 9, 15}, PieLabels -> {"Москвич", "Жигули", "Волга",
"Мерседес"}, PlotLabel -> "Результаты рейтинга"]
```

Реализация функции приведена на рис. 3.28.

```
PieChart[{3, 7, 9, 15}, PieLabels->
{"Москвич", "Жигули", "Волга", "Мерседес"}, PlotLabel->
"Результаты рейтинга"]
```



Рис. 3.28. Результаты рейтинга автомобилей

3.2. Трехмерная графика

Трехмерная графика — это графика функций двух переменных. Трехмерные графики могут быть следующих видов: контурные, графики поверхностей, графики фигур.

Система Mathematica имеет богатые возможности создания объектов трехмерной графики. Среди универсальных программных средств символьной математики она является лидером в области графики. Большое число встроенных графических функций, опций и директив к ним позволяют создавать графики практически любого вида.

Графические возможности системы таковы, что они выходят далеко за пределы визуализации вычислений.

Графика системы Mathematica имеет самостоятельное значение.

Наша книга посвящена компьютерным технологиям решения математических задач. Для нас графика является средством визуализации вычислений. Для этих

целей в большинстве случаев достаточно двумерной графики. В связи с этим методы создания объектов трехмерной графики здесь не излагаются. Покажем только на нескольких примерах возможности системы по созданию объектов трехмерной графики. При необходимости читатель может самостоятельно изучить технологию построения графических объектов. Эта задача во много раз проще, чем, например, изучить компьютерные технологии решения уравнений. Здесь не нужно знать алгоритмов решения задачи. Достаточно помнить графические встроенные функции, опции и директивы к ним, чтобы без труда создать желаемый трехмерный график.

3.2.1. Создание контурных графиков

Контурные графики позволяют представлять поверхности на плоскости. Система Mathematica имеет около 20 функций и опций построения контурных графиков. Вот одна из них:

`ContourPlot[f, {x, xmin, xmax}, {y, ymin, ymax}]`

где:

- ☐ f — функция переменных x, y ;
- ☐ $x_{\min}, x_{\max}$ — диапазон изменения переменной x ;
- ☐ $y_{\min}, y_{\max}$ — диапазон изменения переменной y .

Функция `ContourPlot` создает контурный график функции $f(x, y)$.

Пример 3.23

Необходимо представить функцию

$$y = x \sin(x + y) + e^{x+y} \cos(x + y)$$

в виде контурного графика в диапазоне изменения x от $x_{\min} = -3$ до $x_{\max} = 1$ и y от $y_{\min} = -3$ до $y_{\max} = 3$.

В данном случае функция `ContourPlot` будет иметь вид:

`ContourPlot[x Sin[x+y]+Exp[x+y] Cos[x+y], {x, -3, 1}, {y, -3, 3}]`

Реализация функции показана на рис. 3.29.

3.2.2. Построение графиков поверхностей

Для построения графиков поверхностей система имеет графическую функцию `Plot3D`, которая имеет вид:

`Plot3D[f, {x, xmin, xmax}, {y, ymin, ymax}]`

Обозначения имеют тот же смысл, что и в функции `ContourPlot`.

```
ContourPlot[x Sin[x+y]+Exp[x+y] Cos[x+y] ,{x,-3,1},{y,-3,3}]
```

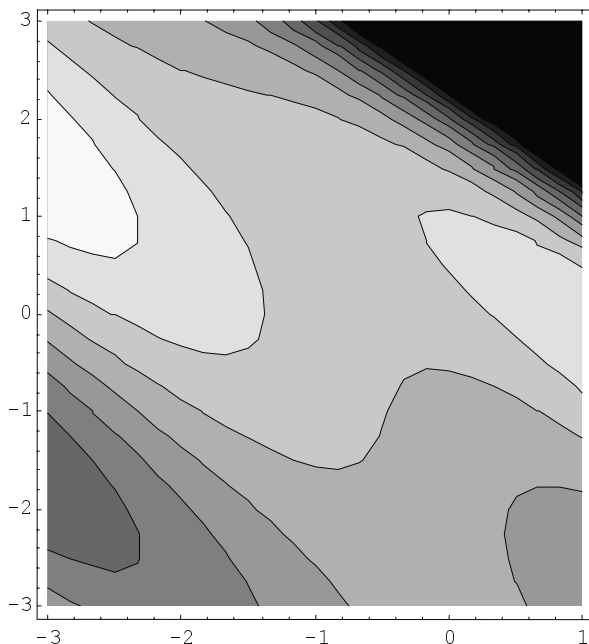


Рис. 3.29. Контурный график функции примера 3.23

Функция `Plot3D` строит пространственный график функции двух переменных $z = f(x, y)$. Для придания видимости объема фигуры применяется способ аксонометрической поверхности.

Покажем действие функции `Plot3D` на примере.

Пример 3.24

Необходимо построить график поверхности, описываемой функцией:

$$z = xe^{x+y} \sin(x+y)/\cos(x+y)$$

при изменении x от $x_{\min}=-3$ до $x_{\max}=1$, а y от $y_{\min}=-3$ до $y_{\max}=3$.

Решение приведено на рис. 3.30.


```
Plot3D[x Exp[x+y] Sin[x+y]/Cos[x+y], {x, -3, 1}, {y, -3, 3}]
```

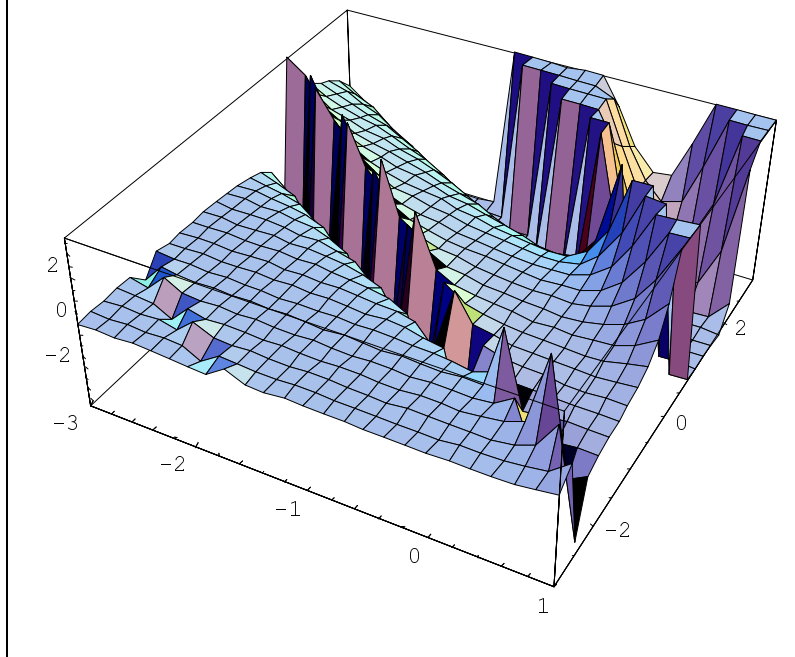


Рис. 3.30. График поверхности примера 3.24

3.2.3. Построение фигур

Наиболее интересные фигуры создаются с помощью графической функции `ParametricPlot3D`, которая имеет несколько модификаций. Одна из них такова:

```
ParametricPlot3D[f1, f2, f3, {x, xmin, xmax}, {y, ymin, ymax}]
```

где:

- f_i — i -я функция, задающая координаты точки x_i, y_i ;
- $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ — минимальное и максимальные значения координат точек.

Данная функция строит график на основании параметрического задания всех трех функций. Покажем результаты ее действия на примерах.

Пример 3.25

Необходимо построить график объекта, параметрические функции которого имеют вид:

$$f_1 = x^2 \cos x (5 + \cos(x + y)),$$

$$f_2 = x^2 \sin x (5 + \cos(x + y)),$$

$$f_3 = x^2 \sin(x + y).$$

Диапазон изменения переменных: для x — от 0 до 3π , для y — от 0 до 2π .

В данном случае функция `ParametricPlot3D` будет иметь вид:

```
ParametricPlot3D[{x^2 Cos[x] (5+Cos[x+y]), x^2 Sin[x] (5+Cos[x+y]),  
x^2 Sin[x+y]}, {x, 0, 3 Pi}, {y, 0, 2 Pi}]
```

Реализация функции показана на рис. 3.31.

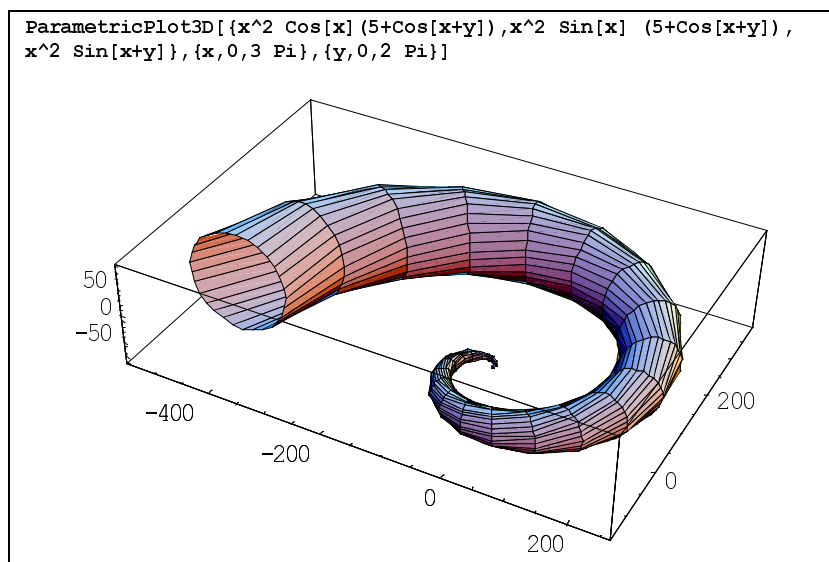


Рис. 3.31. График функции примера 3.25

Приведем еще один типичный пример построения фигуры

Пример 3.26

Необходимо построить график фигуры, параметрические функции которой имеют вид:

$$f_1 = \cos x (3 + \cos y) ,$$

$$f_2 = \sin x (3 + \cos y) ,$$

$$f_3 = \sin 2y .$$

Диапазон изменения переменных: для x — от 0 до 2π , для y — от 0 до 2π .

Функция `ParametricPlot3D` в данном случае будет иметь вид:

```
ParametricPlot3D[{Cos[x](3+Cos[y]), Sin[x](3+Cos[y]), Sin[2 y]}, {x, 0, 2 Pi}, {y, 0, 2 Pi}]
```

Реализация функции приведена на рис. 3.32.

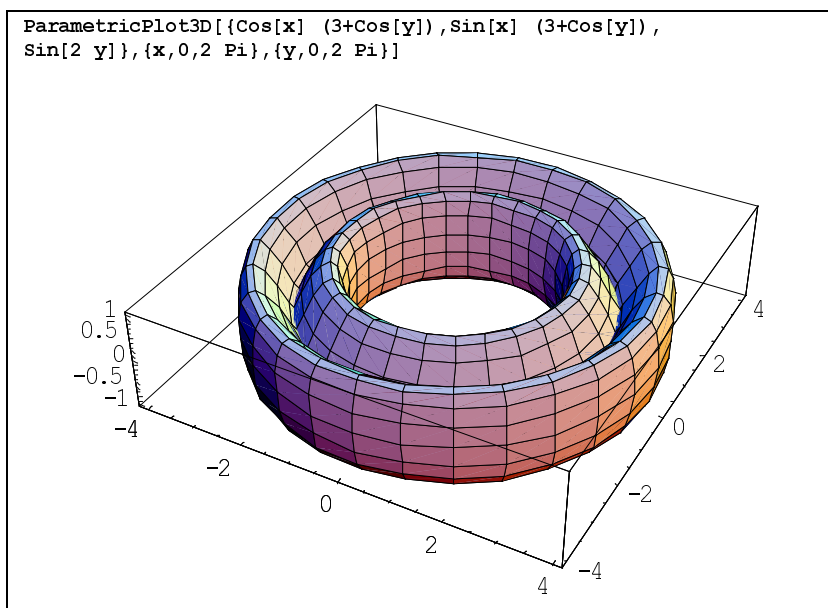
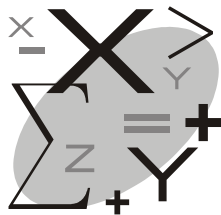


Рис. 3.32. График функции примера 3.26

ГЛАВА 4



Специальные вычисления

В главе излагаются компьютерные технологии решения математических задач. Описаны следующие операции математического анализа в среде Mathematica: вычисление сумм и произведений, способы табулирования функций, вычисление пределов, разложение функции в степенной ряд, дифференцирование. Приводятся примеры решения задач математического анализа.

4.1. Вычисление сумм

Система Mathematica вычисляет суммы вида:

$$S = \sum_{i=i_{\min}}^{i_{\max}} \sum_{j=j_{\min}}^{j_{\max}} \dots \sum_{k=k_{\min}}^{k_{\max}} f_{i,j,\dots,k}.$$

Суммирование осуществляется по неограниченному числу переменных и любому диапазону их изменения до бесконечности включительно. Вычисления осуществляются как в аналитическом, так и в численном виде. При этом функция f может быть списком чисел, символьных переменных или функцией неограниченного числа переменных суммирования.

4.1.1. Вычисление сумм в аналитическом виде

Результатами суммирования в аналитическом виде могут быть выражения (формулы) или точные значения суммы, представленные в нормальной форме (с числителем и знаменателем). Суммирование в аналитическом виде осуществляется с помощью следующих встроженных функций:

`Sum[f1, {i, imin, imax}]`

`Sum[f1, {i, imin, imax}]`

`Sum[f1, {i, imin, imax, Δ1}]`

`Sum[f1,j, {i, imin, imax}, {j, jmin, jmax}, ...]`

В функциях приняты следующие обозначения:

- f — элементы суммирования;
- i — переменная суммирования;
- $i_{\min}, i_{\max}$ — минимальное и максимальное значения индекса i ;
- Δ_i — шаг изменения переменной i .

Приведенные функции реализуют символьные вычисления сумм, а также численные вычисления с выдачей результата в виде точного значения суммы в нормальной форме представления числа.

Рассмотрим подробно все функции суммирования и приведем примеры.

Функция $\text{Sum}[f_i, \{i, i_{\max}\}]$ вычисляет сумму значений f_i при изменении индекса i от 1 до $i_{\max}$ с шагом, равным 1.

Пример 4.1

Необходимо найти сумму чисел от 1 до n , если $f = n$, $f = n^2$, $f = n^3$, $i_{\max} = 1000$, $i_{\max} = 1000000$.

В данном случае следует воспользоваться функцией $\text{Sum}[f, \{i, i_{\max}\}]$, где $i = n$, $i_{\max} = n$, или $i_{\max} = 1000$, или $i_{\max} = 1000000$.

Решение приведено на рис. 4.1.

$$\begin{aligned} &\text{Sum}[n, \{n, n\}] \\ &\frac{1}{2} n (1 + n) \\ &\text{Sum}[n^2, \{n, n\}] \\ &\frac{1}{6} n (1 + n) (1 + 2n) \\ &\text{Sum}[n^3, \{n, n\}] \\ &\frac{1}{4} n^2 (1 + n)^2 \\ &\text{Sum}[n, \{n, 1000\}] \\ &500500 \\ &\text{Sum}[n, \{n, 1000000\}] \\ &500000500000 \end{aligned}$$

Рис. 4.1. Решение примера 4.1

В первых трех функциях Mathematica выдала формулы вычисления сумм натурального ряда чисел, их квадратов и кубов. Теперь вычисления можно выпол-

нить даже "вручную". Значение символьной математики в ее компьютерной реализации трудно переоценить. В последних двух примерах функция вычислила точное значение суммы чисел от 1 до 1000 и от 1 до 1 000 000.

Функция $\text{Sum}[f_i, \{i, i_{\min}, i_{\max}\}]$ вычисляет сумму значений f_i при изменении индекса i от значения $i_{\min}$ до значения $i_{\max}$ с шагом, равным 1.

Пример 4.2

Необходимо найти сумму чисел функции $f = \frac{x^n}{n!}$ при изменении n от 0 до ∞ , от 1 до 10 и от 10 до 15. Определить также значение следующей суммы:

$$\sum_{k=1}^{30} \left(\frac{3^k}{k!} - \frac{2^k}{(k-1)!} \right).$$

В данном случае указан диапазон изменения переменной, поэтому для вычисления сумм необходимо воспользоваться функцией $\text{Sum}[f, \{i, i_{\min}, i_{\max}\}]$.

Решение приведено на рис. 4.2

```
Sum[x^n/n!, {n, 0, ∞}]
e^x
Sum[x^n/n!, {n, 1, 5}]
x + x^2/2 + x^3/6 + x^4/24 + x^5/120
Sum[x^n/n!, {n, 10, 13}]
x^10/3628800 + x^11/39916800 + x^12/479001600 + x^13/6227020800
Sum[3^k/k! - 2^k/(k-1)!, {k, 1, 30}]
380852242272853492818892813085603
88417619937397019545436160000000
N[%]
4.30742
```

Рис. 4.2. Решение примера 4.2

В первой задаче примера 4.2 функция $\text{Sum}[f_i, \{i, i_{\min}, i_{\max}\}]$ нашла сумму бесконечного ряда показательной функции. Во второй и третьей задачах она нашла частичные суммы членов ряда. В последнем примере функция нашла точное значение суммы 30 членов ряда, вычисленных по формуле:

$$\sum_{k=1}^{30} \left(\frac{3^k}{k!} - \frac{2^k}{(k-1)!} \right).$$

Функция $\text{Sum}[f_i, \{i, i_{\min}, i_{\max}, \Delta_i\}]$ вычисляет сумму значений f_i при изменении индексной переменной i от $i_{\min}$ до $i_{\max}$ с постоянным шагом Δ_i .

Функция $\text{Sum}[f_i, j \dots, \{i, i_{\min}, i_{\max}\}, \{j, j_{\min}, j_{\max}\}, \dots]$ вычисляет сумму многоиндексной функции при известных диапазонах значений индексных переменных и постоянном шаге, равном 1.

Пример 4.3

Необходимо вычислить сумму функции $f = n!$ при изменении n от 0 до 10 с шагом $\Delta_i = 0.2$, а также суммы вида:

$$\sum_{i=1}^{50} \sum_{j=1}^{10} (x_i^2 + y_j^2), \quad \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} \frac{x^n}{n!} \frac{y^m}{m!}, \quad \sum_{n=1}^3 \sum_{m=1}^3 \frac{x^n}{n!} \frac{y^m}{m!},$$

вычислив последнюю сумму при $x = y = 1$.

В данном случае для вычисления суммы $n!$ необходимо использовать встроенную функцию $\text{Sum}[f, \{i, i_{\min}, i_{\max}, \Delta_i\}]$, а для вычисления многоиндексных сумм — функцию $\text{Sum}[f, \{\{i, i_{\min}, i_{\max}\}, \{j, j_{\min}, j_{\max}\}\}]$. Решение задачи приведено на рис. 4.3.

```
Sum[n!, {n, 0, 10, 0.2}]
```

```
9.81876 × 106
```

```
Sum[x^2 + y^2, {x, 1, 50}, {y, 1, 10}]
```

```
448500
```

```
Sum[x^n/n! y^m/m!, {n, 0, ∞}, {m, 0, ∞}]
```

```
ex+y
```

```
Sum[x^n/n! y^m/m!, {n, 1, 3}, {m, 1, 3}]
```

$$x y + \frac{x^2 y}{2} + \frac{x^3 y}{6} + \frac{x y^2}{2} + \frac{x^2 y^2}{4} + \frac{x^3 y^2}{12} + \frac{x y^3}{6} + \frac{x^2 y^3}{12} + \frac{x^3 y^3}{36}$$

```
x := 1
```

```
y := 1
```

```
N[%%]
```

```
2.77778
```

Рис. 4.3. Процедуры решения примера 4.3

4.1.2. Вычисление сумм в численном виде

Вычисление сумм в численном виде осуществляется в системе Mathematica с помощью встроенных функций, имеющих тот же вид, что и функции символьного суммирования, с добавлением перед именем функции символа N:

`NSum[fi, {i, imax}]`

`NSum[fi, {i, imin, imax}]`

`NSum[fi, {i, imin, imax, Δi}]`

`NSum[fi,j..., {i, imin, imax}, {j, jmin, jmax}, ...]`

Технология вычисления сумм здесь та же, что и в случае символьных переменных. Результатом вычислений является число, представленное в естественной форме.

Пример 4.4

Необходимо вычислить следующие суммы:

$$\sum_{z=1}^{100} \frac{z^2}{2}, \quad \sum_{z=1}^{\infty} \frac{1}{z^3}, \quad \sum_{x=1}^2 x^2 e^{-x}, \quad \sum_{x=1}^{20} \sum_{y=1}^{30} (x+y)^2.$$

Последнюю сумму вычислить при изменении x от $x=1$ до $x=2$ с шагом $h=0.1$.

Вычисления сумм в системе Mathematica показаны на рис. 4.4.

```
NSum[z^2 / 2, {z, 100}]
169175.
NSum[1 / z^3, {z, 1, ∞}]
1.20206
NSum[x^2 Exp[-x], {x, 1, 2, 0.1}]
5.31199
NSum[(x+y)^2, {x, 1, 20}, {y, 1, 30}]
470500.
```

Рис. 4.4. Решение примера 4.4

4.1.3. Использование символа суммирования $\sum_{\oplus}^{\otimes}$

При решении практических задач удобно и привычно вместо встроенной функции `Sum` использовать символ суммирования $\sum_{\oplus}^{\otimes}$. В этом случае технология суммирования будет состоять из выполнения следующих пунктов:

1. Вызов символа суммирования панели инструментов путем обращения к командам меню **File: File | Palettes | BasicInput** | знак суммирования $\sum^{\oplus} \otimes$.
2. Ввод функции f_1 и пределов суммирования.
3. Нажатие комбинации клавиш <Shift>+<Enter> для получения ответа.

Пример 4.5

Необходимо вычислить, используя символ суммирования, следующие суммы:

$$\sum_{n=1}^n n, \quad \sum_{i=0}^{\infty} \frac{x^i}{i!}, \quad \sum_{k=1}^{30} \left(\frac{2^k}{k!} - \frac{1}{(k-1)!} \right), \quad \sum_{n=1}^3 \sum_{m=1}^3 \frac{x^n}{n!} \frac{y^m}{m!}, \quad \sum_{x=1}^{20} \sum_{y=1}^{30} (x+y)^2.$$

Решение задачи приведено на рис. 4.5.

$$\begin{aligned} & \sum_{n=1}^n n \\ & \frac{1}{2} n (1 + n) \\ & \sum_{i=0}^{\infty} x^i / i! \\ & e^x \\ & \sum_{k=1}^{30} ((2^k) / k! - 1 / (k-1)!) \\ & \frac{243420843241900635790370532190351}{66313214953047764659077120000000} \\ & \sum_{n=1}^3 \sum_{m=1}^3 x^n / n! \cdot x^m / m! \\ & x^2 + x^3 + \frac{7x^4}{12} + \frac{x^5}{6} + \frac{x^6}{36} \\ & \sum_{x=1}^{20} \sum_{y=1}^{30} (x+y)^2 \\ & 470500 \end{aligned}$$

Рис. 4.5. Решение примера 4.5

4.1.4. Примеры вычисления сумм

В табл. 4.1 представлены функции и пределы суммирования, а также ответы, которые приводятся в математическом справочнике [7]. Необходимо найти

суммы приведенных функций с помощью системы Mathematica и сравнить их с ответами. Если они не совпадают, то установить, какой из ответов верный. Примеры полезно решать не только с целью усвоения технологии суммирования. Функции суммирования являются аналитическими, содержащими символьные переменные. Вычисление сумм позволяет судить об интеллектуальности системы Mathematica.

Таблица 4.1. Функции и пределы их суммирования

№ варианта	Функция суммирования	$i_{\min}$	$i_{\max}$	Ответ по справочнику
1	$2k-1$	$k=1$	n	n^2
2	$k!k$	$k=1$	n	$(n-1)!-1$
3	$\frac{k^2+k-1}{(k+2)!}$	$k=1$	n	$\frac{1}{2}-\frac{n+1}{(n+2)!}$
4	aq^k	$k=0$	∞	$\frac{a}{1-q}$
5	$\frac{1}{(2k-1)^2}$	$k=1$	∞	$\frac{\pi^2}{8}$
6	$\frac{1}{(2k-1)^4}$	$k=1$	∞	$\frac{\pi^2}{96}$
7	$\frac{1}{4k^2-1}$	$k=1$	∞	$\frac{1}{2}$
8	$\frac{1}{k2^k}$	$k=1$	∞	$\ln 2$
9	$\frac{k}{(4k^2-1)^2}$	$k=1$	∞	$\frac{1}{8}$
10	$\frac{1}{k^2-1}$	$k=2$	∞	$\frac{3}{4}$
11	$\frac{2k}{(2k+1)!}$	$k=1$	∞	$\frac{1}{e}$

Таблица 4.1 (окончание)

№ варианта	Функция суммирования	$i_{\min}$	$i_{\max}$	Ответ по справочнику
12	$\frac{1}{(2k+1)!}$	$k=0$	∞	$\frac{1}{2}\left(e - \frac{1}{e}\right)$
13	$\frac{(-1)^k}{(2k)!}$	$k=0$	∞	$\cos 1 = 0.54030$
14	kx^k	$k=1$	∞	$\frac{x}{(1-x)^2}$
15	$\sin \frac{\pi k}{n}$	$k=1$	$n-1$	$\operatorname{ctg} \frac{\pi}{2n}$
16	$\frac{\cos kx}{k^2}$	$k=1$	∞	$\frac{\pi^2}{6} - \frac{\pi x}{2} + \frac{x^2}{4}$
17	$\frac{1}{kx^k}$	$k=1$	∞	$\ln \frac{x}{x-1}$
18	$\frac{x^k}{k}$	$k=1$	∞	$\ln \frac{1}{1-x}$

Примечание

Система Mathematica может выдать решение, совпадающее с ответом, верное, но в ином виде, наконец, решение со специальными функциями в ответе.

Для упрощения решения и совпадения результата с ответом полезно применять функции упрощения: `Simplify`, `FullSimplify`, `Expand`.

4.2. Вычисление произведений

Система Mathematica позволяет вычислять произведения неограниченного числа переменных как в аналитическом, так и в численном виде. При этом функция `f1` может быть списком чисел, символьных переменных или функцией неограниченного числа переменных.

Рассмотрим технологии вычисления произведений в среде системы Mathematica.

4.2.1. Вычисление произведений в аналитическом виде

Результатами вычисления произведений в аналитическом виде могут быть формулы или точные численные значения произведения, представленные в нормальной форме (с числителем и знаменателем).

Вычисление произведений в аналитическом виде осуществляется в системе Mathematica с помощью следующих встроенных функций:

`Product[fi, {i, imin, imax}]`

`Product[fi, {i, imin, imax}]`

`Product[fi, {i, imin, imax, Δ}]`

`Product[fi,j, ..., {i, imin, imax}, {j, jmin, jmax}, ...]`

В функциях приняты следующие обозначения:

- ☐ f — элементы произведения;
- ☐ i — индексная переменная элементов произведения;
- ☐ $i_{\min}, i_{\max}$ — минимальное и максимальное значения индекса i .

Приведенные функции вычисляют произведения в символьном или численном виде с выдачей результата в виде точного решения.

Рассмотрим возможности перечисленных функций и приведем примеры их реализации.

- ☐ Функция `Product[fi, {i, imin, imax}]` — вычисляет произведения значений f при изменении индекса i от 1 до $i_{\max}$ с шагом, равным 1.
- ☐ Функция `Product[fi, {i, imin, imax}]` — вычисляет произведение значений f при изменении i от $i_{\min}$ до $i_{\max}$ с шагом, равным 1.
- ☐ Функция `Product[fi, {i, imin, imax, Δ}]` — вычисляет произведение значений f при изменении i от $i_{\min}$ до $i_{\max}$ с шагом, равным Δ .
- ☐ Функция `Product[fi,j, ..., {i, imin, imax}, {j, jmin, jmax}, ...]` — вычисляет произведение значений многопараметрической функции $f_{i,j}, \dots$ при изменении i от $i_{\min}$ до $i_{\max}$, j от $j_{\min}$ до $j_{\max}$ и т. д.

Приведем примеры вычисления произведений аналитическими методами.

Пример 4.6

Требуется вычислить произведения следующих видов:

$$\prod_{x=1}^5 (x+1)^3, \quad \prod_{k=1}^5 (1+x)^k, \quad \prod_{k=1}^k x^k, \quad \prod_{x=2}^{10} \ln x, \quad \prod_{x=1}^3 \prod_{y=1}^3 (x+y)^2.$$

Произведение $\prod_{x=2}^{10} \ln x$ необходимо вычислить с шагом $h = 2$.

Решение задач приведено на рис. 4.6.

```

Product[(1 + x) ^3, {x, 5}]
373248000

Product[(1 + x) ^k, {k, 5}]
(1 + x) ^15

Product[x ^k, {k, 1, k}]
x ^ $\frac{1}{2} k (1+k)$ 

Product[Log[x], {x, 2, 10, 2}]
Log[2] Log[4] Log[6] Log[8] Log[10]
N[%]
8.24372

Product[(x + y) ^2, {x, 1, 3}, {y, 1, 2}]
2073600

```

Рис. 4.6. Решение задач примера 4.6

Из рис. 4.6 следует, что в первых трех задачах решение получено в аналитическом виде. Система не вычислила сумму логарифмов, хотя аргумент задан в виде чисел. Для получения суммы пришлось использовать команду **N[%]**. А вот в последнем примере система нашла сумму в численном виде, несмотря на то, что решала аналитическим методом. Это объясняется тем, что в данном примере получено решение абсолютно точно.

4.2.2. Вычисление произведений в численном виде

Вычисление произведений в численном виде осуществляется системой Mathematica с помощью встроенных функций, имеющих тот же вид, что и аналитические методы, с добавлением перед именем функции символа **N**:

NProduct[f_i, {i, i_{max}}]

NProduct[f_i, {i, i_{min}, i_{max}}]

NProduct[f_i, {i, i_{min}, i_{max}, Δ}]

NProduct[f_{i,j}, ..., {i, i_{min}, i_{max}}, {j, j_{min}, j_{max}}, ...]

Приведем примеры вычисления произведений численными методами.

Пример 4.7

Необходимо вычислить в численном виде следующие произведения:

$$\prod_{x=1}^{10} \frac{x-0.5}{x+0.5}, \quad \prod_{x=2}^6 \frac{1}{x!}, \quad \prod_{x=1}^{10} (xe^{-x} + 1), \quad \prod_{x=1}^5 \prod_{y=1}^3 xy.$$

Решение задач приведено на рис. 4.7.

```

NProduct[(x - 0.5) / (x + 0.5), {x, 10}]
0.047619

NProduct[1 / x!, {x, 2, 6}]
4.01878 × 10-8

NProduct[(x Exp[-x] + 1), {x, 1, 10, 0.5}]
4.41215

NProduct[x y, {x, 1, 5}, {y, 1, 3}]
1.34369 × 1010

Product[x y, {x, 1, 5}, {y, 1, 3}]
13436928000

```

Рис. 4.7. Решение задач примера 4.7

4.2.3. Использование символа произведения Π

Панель инструментов содержит символ произведения Π , который целесообразно использовать при аналитическом методе суммирования. При этом технология вычисления произведения состоит в выполнении следующих процедур:

1. Обращение к символу Π : **File** | **Palettes** | **BasicInput** | Π .
2. Ввод функции f и пределов переменной произведения.
3. Нажатие комбинации клавиш <Shift>+<Enter> для получения решения.

Приведем примеры использования символа произведения.

Пример 4.8

Необходимо вычислить, используя символ Π , произведения функций из примера 4.7.

Решение задач приведено на рис. 4.8.

$$\prod_{x=1}^{10} (x - 0.5) / (x + 0.5)$$

$$0.047619$$

$$\prod_{x=2}^6 1/x!$$

$$\frac{1}{24883200}$$

$$N[\%]$$

$$4.01878 \times 10^{-8}$$

$$\prod_{x=1}^5 \prod_{y=1}^3 x y$$

$$13436928000$$

Рис. 4.8. Решение задач примера 4.8

Из примера 4.8 видно, что решение получено в виде точных значений произведений. Произведение $\prod_{x=2}^6 \frac{1}{x!}$ получено в виде точного числа, представленного в нормальной форме. Для получения приближенного числа в естественной форме пришлось использовать символ $N[\%]$.

4.2.4. Примеры вычисления произведений

В табл. 4.2 приведены функции и пределы вычисления произведений. Их решение полезно для приобретения опыта решения задач в системе Mathematica, а также с целью изучения возможностей системы и ее интеллектуальных способностей.

Таблица 4.2. Функции и пределы вычисления произведений

№ варианта	Функция	$i_{\min}$	$i_{\max}$	Ответ
1	$1 + \frac{(-1)^{k+1}}{2k-1}$	$k = 1$	∞	$\sqrt{2}$
2	$1 - \frac{1}{k^2}$	$k = 2$	∞	$\frac{1}{2}$
3	$1 - \frac{1}{(2k)^2}$	$k = 1$	∞	$\frac{2}{\pi}$

Таблица 4.2 (окончание)

№ варианта	Функция	$i_{\min}$	$i_{\max}$	Ответ
4	$1 - \frac{1}{(2k-1)^2}$	$k = 2$	∞	$\frac{\pi}{4}$
5	$1 + x^{(2^k)}$	$k = 0$	∞	$\frac{1}{1-x}$
6	$1 - \frac{4x^2}{(2k+1)^2 \pi^2}$	$k = 1$	∞	$\cos x$
7	$x \prod_{k=1}^{\infty} \left(1 + \frac{x^2}{k^2 \pi^2} \right)$	—	—	$\sinh x$
8	$2^{n-1} \prod_{k=0}^{n-1} \sin \left(x + \frac{k\pi}{n} \right)$	—	—	$\sin(nx)$
9	$-\prod \left(1 - \left(\frac{2x}{x + k\pi} \right)^2 \right)$	—	—	$\frac{\sin 3x}{\sin x}$
10	$x \prod_{k=1}^{\infty} \cos \frac{x}{2^k}$	—	—	$\sin x$
11	$x \prod_{k=1}^{\infty} \left(1 - \frac{x^2}{k^2 \pi^2} \right)$	—	—	$\sin x$

Примечание

Добивайтесь, чтобы ваши решения совпадали с ответами.

4.3. Табулирование функции

Представление функции в виде таблицы необходимо в следующих случаях:

- ☐ определение погрешности интерполяции;
- ☐ вычисление интегралов специального вида;
- ☐ определение степени интерполяционного полинома по значениям табличных разностей;
- ☐ оценка области изоляции корня.

Таблицу можно получить непосредственными вычислениями значений функции $f(x)$ при различных значениях аргумента. Однако в этом нет необходимости. Система Mathematica для этих целей имеет следующие встроенные функции:

```
Table[f(x), {n}]
```

```
Table[f(x), {x, xmax}]
```

```
Table[f(x), {x, xmin, xmax}]
```

```
Table[f(x), {x, xmin, xmax, Δx}]
```

```
Table[f(x, y, ...), {x, xmin, xmax}, {y, ymin, ymax}, ...]
```

Встроенные функции имеют следующие обозначения:

- $f(x)$ — табулируемая функция;
- x — аргумент табулируемой функции;
- n — число повторений функции $f(x)$;
- $x_{\min}$ — минимальное значение аргумента;
- $x_{\max}$ — максимальное значение аргумента;
- Δx — шаг изменения аргумента;
- $f(x, y, \dots)$ — функция многих переменных, аргументами которых являются $x, y, \dots$

Откликом любой функции `Table` является вектор или матрица, элементы которых есть точные значения функции. При этом нет ограничений на вид функции, диапазон изменения аргумента и шаг таблицы.

Технология табулирования функции в системе Mathematica чрезвычайно проста: вводится функция `Table` и для получения ответа нажимается комбинация клавиш `<Shift>+<Enter>`.

Рассмотрим более подробно встроенные функции и приведем примеры.

Функция `Table[f(x), {n}]` — функция $f(x)$ повторяется n раз и представляется в виде вектора, каждый элемент которого — функция в аналитическом виде.

Если x — число, а функция $f(x)$ имеет точное значение, то откликом являются числа вектора, представленные в нормальной форме.

Пример 4.9

Необходимо представить функции $\ln x$, $\ln 3$ в виде вектора с числом элементов $n = 5$, а функции e^{-1} и $\sin 2$ в виде матрицы с числом элементов $n = 3$.

Решение приведено на рис. 4.9.

Из рис. 4.9 видно, что во всех случаях получаем решение в аналитическом виде. Для получения численного решения используется команда `N[%]`.

```

Table[Log[x], {5}]
{Log[x], Log[x], Log[x], Log[x], Log[x]}
Table[Log[3], {5}]
{Log[3], Log[3], Log[3], Log[3], Log[3]}
N[%]
{1.09861, 1.09861, 1.09861, 1.09861, 1.09861}
Table[{Exp[-1], Sin[2]}, {3}]
{{1/e, Sin[2]}, {1/e, Sin[2]}, {1/e, Sin[2]}}
N[%]
{{0.367879, 0.909297}, {0.367879, 0.909297},
{0.367879, 0.909297}}

```

Рис. 4.9. Табулирование функций примера 4.9

Функция $\text{Table}[f(x), \{x, x_{\max}\}]$ — откликом является вектор, элементы которого есть значения функции $f(x)$ от $x=1$ до $x=x_{\max}$. Решения представляются в аналитическом виде.

Пример 4.10

Необходимо протабулировать следующие функции:

$$\frac{x^2+1}{x}, xe^x, \frac{a^2+b^2}{2a},$$

а также функции x^2 , e^x , представленные в виде матрицы. Максимальное значение аргумента $x_{\max}=5$.

Решение приведено на рис. 4.10.

Функция $\text{Table}[f(x, \{x, x_{\min}, x_{\max}\})]$ — откликом этой функции является вектор, элементы которого есть значения функции $f(x)$ в диапазоне от $x=x_{\min}$ до $x=x_{\max}$, вычисленные с шагом $h=1$.

Пример 4.11

Необходимо протабулировать функции:

□ $\frac{x-1}{x+1}$ при $x_{\min}=4$, $x_{\max}=10$;

□ $\frac{a-1}{b+1}$ при $b_{\min}=2$, $b_{\max}=6$.

Table[{x^2 + 1} / x, {x, 5}]

$\left\{2, \frac{5}{2}, \frac{10}{3}, \frac{17}{4}, \frac{26}{5}\right\}$

Table[x Exp[x], {x, 5}]

$\{e, 2 e^2, 3 e^3, 4 e^4, 5 e^5\}$

N[%]

$\{2.71828, 14.7781, 60.2566, 218.393, 742.066\}$

Table[{a^2 + b^2} / 2 a, {a, 5}]

$\left\{\frac{1}{2} (1 + b^2), 4 + b^2, \frac{3}{2} (9 + b^2), 2 (16 + b^2), \frac{5}{2} (25 + b^2)\right\}$

Table[{x^2, Exp[x]}, {x, 5}]

$\{\{1, e\}, \{4, e^2\}, \{9, e^3\}, \{16, e^4\}, \{25, e^5\}\}$

Рис. 4.10. Решение примера 4.10

Выполнить также табулирование функций x и $\ln x$ в диапазоне $x_{\min}=5$, $x_{\max}=8$, представленных в виде матрицы.

Решение приведено на рис. 4.11.

Table[{x - 1} / (x + 1), {x, 4, 10}]

$\left\{\frac{3}{5}, \frac{2}{3}, \frac{5}{7}, \frac{3}{4}, \frac{7}{9}, \frac{4}{5}, \frac{9}{11}\right\}$

Table[{a - 1} / (b + 1), {b, 2, 6}]

$\left\{\frac{1}{3} (-1 + a), \frac{1}{4} (-1 + a), \frac{1}{5} (-1 + a), \frac{1}{6} (-1 + a), \frac{1}{7} (-1 + a)\right\}$

Table[{x, Log[x]}, {x, 5, 8}]

$\{\{5, \text{Log}[5]\}, \{6, \text{Log}[6]\}, \{7, \text{Log}[7]\}, \{8, \text{Log}[8]\}\}$

N[%]

$\{\{5., 1.60944\}, \{6., 1.79176\}, \{7., 1.94591\}, \{8., 2.07944\}\}$

Рис. 4.11. Табулирование функций примера 4.11

Одним из недостатков табулирования с помощью функции **Table** является отсутствие в ответе значений аргумента, при котором вычисляется функция. Этот недостаток затрудняет определение ответа, если таблица длинная. Возможность табулирования одновременно нескольких функций позволяет устранить этот недостаток. Для этого достаточно функцию $f(x)$ представить в виде вектора $\{x, f(x)\}$. Тогда ответ будет получен в виде матрицы, элементами которой

являются значения x и $f(x)$. Этот случай показан на рис. 4.13 при табулировании функции $\log x$.

Функция `Table[f(x), {x, xmin, xmax, Δx}]` — откликом функции является вектор, элементами которого являются значения функции $f(x)$ в диапазоне от $x=x_{\min}$ до $x_{\max}$ при шаге $h=\Delta x$.

Пример 4.12

Необходимо протабулировать следующие функции:

- $f(x) = \ln x$ при $x_{\min}=1$, $x_{\max}=5$, $\Delta x=0.1$;
- $f(x) = \sqrt{x}$ при $x_{\min}=1$, $x_{\max}=5$, $\Delta x=0.5$, представив ответ в виде матрицы, элементами которой являются значения аргументов x и функции $f(x)$;
- $f(x) = x^2$ и $f(x) = \sqrt{x}$ при $x_{\min}=1$, $x_{\max}=2$, $\Delta x=0.2$, представив ответ в виде матрицы, элементами которой являются значения аргументов x и функций $f(x)$.

Решения приведены на рис. 4.12.

Из рис. 4.14 видно, что функция `Table[f(x), {x, xmin, xmax, Δx}]` выдает решение в численном виде, в то время как табулирование с постоянным шагом дает решение в аналитическом виде.

Функция `Table[f(x,y,...), {x, xmin, xmax}, {y, ymin, ymax}, ...]` осуществляет табулирование функций многих переменных, число функций не ограничено. Решение выдает в аналитическом виде.

Table [Log [x] , {x,1,5,0.1}]

```
{0,0.0953102,0.182322,0.262364,0.336472,0.405465,0.470004,0.530628,0.587787,0.641854,0.693147,0.741937,0.788457,0.832909,0.875469,0.916291,0.955511,0.993252,1.02962,1.06471,1.09861,1.1314,1.16315,1.19392,1.22378,1.25276,1.28093,1.30833,1.335,1.36098,1.38629,1.41099,1.43508,1.45862,1.4816,1.50408,1.52606,1.54756,1.56862,1.58924,1.60944}
```

Table [{x,Sqrt [x]} ,{x,1,5,0.5}]

```
{{1,1},{1.5,1.22474},{2.,1.41421},{2.5,1.58114},{3.,1.73205},{3.5,1.87083},{4.,2.},{4.5,2.12132},{5.,2.23607}}
```

Table [{x,x^2,Sqrt [x]} ,{x,1,2,0.2}]

```
{{1,1,1},{1.2,1.44,1.09545},{1.4,1.96,1.18322},{1.6,2.56,1.26491},{1.8,3.24,1.34164},{2.,4.,1.41421}}
```

Рис. 4.12. Решение примера 4.12

Пример 4.13

Необходимо табулировать следующие функции: $f(a, b) = \sqrt{a^2 + b^2}$, $f(n, m) = n!m!$ при изменении аргумента a от 1 до 4, а всех остальных аргументов — от 1 до 5 с шагом, равным 1.

Решение приведено на рис. 4.13.

```
Table[Sqrt[a^2 + b^2], {a, 1, 4}, {b, 1, 5}]
{{Sqrt[2], Sqrt[5], Sqrt[10], Sqrt[17], Sqrt[26]}, {Sqrt[5], 2 Sqrt[2], Sqrt[13], 2 Sqrt[5], Sqrt[29]},
 {Sqrt[10], Sqrt[13], 3 Sqrt[2], 5, Sqrt[34]}, {Sqrt[17], 2 Sqrt[5], 5, 4 Sqrt[2], Sqrt[41]}}
```

```
Table[n! m!, {n, 1, 5}, {m, 1, 5}]
{{1, 2, 6, 24, 120}, {2, 4, 12, 48, 240}, {6, 12, 36, 144, 720},
 {24, 48, 144, 576, 2880}, {120, 240, 720, 2880, 14400}}
```

Рис. 4.13. Решение примера 4.13

4.4. Вычисление пределов

4.4.1. Технология вычисления пределов системой Mathematica

Существуют математические функции, значения которых при определенных значениях аргумента являются неопределенными. Это случается, например, в тех случаях, когда в функции имеет место разрыв непрерывности или при

определенных значениях аргумента она имеет значения: $\frac{0}{0}$, $\frac{0}{\infty}$, $\frac{\infty}{0}$, $\frac{\infty}{\infty}$.

В таких случаях могут возникать трудности решения математических задач.

Вот один из примеров.

Пример 4.14

Необходимо выполнить табулирование функции $y = \frac{x-1}{\ln x}$ при $x_{\min}=1$, $x_{\max}=5$ с шагом, равным 1. Решение с помощью функции Table показано на рис. 4.14.

Из рис. 4.14 видно, что решение не получено, т. к. при $x=1$ значение функции не вычислено. Между тем при $x=1$ значение функции равно 1. Это видно из

последней строки рис. 4.16, где находится результат вычисления предела исходной функции.

```
Table[{x, (x-1)/Log[x]}, {x, 1, 5}]
{ {1, Indeterminate}, {2, 1/Log[2]},
  {3, 2/Log[3]}, {4, 3/Log[4]}, {5, 4/Log[5]} }
N[%]
{{1., Indeterminate}, {2., 1.4427}, {3., 1.82048}, {4., 2.1640
4}, {5., 2.48534}}
Limit[(x-1)/Log[x], x->1]
1
```

Рис. 4.14. Решение задачи примера 4.14

Подобных примеров великое множество. В каждом из таких случаев пользователю приходится анализировать решение, вычислять пределы. Для определения предела во многих случаях пользуются правилом Лопиталья: предел функции $f(x)$ равен пределу отношения производной числителя, деленной на производную знаменателя. Например:

$$\lim_{x \rightarrow 1} \frac{x-1}{\ln x} = \lim_{x \rightarrow 1} \frac{(x-1)'}{(\ln x)'} = \lim_{x \rightarrow 1} \frac{1}{\frac{1}{x}} = \lim_{x \rightarrow 1} x = 1.$$

Заметим, что до появления систем компьютерной алгебры задача вычисления пределов с помощью компьютерной техники вообще не ставилась, т. к. компьютер тех времен с его программным обеспечением такие задачи не мог решать принципиально.

В системе Mathematica пределы определяются с помощью встроенной функции `Limit`, имеющей вид:

`Limit[f(x), x -> x0]`

где:

- $f(x)$ — функция, предел которой необходимо определить;
- x — аргумент функции $f(x)$;
- x_0 — предельное значение x .

Эта функция позволяет находить предел не только в виде числа, но также в виде аналитического выражения, что свидетельствует о высоких интеллектуальных возможностях системы Mathematica.

Рассмотрим примеры применения функции `Limit`.

Пример 4.15

Необходимо определить следующие пределы:

$$\lim_{t \rightarrow 0} \frac{1 - e^{-2t}}{\ln(1 - 3t)}, \quad \lim_{n \rightarrow 0} \frac{1 - a^n}{1 - e^n}, \quad \lim_{x \rightarrow 0} \frac{\operatorname{tg} x}{x}, \quad \lim_{x \rightarrow 0} \frac{\sin x}{x}, \quad \lim_{x \rightarrow 0} \frac{1 + e^x}{1 - e^{-x}}.$$

Решение приведено на рис. 4.15.

```

Limit[(1 - Exp[-2 t]) / Log[1 - 3 t], t -> 0]
- 2/3

Limit[(1 - a^n) / (1 - Exp[n]), n -> 0]
Log[a]

Limit[Tan[x] / x, x -> 0]
1

Limit[Sin[x] / x, x -> 0]
1

Limit[(1 + Exp[x]) / (1 - Exp[-x]), x -> 0]
∞

```

Рис. 4.15. Решение примера 4.15

Функция `Limit` имеет следующие опции: `Direction` и `Analytic`.

Опция `Direction` указывает направление приближения к пределу, она имеет вид:

`Direction -> +1`

`Direction -> -1`

Значение `+1` означает, что приближение к пределу осуществляется слева, а `-1` — справа. По умолчанию направление приближения выбирает система.

Функция `Limit` с опцией `Direction` имеет вид:

`Limit[f(x), x->x0, Direction -> +1]`

`Limit[f(x), x->x0, Direction -> -1]`

Пример 4.16

Требуется определить следующий предел: $\lim_{x \rightarrow 0} \operatorname{arctg} \frac{1}{x}$ при приближении к пределу слева и справа.

Решение приведено на рис. 4.16.

```

Limit[ArcTan[1/x], x -> 0, Direction -> +1]
-  $\frac{\pi}{2}$ 

Limit[ArcTan[1/x], x -> 0, Direction -> -1]
 $\frac{\pi}{2}$ 

Limit[ArcTan[1/x], x -> 0]
 $\frac{\pi}{2}$ 

```

Рис. 4.16. Определение предела при приближении слева и справа

Из рис. 4.16 видно, что пределы при приближении к ним слева и справа разные. При отсутствии опции `Direction` система выбрала приближение к пределу слева.

Рис. 4.17, на котором приведен график функции $\arctg(1/x)$, дает объяснение приближений и ответов.

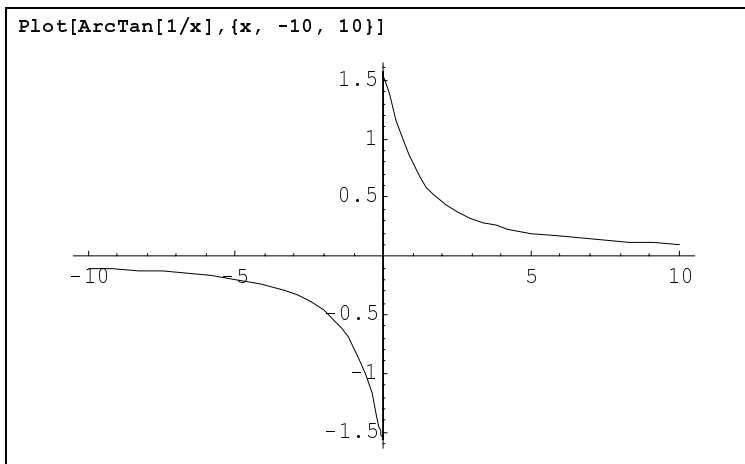


Рис. 4.17. График функции $\arctg(1/x)$

Из графика рис. 4.17 видно, что функция имеет разрыв непрерывности и при приближении к нему слева (+1) пределом будет отрицательное значение функции ($-\pi/2$), а при приближении справа (-1) — положительное ($\pi/2$).

Функция `Limit` имеет еще одну опцию — `Analytic`. С этой опцией она представляется в следующих видах:


```
Limit[f(x), x->x0, Analytic -> True]
```

```
Limit[f(x), x->x0, Analytic -> False]
```

Эта опция указывает, является ли неизвестная функция аналитической. Большого практического значения эта опция не имеет.

4.4.2. Примеры вычисления пределов

В табл. 4.3 приведены задачи на определение пределов функции. Они иллюстрируют возможности системы Mathematica. Пользователю полезно обратить внимание на аналитические решения, которые определяют интеллектуальность системы.

Таблица 4.3. Функции и значения их пределов

№	Функция	Предельное значение аргумента	Ответ
1	$(1+x)^{\frac{1}{x}}$	∞	1
2	$\left(\frac{a^x + b^x}{2}\right)^{\frac{1}{x}}$	0	$\sqrt{a}, \sqrt{b}$
3	$\left(1 + \frac{1}{x}\right)^{\frac{1}{x}}$	∞	1
4	$(1-x)^{\frac{1}{1-x}}$	1	∞
5	$(1-x)^{\frac{1}{1-x}}$	∞	1
6	$\frac{1 - e^{-at}}{\log(1-t)}$	0	$-a$
7	$\frac{1 - a^n}{a^n}$	0	$-\frac{\ln a}{a}$
8	$2a \frac{1 - e^{-ax}}{2 - e^{-ax}}$	∞	a
9	$\frac{x - a}{\ln(x - a + 1)}$	a	1

Таблица 4.3 (окончание)

№	Функция	Предельное значение аргумента	Ответ
10	$\frac{n^2 - n^x}{x - 2}$	2	$-n^2 \ln n$

4.5. Разложение функции в степенной ряд

4.5.1. Технология разложения функции в ряд Тейлора в системе Mathematica

Разложение функции в степенной ряд имеет в математике большое значение. Оно позволяет:

- ☐ упрощать математические расчеты;
- ☐ получать решения задач в аналитическом виде;
- ☐ находить математические модели различных объектов и явлений путем полиномиальной интерполяции.

Вот один простой пример. Дана функция

$$y(x) = xe^{-0.5x} + 5x^2.$$

Разложив ее вокруг $x_0 = 0.75$ в ряд Тейлора, получим:

$$y(x) = 4.7x^2 + 0.854x + 0.036.$$

Вычислим теперь определенный интеграл от обеих функций в пределах от $x = -2$ до $x = 5$. Получим:

$$\int_{-2}^5 (xe^{-0.5x} + 5x^2) dx = 220.5, \quad \int_{-2}^5 (4.7x^2 + 0.85x + 0.036) dx = 218.45.$$

Решения практически совпадают, над полученным в результате разложения в ряд полиномом теперь можно выполнять любые математические действия в диапазоне значений x от -2 до 5 . Например, производная от исходной функции имеет вид:

$$y'(x) = \frac{e^{-\frac{x}{2}}(2-x)}{2} + 10x,$$

а в случае полинома $y'(x) = 9.4x + 0.85$ — прямая линия.

Разложение функции в степенной ряд осуществляется в системе Mathematica по формуле Тейлора, которая имеет вид:

$$f(x) = f(a) + (x-a) \frac{f'(a)}{1!} + (x-a)^2 \frac{f''(a)}{2!} + \dots + (x-a)^n \frac{f^{(n)}(a)}{n!}, \quad (4.1)$$

где a — значение аргумента x функции $y = f(x)$, вокруг которого происходит разложение в ряд.

Формула (4.1) справедлива для любого вещественного числа a , в том числе и для $a = 0$. При $a = 0$ формула Тейлора принимает вид:

$$f(x) = f(0) + x \frac{f'(0)}{1!} + x^2 \frac{f''(0)}{2!} + \dots + x^n \frac{f^{(n)}(0)}{n!}. \quad (4.2)$$

Формула (4.2) называется *рядом Маклорена*.

Заметим, что формулы (4.1) и (4.2) Mathematica выдает, если воспользоваться встроенной функцией `Series`, описание которой приводится далее. Решение показано на рис. 4.18.

Series[f[x], {x, a, 3}]

$$f[a] + f'[a] (x - a) + \frac{1}{2} f''[a] (x - a)^2 + \frac{1}{6} f^{(3)}[a] (x - a)^3 + O[x - a]^4$$

Series[f[x], {x, 0, 3}]

$$f[0] + f'[0] x + \frac{1}{2} f''[0] x^2 + \frac{1}{6} f^{(3)}[0] x^3 + O[x]^4$$

Рис. 4.18. Отклик в виде формул Тейлора и Маклорена

Разложение функций в степенной ряд в системе Mathematica осуществляется с помощью следующих встроенных функций:

```
Series[f, {x, x0, n}]
Series[f(x,y), {x, x0, nx}, {y, y0, ny}]
Collect[Series[f, {x, x0, n}], x]
Normal[Series[f, {x, x0, n}]]
SeriesCoefficient[s, n]
```

В этих функциях приняты следующие обозначения:

- ☐ f — функция, разлагаемая в ряд;
- ☐ $f(x, y)$ — функция двух переменных x и y ;

- x — аргумент функции;
- x_0 — окрестность точки разложения;
- n — степень многочлена;
- s — ряд степени n .

Технология разложения функции в степенной ряд исключительно проста: ввод встроенной функции и нажатие комбинации клавиш <Shift>+<Enter> для получения ответа.

Рассмотрим подробно встроенные функции и приведем примеры.

Функция `Series[f, {x, x0, n}]` осуществляет разложение функции f в ряд вокруг $x=x_0$ с числом членов n .

Пример 4.17

Необходимо разложить в степенной ряд следующие функции: e^x , $\sin x$, $\arcsin x$, $\sinh x$, $\ln x$. Разложение выполнить при числе членов $n = 7$.

Решение приведено на рис. 4.19.

Series[Exp[x], {x, 0, 7}]

$$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \frac{x^7}{5040} + O[x]^8$$

Series[Sin[x], {x, 0, 7}]

$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + O[x]^8$$

Series[ArcSin[x], {x, 0, 7}]

$$x + \frac{x^3}{6} + \frac{3x^5}{40} + \frac{5x^7}{112} + O[x]^8$$

Series[Sinh[x], {x, 0, 7}]

$$x + \frac{x^3}{6} + \frac{x^5}{120} + \frac{x^7}{5040} + O[x]^8$$

Series[Log[x], {x, 0, 7}]

$$\text{Log}[x] + O[x]^8$$

Series[Log[x], {x, 1, 5}]

$$(x-1) - \frac{1}{2}(x-1)^2 + \frac{1}{3}(x-1)^3 - \frac{1}{4}(x-1)^4 + \frac{1}{5}(x-1)^5 + O[x-1]^6$$

Рис. 4.19. Разложение функций в степенной ряд

Из рис. 4.19 видны следующие особенности разложения функций в степенной ряд:

- ☐ в каждом ответе указывается порядок погрешности;
- ☐ степенные ряды функций $\sin x$, $\arcsin x$, $\sinh x$ имеют только четыре члена, в то время как функция Series содержит $n = 7$ членов; это объясняется тем, что эти функции являются нечетными, их четные члены равны нулю;
- ☐ функцию $\ln x$ система не разложила в ряд вокруг $x_0=0$, т. к. $\ln 0$ не существует;
- ☐ решение представляется в аналитическом виде.

Функция Series[y(x,y),{x, x₀, n_x}, {y, y₀, n_y}] осуществляет разложение в ряд функции y(x,y) сначала по y, а затем по x вокруг y₀, x₀ с числом членов, соответственно, n_y, n_x.

Пример 4.18

Необходимо разложить в ряд функции $e^{(x+y)}$, e^{xy} вокруг $x_0=0$ с числом членов n_x, n_y, указанных в решении.

Решение задачи приведено на рис. 4.20.

Series[Exp[x + y], {x, 0, 3}, {y, 0, 3}]

$$\left(1 + y + \frac{y^2}{2} + \frac{y^3}{6} + O[y]^4\right) + \left(1 + y + \frac{y^2}{2} + \frac{y^3}{6} + O[y]^4\right) x + \left(\frac{1}{2} + \frac{y}{2} + \frac{y^2}{4} + \frac{y^3}{12} + O[y]^4\right) x^2 + \left(\frac{1}{6} + \frac{y}{6} + \frac{y^2}{12} + \frac{y^3}{36} + O[y]^4\right) x^3 + O[x]^4$$

Series[Exp[x + y], {x, 0, 4}, {y, 0, 2}]

$$\left(1 + y + \frac{y^2}{2} + O[y]^3\right) + \left(1 + y + \frac{y^2}{2} + O[y]^3\right) x + \left(\frac{1}{2} + \frac{y}{2} + \frac{y^2}{4} + O[y]^3\right) x^2 + \left(\frac{1}{6} + \frac{y}{6} + \frac{y^2}{12} + O[y]^3\right) x^3 + \left(\frac{1}{24} + \frac{y}{24} + \frac{y^2}{48} + O[y]^3\right) x^4 + O[x]^5$$

Series[Exp[xy], {x, 0, 3}, {y, 0, 3}]

$$1 + (y + O[y]^4) x + \left(\frac{y^2}{2} + O[y]^4\right) x^2 + \left(\frac{y^3}{6} + O[y]^4\right) x^3 + O[x]^4$$

Рис. 4.20. Разложение в ряд функций $e^{(x+y)}$, e^{xy}

Из рис. 4.20 видно, что решение получено в аналитическом виде.

Функция Collect[Series[f, {x, x₀, n}], x] осуществляет разложение в степенной ряд с удалением члена с остаточной погрешностью.

Функция `Normal[Series[f, {x, x0, n}]]` осуществляет разложение в степенной ряд с удалением члена с остаточной погрешностью, отличается от предыдущей функции лишь формой представления.

Приведем примеры разложения функций в ряд с удалением члена с остаточной погрешностью.

Пример 4.19

Необходимо разложить в ряд Тейлора вокруг $x_0=0$ следующие функции:

$$\sin x, \sinh x, e^x, e^{xy}.$$

Число членов ряда видно из решения.

Решение приведено на рис. 4.21.

```
Collect[Series[Sin[x], {x, 0, 5}], x]
x -  $\frac{x^3}{6}$  +  $\frac{x^5}{120}$ 

Normal[Series[Sin[x], {x, 0, 5}]]
x -  $\frac{x^3}{6}$  +  $\frac{x^5}{120}$ 

Collect[Series[Exp[x y], {x, 0, 5}, {y, 0, 5}], {x, y}]
1 + x y +  $\frac{x^2 y^2}{2}$  +  $\frac{x^3 y^3}{6}$  +  $\frac{x^4 y^4}{24}$  +  $\frac{x^5 y^5}{120}$ 

Normal[Series[Exp[x y], {x, 0, 5}, {y, 0, 5}]]
1 + x y +  $\frac{x^2 y^2}{2}$  +  $\frac{x^3 y^3}{6}$  +  $\frac{x^4 y^4}{24}$  +  $\frac{x^5 y^5}{120}$ 
```

Рис. 4.21. Разложение функций в ряд с удалением члена с остаточной погрешностью

Из рис. 4.21 видно, что во всех случаях решение получено в аналитическом виде без члена с остаточной погрешностью.

При практическом использовании формулы Тейлора у пользователя возникают следующие вопросы:

- ☐ Какие функции допускают разложение в ряд Тейлора?
- ☐ Каким числом членов ряда можно ограничиться в данном конкретном случае?
- ☐ Как найти погрешность разложения?

Ответим на эти вопросы.

Разложить в ряд Тейлора в ограниченном диапазоне аргумента можно любую функцию, имеющую n производных. При этом функция и производные должны иметь численные значения при значении аргумента, вокруг которого происходит разложение. Так, например, функцию $\ln x$ нельзя разложить в степенной ряд вокруг $x_0=0$, т. к. $\ln 0$ не существует. А вот функции $\sin x/x$, $(1-e^x)/x$, $(1-e^{-ax})/bx$ разложить в ряд Тейлора вокруг $x_0=0$ можно, несмотря на то, что здесь имеют место случаи неопределенности вида $0/0$. Ряды этих функций приведены на рис. 4.22.

The image shows a Mathematica notebook output with four series expansions. Each expansion is preceded by a command in bold. The first command is **Series[Log[x], {x, 0, 7}]** and the result is $\text{Log}[x] + O[x]^8$. The second command is **Series[Sin[x]/x, {x, 0, 7}]** and the result is $1 - \frac{x^2}{6} + \frac{x^4}{120} - \frac{x^6}{5040} + O[x]^8$. The third command is **Series[(1-Exp[x])/x, {x, 0, 7}]** and the result is $-1 - \frac{x}{2} - \frac{x^2}{6} - \frac{x^3}{24} - \frac{x^4}{120} - \frac{x^5}{720} - \frac{x^6}{5040} - \frac{x^7}{40320} + O[x]^8$. The fourth command is **Series[(1-Exp[-a x])/(b x), {x, 0, 7}]** and the result is $\frac{a}{b} - \frac{a^2 x}{2b} + \frac{a^3 x^2}{6b} - \frac{a^4 x^3}{24b} + \frac{a^5 x^4}{120b} - \frac{a^6 x^5}{720b} + \frac{a^7 x^6}{5040b} - \frac{a^8 x^7}{40320b} + O[x]^8$.

Рис. 4.22. Разложение в степенной ряд функций особого вида

Из рис. 4.22 видно, что ряд Тейлора функции $\ln x$ не существует, т. к. $\ln 0$ равен $-\infty$. В остальных случаях решение получено, что свидетельствует о высокой интеллектуальности системы Mathematica, которая раскрыла неопределенности вида $0/0$ и представила решение в аналитическом виде.

4.5.2. Погрешности степенных рядов Тейлора

В процессе вычисления функции бесконечный степенной ряд заменяется частичной суммой. В связи с этим следует знать, сколько членов ряда необходимо сохранить, чтобы вычислить функцию с заданной точностью. Решение этой задачи зависит от вида функции и значения аргумента.

Случай 1. Ряд знакопеременный, члены ряда быстро убывают, $x < 1$

В этом случае ряд обладает следующим свойством: сумма отброшенных членов ряда по абсолютной величине не превышает значения последнего оставленного члена. Для такого случая определить число членов ряда легко. Например, для

ряда e^{-x} общий член ряда имеет вид: $(-1)^n x^n / n!$. Тогда число n можно найти из условия: $x^n / n! \leq \varepsilon$, где ε — погрешность вычисления функции. Если, например, $\varepsilon = 0.001$, то $x^n / n! \leq 0.001$. Тогда при $x = 0.1$ достаточно оставить три члена ряда, $n = 3$.

Случай 2. Ряд знакопеременный, $x > 1$

В этом случае свойство частичной суммы ряда сохраняется, но ряд сходится медленно и необходимое число членов ряда должно быть слишком большим. Так, например, если в случае 1 положить $x = 1$, то условием выбора числа членов ряда будет: $1^n / n! \leq 0.001$ или $n! = 1000$, т. е. $n \geq 7$.

Случай 3. Ряд не знакопеременный

В этом случае общих правил выбора числа членов из условий точности не существует.

4.5.3. Компьютерные технологии оценки погрешностей рядов

Способ 1. Табулирование функций

Сущность этого способа состоит в следующем: осуществляется табулирование исходной функции $f(x)$ и полученной в результате разложения в ряд Тейлора. Начальное и конечное значения x выбираются из диапазона разложения функции в ряд. Сравнение численных значений функций позволяет судить о точности разложения. Более того, можно установить область значений x , в которой погрешность степенного ряда будет не более допустимой при данном числе членов ряда.

Пример 4.20

Необходимо оценить погрешность ряда, полученного в результате разложения функции e^x в области $x = 0 \text{—} 2$ при числе членов разложения $n = 3, 4, 5$. За начальное значение принять $x_0 = 0$.

Технология решения задачи.

□ Разложение e^x в ряд с помощью функции

```
Normal[Series[f(x), {x, 0, n}]]
```

В результате разложения получим:

```
Normal[Series[Exp[x], {x, 0, 3}]]
```

$$1 + x + \frac{x^2}{2} + \frac{x^3}{6}$$

`Normal[Series[Exp[x], {x, 0, 4}]]`

$$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24}$$

`Normal[Series[Exp[x], {x, 0, 5}]]`

$$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120}$$

- Табулирование e^x и функций разложения. Функция табуляции в данном случае будет иметь вид:

`Table[{x, f(x)}, {x, xmin, xmax, h}]`

Пусть $x_{\min}=0$, $x_{\max}=2$, а шаг $h=0.2$. Тогда функция `Table` будет иметь вид:

`Table[{x, Exp[x], f3(x), f4(x), f5(x)}, {x, 0, 2, 0.2}]`

где $f_3(x)$, $f_4(x)$, $f_5(x)$ — функция e^x при $n=3, 4, 5$. Процедуры вычисления и результаты табулирования приведены на рис. 4.23.

Table[{x, Exp[x], Out[6], Out[7], Out[8]}, {x, 0, 2, 0.2}]

```
{ {0, 1, 1, 1, 1}, {0.2, 1.2214, 1.22133, 1.2214, 1.2214}, {0.4, 1.49182, 1.49067, 1.49173, 1.49182}, {0.6, 1.82212, 1.816, 1.8214, 1.82205}, {0.8, 2.22554, 2.20533, 2.2224, 2.22513}, {1., 2.71828, 2.66667, 2.70833, 2.71667}, {1.2, 3.32012, 3.208, 3.2944, 3.31514}, {1.4, 4.0552, 3.83733, 3.9974, 4.04222}, {1.6, 4.95303, 4.56267, 4.83573, 4.92311}, {1.8, 6.04965, 5.392, 5.8294, 5.98686}, {2., 7.38906, 6.33333, 7., 7.26667} }
```

Рис. 4.23. Результаты табулирования исходной функции e^x и функций ряда Тейлора

Обозначения `Out[6]`, `Out[7]`, `Out[8]` означают номера строк, в которых находится функция e^x с числом членов разложения, равным, соответственно, 3, 4, 5.

Из решения видно, что погрешности рядов при числе членов $n=3, 4, 5$ в диапазоне $x=0—2$ достаточно большие. Они уменьшаются с увеличением n . Можно показать, что для вычисления значения e с точностью шесть знаков необходимо иметь ряд Тейлора с числом членов разложения, равным 9.

Способ 2. Визуализация решения

Сущность этого способа состоит в следующем.

Исходная функция $f(x)$ и степенной ряд, ей соответствующий, представляются в виде графика. Сравнение графиков позволяет судить о точности степенного ряда и допустимом диапазоне значений аргумента.

Технология этого способа состоит в следующем:

- ☐ разложение $f(x)$ в ряд Тейлора;
- ☐ построение на одном графике исходной функции $f(x)$ и полученной в результате ее разложения в ряд;
- ☐ сравнение графиков.

При желании можно построить график разности функций, который будет характеризовать погрешность ряда Тейлора.

На рис. 4.24 приведены графики функций e^x и ряда Тейлора с числом членов $n = 5$, построенные по данным предыдущего примера.

Из рис. 4.24 видна погрешность степенной функции, возрастающей с ростом x .

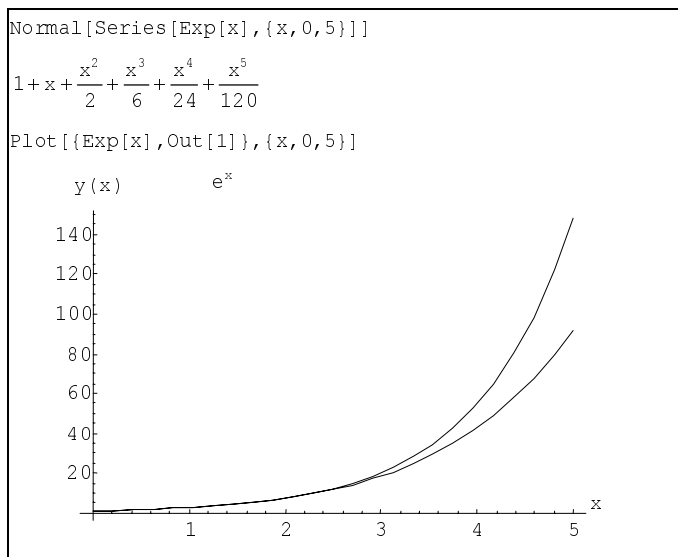


Рис. 4.24. Графики функций e^x и ряда Тейлора при $n = 5$

Способ 3. Вычисление погрешностей

Способ вычисления погрешности степенной функции определяется выбранным критерием, например, максимальной погрешностью, абсолютной среднеквадратической, относительной среднеквадратической. Методы расчета и компьютерные технологии подробно рассматриваются в *главе 12*.

4.6. Вычисление производных

Система Mathematica вычисляет производные в аналитическом и численном виде. При этом дифференцируемыми функциями могут быть функции элементарные, специальные и любые пользовательские.

Вычисление производных осуществляется с помощью следующих встроенных функций:

```
D[f, x]
D[f, {x, n}]
D[f, x1, x2, ...]
Dt[f, x]
Dt[f]
Derivative[n1, n2, ...][f]
```

В функциях приняты следующие обозначения:

- f — дифференцируемая функция;
- x — аргумент дифференцируемой функции (переменная дифференцирования);
- n — порядок производной;
- $x_1, x_2, \dots$ — переменные дифференцирования;
- $n_1, n_2, \dots$ — порядок производной при n -кратном дифференцировании.

Рассмотрим подробно встроенные функции и приведем примеры.

Функция $D[f, x]$ вычисляет частную производную функции f по переменной x .

Пример 4.21

Необходимо вычислить частную производную по x следующих функций:

$$ax^3 + bx - c, \ln ax/x^2, \sinh ax, (x-1)/(x+1), x!, (\sin x + \cos x)^2.$$

Решение приведено на рис. 4.25.

Из рис. 4.24 можно сделать следующие выводы:

- получаемые выражения производных программой не упрощаются. Для упрощения результата следует использовать функцию `simplify`;
- система позволяет получать решения в сложных случаях, когда производная выражается через специальные функции (см. вычисление производной $x!$);
- вычисление производных система осуществляет в аналитическом виде при неограниченном числе символьных переменных.

Функция $D[f, \{x, n\}]$ вычисляет частную производную n -го порядка функции f по переменной x .

```

D[a x^3 + b x - c, x]
b + 3 a x^2
D[Log[a x] / x^2, x]
1/x^3 - 2 Log[a x]/x^3
D[Sinh[a x], x]
a Cosh[a x]
D[(x - 1) / (x + 1), x]
-1/x^2 + 1/(x + 1)^2
Simplify[%]
2/(1 + x)^2
D[x!, x]
Gamma[1 + x] PolyGamma[0, 1 + x]
D[(Sin[x] + Cos[x])^2, x]
2 (Cos[x] - Sin[x]) (Cos[x] + Sin[x])
Simplify[%]
2 Cos[2 x]

```

Рис. 4.25. Решение примера 4.21

Пример 4.22

Определить третью производную функций примера 4.21.

Решение приведено на рис. 4.26.

Функция $D[f, x_1, x_2, \dots]$ возвращает производную функции f по параметрам $x_1, x_2, \dots$

Пример 4.23

Необходимо найти производные по всем параметрам следующих функций:

$$a^2 x^2, (\sin ax + \cos ax)^2, a^3 b^3 x^3, a^3 + b^3 + x^3, \lg(a^2 b^2 x^2) / axb.$$

Решение приведено на рис. 4.27.

Функция $Dt[f, x]$ есть обобщенная производная функции f по переменной x .

Функция $Dt[f]$ возвращает дифференциал функции f .

D[a x^3 + b x - c, {x, 3}]

6 a

D[Log[a x] / x^2, {x, 3}]

$$\frac{26}{x^5} - \frac{24 \operatorname{Log}[a x]}{x^5}$$

D[Sinh[a x], {x, 3}]

a^3 Cosh[a x]

D[(x - 1) / (x + 1), {x, 3}]

$$-\frac{6(-1+x)}{(1+x)^4} + \frac{6}{(1+x)^3}$$

Simplify[%]

$$\frac{12}{(1+x)^4}$$

D[x!, {x, 3}]

Gamma[1 + x] PolyGamma[0, 1 + x]^3 +
3 Gamma[1 + x] PolyGamma[0, 1 + x] PolyGamma[1, 1 + x] +
Gamma[1 + x] PolyGamma[2, 1 + x]

D[(Sin[x] + Cos[x])^2, {x, 3}]

$$6(-\operatorname{Cos}[x] - \operatorname{Sin}[x])(\operatorname{Cos}[x] - \operatorname{Sin}[x]) +$$

$$2(-\operatorname{Cos}[x] + \operatorname{Sin}[x])(\operatorname{Cos}[x] + \operatorname{Sin}[x])$$

Simplify[%]

-8 Cos[2 x]

Рис. 4.26. Определение третьей производной функций примера 4.21

D[a^2 x^2, a, x]

4 a x

D[(Sin[a x] + Cos[a x])^2, a, x]

$$2(a \operatorname{Cos}[a x] - a \operatorname{Sin}[a x])(x \operatorname{Cos}[a x] - x \operatorname{Sin}[a x]) +$$

$$2(\operatorname{Cos}[a x] + \operatorname{Sin}[a x])(\operatorname{Cos}[a x] - a x \operatorname{Cos}[a x] - \operatorname{Sin}[a x] - a x \operatorname{Sin}[a x])$$

Simplify[%]

$$2(\operatorname{Cos}[2 a x] - 2 a x \operatorname{Sin}[2 a x])$$

D[a^3 b^3 x^3, a, b, x]

27 a^2 b^2 x^2

D[a^3 + b^3 + x^3, a, b, x]

0

D[Log[a^2 b^2 x^2] / a b x, a, b, x]

$$-\frac{2}{a^2} - \frac{\operatorname{Log}[a^2 b^2 x^2]}{a^2}$$

Рис. 4.27. Производные многопараметрических функций

Приведем примеры получения дифференциала и обобщенной производной функции f .

Пример 4.24

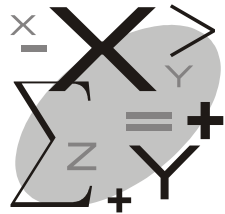
Пусть необходимо найти дифференциал функций $x \sin x$, $\lg(3x/4)$, а также обобщенную производную функций x^n , xe^x , $(x-1)/(x+1)$. Производную последней функции найти третьего порядка.

Решение примера показано на рис. 4.28.

$$\begin{aligned}
 &\mathbf{Dt[x Sin[x]]} \\
 &x \cos[x] \, Dt[x] + Dt[x] \sin[x] \\
 &\mathbf{Dt[Log[3 x/4]]} \\
 &\frac{Dt[x]}{x} \\
 &\mathbf{Dt[x^n, x]} \\
 &x^n \left(\frac{n}{x} + Dt[n, x] \log[x] \right) \\
 &\mathbf{Dt[x Exp[x], x]} \\
 &e^x + e^x x \\
 &\mathbf{Dt[(x-1)/(x+1), \{x, 3\}]} \\
 &-\frac{6(-1+x)}{(1+x)^4} + \frac{6}{(1+x)^3}
 \end{aligned}$$

Рис. 4.28. Примеры реализации функции Dt

ГЛАВА 5



Представление данных. Создание векторов и матриц

5.1. Типы данных

Основными типами данных, которыми оперирует система Mathematica, являются:

- ☐ числовые данные;
- ☐ символьные данные;
- ☐ списки.

Числовые данные могут быть целочисленными (Integer), рациональными (Rational), вещественными (Real) и комплексными (Complex). При этом числа могут быть представлены с произвольным основанием. В информатике и вычислительной технике чаще всего применяются двоичная, восьмеричная, десятичная и шестнадцатеричная системы счисления (СС). В табл. 5.1 приведены примеры чисел перечисленных типов в различных системах счисления.

Таблица 5.1. Типы чисел в различных системах счисления

Обозначение Тип чисел	Integer (целочис- ленные)	Rational (рацио- нальные)	Real (вещест- венные)	Complex (комплекс- ные)
Десятичная систе- ма счисления	179	38/29	45.375	-5.5+2.5 i
Двоичная система счисления	10110011	100110/11101	101101.011	-101.1+10.1 i
Восьмеричная система счисления	263	46/35	55.3	5.4+2.4 i
Шестнадцатерич- ная система счис- ления	b3	26/1d	2d.6	-5.8+2.8 i

Перевод чисел из десятичной СС в систему с основанием n (до 32) осуществляется в системе Mathematica с помощью функции

`BaseForm[N10, Nn]`

где:

□ N_{10} — число, представленное в десятичной СС;

□ N_n — число в СС с основанием n .

Пример 5.1

Представить числа 328, 625.15, 105/37, $-79+12.6i$ в двоичной, восьмеричной и шестнадцатеричной системах счисления, используя функцию `BaseForm[N0, Nn]`.

Решение приведено на рис. 5.1.

```
BaseForm[328,2]
1010010002
BaseForm[328,8]
5108
BaseForm[328,16]
14816
BaseForm[625.15,2]
1.0011100010010011012 × 29
BaseForm[625.15,8]
1161.1158
BaseForm[625.15,16]
271.2616
BaseForm[105/37,2]
11010012
1001012
BaseForm[105/37,8]
1518
458
BaseForm[105/37,16]
6916
2516
BaseForm[-79+12.6 i,2]
-10011112 + 1100.1001100110011012 i
BaseForm[-79+12.6 i,8]
-1178 + 14.463158 i
BaseForm[-79+12.6 i,16]
-4f16 + c.999a16 i
```

Рис. 5.1. Перевод чисел в различные системы счисления

Для перевода чисел из системы счисления с основанием n в десятичную используется функция $n^{\wedge}N_n$.

Пример 5.2

Перевести в десятичную систему счисления следующие числа:

$$N_2 = 1100101,101; N_8 = 372164; N_{16} = 125abe.$$

Решение приведено на рис. 5.2.

$2^{\wedge}1100101.101$
101.625
$8^{\wedge}372164$
128116
$16^{\wedge}125abe$
1202878

Рис. 5.2. Перевод чисел в десятичную систему счисления

Система Mathematica относится к классу систем символьной математики (компьютерной алгебры). Поэтому она выполняет математические действия над целыми числами без погрешностей и ограничений на разрядность. Арифметические операции над рациональными числами она выполняет абсолютно точно. Так, например, суммируя числа $2/3 + 5/7 + 1/3$, она выдает результат: $12/7$. Программа привела числа к общему знаменателю и выполнила операцию сокращения.

5.1.1. Вещественные числа

Вещественное число состоит из мантиссы и порядка. Мантисса содержит целую и дробную части, порядок представляется как степень числа 10. Они могут иметь различную форму представления и в теории информации часто называются числами с плавающей запятой (точкой). Например, 216.35 , $.21635$, $103,21635 \cdot 10^{-2}$. Целая часть мантиссы от дробной отделяется точкой, мантисса от порядка — пробелом или знаком умножения, перед дробной мантиссой ноль целых можно опустить, оставив лишь точку, отделяющую целую часть от дробной.

Возможные формы представления вещественных чисел показаны на рис. 5.3 при умножении числа 0.25 на 3 .

Обратите внимание на то, что при представлении произведения в виде $250 \cdot 10^{-3} \cdot 3$ и $2500 \cdot 10^{-4} \cdot 3$ ответом является число, представленное в рациональной форме. Это объясняется тем, что вещественное число $250 \cdot 10^{-3}$ программа воспринимает как рациональное.

Любое целое число при наличии разделительной точки воспринимается системой как вещественное число (последняя строка примера).

```

0.25 3
0.75
.25 3
0.75
250 10^-3 3
  3
  4
2500*10^-4*3
  3
  4
.0025 10^2 3
0.75
250. 10^-3 3
0.75

```

Рис. 5.3. Представление числа в вещественной форме

Для представления выражения в форме вещественного числа в системе имеются функции:

$N[D]$

$N[D, n]$

где:

☐ D — выражение;

☐ n — число цифр результата.

На рис. 5.4 показаны все способы представления вещественного числа.

```

12/7
  12
  7
12./7
1.71429
N[12/7]
1.71429
N[12/7,30]
1.71428571428571428571428571429
N[E^1,100]
2.718281828459045235360287471352662497757247093
69959574966967627724076630353547594571382178525
1664274274498344858402`100.

```

Рис. 5.4. Способы представления вещественного числа

Максимальные и минимальные числа, которыми может оперировать система Mathematica, принимают значения: 1.78769×10^{308} — 1.78769×10^{-308} . Эти числа можно вывести на экран с помощью команд `$MaxMachineNumber`, `$MinMachineNumber`.

5.1.2. Комплексные числа

Комплексные числа в системе Mathematica задаются в общепринятой форме:

$$z = \text{Re}(z) + i \cdot \text{Im}(z)$$

или

$$z = \text{Re}(z) + i \text{Im}(z)$$

Мнимая единица отделяется от мнимой части комплексного числа знаком умножения или пробелом.

Функции `Re(z)` и `Im(z)` позволяют выделить вещественную и мнимую части комплексного числа, например:

<code>Re[7-5 I]</code>	<code>Im[7-5 I]</code>
7	-5

В системе Mathematica математические операции с комплексными числами в большинстве случаев выполняются так же, как и с вещественными числами.

5.1.3. Символьные переменные

Символьными переменными могут быть отдельные символы (a, b, c, ..), строки ("z z z z"), выражения (`Sin[x]`, `(a+b)^2`, `2 Log[x+1]`). При записи математических операций над выражениями применяются следующие синтаксические правила:

- ☐ знак умножения можно заменять пробелом;
- ☐ аргументы функций пишутся в квадратных скобках;
- ☐ встроенные функции начинаются с заглавной буквы;
- ☐ круглые скобки используются при выделении части математического выражения `((x-1)/(x+1))`;
- ☐ фигурные скобки используются при задании списков `Cos[{1, 3, 2, 5, 4}]`.

5.1.4. Списки и массивы

Списком называется совокупность данных, заключенных в фигурные скобки. При этом данными могут быть числа, символьные переменные, выражения. Примеры списков:

- ☐ `{1, 3, 5, 7, 9}` — список чисел;
- ☐ `{a, b, c, d, e}` — список символьных переменных;

- $\{\text{Sin}[x], \text{Cos}[x], \text{Log}[x+1], \text{Tanh}[x^2+3]\}$ — список функций;
- $\{(a+b)^2, (x-1)/(x+1), x \text{ Exp}[x]+1, x+2 \ y+z^2\}$ — список математических выражений.

Списки формируют данные, называемые *массивами*. Одноуровневым массивом является вектор, двухуровневым — матрица.

Рассмотрим основные математические действия над списками, уделив главное внимание векторам и матрицам.

5.2. Представление и образование векторов и матриц

Вектор и *матрица* представляют собой, соответственно, одноуровневый и двухуровневый списки. Элементами вектора и матрицы могут быть вещественные и комплексные числа, символьные переменные, функции и даже математические выражения. На рис. 5.5 приведены векторы и матрицы из различных элементов.

```
f1={1,2,3,4,5}
{1,2,3,4,5}
f2={a,b,c,d,e}
{a,b,c,d,e}
f3={2+3 i,1-2 i,5,3+7 i,7}
{2+3 i,1-2 i,5,3+7 i,7}
f4={{1,2,3},{4,7,0},{-5,1,8}}
{{1,2,3},{4,7,0},{-5,1,8}}
f5={{Sin[x],E^-x,(x-1)/(x+1),Log[x],5},
{1+2 i,3,5,Tan[x+1],-8},{4,a,b,Cos[x],2-i}}
{{{Sin[x],e^-x,-1+x/(1+x),Log[x],5},{1+2 i,3,5,Tan[1+x],-8},{4,a,b,Cos[x],2-i}}}
```

Рис. 5.5. Векторы и матрицы из различных элементов

Представление векторов и матриц в табличном виде возможно с помощью функций `TableForm` и `MatrixForm`, имеющих вид:

```
TableForm[f]
Out[n] // MatrixForm
```

где:

- f — имя вектора или матрицы;
- n — номер строки, в которой находится вектор или матрица;

- `%` — применяется в случае, если функция `% // MatrixForm` представления вектора или матрицы в табличном виде располагается вслед за вектором (матрицей).

Вектор или матрицу можно также создать с помощью функции `List`:

- `List[a, b, c, ...]` — создает вектор $\{a, b, c, \dots\}$;
- `List[{a, b, c, ...}, {d, e, f, ...}, {d, h, k, ...}]` — создает матрицу $\{\{a, b, c, \dots\}, \{d, e, f, \dots\}, \{g, y, k, \dots\}\}$.

На рис. 5.6 приведены в табличной форме вектор `f3` и матрицы `f4`, `f5` из рис. 5.5, образованные с помощью функций `TableForm` и `MatrixForm`.

TableForm[f3]				
2+3 i				
1-2 i				
5				
3+7 i				
7				
TableForm[f4]				
1	2	3		
4	7	0		
-5	1	8		
TableForm[f5]				
Sin[x]	e^{-x}	$\frac{-1+x}{1+x}$	Log[x]	5
1+2 i	3	5	Tan[1+x]	-8
4	a	b	Cos[x]	2-i
Out[29]/MatrixForm				
$\left(\begin{array}{ccccc} \text{Sin}[x] & e^{-x} & \frac{-1+x}{1+x} & \text{Log}[x] & 5 \\ 1+2 i & 3 & 5 & \text{Tan}[1+x] & -8 \\ 4 & a & b & \text{Cos}[x] & 2-i \end{array} \right)$				

Рис. 5.6. Представление векторов и матриц в табличной форме

5.2.1. Генерация векторов и матриц с помощью функции *Range*

Функция `Range` применяется для создания числовых списков и представляется в следующих видах:

- `Range[nmax]` — генерирует вектор числовых элементов вида $\{1, 2, \dots, n_{\max}\}$;
- `Range[nmin, nmax]` — генерирует вектор из числовых элементов вида $\{n_{\min}, n_{\max}\}$;

- `Range[nmin, nmax, dn]` — генерирует вектор числовых элементов от $n_{\min}$ до $n_{\max}$ с шагом dn .

Примеры реализации функции приведены на рис. 5.7.

```
Range[7]
{1, 2, 3, 4, 5, 6, 7}

Range[4, 10]
{4, 5, 6, 7, 8, 9, 10}

Range[3, 8, 0.5]
{3, 3.5, 4., 4.5, 5., 5.5, 6., 6.5, 7., 7.5, 8.}
```

Рис. 5.7. Генерация векторов с помощью функции `Range`

5.2.2. Генерация векторов и матриц с помощью функций *Table*

Создание списков типа вектора и матрицы в системе Mathematica эффективно с помощью функции `Table`, имеющей вид:

- `Table[f, {nmax}]` — создает $n_{\max}$ экземпляров выражения f ;
- `Table[f, {1, nmax}]` — создает список функций f при изменении переменной n от 1 до $n_{\max}$;
- `Table[f, {n, nmin, nmax}]` — создает список функций f при изменении переменной n от $n_{\min}$ до $n_{\max}$;
- `Table[f, {n, nmin, nmax, dn}]` — создает список функций f при изменении переменной n от $n_{\min}$ до $n_{\max}$ с шагом dn ;

```
Table[Log[x], {5}]
{Log[x], Log[x], Log[x], Log[x], Log[x]}

Table[Log[x], {x, 5}]
{0, Log[2], Log[3], Log[4], Log[5]}

Table[Log[x], {x, 5, 10}]
{Log[5], Log[6], Log[7], Log[8], Log[9], Log[10]}

Table[Log[x], {x, 1, 3, 0.5}]
{0, 0.405465, 0.693147, 0.916291, 1.09861}

Table[i+j, {i, 2, 4}, {j, 2, 4}]
{{4, 5, 6}, {5, 6, 7}, {6, 7, 8}}
```

Рис. 5.8. Генерация векторов и матриц с помощью функций `Table`

- `Table[f, {n1, n1min, n1max}, {n2, n2min, n2max}, ...]` — откликом является вложенный список по переменной `n`.

Примеры генерации списков приведены на рис. 5.8.

5.2.3. Выделение и вывод элементов вектора и матрицы

Система Mathematica имеет богатые возможности выделения элементов векторов и матриц с расположением их на экране монитора. Выделение осуществляется с помощью следующих трех способов:

- использование двойных квадратных скобок;
- применение функции `Part`;
- применение функции `Select`.

Рассмотрим эти способы и приведем примеры.

Использование двойных квадратных скобок

В этом случае выражение, выделяющее элементы вектора или матрицы, представляется в виде:

`f[[n]]`

`f[[n1, n2, ...]]`

где:

- `f` — имя вектора или матрицы;
- `n` — выделяемый элемент;
- `ni` — *i*-й элемент из совокупности выделяемых элементов.

Пример 5.3

Заданы следующие векторы и матрица:

`f1={2, 1, 4, 3, 5}`

`f2={{1, 2, 3}, {3, 5, 7},{2, 4, 6}}`

`f3={a, 1, b, 2, 3, c}`

`f4={2, a, Sin[x+y^2], b, 1}`

Необходимо выделить четвертый элемент вектора `f1`, элемент второй строки и второго столбца матрицы `f2`, второй и третий элементы вектора `f1`, второй элемент первой строки и первый элемент третьей строки матрицы `f2`.

Решение приведено на рис. 5.9.

```

f1={2,1,4,3,5}
{2,1,4,3,5}
f2={{1,2,3},{3,5,7},{2,4,6}}
{{1,2,3},{3,5,7},{2,4,6}}
f3={a,1,b,2,3,c}
{a,1,b,2,3,c}
f4={2,a,Sin[x+y^2],b,1}
{2, a, Sin[x+y^2], b, 1}
f1[[4]]
3
f2[[2,2]]
5
f1[[{2,3}]]
{1,4}
{f2[[1,2]],f2[[3,1]]}
{2,2}

```

Рис. 5.9. Выделение элементов векторов и матрицы с помощью квадратных скобок

Выделение элементов вектора и матрицы с помощью функции *Part*

Функция *Part* позволяет выделить не только элемент вектора или матрицы. Она дает возможность выделить элемент из сложного выражения, которое является элементом вектора или матрицы.

Функция *Part* представляется в следующем виде:

```
{Part[f, n1], Part[f, n2], ...}
```

где:

- *f* — имя вектора;
- *n_i* — *i*-й элемент вектора *f*.

Символ *n_i* может быть положительным и отрицательным. Если *n_i* > 0, то отсчет элементов вектора идет от начала вектора, а при *n_i* < 0 — с конца.

В случае выделения элементов матрицы функция *Part* имеет вид:

```
{Part[f, n1, m1], Part[f, n2, m2], ...}
```

где:

- *n_i* — *i*-й элемент строки матрицы;
- *m_i* — *i*-й элемент столбца матрицы.

В случае выделения элементов из сложных элементов вектора или матрицы функция `Part` представляется в следующем виде:

```
Part[f, n, m, l]
```

где:

- `f` — имя вектора или матрицы;
- `n` — номер элемента вектора `f`;
- `m` — уровень выражения (`m=1` в случае вектора, `m=2` в случае матрицы);
- `l` — номер элемента в векторе или матрице.

Пример 5.4

Даны векторы `f3` и `f4` из примера 5.3. Необходимо выделить из вектора `f3` элементы с номерами 2, 5, 6 и -3, -5, -6. Из вектора `f4` выделить четвертый элемент, а также x и y^2 из третьего элемента.

Решение приведено на рис. 5.10.

```
f3={a,1,b,2,3,c}
{a,1,b,2,3,c}
f4={2,a,Sin[x+y^2],b,1}
{2,a,Sin[x+y^2],b,1}
{Part[f3,2],Part[f3,5],Part[f3,6]}
{1,3,c}
{Part[f3,-3],Part[f3,-5],Part[f3,-6]}
{2,1,a}
{Part[f4,4],Part[f4,3,1,1],Part[f4,3,1,2]}
{b,x,y^2}
```

Рис. 5.10. Выделение элементов вектора с помощью функции `Part`

Вывод элементов векторов и матриц осуществляется с помощью функций `MatrixForm` и `TableForm`.

Функция `MatrixForm` выводит матрицу в матричной форме, функция `TableForm` — в виде таблицы. Разница этих форм вывода видна на рис. 5.11.

Функции `MatrixForm` и `TableForm` имеют большое число опций, позволяющих располагать векторы и матрицы на экране дисплея в желаемой форме.

Все используемые опции можно вывести на экран, набрав функции `Options[MatrixForm]` и `Options[TableForm]`.

```

F={ {a,1,4} , {b,2,5} , {c,3,6} }
  {{a,1,4},{b,2,5},{c,3,6}}
MatrixForm[F]

$$\begin{pmatrix} a & 1 & 4 \\ b & 2 & 5 \\ c & 3 & 6 \end{pmatrix}$$

TableForm[F]

|   |   |   |
|---|---|---|
| a | 1 | 4 |
| b | 2 | 5 |
| c | 3 | 6 |


```

Рис. 5.11. Представление списка элементов в виде матрицы и таблицы

```

s={5,73426813438765,34}
  {5,73426813438765,34}
TableForm[s,TableAlignments->Left]
  5
  73426813438765
  34
TableForm[s,TableAlignments->Center]
      5
  73426813438765
      34
TableForm[s,TableAlignments->Right]
      5
  73426813438765
      34

s1={{1,3,4},{5,1,7},{3,2,1}}
  {{1,3,4},{5,1,7},{3,2,1}}
TableForm[s1]

|   |   |   |
|---|---|---|
| 1 | 3 | 4 |
| 5 | 1 | 7 |
| 3 | 2 | 1 |

TableForm[s1,TableSpacing->{1,1}]

|   |   |   |
|---|---|---|
| 1 | 3 | 4 |
| 5 | 1 | 7 |
| 3 | 2 | 1 |

TableForm[s1,TableSpacing->{5,2}]

|   |   |   |
|---|---|---|
| 1 | 3 | 4 |
| 5 | 1 | 7 |
| 3 | 2 | 1 |

TableForm[s1,TableSpacing->{2,5}]

|   |   |   |
|---|---|---|
| 1 | 3 | 4 |
| 5 | 1 | 7 |
| 3 | 2 | 1 |


```

Рис. 5.12. Расположение векторов и матриц в желаемом виде

На рис. 5.12 показано использование опций `TableAlignments` и `TableSpacing`, позволяющих расположить вектор и матрицу на экране в желаемом виде. Технология очевидна и объяснений не требует.

5.3. Работа со списками.

Создание векторов и матриц

Списки представляют собой сложную структуру данных. Система *Mathematica* позволяет осуществлять контроль структуры данных, изменять порядок расположения элементов, осуществлять их удаление, включать новые элементы. Для реализации этих процедур существует большое число встроенных функций. Рассмотрим эти функции и приведем примеры. Главное внимание уделим спискам типа "вектор" и "матрица".

5.3.1. Выявление структуры вектора или матрицы

Для выявления структуры вектора или матрицы служат следующие функции:

- ☐ `VectorQ[V]` — проверяет, является ли `V` вектором, и выдает `True`, если да, и `False`, если нет;
- ☐ `MatrixQ[M]` — проверяет, является ли `M` матрицей, и выдает `True`, если да, и `False`, если нет;
- ☐ `Length[V]` — возвращает число элементов вектора `V`;
- ☐ `Length[M]` — возвращает число строк матрицы `M`;
- ☐ `MemberQ[V, n]` — проверяет, имеется ли в векторе `V` элемент `n`; если да, то выдает `True`, если нет — `False`;
- ☐ `FreeQ[V, n]` — проверяет, имеет ли вектор `V` элемент `n`; если содержит, то выдает `False`, если нет — `True`;
- ☐ `FreeQ[M, n]` — проверяет, имеет ли матрица `M` элемент `n`; если содержит, то выдает `False`, если нет — `True`;
- ☐ `Dimensions[V]` — возвращает число элементов вектора `V`;
- ☐ `Dimensions[M]` — возвращает размер матрицы (число строк и число столбцов);
- ☐ `Position[V, n]` — возвращает номера позиций элемента `n` вектора `V`;
- ☐ `Count[V, n]` — возвращает число элементов в векторе `V`, имеющих значение `n`;
- ☐ `TensorRank[V]` — выдает ранг вектора `V`, если `V` является тензором;
- ☐ `TensorRank[M]` — выдает ранг матрицы `M`, если `M` является тензором.

Пример 5.5

Дан вектор V и матрица M :

$V = \{1, 3, 4, 5, 7, 2, 4\}$, $M = \{\{1, 3, 5\}, \{2, 4, 6\}, \{1, 2, 3\}\}$

Выявить структуру вектора и матрицы, воспользовавшись функциями `Count`, `Dimensions`, `Free`, `Length`, `Matrix`, `Member`, `Position`, `Vector`, `TensorRank`.

Решение приведено на рис. 5.13.

```

V={1,3,4,5,7,2,4}
{1,3,4,5,7,2,4}
M={{1,3,5},{2,4,6},{1,2,3}}
{{1,3,5},{2,4,6},{1,2,3}}
VectorQ[V]
True
MatrixQ[V]
False
{MatrixQ[V],MatrixQ[M]}
{False,True}
Length[V]
7
Length[M]
3
MemberQ[V,0]
False
FreeQ[V,3]
False
FreeQ[M,8]
True
Dimensions[V]
{7}
Dimensions[M]
{3,3}
Position[V,4]
{{3},{7}}
Count[V,4]
2
TensorRank[V]
1
TensorRank[M]
2

```

Рис. 5.13. Структура вектора и матрицы

5.3.2. Преобразование и создание векторов и матриц

Система Mathematica имеет богатые возможности преобразования и создания новых векторов и матриц (и списков любого уровня). Это достигается путем выполнения следующих операций:

- ☐ удаление элементов;
- ☐ добавление элементов;
- ☐ изменение порядка расположения элементов;
- ☐ комбинирование векторов и матриц.

Эти действия выполняются с помощью большого числа функций. Приведем основные из них:

- ☐ `Drop[V, n]` — удаляет первые n элементов вектора V ;
- ☐ `Drop[V, -n]` — удаляет последние n элементов вектора V ;
- ☐ `Drop[V, {n}]` — удаляет n -й элемент вектора V ;
- ☐ `Drop[V, {m, n}]` — удаляет элементы от m -го до n -го вектора V ;
- ☐ `Last[f]` — возвращает последний элемент вектора (матрицы) f ;
- ☐ `Rest[V]` — возвращает вектор V с отброшенным первым элементом;
- ☐ `Take[V, n]` — возвращает вектор V с первыми n элементами;
- ☐ `Take[V, -n]` — возвращает вектор V с последними n элементами;
- ☐ `Take[V, {m, n}]` — возвращает вектор V с номерами элементов от m до n ;
- ☐ `Append[V, a]` — добавляет элемент a в конец вектора V ;
- ☐ `Prepend[V, -a]` — добавляет элемент a в начало вектора V ;
- ☐ `Insert[V, a, n]` — вставляет элемент a в позицию n с отсчетом из начала вектора V ;
- ☐ `Insert[V, a, -n]` — вставляет элемент a в позицию n с отсчетом из конца вектора V ;
- ☐ `Delete[V, n]` — удаляет элемент n из вектора V ;
- ☐ `{Delete[f, n1], Delete[f, n2], Delete[f, n3], ...}` — удаляет из вектора (матрицы) f элементы n_1 и образует новую матрицу.

Пример 5.6

Дан вектор V и матрица M из примера 5.5. Необходимо образовать новые вектор и матрицу путем удаления и увеличения их элементов. Следует воспользоваться функциями `Drop`, `Delete`, `Last`, `Rest`, `Take`, `Append`, `Prepend`, `Insert`.

Решение задачи приведено на рис. 5.14.

```

V={1,3,4,5,7,2,4}
{1,3,4,5,7,2,4}
M={{1,3,5},{2,4,6},{1,2,3}}
{{1,3,5},{2,4,6},{1,2,3}}
Drop[V,3],Drop[V,−4],Drop[V,{5}],Drop[V,{2,5}]
{{5,7,2,4},{1,3,4},{1,3,4,5,2,4},{1,2,4}}
Delete[V,6]
{1,3,4,5,7,4}
Delete[V,1],Delete[V,−3],Delete[V,7]}
{{3,4,5,7,2,4},{1,3,4,5,2,4},{1,3,4,5,7,2}}
Delete[M,{2,3}]
{{1,3,5},{2,4},{1,2,3}}
Delete[M,{3,1}]
{{1,3,5},{2,4,6},{2,3}}
Last[V]
4
Last[M]
{1,2,3}
Rest[V]
{3,4,5,7,2,4}
Take[V,4],Take[V,−4],Take[V,{3,6}]
{{1,3,4,5},{5,7,2,4},{4,5,7,2}}
Append[V,Sin[x]]
{1,3,4,5,7,2,4,Sin[x]}
Prepend[V,Sin[x]]
{Sin[x],1,3,4,5,7,2,4}
Insert[V,a,5]
{1,3,4,5,a,7,2,4}
Insert[V,{3,a,Sin[x]},5]
{1,3,4,5,{3,a,Sin[x]},7,2,4}.

```

Рис. 5.14. Образование новых векторов и матриц путем изменения числа элементов исходных списков

Образование новых векторов и матриц в системе Mathematica возможно также путем изменения расположения элементов вектора или матрицы. Для этих целей система имеет следующие функции:

- ☐ **Flatten**[**M**] — образует вектор из любого списка, например, матрицы;
- ☐ **Flatten**[**M**, **n**] — образует вектор в строке **n** матрицы **M**;
- ☐ **Sort**[**f**] — сортирует элементы вектора (матрицы) **f** в естественном порядке;

- ❑ `Reverse[f]` — возвращает вектор (матрицу) `f` в обратном порядке расположения элементов;
- ❑ `RotateLeft[f]` — возвращает элементы вектора (матрицы) `f` после однократного поворота влево;
- ❑ `RotateLeft[f, n]` — возвращает элементы вектора (матрицы) `f` после `n`-кратного поворота влево;
- ❑ `RotateRight[f]` — возвращает элементы вектора (матрицы) `f` после однократного поворота вправо;
- ❑ `RotateRight[f, n]` — возвращает элементы вектора (матрицы) `f` после `n`-кратного поворота вправо;
- ❑ `Transpose[M]` — смена строк и столбцов матрицы.

```

V={1,3,6,2,4,5}
{1,3,6,2,4,5}
M={{2,1,4},{3,8,6},{1,2,3}}
{{2,1,4},{3,8,6},{1,2,3}}
Flatten[M]
{2,1,4,3,8,6,1,2,3}
FlattenAt[M,2]
{{2,1,4},{3,8,6},{1,2,3}}
Sort[V]
{1,2,3,4,5,6}
Sort[M]
{{1,2,3},{2,1,4},{3,8,6}}
Reverse[V]
{5,4,2,6,3,1}
Reverse[M]
{{1,2,3},{3,8,6},{2,1,4}}
RotateLeft[V,3]
{2,4,5,1,3,6}
RotateRight[V,2]
{4,5,1,3,6,2}
RotateLeft[M,1]
{{3,8,6},{1,2,3},{2,1,4}}
RotateRight[M,1]
{{1,2,3},{2,1,4},{3,8,6}}
Transpose[M]
{{2,3,1},{1,8,2},{4,6,3}}

```

Рис. 5.15. Создание новых векторов и матриц изменением порядка расположения элементов

Пример 5.7

Даны вектор V и матрица M :

$$V = \{1, 3, 6, 2, 4, 5\}$$
$$M = \{\{2, 1, 4\}, \{3, 8, 6\}, \{1, 2, 3\}\}$$

Необходимо создать новые векторы и матрицы путем изменения порядка расположения элементов вектора V и матрицы M , используя следующие функции: Flatten, Sort, Reverse, Rotate, Transpose.

Решение задачи приведено на рис. 5.15.

5.3.3. Комбинирование векторов и матриц

Комбинирование векторов и матриц позволяет создавать новые векторы и матрицы, т. е. создавать сложные структуры данных, выполнять операции с множествами.

Комбинирование векторов и матриц осуществляется с помощью следующих функций:

- ☐ Union[F] — возвращает новый вектор или матрицу, из которого удалены повторяющиеся элементы;
- ☐ Union[f₁, f₂, ...] — объединяет f₁, f₂, ..., удаляя повторяющиеся элементы векторов и матриц;
- ☐ Join[f₁, f₂, ...] — объединяет f₁, f₂, ... в единую цепь (конкатенация);
- ☐ Complement[f₁, f₂, ...] — возвращает список f, элементы которого не содержатся в f₁, f₂, ...;
- ☐ Intersection[f₁, f₂, ...] — возвращает упорядоченный список элементов, общих для всех списков.

Пример 5.8

Даны следующие векторы V_1 , V_2 , V_3 и матрицы M_1 , M_2 :

$$V_1 = \{a, 3, 1, 7, 2, 4, 6\}$$
$$V_2 = \{1, b, 3, 4, 5, 8\}$$
$$V_3 = \{4, 5, c, 7, 8, 1\}$$
$$M_1 = \{\{1, 2, 4\}, \{3, 5, 8\}, \{7, 2, 3\}\}$$
$$M_2 = \{\{a, b, c\}, \{1, 2, 3\}, \{4, 5, 6\}\}$$

Необходимо выполнить комбинирование векторов и матриц, пользуясь функциями: Union, Join, Complement, Intersection.

Решение приведено на рис. 5.16.


```

V1={a,3,1,7,2,4,6}
{a,3,1,7,2,4,6}
V2={1,b,3,4,5,8}
{1,b,3,4,5,8}
V3={4,5,c,7,8,1}
{4,5,c,7,8,1}
M1={{1,2,4},{3,5,8},{7,2,3}}
{{1,2,4},{3,5,8},{7,2,3}}
M2={{a,b,c},{1,2,3},{4,5,6}}
{{a,b,c},{1,2,3},{4,5,6}}
Union[V1,V2,V3,M1,M2]
{1,2,3,4,5,6,7,8,a,b,c,{1,2,3},{1,2,4},{3,5,8},{4,5,6},
{7,2,3},{a,b,c}}
Union[{1,2,3,1,4,5,2,4,1,7}]
{1,2,3,4,5,7}
Join[V1,V2,V3,M1,M2]
{a,3,1,7,2,4,6,1,b,3,4,5,8,4,5,c,7,8,1,{1,2,4},{3,5,8},
{7,2,3},{a,b,c},{1,2,3},{4,5,6}}
Complement[V1,V2,V3]
{2,6,a}
Intersection[V1,V2,V3]
{1,4}

```

Рис. 5.16. Комбинирование векторов и матриц

5.3.4. Создание векторов и матриц

Система Mathematica позволяет создавать массивы данных любых размерностей. Создание векторов и матриц осуществляется с помощью следующих функций:

```

Array[f, n]
Array[f, {n1, n2, ...}]
Array[f, k, l]
Array[f, k, l, h]

```

Рассмотрим эти функции и приведем примеры.

- ☐ `Array[f, n]` — создает вектор из n элементов $f[1], f[2], \dots, f[n]$;
- ☐ `Array[f, {n1, n2}]` — создает вектор из n_1 элементов, начиная с $f[n_2]$, при этом n_2 может быть числом, функцией, выражением;
- ☐ `Array[f, {n1, n2}]` — создает матрицу $n_1 \times n_2$ из элементов функции $f(n_1, n_2)$;

□ `Array[f, n1, n2, h]` — создает вектор из n_1 элементов, начиная с $f(n_2)$, с уровнем массива h .

Примеры создания векторов с помощью функций `Array[f, n]`, `Array[f, n1, n2]`, `Array[f, {n1, n2}]`, `Array[f, n1, n2, h]` приведены на рис. 5.17 и 5.18. Обратите внимание на то, что эти функции позволяют осуществлять суммирование и умножение элементов вектора с помощью функций `Plus` и `Times`.

```

Array[Tan, 5]
{Tan[1], Tan[2], Tan[3], Tan[4], Tan[5]}

Array[Sin, 5, 3]
{Sin[3], Sin[4], Sin[5], Sin[6], Sin[7]}

Array[Sin, 5, Sqrt[3]]
{Sin[Sqrt[3]], Sin[1 + Sqrt[3]], Sin[2 + Sqrt[3]], Sin[3 + Sqrt[3]], Sin[4 + Sqrt[3]]}

N[%]
{0.987027, 0.398189, -0.556742, -0.999807, -0.523654}

Array[Sin, 5, Sqrt[3], Plus]
Sin[Sqrt[3]] + Sin[1 + Sqrt[3]] + Sin[2 + Sqrt[3]] + Sin[3 + Sqrt[3]] + Sin[4 + Sqrt[3]]

N[%]
-0.694987

Array[Sin, 5, Sqrt[3], Times]
Sin[Sqrt[3]] Sin[1 + Sqrt[3]] Sin[2 + Sqrt[3]] Sin[3 + Sqrt[3]] Sin[4 + Sqrt[3]]

N[%]
-0.11456

```

Рис. 5.17. Создание векторов и матриц с помощью функций `Array[f, n]` и `Array[f, n1, n2]`

```

Array[Tan, {3, 4}]
{{Tan[1, 1], Tan[1, 2], Tan[1, 3], Tan[1, 4]}, {Tan[2, 1], Tan[2, 2], Tan[2, 3], Tan[2, 4]}, {Tan[3, 1], Tan[3, 2], Tan[3, 3], Tan[3, 4]}}

Array[Exp, 5, 3, 2]
2[e3, e4, e5, e6, e7]

N[%]
2[e3, e4, e5, e6, e7]

Array[Exp, 5, a, h]
h[ea, e1+a, e2+a, e3+a, e4+a]

```

Рис. 5.18. Образование векторов и матрицы с помощью функций `Array[f, {n1, n2}]` и `Array[f, n1, n2, h]`

5.4. Математические операции над векторами и матрицами

5.4.1. Арифметические операции

Математические операции над векторами и матрицами осуществляются так же, как и над числами. Результатами этих операций являются векторы и матрицы с новыми значениями элементов.

Рассмотрим эти операции на примерах.

Пример 5.9

Даны векторы $v1$, $v2$ и матрицы $m1$, $m2$:

$$v1 = \{1, 3, 5, 2, 6, 4\}$$
$$v2 = \{2, 7, 5, 8, 1, 3\}$$
$$m1 = \{\{1, 2, 3\}, \{6, 5, 4\}, \{1, 3, 5\}\}$$
$$m2 = \{\{3, 2, 1\}, \{4, 5, 6\}, \{5, 3, 1\}\}$$

Необходимо:

- ☐ сложить, вычесть, умножить и разделить вектор $v1$ и матрицу $m1$ на число 3;
- ☐ возвести в квадрат вектор $v2$ и матрицу $m2$;
- ☐ извлечь квадратный корень из вектора $v1$ и матрицы $m1$;
- ☐ вычислить e^{v1} , $\sin v2$, $\ln M1$, $\cosh M2$.

Решение приведено на рис. 5.19.

Из рис. 5.19 видно, что во всех случаях выполняются операции с элементами векторов и матриц как с обычными числами. Следует иметь в виду, что система может выполнять так же операции с символьными переменными и даже с функциями.

5.4.2. Матричное умножение

Векторные операции умножения и возведения в степень в системе Mathematica осуществляются с помощью встроенных функций. В частности, векторное произведение реализуется с помощью функции `Dot[v1, v2]`. Оно также реализуется знаком умножения в виде точки: $v1.v2$ или $m1.m2$.

Пример 5.10

Даны два вектора $v1$, $v2$ и две матрицы $m1$, $m2$ из примера 5.9. Необходимо выполнить операции почленного и матричного умножения векторов и матриц. Решение приведено на рис. 5.20.

```

V1={1,3,5,2,6,4}
{1,3,5,2,6,4}
V2={2,7,5,8,1,3}
{2,7,5,8,1,3}
M1={{1,2,3},{6,5,4},{1,3,5}}
{{1,2,3},{6,5,4},{1,3,5}}
M2={{3,2,1},{4,5,6},{5,3,1}}
{{3,2,1},{4,5,6},{5,3,1}}
{V1+3,V1-3,V1*3,V1/3}

{{4,6,8,5,9,7},{-2,0,2,-1,3,1},{3,9,15,6,18,12},
{1/3,1,5/3,2/3,2,4/3}}
{M1+3,M1-3,M1*3,M1/3}
{{{4,5,6},{9,8,7},{4,6,8}},{{-2,-1,0},{3,2,1},
{-2,0,2}},{{3,6,9},{18,15,12},{3,9,15}},
{{{1/3,2/3,1},{2,5/3,4/3},{1/3,1,5/3}}}}
{V2^2,M2^2,Sqrt[V1],Sqrt[M1]}
{{4,49,25,64,1,9},{9,4,1},{16,25,36},{25,9,1}},
{1,Sqrt[3],Sqrt[5],Sqrt[2],Sqrt[6],2},{{1,Sqrt[2],Sqrt[3],
{Sqrt[6],Sqrt[5],2},{1,Sqrt[3],Sqrt[5]}}}
{E^V1,Sin[V2],Log[M1],Cosh[M2]}
{{e,e^2,e^3,e^4,e^5},{Sin[2],Sin[7],Sin[5],Sin[8],
Sin[1],Sin[3]},{{0,Log[2],Log[3]},{Log[6],Log[5],
Log[4]},{0,Log[3],Log[5]}},{Cosh[3],Cosh[2],Cosh[1]},
{Cosh[4],Cosh[5],Cosh[6]},{Cosh[5],Cosh[3],Cosh[1]}}}
N[%]

{{2.71828,20.0855,148.413,7.38906,403.429,54.5982}
, {0.909297,0.656987,-
0.958924,0.989358,0.841471,0.14112},{0.,0.693147,
1.09861},{1.79176,1.60944,1.38629},{0.,1.09861,1.6
0944}},{{10.0677,3.7622,1.54308},{27.3082,74.2099,
201.716},{74.2099,10.0677,1.54308}}}

```

Рис. 5.19. Решение примера 5.9

5.4.3. Математические операции с векторами и матрицами

Математические операции с векторами и матрицами (вычисление определителя, преобразования матриц, создание специальных матриц и т. п.) выполняются в системе Mathematica с помощью встроенных функций.

```

V1={1,3,5,2,6,4}
{1,3,5,2,6,4}
V2={2,7,5,8,1,3}
{2,7,5,8,1,3}
M1={{1,2,3},{6,5,4},{1,3,5}}
{{1,2,3},{6,5,4},{1,3,5}}
M2={{3,2,1},{4,5,6},{5,3,1}}
{{3,2,1},{4,5,6},{5,3,1}}
V1*V2
{2,21,25,16,6,12}
V1.V2
82
M1.M2
{{26,21,16},{58,49,40},{40,32,24}}
Dot[V1,V2]
82
Dot[M1,M2]
{{26,21,16},{58,49,40},{40,32,24}}

```

Рис. 5.20. Решение примера 5.10

Рассмотрим основные из них:

- ☐ **Det[M]** — возвращает детерминант (главный определитель) матрицы;
- ☐ **IdentityMatrix[M]** — возвращает единичную матрицу: матрицу с диагональными элементами, равными 1, и остальными нулями;
- ☐ **Transpose[M]** — возвращает транспонированную матрицу, у которой строки заменены столбцами, а столбцы строками;
- ☐ **Inverse[M]** — возвращает обратную матрицу, т. е. такую, которая будучи умноженной на исходную дает единичную матрицу;
- ☐ **Tr[M]** — возвращает след матрицы (сумму диагональных элементов);
- ☐ **LinearSolve[M,b]** — возвращает вектор неизвестных матричного уравнения $M \cdot x = b$, где M — матрица коэффициентов системы уравнений, x — вектор неизвестных, b — вектор свободных членов;
- ☐ **Eigensystem[M]** — возвращает список собственных значений и собственных векторов квадратной матрицы M ;
- ☐ **Eigenvalues[M]** — возвращает список собственных значений квадратной матрицы M ;
- ☐ **Eigenvectors[M]** — возвращает список собственных векторов квадратной матрицы M ;
- ☐ **PseudoInverse[M]** — ищет псевдообратную матрицу для матрицы M .

Все перечисленные выше матричные операции иллюстрируются на рис. 5.21.

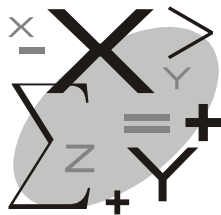
```

M1={{1,2,3},{3,5,4},{7,2,1}}
{{1,2,3},{3,5,4},{7,2,1}}
M2={{4,a,-2},{1,2,b},{3,5,c}}
{{4,a,-2},{1,2,b},{3,5,c}}
{Det[M1],Det[M2]}
{-40,2-20 b+3 a b+8 c-a c}
IdentityMatrix[3]
{{1,0,0},{0,1,0},{0,0,1}}
{Transpose[M1],Transpose[M2]}
{{{1,3,7},{2,5,2},{3,4,1}},{4,1,3},{a,2,5},{-2,b,c}}}
{Inverse[M1],Inverse[M2]}
{{{3/40,-1/10,7/40},{-5/8,1/2,-1/8},{29/40,-3/10,1/40}},
{{(2-20b+3ab+8c-ac)/(3b-c),(-10-ac)/(6+4c),(4+ab)/(-2-4b)},
{(2-20b+3ab+8c-ac)/(2-20b+3ab+8c-ac),(-2-4b)/(2-20b+3ab+8c-ac),
{1/(2-20b+3ab+8c-ac),(-20+3a)/(2-20b+3ab+8c-ac),(8-a)/(2-20b+3ab+8c-ac)}}}
{Tr[M1],Tr[M2]}
{7,6+c}
{LinearSolve[M1,{1,3,7}],LinearSolve[M2,{1,2,3}]}
{{1,0,0},{(-8-5b+3ab+2c-2ac)/(2-20b+3ab+8c-ac),(6-9b+7c)/(2-20b+3ab+8c-ac),(-17+3a)/(2-20b+3ab+8c-ac)}}
PseudoInverse[M1]
{{{3/40,-1/10,7/40},{-5/8,1/2,-1/8},{29/40,-3/10,1/40}}}
Eigenvalues[M1]
{Root[40-24#1-7#1^2+#1^3&,3],Root[40-24#1-7#1^2+#1^3&,1],Root[40-24#1-7#1^2+#1^3&,2]}
N[%]
{9.14592,-3.42345,1.27752}
Eigenvectors[M1]
{{{17/71+13/71 Root[40-24#1-7#1^2+#1^3&,3]-1/71 Root[40-24#1-7#1^2+#1^3&,3]^2,
-95/71-10/71 Root[40-24#1-7#1^2+#1^3&,3]+7/142 Root[40-24#1-7#1^2+#1^3&,3]^2,1},
{17/71+13/71 Root[40-24#1-7#1^2+#1^3&,1]-1/71 Root[40-24#1-7#1^2+#1^3&,1]^2,
-95/71-10/71 Root[40-24#1-7#1^2+#1^3&,1]+7/142 Root[40-24#1-7#1^2+#1^3&,1]^2,1},
{17/71+13/71 Root[40-24#1-7#1^2+#1^3&,2]-1/71 Root[40-24#1-7#1^2+#1^3&,2]^2,
-95/71-10/71 Root[40-24#1-7#1^2+#1^3&,2]+7/142 Root[40-24#1-7#1^2+#1^3&,2]^2,1}}
N[%]
{{0.735903,1.4973,1.},{-0.552462,-0.278106,1.},{0.450362,-1.43751,1.}}

```

Рис. 5.21. Преобразование матриц

ГЛАВА 6



Математические функции

Математические функции принято классифицировать на функции элементарные, специальные и функции пользователя.

Рассмотрим эти функции, способы их представления и вычисления.

6.1. Элементарные математические функции

К элементарным функциям принято относить следующие:

- ☐ `Abs[z]` — вычисляет абсолютное значение действительного числа и модуль комплексного числа z ;
- ☐ `Exp[z]` — вычисляет экспоненциальную функцию;
- ☐ `Log[z]` — вычисляет логарифм числа z ;
- ☐ `Log[b, z]` — вычисляет логарифм числа z с основанием b ;
- ☐ `Sqrt[z]` — вычисляет квадратный корень числа z ;
- ☐ `sign[z]` — возвращает -1 , если z отрицательное, 0 , если $z=0$, и 1 , если z положительное; если z — число комплексное, то возвращает отношение $z/\text{Abs}[z]$;
- ☐ `Sin[z]` — возвращает синус числа z ;
- ☐ `Sinh[z]` — возвращает гиперболический синус числа z ;
- ☐ `ArcSin[z]` — возвращает обратный синус числа z ;
- ☐ `ArcSinh[z]` — возвращает обратный гиперболический синус числа z ;
- ☐ `Cos[z]` — возвращает косинус числа z ;
- ☐ `Cosh[z]` — возвращает гиперболический косинус числа z ;
- ☐ `ArcCos[z]` — возвращает обратный косинус числа z ;

- ☐ `ArcCosh[z]` — возвращает обратный гиперболический синус числа z ;
- ☐ `Tan[z]` — возвращает тангенс числа z ;
- ☐ `Tanh[z]` — возвращает гиперболический тангенс числа z ;
- ☐ `ArcTan[z]` — возвращает обратный тангенс числа z ;
- ☐ `ArcTanh[z]` — возвращает обратный гиперболический тангенс числа z ;
- ☐ `Cot[z]` — возвращает котангенс числа z ;
- ☐ `Coth[z]` — возвращает гиперболический котангенс числа z ;
- ☐ `ArcCot[z]` — возвращает обратный котангенс числа z ;
- ☐ `ArcCoth[z]` — возвращает обратный гиперболический котангенс числа z ;
- ☐ `Sec[z]` — возвращает секанс числа z ;
- ☐ `sech[z]` — возвращает гиперболический секанс числа z ;
- ☐ `ArcSec[z]` — возвращает обратный секанс числа z ;
- ☐ `ArcSech[z]` — возвращает обратный гиперболический секанс числа z ;
- ☐ `Csc[z]` — возвращает косеканс числа z ;
- ☐ `Csch[z]` — возвращает гиперболический косеканс числа z ;
- ☐ `ArcCsc[z]` — возвращает обратный косеканс числа z ;
- ☐ `ArcCsch[z]` — возвращает обратный гиперболический косеканс числа z .

Существуют два основных способа вычисления элементарных функций: непосредственные вычисления и путем присвоения аргументу численного значения.

6.1.1. Непосредственное вычисление функции

По этому способу вычисляемая функция вводится с численным значением аргумента z , и для получения результата нажимается комбинация клавиш `<Shift>+<Enter>`. Пусть, например, необходимо вычислить значение функции $y = e^2$. Процедуры непосредственного вычисления функции будут иметь вид:

`Exp[2.]`

После нажатия комбинации клавиш `<Shift>+<Enter>` на экране появится ответ:

7.38906

Следует иметь в виду, что если при данном аргументе функция имеет точное значение, то она будет вычислена без символа "." (точка) в конце числа. Например, `Exp[0]`, `Cos[Pi/2]`, `Log[Exp[5]]`.

6.1.2. Присвоение аргументу численного значения

Пусть, например, необходимо вычислить значение функции $y = \ln 2$. Процедуры вычисления функции по этому способу будут иметь вид:

x:=2.

Log[x]

0.693147

Изложенные способы вычисления элементарных функций приведены на рис. 6.1.

```

Exp [2.]
7.38906
Log[Exp [5]]
5
x1:=2.
Log [x]
0.693147
x2:=Pi/2
Cos [x2]
0

```

Рис. 6.1. Способы вычисления элементарных функций

Описанные способы являются универсальными и могут применяться при вычислении любых функций.

6.2. Специальные математические функции

6.2.1. Ортогональные полиномы

Система Mathematica имеет встроенные функции вычисления ортогональных полиномов Чебышева, Лежандра, Эрмита, Лагерра, Гегенбауера, Якоби. Эти функции имеют вид:

- ☐ ChebyshevT[n, x] — вычисляет полином Чебышева первого рода;
- ☐ ChebyshevU[n, x] — вычисляет полином Чебышева второго рода;
- ☐ LegendreP[n, x] — вычисляет полином Лежандра;
- ☐ LegendreP[n, m, x] — вычисляет присоединенный полином Лежандра;

- `LegendreQ[n, x]` — вычисляет функцию Лежандра второго рода;
- `LegendreQ[n, m, x]` — вычисляет присоединенную функцию Лежандра второго рода;
- `HermiteH[n, x]` — вычисляет полином Эрмита;
- `LaguerreL[n, x]` — вычисляет полином Лагерра;
- `LaguerreL[n, m, x]` — вычисляет обобщенный полином Лагерра;
- `GegenbauerC[n, m, x]` — вычисляет полином Гегенбауэра;
- `JacobiP[n, a, b, x]` — вычисляет полином Якоби.

В функциях приняты следующие обозначения:

- n — степень полинома;
- x — аргумент полинома;
- m, a, b — параметры полинома.

Значения любого из полиномов вычисляются так же, как и любой функции, например, элементарной: вводится функция с численными значениями степени, аргумента и ее параметров и нажимается комбинация клавиш решения `<Shift>+<Enter>`.

Пример 6.1

Необходимо вычислить значения полиномов четвертой степени Лежандра, Эрмита и Якоби при $x=1.2$, $a=1$, $b=3$.

В данном случае функции будут иметь вид:

`LegendreP [4, 1.2]`

`HermiteH [4, 1.2]`

`JacobiP [4, 1, 3, 1.2]`

Решение приведено на рис. 6.2.

При необходимости можно получить выражение любого полинома n -й степени. Для этого достаточно ввести функцию с указанием степени полинома n и нажать клавиши решения. Для получения множества вариантов следует использовать табулирование функции по параметру n .

Пример 6.2

Необходимо получить выражения ортогональных полиномов Чебышева первого рода, обобщенного полинома Лагерра и полинома Гегенбауэра. Полиномы получить при значениях $n = 1, 2, \dots, 5$. В полиноме Гегенбауэра параметр $m=2$.

LegendreP[4,1.2]

4.047

HermiteH[4,1.2]

-23.9424

JacobiP[4,1,3,1.2]

19.3745

LegendreP[4,z]

$$\frac{3}{8} - \frac{15}{4} z^2 + \frac{35}{8} z^4$$

HermiteH[4,z]

$$12 - 48 z^2 + 16 z^4$$

JacobiP[4,a,b,z]

$$\begin{aligned} & \frac{1}{24} (1+a) (2+a) (3+a) (4+a) + \\ & \frac{1}{12} (2+a) (3+a) (4+a) (5+a+b) (-1+z) + \\ & \frac{1}{16} (3+a) (4+a) (5+a+b) (6+a+b) (-1+z)^2 + \\ & \frac{1}{48} (4+a) (5+a+b) (6+a+b) (7+a+b) (-1+z)^3 + \\ & \frac{1}{384} (5+a+b) (6+a+b) (7+a+b) (8+a+b) (-1+z)^4 \end{aligned}$$

Рис. 6.2. Вычисление функций Лежандра, Эрмита и Якоби

В нашем случае функции возвращения полиномов будут иметь вид:

```
Table[ChebyshevT[n,z],{n,1,5,1}]
```

```
Table[LaguerreL[n,2,z],{n,1,5,1}]
```

```
Table[GegenbauerC[n,2,z],{n,1,5,1}]
```

Реализация функций показана на рис. 6.3.

Ортогональные многочлены полезно представлять в графическом виде. Графики дают наглядное представление о полиноме как функции, зависящей от степени n и значений аргументов. Убедимся в этом на примерах.

Пример 6.3

Необходимо представить в виде графиков полиномы Чебышева, Лагерра и Лежандра в диапазоне изменения аргумента от $x=-1$ до $x=2$ при значениях $n = 1, 3, 5$. Для обеспечения наглядности использовать различные стили графиков.

В данном случае графические функции будут иметь следующий вид:

```
Plot[{ChebyshevT[1,x],ChebyshevT[3,x],ChebyshevT[5,x]},  
{x,-1,2},PlotStyle->{AbsoluteDashing[{1,5]},None,Thickness[0.01]}];
```

```
Plot[{LaguerreL[1,x],LaguerreL[3,x],LaguerreL[5,x]},
{x,-1,2},PlotStyle->{AbsoluteDashing[{1,5]},None,Thickness[0.01]};
Plot[{LegendreP[1,x],LegendreP[3,x],LegendreP[5,x]},
{x,-1,2},PlotStyle->{AbsoluteDashing[{1,5]},None,Thickness[0.01]}].
```

Реализация функций приведена на рис. 6.4—6.6.

```
Table[ChebyshevT[n,z],{n,1,5,1}]
{z, -1+2 z^2, -3 z+4 z^3, 1-8 z^2+8 z^4, 5 z-20 z^3+16 z^5}
Table[LaguerreL[n,2,z],{n,1,5,1}]
{3-z, 1/2 (12-8 z+z^2), 1/6 (60-60 z+15 z^2-z^3),
1/24 (360-480 z+180 z^2-24 z^3+z^4),
1/120 (2520-4200 z+2100 z^2-420 z^3+35 z^4-z^5)}
Table[GegenbauerC[n,2,z],{n,1,5,1}]
{4 z, -2+12 z^2, -12 z+32 z^3, 3-48 z^2+
80 z^4, 24 z-160 z^3+192 z^5}
```

Рис. 6.3. Ортогональные полиномы Чебышева, Лагерра и Гегенбауэра при различных значениях n

```
Plot[{ChebyshevT[1,x],ChebyshevT[3,x],ChebyshevT[5,x]},
{x,-1,2},PlotStyle->{AbsoluteDashing[{1,5]},
None,Thickness[0.01]}]
```

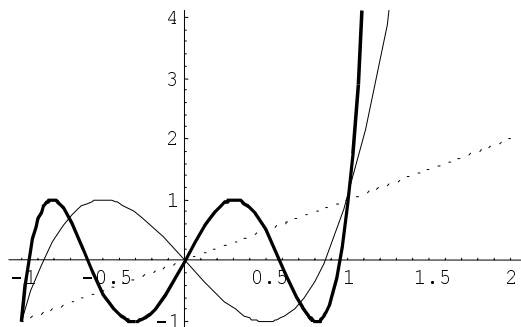


Рис. 6.4. Графическое представление функции Чебышева

```
Plot[{LaguerreL[1,x],LaguerreL[3,x],LaguerreL[5,x]},
{x,-1,2},PlotStyle->{AbsoluteDashing[{1,5]},
None,Thickness[0.01]}]
```

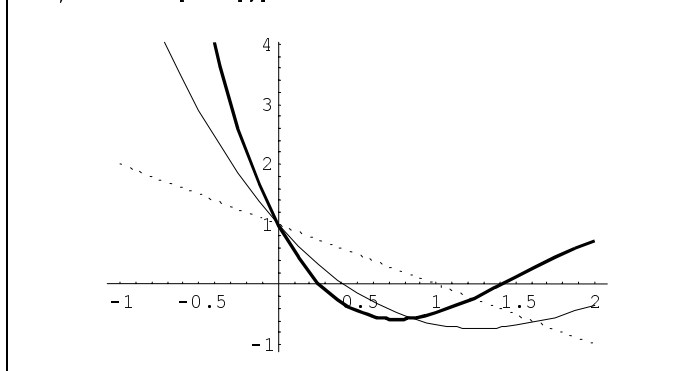


Рис. 6.5. Графическое представление функции Лагерра

```
Plot[{LegendreP[1,x],LegendreP[3,x],LegendreP[5,x]},
{x,-1,2},PlotStyle->{AbsoluteDashing[{1,5]},
None,Thickness[0.01]}]
```

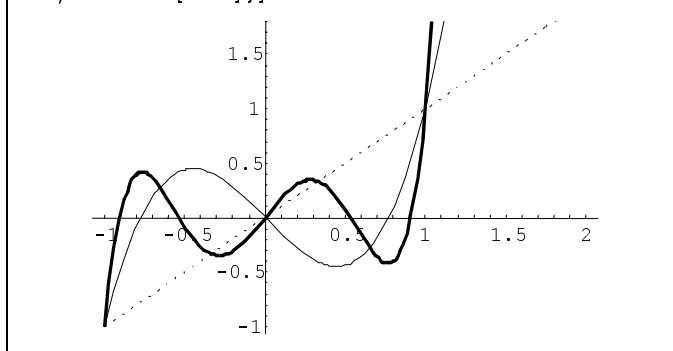


Рис. 6.6. Графическое представление функции Лежандра

6.2.2. Интегральные показательные функции

В системе Mathematica интегральные показательные функции вычисляются с помощью следующих встроенных функций:

□ `SinIntegral[x]` — интегральный синус

$$Si = - \int_{-x}^{\infty} \frac{\sin t}{t} dt = - \frac{\pi}{2} + \int_0^x \frac{\sin t}{t} dt ;$$

□ `CosIntegral[x]` — интегральный косинус

$$Ci = -\int_x^{\infty} \frac{\cos t}{t} dt = \int_0^x \frac{\cos t}{t} dt ;$$

□ `sinhIntegral[x]` — интегральный синус гиперболический;

□ `CoshIntegral[x]` — интегральный косинус гиперболический;

□ `ExpIntegralEi[x]` — интегральная показательная функция;

□ `ExpIntegralE[n, x]` — интегральная показательная функция

$$Ei(n, x) = -\int_{-x}^{\infty} \frac{e^{-t}}{t} dt ,$$

вычисляемая методом разложения в ряд при числе членов, равном n ;

□ `LogIntegral[x]` — интегральный логарифм

$$Li = \int_0^x \frac{dt}{\ln t} = Ei(\ln x) .$$

Технология вычисления интегральных функций аналогична технологии вычисления элементарных функций. Приведем примеры.

Пример 6.4

Необходимо вычислить все интегральные функции в диапазоне x от 1 до 7 с шагом, равным 1. Решение нужно получить в табличной форме.

Решение приведено на рис. 6.7.

Из рис. 6.6 видно, что табулирование выполнено в два этапа: вначале получено решение для тригонометрических и гиперболических функций, а затем для интегральных показательных и интегральной геометрической функций. Это осуществлено лишь с целью удобств расположения таблиц на экране монитора. Табличное представление решений выполнено с помощью функции `TableForm`.

Теперь представим интегральные показательные функции в виде графиков.

Пример 6.5

Необходимо представить все интегральные показательные функции в виде графиков. Диапазон изменения аргумента $x_{\min}=0$, $x_{\max}=6$ для геометрических и гиперболических функций и $x_{\min}=0$, $x_{\max}=1.5$ для остальных функций. В целях обеспечения высокой наглядности необходимо использовать различные стили графиков.

Решение задачи представлено на рис. 6.8 и 6.9.

```

Table[{x, SinIntegral[x], SinhIntegral[x],
CosIntegral[x], CoshIntegral[x]}, {x, 1., 7.}]
{{1., 0.946083, 1.05725, 0.337404, 0.837867}, {2., 1.60541,
2.50157, 0.422981, 2.45267}, {3., 1.84865, 4.97344, 0.11963
, 4.96039}, {4., 1.7582, 9.81733, -
0.140982, 9.81355}, {5., 1.54993, 20.0932, -
0.19003, 20.0921}, {6., 1.42469, 42.9951, -
0.0680572, 42.9947}, {7., 1.4546, 95.7524, 0.0766953, 95.75
23}}

TableForm[Out [14]]
1.      0.946083      1.05725      0.337404      0.837867
2.      1.60541      2.50157      0.422981      2.45267
3.      1.84865      4.97344      0.11963      4.96039
4.      1.7582      9.81733      -0.140982     9.81355
5.      1.54993     20.0932     -0.19003     20.0921
6.      1.42469     42.9951     -0.0680572   42.9947
7.      1.4546     95.7524     0.0766953    95.7523

Table[{x, ExpIntegralEi[x], ExpIntegralE[3, x],
LogIntegral[x]}, {x, 1., 7.}]
{{1., 1.89512, 0.109692, -
∞}, {2., 4.95423, 0.0301334, 1.04516}, {3., 9.93383, 0.0089
3065, 2.16359}, {4., 19.6309, 0.00276136, 2.96759}, {5., 40.
1853, 0.000877801, 3.63459}, {6., 85.9898, 0.000284604, 4.2
2222}, {7., 191.505, 0.0000936565, 4.75705}}

TableForm[Out [16]]
1.      1.89512      0.109692      -∞
2.      4.95423      0.0301334     1.04516
3.      9.93383      0.00893065    2.16359
4.      19.6309      0.00276136    2.96759
5.      40.1853      0.000877801   3.63459
6.      85.9898      0.000284604   4.22222
7.      191.505      0.0000936565  4.75705

```

Рис. 6.7. Табулирование интегральных показательных функций

6.2.3. Гамма-функция

Гамма-функция имеет вид:

$$\Gamma(n) = \int_0^{\infty} e^{-t} t^{n-1} dt.$$

Если n — целое, то гамма-функция отождествляется с факториалом.

Гамма-функция существует для целых и дробных, положительных и отрицательных, вещественных и комплексных чисел. Математически она может быть представлена в следующих видах:

$$\Gamma(n+1) = n\Gamma(n)$$

$$\Gamma(1) = \Gamma(2) = 1$$

$$\Gamma(n) = (n-1)!$$

$$\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$$

$$\Gamma\left(-\frac{1}{2}\right) = 2\sqrt{\pi}$$

$$\Gamma\left(n + \frac{1}{2}\right) = \left(n - \frac{1}{2}\right)!!$$

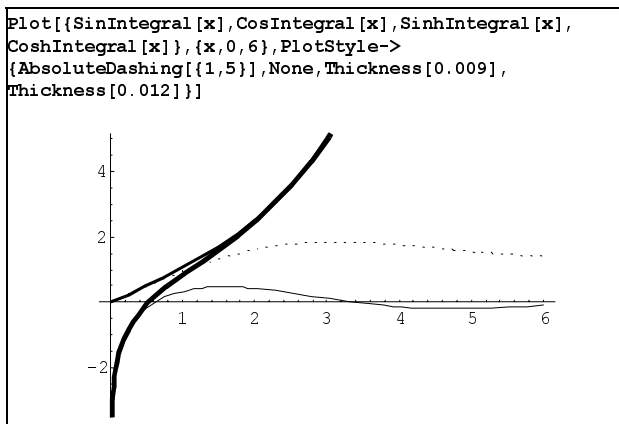


Рис. 6.8. Графики интегральных геометрических и гиперболических функций

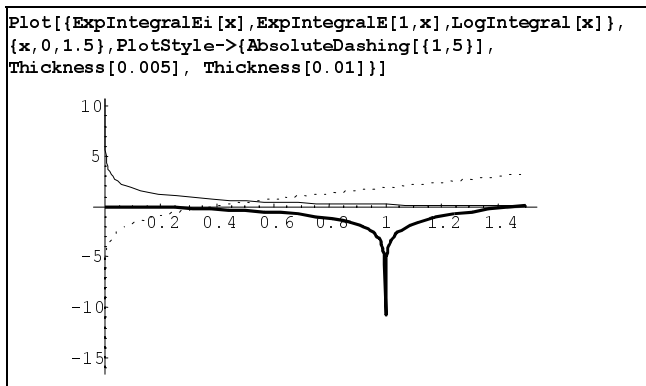


Рис. 6.9. Графики логарифмической и интегральных показательных функций

В системе Mathematica для вычисления гамма-функции имеются следующие встроенные функции:

- ☐ $\text{Gamma}[n]$ — эйлерова гамма-функция;
- ☐ $\text{Gamma}[a, x]$ — неполная гамма-функция;
- ☐ $\text{Gamma}[a, z_0, z_1]$ — обобщенная неполная гамма-функция:
 $\text{Gamma}(a, z_0) - \text{Gamma}(a, z_1)$
- ☐ $\text{GammaRegularized}[a, z]$ — регулярная неполная гамма-функция:
 $Q[a, z] = \text{Gamma}[a, z] / \text{Gamma}[a]$
- ☐ $\text{GammaRegularized}[a, z_0, z_1]$ — обобщенная неполная гамма-функция:
 $Q[a, z_0] - Q[a, z_1]$
- ☐ $\text{LogGamma}[z]$ — логарифм эйлеровой гамма-функции;
- ☐ $\text{PolyGamma}[z]$ — дигамма-функция $\Psi(z)$;
- ☐ $\text{PolyGamma}[n, z]$ — n -я производная от гамма-функции.

Технология вычисления этих функций особенностей не имеет, она такая же, как и других функций. Приведем примеры.

Пример 6.6

Необходимо вычислить:

- ☐ гамма-функцию вещественных чисел $1/2, -1/2, 3, 5.5, 10$;
- ☐ гамма-функцию комплексных чисел $i, 2+3i$;
- ☐ логарифм гамма-функции комплексного числа $2+3i$;
- ☐ дигамма-функцию комплексного числа $2+3i$;
- ☐ вторую производную от дигамма-функции комплексного числа $2+3i$;
- ☐ регулярную неполную гамма-функцию комплексного числа $2+3i$ при $a=2$;
- ☐ обобщенную неполную гамма-функцию чисел $z_0=2+2i, z_1=3+3i$ при $a=1$.

Решение примера приведено на рис. 6.10.

Обратим внимание на следующие особенности вычисления гамма-функции. Символ i комплексного числа представляется в виде заглавной буквы I , при этом между числом и символом I ставится знак умножения (или пробел). Целые числа должны заканчиваться точкой, независимо от их расположения в числе.

Гамма-функция является достаточно сложной функцией, в чем можно убедиться по ее графическому представлению. Приведем пример.

```

Gamma[{1/2,-1/2,3,5.5,10}]
{ $\sqrt{\pi}$ , -2  $\sqrt{\pi}$ , 2, 52.3428, 362880}
Gamma[1. I]
-0.15495-0.498016 i
Gamma[2.+3. I]
-0.0823953+0.0917743 i
LogGamma[2.+3. I]
-2.09285+2.3024 i
PolyGamma[2.+3. I]
1.20798 +1.10413 i
GammaRegularized[3.,2.+3. I]
-0.98697-0.119567 i
GammaRegularized[1.,2+2. I,3.+3. I]
-0.00703053-0.116034 i
PolyGamma[2,2.+3. I]
0.0526762 +0.0730362 i

```

Рис. 6.10. Результаты решения примера 6.6

Пример 6.7

Необходимо представить в виде графиков следующие функции: $\Gamma[1, x]$, $\Gamma[2, x]$, $\Gamma[x]$. В целях наглядности использовать графические стили.

В данном случае функция Plot будет иметь вид:

```
Plot[{Gamma[1, x], Gamma[2, x], Gamma[x]}, {x, -4, 5}, PlotStyle->
{AbsoluteDashing[{1, 3}], None, Thickness[0.005]}]
```

Реализация этой функции приведена на рис. 6.11.

Пример 6.8

Необходимо представить гамма-функцию в виде графика на комплексной плоскости в диапазоне изменения x от 1 до 2, а y от -0.5 до 0.5 .

В данном случае функция построения контурного графика будет иметь вид:

```
ContourPlot[Abs[Gamma[-x+I y]], {x,1,2}, {y,-0.5,0.5}]
```

Ее реализация приведена на рис. 6.12.

```
Plot[{Gamma[1,x],Gamma[2,x],Gamma[x]}, {x,-4,5},
PlotStyle -> {AbsoluteDashing[{1,3}],None,
Thickness[0.007]}]
```

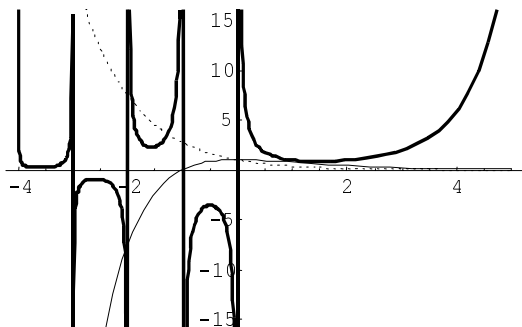


Рис. 6.11. Гамма-функции примера 6.7

```
ContourPlot[Abs[Gamma[-x+I y]], {x,1,2}, {y,-0.5,0.5}]
```

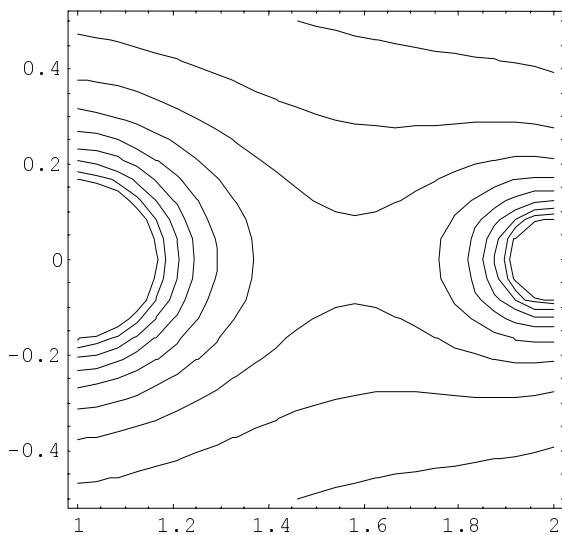


Рис. 6.12. Контурный график гамма-функции

6.2.4. Функции Бесселя

Функции Бесселя являются решением дифференциального уравнения следующего вида:

$$z^2 y'' + zy' + (z^2 - n^2)y = 0.$$

В системе Mathematica имеются следующие встроенные функции вычисления функций Бесселя:

- ☐ `BesselJ[n, z]` — функция Бесселя первого рода;
- ☐ `BesselI[n, z]` — модифицированная функция Бесселя первого рода;
- ☐ `BesselY[n, z]` — функция Бесселя второго рода;
- ☐ `BesselK[n, z]` — модифицированная функция Бесселя второго рода.

Встроенные функции вычисляют значения функций Бесселя при заданных n и z . При этом n является вещественным числом, а z может быть и вещественным и комплексным.

Технология вычислений очевидна. Приведем примеры.

Пример 6.9

Необходимо вычислить значения функций Бесселя первого и второго рода, в том числе и модифицированных при $n = 1, 2, \dots, 5$, $z = 1.2$.

В данном случае целесообразно не вычислять пять раз функции при различных значениях n , а выполнить табулирование функций по переменной n .

Программа вычисления функций будет иметь вид:

```
Table[{n, BesselJ[n, 1.2]}, {n, 1, 5}]
Table[{n, BesselI[n, 1.2]}, {n, 1, 5}]
Table[{n, BesselY[n, 1.2]}, {n, 1, 5}]
Table[{n, BesselK[n, 1.2]}, {n, 1, 5}]
```

Для получения решения в табличной форме воспользуемся функцией `TableForm[%]`.

Результаты вычисления функций приведены на рис. 6.13.

Пример 6.10

Необходимо вычислить значения функций Бесселя при $n = 1, 2, \dots, 5$ и $z = 2 + 3i$.

Технология вычисления функций Бесселя при комплексном аргументе остается прежней. Решение приведено на рис. 6.14. Решения представлены в табличной форме с помощью функции `TableForm`.

Представим функции Бесселя первого и второго рода в виде графиков при различных значениях n .

```
Table[{n,BesselJ[n,1.2]},{n,1,5}]
{{1,0.498289},{2,0.159349},{3,0.0328743},{4,0.00502267},{5,0.000610105}}
```

```
TableForm[Out[%]]
```

```
1      0.498289
2      0.159349
3      0.0328743
4      0.00502267
5      0.000610105
```

```
Table[{n,BesselI[n,1.2]},{n,1,5}]
```

```
{{1,0.714678},{2,0.202596},{3,0.039359},{4,0.00580067},{5,0.000687895}}
```

```
TableForm[%]
```

```
1      0.714678
2      0.202596
3      0.039359
4      0.00580067
5      0.000687895
```

```
Table[{n,BesselY[n,1.2]},{n,1,5}]
```

```
{{1,-0.621136},{2,-1.26331},{3,-3.5899},{4,-16.6862},{5,-107.651}}
```

```
TableForm[%]
```

```
1      -0.621136
2      -1.26331
3      -3.5899
4      -16.6862
5      -107.651
```

```
Table[{n,BesselK[n,1.2]},{n,1,5}]
```

```
{{1,0.434592},{2,1.04283},{3,3.91069},{4,20.5963},{5,141.219}}
```

```
TableForm[%]
```

```
1      0.434592
2      1.04283
3      3.91069
4      20.5963
5      141.219
```

Рис. 6.13. Вычисление функций Бесселя

Пример 6.11

Пусть необходимо представить функцию Бесселя первого и второго рода в виде графика в диапазоне z от 0 до 8 при $n = 1, 3, 7$. Для повышения наглядности использовать различные стили для кривых.

```
Table[{n,BesselJ[n,2.+3. I]},{n,1,5}]
```

```
TableForm[%]
```

```
1      3.78068 - 0.812781 i
2      1.25767 + 2.31877 i
3      -0.866325 + 1.07879 i
4      -0.563654 - 0.123441 i
5      -0.0552951 - 0.190122 i
```

```
Table[{n,BesselI[n,2.+3. I]},{n,1,5}]
```

```
TableForm[%]
```

```
1      -1.26098 + 0.780149 i
2      -1.22131 + 0.125946 i
3      -0.625665 - 0.424719 i
4      -0.0556997 - 0.348311 i
5      0.0859247 - 0.098858 i
```

```
Table[{n,BesselY[n,2.+3. I]},{n,1,5}]
```

```
TableForm[%]
```

```
1      0.796502 + 3.76489 i
2      -2.3443 + 1.27581 i
3      -1.06148 - 0.815805 i
4      0.234897 - 0.559111 i
5      0.318383 - 0.305987 i
```

```
Table[{n,BesselK[n,2.+3. I]},{n,1,5}]
```

```
TableForm[%]
```

```
1      -0.0865 + 0.0390614 i
2      -0.0915555 + 0.0798916 i
3      -0.0690958 + 0.172738 i
4      0.0838398 + 0.335013 i
5      0.652578 + 0.430281 i
```

Рис. 6.14. Функции Бесселя при комплексном аргументе

В данном случае функции Plot будут иметь вид:

```
Plot[{BesselJ[1,z],BesselJ[3,z],BesselJ[7,z]},{z,0,8},
PlotStyle->{AbsoluteDashing[{1,3}],None,Thickness[0.008]}]
```

```
Plot[{BesselI[1,z],BesselI[3,z],BesselI[7,z]},{z,0,8},
PlotStyle->{AbsoluteDashing[{1,3}],None,Thickness[0.008]}]
```

Реализация этих функций показана на рис. 6.15 и 6.16.

```
Plot[{BesselJ[1,z],BesselJ[3,z],BesselJ[7,z]},{z,0,8},
PlotStyle->{AbsoluteDashing[{1,3}],None,
Thickness[0.008]}]
```

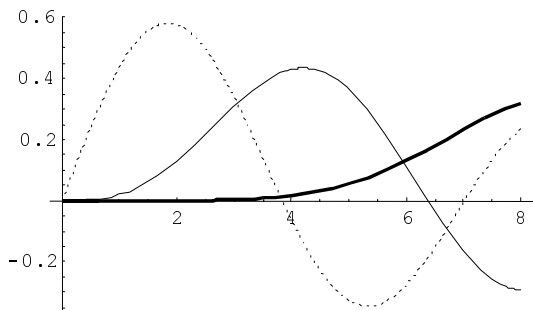


Рис. 6.15. Графики функций Бесселя первого рода

```
Plot[{BesselI[1,z],BesselI[3,z],BesselI[7,z]},{z,0,8},
PlotStyle->{AbsoluteDashing[{1,3}],None,
Thickness[0.008]}]
```

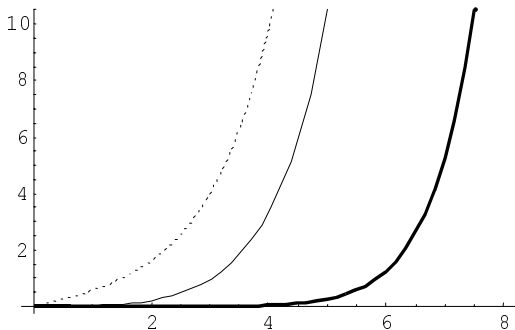


Рис. 6.16. Графики функций Бесселя второго рода

6.2.5. Функции Эйри

В системе Mathematica функции Эйри возвращают следующие встроенные функции:

- ☐ `AiryAi[x]` — вычисляет *Ai*-функцию Эйри;
- ☐ `AiryBi[x]` — вычисляет *Bi*-функцию Эйри;

□ `AiryAiPrime[x]` — вычисляет производную *Ai*-функции Эйри;

□ `AiryBiPrime[x]` — вычисляет производную *Bi*-функции Эйри.

Они позволяют вычислять функции Эйри и их производные при вещественном и комплексном значениях аргумента x .

Приведем примеры вычисления функций Эйри и их графическое представление.

Пример 6.12

Необходимо вычислить функции Эйри и их производные при $x = 1, 2, \dots, 5$.

В данном случае вычисления целесообразно выполнить путем табулирования функций Эйри и их производных. Решения приведены на рис. 6.17 в табличной форме.

Table[{x,AiryAi[x]},{x,1.,5.}]

TableForm[%]

1.	0.135292
2.	0.0349241
3.	0.00659114
4.	0.000951564
5.	0.000108344

Table[{x,AiryBi[x]},{x,1.,5.}]

TableForm[%]

1.	1.20742
2.	3.29809
3.	14.0373
4.	83.8471
5.	657.792

Table[{x,AiryAiPrime[x]},{x,1.,5.}]

TableForm[%]

1.	-0.159147
2.	-0.0530904
3.	-0.011913
4.	-0.00195864
5.	-0.000247414

Table[{x,AiryBiPrime[x]},{x,1.,5.}]

TableForm[%]

1.	0.932436
2.	4.10068
3.	22.9222
4.	161.927
5.	1435.82

Рис. 6.17. Табличное представление функций Эйри и их производных

Пример 6.13

Необходимо вычислить функции Эйри и их производные при следующих значениях комплексного аргумента: $1+i$, $1-3i$, $3+2i$.

В данном случае функции Эйри вычисляются путем представления комплексных аргументов в виде вектора:

```
AiryAi[{1.+I,1.-3. I,3.+2. I}]
```

```
AiryBi[{1.+I,1.-3. I,3.+2. I}]
```

```
AiryAiPrime[{1.+I,1.-3. I,3.+2. I}]
```

```
AiryBiPrime[{1.+I,1.-3. I,3.+2. I}]
```

Реализация этих функций показана на рис. 6.18.

```
AiryAi [{1.+I,1.-3. I,3.+2. I}]
{0.0604583 -0.15189 i,-0.483349-0.433065 i,
-0.0096772+0.00552469 i}

AiryBi [{1.+I,1.-3. I,3.+2. I}]
{0.716658 +0.619889 i,-0.569634+0.498224 i,
-7.33735-1.66152 i}

AiryAiPrime [{1.+I,1.-3. I,3.+2. I}]
{-0.130628+0.163068 i,1.13237 +0.171739
i,0.0209901 -0.00534747 i}

AiryBiPrime [{1.+I,1.-3. I,3.+2. I}]
{0.0756628 +0.783701 i,-0.00635607-0.959128
i,-11.9181-7.25463 i}
```

Рис. 6.18. Функции Эйри комплексного аргумента

Пример 6.14

Необходимо представить функции Эйри в виде графиков в диапазоне x от -8 до 5 . Кривые изобразить в разных стилях.

Представим функции Plot в следующем виде:

```
Plot[{AiryAi[x],AiryBi[x]},{x,-8,5}, PlotStyle->
{AbsoluteDashing[{1,5]},Thickness[0.01]}]
```

```
Plot[{AiryAiPrime[x],AiryBiPrime[x]},{x,-8,5}, PlotStyle->
{AbsoluteDashing[{1,5]},Thickness[0.01]}]
```

Реализация этих функций показана на рис. 6.19 и 6.20.

```
Plot[{AiryAi[x],AiryBi[x]},{x,-8,5},
PlotStyle->{AbsoluteDashing[{1,5]},Thickness[0.01]}]
```

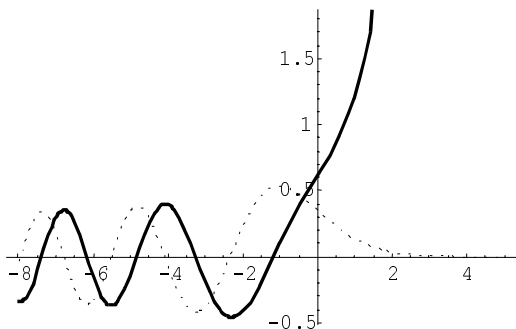


Рис. 6.19. Графики функций Эйри

```
Plot[{AiryAiPrime[x],AiryBiPrime[x]},{x,-8,5},
PlotStyle->{AbsoluteDashing[{1,5]},Thickness[0.01]}]
```

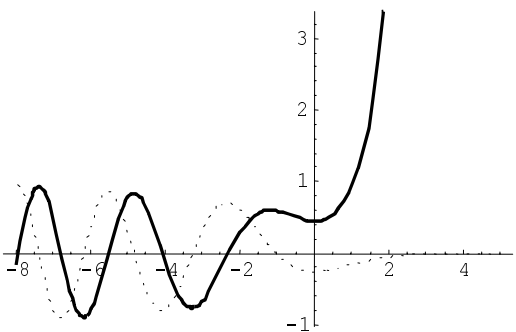


Рис. 6.20. Графики производных функций Эйри

6.2.6. Бета-функция (эйлеров интеграл первого рода)

Бета-функция представляется в виде следующего интеграла:

$$B(x, y) = \int_0^1 t^{x-1} (1-t)^{y-1} dt.$$

Она имеет много других интегральных представлений.

При практических расчетах бета-функция вычисляется через гамма-функцию следующим образом:

$$B(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}.$$

Бета-функция существует для вещественных и комплексных аргументов.

В системе Mathematica бета-функция и родственные ей вычисляются с помощью следующих встроенных функций:

- `Beta[x, y]` — бета-функция;
- `Beta[z, x, y]` — неполная бета-функция;
- `Beta[z0, z1, x, y]` — обобщенная неполная бета-функция:
`Beta[z1, x, y]-Beta[z0, x, y]`
- `BetaRegularized[z, x, y]` — регуляризованная неполная бета-функция:
`Beta[z, x, y]/Beta[x, y]`
- `BetaRegularized[z0, z1, x, y]` — регуляризованная обобщенная бета-функция:
`Beta[z1, x, y]-Beta[z0, x, y]`

Технология вычисления бета-функции и всех ее родственных функций очевидна, она не отличается от технологии вычисления ранее рассмотренных функций. Приведем примеры.

```
{Beta[1.5,3.],Beta[1.,3.],Beta[2.-3. I,2.+4. I]}
{0.152381,0.333333,0.000910922 +0.000058345 i}
Beta[3.,2.5,1.25]
4.50672 +4.2343 i
Beta[1.,2.,2.5,1.25]
1.11054 +1.11054 i
BetaRegularized[3.,2.5,1.25]
16.5432 +15.5432 i
BetaRegularized[1.,2.,2.5,1.25]
4.07656 +4.07656 i
```

Рис. 6.21. Вычисление бета-функций

Пример 6.15

Необходимо вычислить бета-функцию и все ее родственные функции при следующих значениях переменных:

□ для бета-функции: $x = \{[1.5, 3], [1, 3], [2-3i, 2+4i]\}$;

□ для всех родственных функций: $x=2.5, y=1.25, z=3, z_0=1, z_1=2$.

Вычисления приведены на рис. 6.21.

6.2.7. Функции статистических распределений и функции ошибок

Основными параметрами статистических распределений являются: среднее значение, дисперсия, медиана, асимметрия данных. Система Mathematica имеет большое число встроенных функций вычисления параметров статистических распределений. Они находятся в подпакете `DescriptiveStatistics` пакета `Statistics`. Обращение к подпакету имеет вид

```
<< Statistics` DescriptiveStatistics`
```

Основными функциями подпакета `DescriptiveStatistics` являются:

□ `Mean[D]` — вычисляет среднее значение данных `D`;

□ `GeometricMean[D]` — вычисляет геометрическое среднее данных `D`;

□ `HarmonicMean[D]` — вычисляет гармоническое среднее данных `D`;

□ `CentralMoment[D, r]` — возвращает центральный момент данных `D` порядка `r`;

□ `MeanDeviation[D]` — вычисляет среднее отклонение данных `D`;

□ `Median[D]` — вычисляет медиану (центральное значение) данных `D`;

□ `MedianDeviation[D]` — вычисляет абсолютное отклонение данных `D` (отклонение от медианы);

□ `StandardDeviation[D]` — вычисляет стандартное отклонение данных `D`;

□ `RootMeanSquare[D]` — вычисляет среднеквадратическое значение данных `D`;

□ `VarianceData[D]` — вычисляет среднеквадратическое отклонение данных `D`;

□ `srewness[D]` — вычисляет коэффициент асимметрии данных `D`;

□ `Quantile[D, q]` — вычисляет q -й квантиль данных `D`;

□ `InterpolatingQuantile[D, q]` — вычисляет q -й квантиль данных `D`, применяя интерполяцию данных в процессе вычислений.

Определение этих функций можно получить, если их вычислять для случая символьных переменных. Убедимся в этом на примерах.

Встроенные функции ошибок находятся в ядре системы Mathematica. Они имеют вид:

□ `Erf[x]` — функция ошибок (интеграл вероятности);

□ `Erf[x0, x]` — обобщенная функция ошибок: $\text{Erf}[x] - \text{Erf}[x_0]$;

- `Erfc[x]` — дополнительная функция ошибок: $1 - \text{Erf}[x]$;
- `Erfi[x]` — мнимое значение функции ошибок: $\text{Erf}[i \cdot x] / i$;
- `InverseErf[x]` — инверсная функция ошибок;
- `InverseErfC[x]` — инверсная дополнительная функция ошибок.

Приведем примеры вычисления функций статистических распределений и функций ошибок.

Пример 6.16

Пусть данные являются символьными переменными, представленными в виде вектора $f=\{a, b, c, d, e\}$. Необходимо вычислить среднее значение данных, геометрическое среднее, гармоническое среднее, среднеквадратическое отклонение, стандартное отклонение.

Соответствующие встроенные функции в данном случае имеют вид:

```
Mean[{a,b,c,d,e}]
GeometricMean[{a,b,c,d,e}]
HarmonicMean[{a,b,c,d,e}]
RootMeanSquare[{a,b,c,d,e}]
StandardDeviation[{a,b,c,d,e}]
```

Решение примера приведено на рис. 6.22.

На рис. 6.22 представлены формулы, по которым вычисляются пять показателей статистических распределений. Остальные показатели не приводятся в виду большого размера формул. Пользователь при необходимости может их получить самостоятельно.

Пример 6.17

Данные представлены в виде следующего списка: $D=\{1, 3.2, 2.7, 4, 5.6, 6, 7.3, 8, 9, 12, 14.5, 15\}$. Необходимо вычислить все показатели статистических распределений.

Решение примера приведено на рис. 6.23.

Пример 6.18

Необходимо вычислить функции ошибок `Erf[x]`, `Erf[x0, x]`, `Erfc[x]`, `InverseErf[x]` при значениях x от 0 до 1 с шагом 0.1. Решение представить в виде таблицы.

В данном случае необходимо выполнить табулирование перечисленных функций с помощью одной функции `Table`. Решение приведено на рис. 6.24.

```
<<Statistics`DescriptiveStatistics`
f={a,b,c,d,e}
{a,b,c,d,e}
Mean[f]

$$\frac{1}{5} (a+b+c+d+e)$$

GeometricMean[f]

$$(a b c d e)^{1/5}$$

HarmonicMean[f]

$$\frac{5}{\frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d} + \frac{1}{e}}$$

RootMeanSquare[f]

$$\frac{\sqrt{a^2+b^2+c^2+d^2+e^2}}{\sqrt{5}}$$

StandardDeviation[f]

$$\frac{1}{2} \sqrt{\left(a - \frac{1}{5} (a+b+c+d+e)\right)^2 + \left(b - \frac{1}{5} (a+b+c+d+e)\right)^2 + \left(c - \frac{1}{5} (a+b+c+d+e)\right)^2 + \left(d - \frac{1}{5} (a+b+c+d+e)\right)^2 + \left(e - \frac{1}{5} (a+b+c+d+e)\right)^2}$$

```

Рис. 6.22. Получение формул статистических распределений

```
<<Statistics`DescriptiveStatistics`
F={1,2,7,4,3,2,9,14.5,5.6,12,7.3,6,8,15}
{1,2,7,4,3,2,9,14.5,5.6,12,7.3,6,8,15}
Mean[F]
7.35833
GeometricMean[F]
5.84589
HarmonicMean[F]
4.18093
CentralMoment[F,2]
19.1408
MeanDeviation[F]
3.61806
Median[F]
6.65
```

Рис. 6.23. (Часть 1 из 2) Параметры статистических распределений данных примера 6.17

```

MedianDeviation[F]
3.05
StandardDeviation[F]
4.56956
RootMeanSquare[F]
8.56071
Variance[F]
20.8808
Skewness[F]
0.437172
Quantile[F,1]
15.
InterpolatingQuantile[F,1]

InterpolatingQuantile[{1,2.7,4,3.2,9,14.5,5.6,12,7
.3,6,8,15},1]

```

Рис. 6.23. (Часть 2 из 2)

```

z1=Erf[x]
z2=Erf[1.,x]
z3=Erfc[x]
z4=InverseErf[x]
Table[{x,z1,z2,z3,z4},{x,0.,1.,0.1}]

TableForm[%]

```

0.	0.	-0.842701	1.	0.
0.1	0.112463	-0.730238	0.887537	0.088856
0.2	0.222703	-0.619998	0.777297	0.179143
0.3	0.328627	-0.514074	0.671373	0.272463
0.4	0.428392	-0.414308	0.571608	0.370807
0.5	0.5205	-0.322201	0.4795	0.476936
0.6	0.603856	-0.238845	0.396144	0.595116
0.7	0.677801	-0.1649	0.322199	0.732869
0.8	0.742101	-0.1006	0.257899	0.906194
0.9	0.796908	-0.0457926	0.203092	1.16309
1.	0.842701	0.	0.157299	∞

Рис. 6.24. Вычисление функций ошибок

Пример 6.19

Необходимо вычислить функции ошибок $\text{Erf}[x]$, $\text{Erfc}[x]$, $\text{InverseErf}[x]$ и представить их в виде точечных графиков в диапазоне изменения x от 0 до 1 с шагом 0.1. Решение получить на одном рисунке.

В данном случае первоначально необходимо выполнить табулирование функций, а затем построить многофункциональный график с помощью графической функции `MultipleListPlot`, которая находится в пакете расширения `Graphics`.

Программа решения задачи и ее реализация показаны на рис. 6.25.

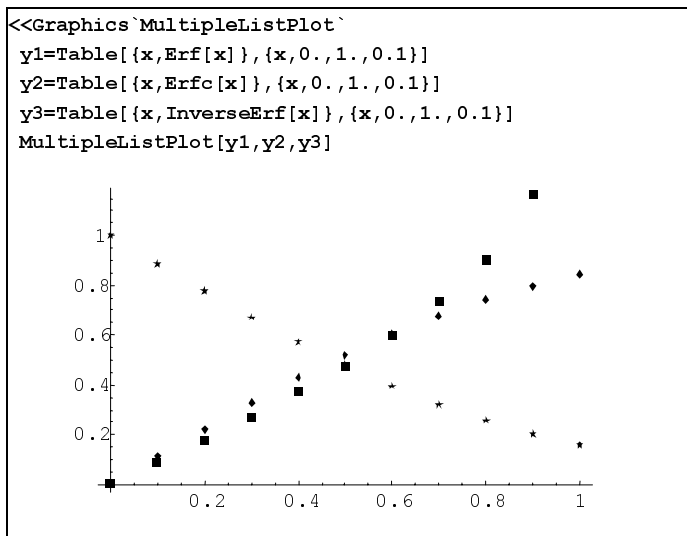


Рис. 6.25. Графическое представление функций ошибок

Из приведенных примеров видно, что процедуры вычисления статистических характеристик исключительно просты. Кроме того, они позволяют пользователю получить расчетные формулы, по которым ведутся расчеты. Это еще раз свидетельствует о высокой интеллектуальности системы Mathematica.

6.2.8. Функции генерации случайных чисел

Образование случайных чисел, подчиненных определенному закону распределения случайных величин, необходимо при статистических методах моделирования.

Система Mathematica имеет генератор случайных чисел. Обращение к генератору осуществляется с помощью встроенных функций `Random` и `SeedRandom`. Функции имеют следующий вид и назначение:

- ❑ `Random[ ]` — генерирует псевдослучайное действительное число из интервала $0—1$, распределенное по нормальному закону;
- ❑ `Random[T, x]` — генерирует псевдослучайное число типа `T` из диапазона `x`. Типами чисел могут быть: целые (`Integer`), действительные (`Real`) и комплексные (`Complex`). По умолчанию интервал генерируемых чисел $0—1$. Задать интервал генерирования чисел можно, представив функцию в одном из следующих видов:
 - `Random[T, xmax]` — генерация случайных чисел из диапазона $0—x_{\max}$;
 - `Random[T, {xmin, xmax}]` — генерация случайных чисел из диапазона $x_{\min}—x_{\max}$;
- ❑ `SeedRandom[n]` — устанавливает генератор в начальное состояние, принимая `n` за начальное число;
- ❑ `SeedRandom[ ]` — устанавливает генератор в начальное состояние, принимая за начальное значение текущее время.

С помощью функции `Random[ ]` можно генерировать множество чисел из диапазона $0—1$, распределенных по нормальному закону. Для этого необходимо табулировать функцию `Random[ ]`. Функция `Table` при этом имеет вид:

```
Table[{n, Random[]}, {n, 1, xmax}]
```

где `n` — число повторений функции `Random` от 1 до $x_{\max}$.

Пример 6.20

Необходимо получить 10 случайных чисел из диапазона $0—1$, распределенных по нормальному закону.

В данном случае функция генерации чисел будет иметь вид:

```
Table[{n, Random[]}, {n, 1, 10}]
```

Решение примера приведено на рис. 6.26 и представлено в форме таблицы.

Пример 6.21

Необходимо сгенерировать 10 случайных целых чисел из диапазона $0—20$.

В данном случае программа генерации целых чисел будет иметь вид:

```
Table[{n, Random[Integer, 20]}, {n, 1, 10}]
```

Реализация программы показана на рис. 6.27. Решение получено в табличной форме.

Table [{n, Random[]}, {n, 1, 10}]	
TableForm[%]	
1	0.248875
2	0.946241
3	0.939069
4	0.742521
5	0.48878
6	0.200137
7	0.197453
8	0.831508
9	0.701402
10	0.450476

Рис. 6.26. Генерация 10 случайных чисел из диапазона 0—1

Table [{n, Random[Integer, 20]}, {n, 1, 10}]	
TableForm[%]	
1	19
2	10
3	11
4	18
5	18
6	3
7	9
8	11
9	6
10	20

Рис. 6.27. Генерация случайных целых чисел из диапазона 0—20

Пример 6.22

Необходимо сгенерировать 10 случайных действительных чисел из диапазона 1—10. Решение представить в табличной форме.

В данном случае программа генерации чисел имеет вид:

```
Table[{n, Random[Real, {1, 10}]}, {n, 1, 10}]
```

Реализация программы приведена на рис. 6.28.

Пример 6.23

Необходимо сгенерировать 10 случайных комплексных чисел. Решение представить в форме таблицы.

Программа генерации случайных комплексных чисел имеет вид:

`Table[{n, Random[Complex] }, {n, 1, 10}]`

Ее реализация приведена на рис. 6.29.

Table[{n, Random[Real, {1, 10}]}], {n, 1, 10}]	
TableForm[%]	
1	6.98841
2	1.74036
3	6.99167
4	8.61901
5	4.81583
6	5.49817
7	3.15366
8	1.15565
9	9.82008
10	6.10199

Рис. 6.28. Генерация случайных действительных чисел из диапазона 1—10

Table[{n, Random[Complex] }, {n, 1, 10}]	
TableForm[%]	
1	0.371845 + 0.537291 i
2	0.383258 + 0.0619815 i
3	0.353799 + 0.393909 i
4	0.239523 + 0.278312 i
5	0.613089 + 0.812574 i
6	0.462254 + 0.376349 i
7	0.337558 + 0.711458 i
8	0.796875 + 0.294087 i
9	0.671817 + 0.864902 i
10	0.372894 + 0.794291 i

Рис. 6.29. Генерация случайных комплексных чисел

6.2.9. Эллиптические интегралы и функции

Система Mathematica имеет большое число встроенных функций вычисления эллиптических интегралов и интегральных функций. Приведем основные из них:

□ `EllipticE[m]` — полный эллиптический интеграл $E(m)$;

□ `EllipticK[m]` — полный эллиптический интеграл первого рода $K(m)$;

- `EllipticE[Pi, m]` — полный эллиптический интеграл второго рода;
- `EllipticPi[n, Pi, m]` — эллиптический интеграл третьего рода;
- `EllipticF[Pi, m]` — эллиптический интеграл первого рода;
- `EllipticPi[n, m]` — полный эллиптический интеграл;
- `EllipticTheta[i, z, q]` — эллиптическая тета-функция при $i = 1, 2, 3, 4$;
- `EllipticThetaC[n, m]` — эллиптическая функция Невилла $G_C(n, m)$;
- `EllipticThetaD[n, m]` — эллиптическая функция Невилла $G_D(n, m)$;
- `EllipticThetaN[n, m]` — эллиптическая функция Невилла $G_N(n, m)$;
- `FresnelC(x)` — интеграл Френеля $C(x)$;
- `FresnelS(x)` — интеграл Френеля $S(x)$;
- `JacobiZeta[Pi, m]` — дзета-функция Якоби;
- `WeierstrassP[u, q1, q2]` — эллиптическая функция Вейерштрасса.

Технология вычисления эллиптических интегралов и интегральных функций не отличается от вычисления других функций, рассмотренных нами ранее.

Пример 6.24

Необходимо вычислить все эллиптические интегралы первого, второго и третьего рода при $m=0.2$.

Решения приведены на рис. 6.30.

```

EllipticE[0.2]
1.48904
EllipticK[0.2]
1.65962
EllipticE[Pi,0.2]
2.97807
EllipticPi[5,Pi,0.2]
-0.0703735-1.60319 i
EllipticF[Pi,0.2]
3.31925
EllipticPi[3,0.2]
-0.0599029-1.14971 i

```

Рис. 6.30. Вычисление эллиптических интегралов

Пример 6.25

Необходимо вычислить:

- ☐ эллиптическую тета-функцию при $i = 1, 2, 3, 4$, $z = 2$, $q = 0.2$;
- ☐ интегралы Френеля $C(x)$ и $S(x)$ при $x = 1, 2, 3, \dots, 7$;
- ☐ эллиптическую функцию Вейерштрасса при $m = 1, 2, 3, \dots, 7$, $q_1 = 3$, $q_2 = 0.2$.

```
Table[{i, EllipticTheta[i, 2, 0.2]}, {i, 1, 4}]
TableForm[%]
1      1.23107
2      -0.505292
3      0.738078
4      1.26099

Table[{x, FresnelC[x]}, {x, 1., 7.}]
TableForm[Out[145]]
1.      0.779893
2.      0.488253
3.      0.605721
4.      0.498426
5.      0.563631
6.      0.499531
7.      0.545467

Table[{x, FresnelS[x]}, {x, 1., 7.}]
Table[{}]
```

```
TableForm[Out[150]]
1.      0.438259
2.      0.343416
3.      0.496313
4.      0.420516
5.      0.499191
6.      0.446961
7.      0.499705

Table[{m, WeierstrassP[m, 3., 0.2]}, {m, 1, 7}]
TableForm[%]
1      1.16513
2      1.86509
3      16.0294
4      0.924128
5      4.03313
6      4.04301
7      0.923861
```

Рис. 6.31. Табличное представление эллиптических функций примера 6.25

Решения представить в виде таблиц путем табулирования функций.

Процедуры вычислений и результаты приведены на рис. 6.31.

6.2.10. Специальные числа и полиномы

Система Mathematica имеет встроенные функции для вычисления ряда специальных чисел и полиномов. Приведем основные из них:

- `Binomial[n, x]` — биномиальный коэффициент;
- `BernoulliB[n]` — n -е число Бернулли;
- `Bernoulli[n, x]` — полином Бернулли n -й степени;

```

Binomial[5,3]
10
N[BernoulliB[10]]
0.0757576
BernoulliB[5,x]
- $\frac{x}{6} + \frac{5x^3}{3} - \frac{5x^4}{2} + x^5$ 
EulerE[2]
-1
EulerE[5,x]
- $\frac{1}{2} + \frac{5x^2}{2} - \frac{5x^4}{2} + x^5$ 
Fibonacci[3]
2
Fibonacci[7,x]
 $1 + 6x^2 + 5x^4 + x^6$ 
StirlingS1[5,3]
35
StirlingS2[5,3]
25
MoebiusMu[5]
-1
Zeta[6]
 $\frac{\pi^6}{945}$ 
Zeta[3,2]
-1+Zeta[3]

```

Рис. 6.32. Специальные числа и полиномы

- ☐ EulerE[n] — n -е число Эйлера;
- ☐ EulerE[n, x] — полином Эйлера n -й степени;
- ☐ Fibonacci[n] — n -е число Фибоначчи;
- ☐ Fibonacci[n, x] — полином n -й степени Фибоначчи;
- ☐ stirlings1[n, m] — число Стирлинга первого рода;
- ☐ stirlings2[n, m] — число Стирлинга второго рода;
- ☐ Cyclotomic[n, x] — циклический полином порядка n ;
- ☐ MoebiusMu[n] — значение функции Мебиуса;
- ☐ Zeta[x] — дзета-функция Римана;
- ☐ Zeta[x, a] — обобщенная дзета-функция Римана.

Пример 6.26

Необходимо вычислить специальные числа и полиномы при произвольных значениях n, m, a, x .

Решения приведены на рис. 6.32.

6.3. Интегральные преобразования

Интегральные преобразования находят широкое применение при решении дифференциальных уравнений, вычислении предельных значений функции $f(x)$, исследовании динамики систем управления, систем массового обслуживания и во многих других технических и научных задачах.

Наиболее популярными являются преобразования Лапласа, Фурье и Z -преобразование.

6.3.1. Преобразование Лапласа

Преобразование Лапласа функции $f(x)$ имеет вид:

$$L\{f(x)\} = \int_{-\infty}^{\infty} f(x)e^{-sx} dx, \quad (6.1)$$

где $f(x)$ — функция, преобразование Лапласа которой необходимо найти.

Если аргументом функции является время t , то преобразование Лапласа имеет вид:

$$L\{f(t)\} = \int_0^{\infty} f(t)e^{-st} dt. \quad (6.2)$$

Получим преобразование Лапласа для некоторых простых функций.

Пусть

$$f(t) = a ,$$

$$f(t) = e^{-at} ,$$

$$f(t) = \sin(\omega t) .$$

Тогда на основании (6.2) имеем:

$$L(a) = \int_0^{\infty} a e^{-st} dt = \frac{a}{s} ,$$

$$L(e^{-at}) = \int_0^{\infty} e^{-at} e^{-st} dt = \frac{1}{s+a} ,$$

$$L(\sin(\omega t)) = \int_0^{\infty} \sin(\omega t) e^{-st} dt = \frac{1}{s^2 + \omega^2} .$$

Преобразование Лапласа производной n -го порядка имеет вид:

$$L\left(\frac{d^n x(t)}{dt^n}\right) = s^n x(s) - x_{(0)}^{(n-1)} - x_{(0)}^{(n-2)} - \dots - x(0) ,$$

где $x_{(0)}^i$ — значение i -й производной при $t = 0$.

Преобразование Лапласа интеграла имеет вид:

$$L\left(\int_0^t f(t) dt\right) = \frac{1}{s} f(s) .$$

С помощью преобразования Лапласа можно существенно упростить решение ряда задач, связанных с определением пределов функций. Для этого служат следующие предельные теоремы:

$$\begin{aligned} \lim_{t \rightarrow 0} f(t) &= \lim_{s \rightarrow \infty} s f(s) , \\ \lim_{t \rightarrow \infty} f(t) &= \lim_{s \rightarrow 0} s f(s) . \end{aligned} \tag{6.3}$$

В книгах и справочниках приводятся преобразования Лапласа многих функций. Эти результаты легко получить с помощью систем компьютерной алгебры.

Преобразование Лапласа позволяет сравнительно просто получить решение задачи. Однако оно будет не в той области независимой переменной, которая нужна пользователю.

Для получения решения в нужной области, например временной, необходимо найти оригинал полученной функции. Для этой цели и служит обратное преобразование Лапласа.

Обратное преобразование Лапласа имеет вид:

$$f(t) = \frac{1}{2\pi i} \int_{\delta-i\infty}^{\delta+i\infty} F(s)e^{st} ds. \quad (6.4)$$

Существуют таблицы обратных преобразований различных функций. Однако при наличии универсальных программных средств символьной математики обращаться к ним нет необходимости. Система Mathematica найдет оригинал искомой функции.

Преобразование Лапласа наиболее часто используется при решении систем дифференциальных уравнений. Компьютерные технологии решения систем линейных дифференциальных уравнений с помощью преобразования Лапласа приводятся в *главе 10*.

Интегральные преобразования Лапласа в системе Mathematica реализуются с помощью следующих встроенных функций:

- `LaplaceTransform[f, t, s]` — возвращает прямое преобразование Лапласа функции $f(x)$ в виде новой функции $f(s)$;
- `InverseLaplaceTransform[f, s, t]` — возвращает преобразование функции $f(s)$ в функцию $f(t)$ (обратное преобразование Лапласа);
- `LaplaceTransform[f(t1, t2, ...), (t1, t2, ...), (s1, s2, ...)]` — возвращает прямое преобразование Лапласа функции $f(t_1, t_2, \dots)$ в виде новой функции $f(s_1, s_2, \dots)$;
- `InverseLaplaceTransform[f(s1, s2, ...), (s1, s2, ...), (t1, t2, ...)]` — возвращает обратное преобразование Лапласа функции $f(s_1, s_2, \dots)$ в функцию $f(t_1, t_2, \dots)$.

Имена переменных s и t могут обозначаться любыми другими символами.

Технология реализации преобразований Лапласа очевидна. Необходимо только помнить, что функции преобразования Лапласа находятся в подпакете `LaplaceTransform` пакета `Calculus` (для случая использования системы Mathematica версий 3, 4).

Приведем примеры образования преобразований Лапласа.

Пример 6.27

Представить в преобразованиях Лапласа следующие функции:

$$y_1 = ab^t,$$

$$y_2 = \sin ax + \cos ax,$$

$$y_3 = \frac{2t+1}{(t-1)(t+1)(2t+3)(2t-3)(t+5)},$$

$$y_4 = [t \ln t + 1, \frac{1}{t^2}, te^t].$$

Преобразование Лапласа функций y_4 представить в виде вектора.

Решение приведено на рис. 6.33.

```

LaplaceTransform[a b^t, t, s]

$$\frac{a}{s - \text{Log}[b]}$$

LaplaceTransform[Sin[a t]+Cos[a t], t, s]

$$\frac{s}{a^2 + s^2} + \frac{\sqrt{a^2} \text{Sign}[a]}{a^2 + s^2}$$

LaplaceTransform[Sin[4 t]+Cos[4 t], t, s]

$$\frac{4}{16 + s^2} + \frac{s}{16 + s^2}$$

LaplaceTransform[(2 t+1)/((t+1) (t-1) (2 t+3) (2 t-3) (t+5)), t, s]

$$\frac{8}{195} e^{-3s/2} \text{Gamma}[0, -\frac{3s}{2}] - \frac{1}{20} e^{-s} \text{Gamma}[0, -s] -$$


$$\frac{1}{40} e^s \text{Gamma}[0, s] + \frac{4}{105} e^{3s/2} \text{Gamma}[0, \frac{3s}{2}] -$$


$$\frac{3}{728} e^{5s} \text{Gamma}[0, 5s]$$

LaplaceTransform[{t Log[t]+1, 1/t^2, t Exp[t]}, t, s]

$$\left\{ \frac{1}{s} - \frac{-1 + \text{EulerGamma} + \text{Log}[s]}{s^2}, s(-1 + \right.$$


$$\left. \text{EulerGamma} + \text{Log}[s]), \frac{1}{(-1+s)^2} \right\}$$

N[%]

$$\left\{ \frac{1}{s} - \frac{1. \cdot (-0.42278433509846713 + \text{Log}[s])}{s^2}, \right.$$


$$\left. s(-0.42278433509846713 + \text{Log}[s]), \frac{1}{(-1. + s)^2} \right\}$$


```

Рис. 6.33. Преобразование Лапласа функций примера 6.27

Обсудим полученные результаты.

Из рис. 6.33 видно, что система успешно решила первую и вторую задачи. При этом во второй задаче она выдала условие, при котором задача может быть решена.

Функция $\text{sign}[a]$ возвращает -1 , 0 или 1 , если аргумент a , соответственно, отрицательный, равный нулю или положительный. Строкой ниже показано преобразование Лапласа при $a=4$.

Преобразование Лапласа функции y_3 в явном виде не получено.

Также не получены преобразования функций y_4 . Для получения ответа в виде вещественных чисел пришлось использовать функцию `N[%]`. Напомним, что константа `EulerGamma` является постоянной Эйлера, равной 0.577216...

Пример 6.28

Необходимо получить обратное преобразование Лапласа функций из примера 6.27. Убедиться в том, что преобразования выполнены верно и соответствуют оригиналам.

Решение приведено на рис. 6.34.

```
InverseLaplaceTransform[a/(s-Log[b]),s,t]
a b^t

InverseLaplaceTransform[(4+s)/(16+s^2),s,t]
Cos[4 t]+Sin[4 t]

V=1/s-(-1+EulerGamma+Log[s])/s^2
InverseLaplaceTransform[V,s,t]
1/s - (-1 + EulerGamma + Log[s]) / s^2
1+t Log[t]

InverseLaplaceTransform[s (-1+EulerGamma+Log[s]),s,t]

Integrate[-DiracDelta'[-s+t]/s,{s,0,t},Assumptions->True]-
DiracDelta'[t]+EulerGamma DiracDelta'[t]
InverseLaplaceTransform[1/(-1+s)^2,s,t]
e^t t
```

Рис. 6.34. Обратные преобразования Лапласа функций примера 6.27

Из рис. 6.34 видно, что в одном из примеров обратное преобразование в явном виде не получено.

Пример 6.29

Получить преобразование Лапласа следующих функций:

☐ $f(t, x) = t \sin x$ с новыми переменными s_1, s_2 ;

☐ $f(t, x, z)$ с новыми переменными s_1, s_2, s_3 .

Решение приведено на рис. 6.35.

$$\begin{aligned}
 &\text{LaplaceTransform}[t \sin[x], \{t, x\}, \{s1, s2\}] \\
 &\quad \frac{1}{s1^2 (1 + s2^2)} \\
 &\text{LaplaceTransform}[t^2 \cos[x] + x \log[z], \{t, x, z\}, \{s1, s2, s3\}] \\
 &\quad \frac{2 s2}{s1^3 (1 + s2^2) s3} - \frac{\text{EulerGamma} + \log[s3]}{s1 s2^2 s3} \\
 &\text{N[\%]} \\
 &\quad \frac{2. s2}{s1^3 (1. + s2^2) s3} - \frac{1. (0.577216 + \log[s3])}{s1 s2^2 s3}
 \end{aligned}$$

Рис. 6.35. Преобразование Лапласа функций примера 6.29

6.3.2. Z-преобразование

Интегральные Z-преобразования в системе Mathematica реализуются с помощью следующих встроенных функций:

- `ZTransform[f, n, z]` — возвращает Z-преобразование выражения $f(n)$, представляющее собой функцию $f(z)$;
- `InverseZTransform[f, z, n]` — обратное Z-преобразование функции $f(z)$.

$$\begin{aligned}
 &\text{ZTransform}[t \exp[t], t, z] \\
 &\quad \frac{e^z}{(e - z)^2} \\
 &\text{InverseZTransform}[\%, z, t] \\
 &\quad e^t t \\
 &\text{ZTransform}[\{3, a n, \exp[-a n], \sinh[a n], \sin[w n]\}, n, t] \\
 &\quad \left\{ \frac{3 t}{-1 + t}, \frac{a t}{(-1 + t)^2}, \frac{e^a t}{-1 + e^a t}, \right. \\
 &\quad \left. \frac{1}{2} t \left(\frac{1}{-e^a + t} + \frac{e^a}{1 - e^a t} \right), \frac{i (-1 + e^{2 i w}) t}{2 (t + e^{2 i w} t - e^{i w} (1 + t^2))} \right\} \\
 &\text{InverseZTransform}[\%, t, n] \\
 &\quad \left\{ 3, a n, e^{a - a (1 + n)}, \frac{1}{2} (-e^{-a n} + e^{a n}), \frac{1}{2} i (e^{-i n w} - e^{i n w}) \right\} \\
 &\text{FullSimplify}[\%] \\
 &\quad \{3, a n, e^{-a n}, \sinh[a n], \sin[n w]\}
 \end{aligned}$$

Рис. 6.36. Z-преобразование функций примера 6.30

Технология Z-преобразований очевидна, она не отличается от технологии преобразования Лапласа. Приведем примеры.

Пример 6.30

Необходимо образовать прямое Z-преобразование следующих функций: te^t , 3 , an , e^{-an} , $\sinh(an)$, $\sin(\omega n)$.

Проверить достоверность результатов путем образования обратного Z-преобразования и сравнения результатов с исходными функциями.

Решение приведено на рис. 6.36.

Из рис. 6.36 видно, что Z-преобразования функций получены верно. Однако некоторые функции обратного Z-преобразования не совпадают с исходными. Пришлось использовать функцию упрощения `FullSimplify`, после реализации которой совпадение полное.

6.3.3. Преобразование Фурье

Система Mathematica имеет богатые возможности спектрального анализа и синтеза сигналов. Большое число встроенных функций позволяют осуществлять:

- ☐ спектральный анализ сигналов;
- ☐ гармонический синтез сигналов;
- ☐ фильтрацию сигналов.

Эти функции имеет смысл рассматривать в том случае, если читатель владеет теоретическими основами спектрального анализа. Так как эти основы здесь не излагаются, то мы приведем лишь функции, которые осуществляют прямое и обратное преобразования Фурье.

Преобразования Фурье выполняются в системе Mathematica с помощью следующих функций:

- ☐ `FourierTransform[F(t), t, w]` — возвращает результат прямого преобразования Фурье функции $F(t)$, представленной через параметр w ;
- ☐ `InverseFourierTransform[F(w), w, t]` — возвращает результат обратного преобразования Фурье выражения $F(w)$, представленного через параметр t ;
- ☐ `FourierSinTransform[F(t), t, w]` — возвращает результат синусного преобразования Фурье функции $F(t)$, представленного через параметр w ;
- ☐ `FourierCosTransform[F(t), t, w]` — возвращает результат косинусного преобразования Фурье функции $F(t)$, представленного через параметр w ;
- ☐ `FourierTransform[F, {t1, t2, ...}, {w1, w2, ...}]` — возвращает результат прямого преобразования Фурье функции $F(t_1, t_2, \dots)$, представленного через параметры $w_1, w_2, \dots$;

- `InverseFourierTransform[F, {w1, w2, ...}, {t1, t2, ...}]` — возвращает результат обратного преобразования Фурье функции $F(w_1, w_2, \dots)$, представленного через параметры $t_1, t_2, \dots$.

Приведем примеры получения преобразований Фурье.

Пример 6.31

Необходимо получить прямое и обратное преобразования Фурье следующих функций:

$$F(t) = te^{at},$$

$$F(t) = t^2 \cos(bt),$$

$$F(t) = t^2 \sin(at),$$

$$F(t_1, t_2) = \cos(t_1, t_2).$$

```

FourierTransform[t Exp[a t], t, w]
-1/2 Sqrt[2 Pi] DiracDelta[-1/2 a + w]
InverseFourierTransform[%, w, t]
e^(a t) t
FourierSinTransform[t^2 Cos[b t], t, w]
1/(Sqrt[2 Pi] (b - w)^3) - 1/(Sqrt[2 Pi] (-b + w)^3) - Sqrt[2/Pi]/(b + w)^3
FourierCosTransform[t^2 Sin[a t], t, w]
-1/(Sqrt[2 Pi] (a - w)^3) + 1/(Sqrt[2 Pi] (-a + w)^3) - Sqrt[2/Pi]/(a + w)^3
FourierTransform[Cos[t1*t2], {t1, t2}, {w1, w2}]
1/2 (e^(-i w1 w2) + e^(i w1 w2))
InverseFourierTransform[%, {w1, w2}, {t1, t2}]
1/2 (e^(-i t1 t2) + e^(i t1 t2))
FullSimplify[%]
Cos[t1 t2]

```

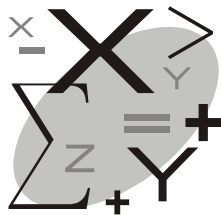
Рис. 6.37. Преобразование функций примера 6.31

Следует использовать все приведенные выше функции Фурье и проверить правильность результатов.

Решение представлено на рис. 6.37.

Из рис. 6.37 видно, что преобразования выполнены верно, т. к. обратные преобразования Фурье совпадают с исходными функциями. В последнем примере для доказательства достоверности решения применена функция упрощения `FullSimplify`.

ГЛАВА 7



Решение оптимизационных задач

Решение оптимизационных задач — одно из наиболее практических приложений математических расчетов.

Значимость задач оптимизации породила большое количество различных математических методов.

Возможности системы Mathematica здесь весьма ограничены: отсутствуют встроенные функции реализации многих из этих методов.

Однако следует иметь в виду, что система Mathematica имеет большие возможности реализации любых методов путем программирования на ее языке.

Более того, имеется возможность решать задачи оптимизации в аналитическом виде.

Возможности системы Mathematica решать задачи оптимизации с помощью ее встроенных функций ограничиваются в основном следующими случаями:

- ☐ поиск максимального и минимального числа из множества чисел, представленных в виде векторов или матриц;
- ☐ определение локального минимума или максимума аналитической функции;
- ☐ определение глобального минимума или максимума аналитической функции, в частности, решение оптимизационных задач линейного программирования.

Изложим эти методы и приведем примеры.

7.1. Поиск минимального и максимального числа в перечне чисел

Задача поиска минимального или максимального числа формулируется следующим образом: даны числа, представленные в виде вектора или матрицы; найти минимальное и максимальное числа.

На первый взгляд кажется, что эта задача слишком проста и большого смысла не имеет. Но это только кажется. Конечно, искать минимальное или максималь-

ное числа из семейства чисел 1, 2, 3, 5, 7, 20, 8 смысла нет, т. к. решение очевидно: максимальным является число 20, а минимальным — 1. А если заданы рациональные числа, например, вида $2/3$, $4/7$, $17/21$, $3/7$, $121/164$, $12/21$ или вида невычисленных функций: $\sin 1.5$, $\cos 2.5$, $e^{-0.25}$, 0.96^6 , $\ln 3.2$. Какое здесь число является минимальным, а какое максимальным?

Система Mathematica имеет следующие функции определения максимального и минимального числа списка:

- ☐ $\text{Max}[x_1, x_2, \dots]$ — возвращает максимальное из чисел x_i ;
- ☐ $\text{Max}[\{x_1, x_2, \dots\}, \{y_1, y_2, \dots\}]$ — возвращает максимальное из матрицы чисел;
- ☐ $\text{Min}[x_1, x_2, \dots]$ — возвращает минимальное из чисел x_i ;
- ☐ $\text{Min}[\{x_1, x_2, \dots\}, \{y_1, y_2, \dots\}]$ — возвращает минимальное из матрицы чисел.

Числа при этом могут быть целыми, рациональными, вещественными. Более того, элементами списка могут быть функции с численными значениями аргументов. При этом откликом во многих случаях может быть число того же типа, что и исходные данные. Это еще раз подтверждает высокую интеллектуальность системы.

Технология поиска искомых чисел элементарна и состоит в следующем:

1. Ввод списка чисел или функций с присвоением ему уникального имени.
2. Ввод функции отыскания максимального или минимального числа.
3. Получение решения нажатием комбинации клавиш <Shift>+<Enter>.

Покажем эту технологию на примерах.

Пример 7.1

Пусть необходимо определить максимальные и минимальные значения чисел вектора $[1, 2.3, -1.7, 3, 4.2, 6.3, 0.25, -1.70001]$ и матрицы $\{\{1, 2, 3\}, \{0.5, 2.8, 7\}, \{4, -1, 3.2\}\}$.

Решение приведено на рис. 7.1.

Приведем пример, когда числа заданы в рациональной форме и в виде функций.

Пример 7.2

Пусть необходимо определить максимальные и минимальные числа, представленные в виде следующих векторов:

```
[2/3, 3/4, 4/5, 4/3, 3/2, 5/4]
[ Sin 1, Sin 2, Sin 0.5, Sin 3.14, Sin 8]
[ e-0.1, e-0.6, 1/2.5, Sin 1, Cos 1, 0.65]
```

Решение приведено на рис. 7.2.

```

f1={1,2.3,-1.7,3,4.2,6.3,0.25,-1.70001}
Max[f1]
  {1,2.3,-1.7,3,4.2,6.3,0.25,-1.70001}
  6.3
Min[f1]
  -1.70001
f2={{1,2,3},{0.5,2.8,7},{4,-1,3.2}}
Max[f2]
  {{1,2,3},{0.5,2.8,7},{4,-1,3.2}}
  7
Min[f2]
  -1

```

Рис. 7.1. Определение максимальных и минимальных чисел вектора и матрицы

```

f3={2/3,3/4,4/5,4/3,3/2,5/4}
Max[f3]
  { 2/3, 3/4, 4/5, 4/3, 3/2, 5/4 }
  3/2
Min[f3]
  2/3
f4={Sin[1],Sin[2],Sin[0.5],Sin[3.14],Sin[8]}
Max[f4]
  {Sin[1],Sin[2],0.479426,0.00159265,Sin[8]}
  Sin[8]
Min[f4]
  0.00159265
f5={E^(-0.1),E^(-0.6),1/2.5,Sin[1],Cos[1],0.6^5}
Max[f5]
  0.904837
Min[f5]
  0.07776
N[f5]
  {0.904837,0.548812,0.4,0.841471,0.540302,0.07776}

```

Рис. 7.2. Решение задач примера 7.2

Из рис. 7.2 видно, что система выдала искомые числа в рациональной форме, в которой они и были заданы в виде вектора.

В случае второго вектора система выдала максимальное число в виде $\sin[8]$. Она не вычислила значение синуса, т. к. аргумент задан в виде целого числа, при котором точное значение синуса не существует.

При определении искомых чисел третьего вектора система выдала решение в численном виде, т. к. аргументы функций представлены в вещественной форме. Однако при этом элементы вектора, соответствующие полученным значениям искомых чисел, нам неизвестны. Пришлось представить элементы вектора в виде чисел с помощью функции $N[f5]$. Теперь видно, что максимальным является элемент $e^{-0.1}$, а минимальным 0.65 .

7.2. Классический метод определения экстремума аналитической функции

В точке экстремума (максимума или минимума) производная функции $y(x)$ равна нулю. Тогда координату x можно найти, если решить уравнение $y'(x)$. Подставив это значение в исходную функцию, получим координаты точки максимума (минимума).

Компьютерная технология решения задачи отыскания экстремума функции этим традиционным методом с помощью системы Mathematica состоит в выполнении следующих процедур:

- ☐ построение графика функции $y(x)$ с целью установления наличия экстремумов функции $y(x)$ и установления приближенного значения координат точек экстремума;
- ☐ определение производной $y'(x)$ с помощью встроенной функции $D[f, x]$;
- ☐ определение вещественных корней уравнения $y'(x) = 0$ с помощью одной из функций системы;
- ☐ вычисление значений функции $y(x)$ при значениях аргументов, равных корням уравнения.

Эта методика достаточно проста и очевидна. Однако при ее практической реализации могут возникнуть трудности с определением корней уравнения $y'(x) = 0$. Здесь возможны случаи, когда система Mathematica решает уравнение и выдает все корни, выдает только один из нескольких корней или вовсе не решает уравнение. Сложности возникают в тех случаях, когда уравнение $y'(x) = 0$ является трансцендентным.

Рассмотрим технологию определения экстремумов аналитической функции на примерах.

Пример 7.3

Определить координаты экстремальных точек функции

$$f = \ln(4 - 2x) + x^2 - 2.$$

Число экстремальных точек определить по виду графика функции, а корни производной $f'(x)$ вычислить с помощью функции `NSolve[f==0, x]`.

Решение задачи приведено на рис. 7.3.

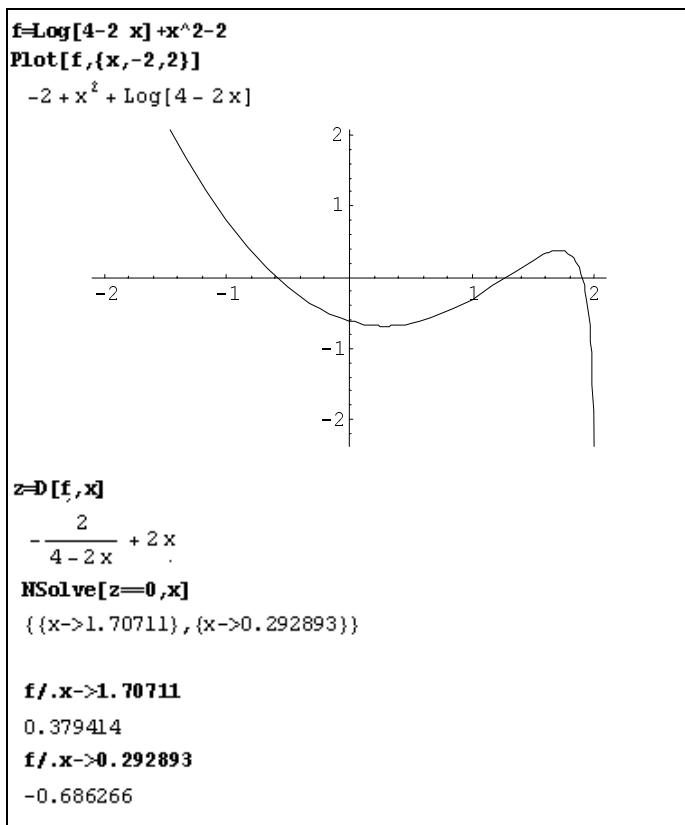


Рис. 7.3. Решение примера 7.3

Из рис. 7.3 видно, что функция имеет максимум и минимум. При этом система с помощью функции `NSolve[f==0, x]` нашла абсциссы точек экстремума, определив два корня уравнения $f'(x) = 0$ за одну команду.

Вычислив значения функции методом подстановки, получим следующие координаты экстремальных точек: максимума — [1.70711, 0.379414], минимума — [0.292893, -0.686266].

Пример 7.4

Необходимо определить координаты экстремальных точек функции:

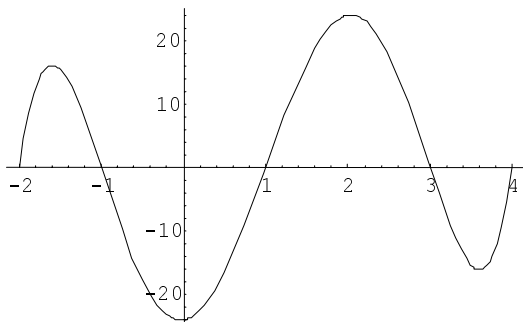
$$f = x^5 - 5x^4 - 3x^3 + 29x^2 + 2x - 24.$$

Решение приведено на рис. 7.4.

```
f=x^5-5 x^4-3 x^3+29 x^2+2 x-24
```

```
Plot[f, {x, -2, 4}]
```

```
- 24 + 2 x + 29 x^2 - 3 x^3 - 5 x^4 + x^5
```



```
y=D[f, x]
```

```
2 + 58 x - 9 x^2 - 20 x^3 + 5 x^4
```

```
NSolve[y, x]
```

```
{{x->-1.59426}, {x->-0.0343141}, {x->2.03431}, {x->3.59426}}
```

```
f/.x->-1.59426
```

```
16.0767
```

```
f/.x->-0.0343141
```

```
-24.0344
```

```
f/.x->2.03431
```

```
24.0344
```

```
f/.x->3.59426
```

```
-16.0767
```

Рис. 7.4. Определение координат точек экстремума функции примера 7.4

Из рис. 7.4 видно, что, как и в предыдущем примере, функция `NSolve[f, x]` определила все корни уравнения $f'(x) = 0$ одной командой, существенно облегчив вычисление координат экстремальных точек. В таких случаях нет необходимости строить график исходной функции $f(x)$. Разве только для контроля правильности определения координат экстремальных точек.

В большинстве случаев уравнение $f'(x) = 0$ является трансцендентным. В таких случаях функция `NSolve[f'(x), x]` не может сразу найти все корни уравнения. Приходится применять иные функции решения уравнений, требующие задания близлежащего значения x или даже области изоляции корня.

Рассмотрим такие примеры.

Пример 7.5

Необходимо определить координаты экстремумов функции:

$$f(x) = (2^x - 4x) \cdot (3^x - 9).$$

Решение приведено на рис. 7.5.

В данном случае определить корни уравнения

$$(3^x - 9x) \cdot (-4 + 2^x \text{Log}[2]) + (2^x - 4x) \cdot (-9 + 3^x \text{Log}[3]) = 0$$

с помощью функции `Solve[f, x]` невозможно. Корни определены с помощью трехкратного применения функции `FindRoot[Z, {x, x0}]`, где x_0 — численное значение x вблизи экстремума. Численные методы определения корней алгебраических и трансцендентных уравнений изложены в главе 8.

7.2.1. Определение координат точек перегиба

К особым точкам аналитической функции относятся также *точки перегиба*. В этих точках вторая производная равна нулю. Классический метод определения экстремальных точек (максимума и минимума) позволяет также определить координаты точек перегиба. Отличие состоит лишь в том, что приходится искать корни не первой, а второй производной функции $y(x)$. В остальном технология определения координат точек перегиба не отличается от технологии определения координат точек максимума и минимума.

Покажем технологию на примере.

Пример 7.6

Необходимо определить координаты точек перегиба функции, приведенной в примере 7.3.

Решение показано на рис. 7.6.

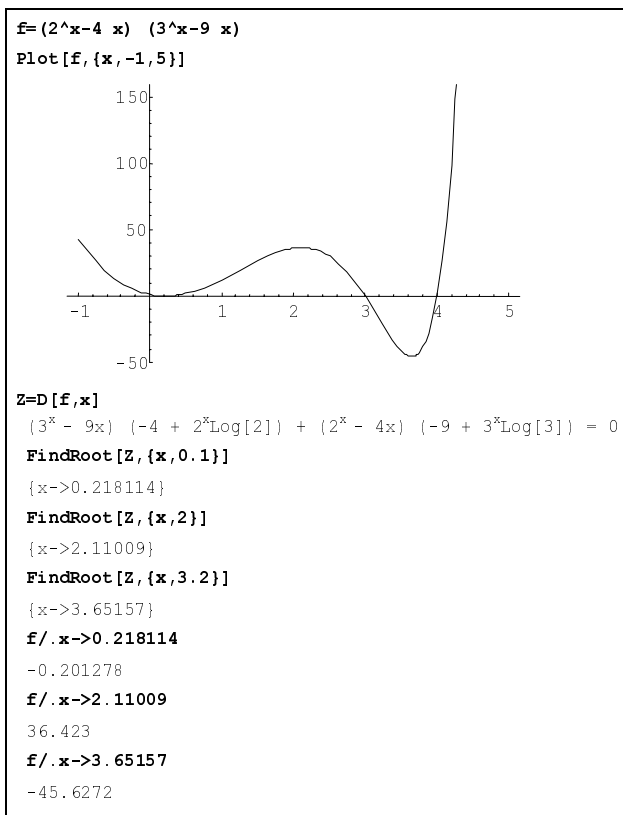


Рис. 7.5. Определение координат экстремальных точек функции примера 7.5

```

f=Log[4-2 x]+x^2-2
Z=D[f,{x,2}]
-2 + x^2 + Log[4 - 2 x]
2 -  $\frac{4}{(4 - 2 x)^2}$ 
NSolve[Z,x]
{{x->2.70711},{x->1.29289}}
f/.x->2.70711
5.67502 +3.14159 i
f/.x->1.29289
0.0181427

```

Рис. 7.6. Определение координат точек перегиба аналитической функции

7.3. Поиск локального минимума аналитической функции с помощью встроенных функций системы Mathematica

Система Mathematica предоставляет встроенную функцию поиска локального минимума аналитической функции. Она имеет вид:

`FindMinimum[f(x), {x, x0}]`

где:

□ $f(x)$ — аналитическая функция аргумента x ;

□ x_0 — значение аргумента вблизи локального минимума.

Функция `FindMinimum[f(x), {x, x0}]` находит координаты всех минимумов функции $f(x)$ путем ее повторений столько раз, сколько имеется локальных минимумов. При этом каждый раз меняется значение x_0 .

Технология отыскания минимума функции проста и состоит в выполнении следующих действий:

1. Построение графика аналитической функции $f(x)$ с целью определения числа экстремальных точек и выбора значений x_0 .
2. Ввод встроенной функции `FindMinimum[f(x), {x, x0}]`.
3. Получение решения путем нажатия комбинации клавиш <Shift>+<Enter>.

Если аналитическая функция имеет максимум, то технология остается прежней. Необходимо только изменить знак функции $f(x)$ на обратный, умножив ее на -1 .

Рассмотрим технологию определения координат минимума аналитической функции на примерах.

Пример 7.7

Необходимо определить координаты экстремумов следующей аналитической функции:

$$x^4 - 13x^3 + 35x^2 + 13x - 36.$$

Процедуры решения задачи приведены на рис. 7.7.

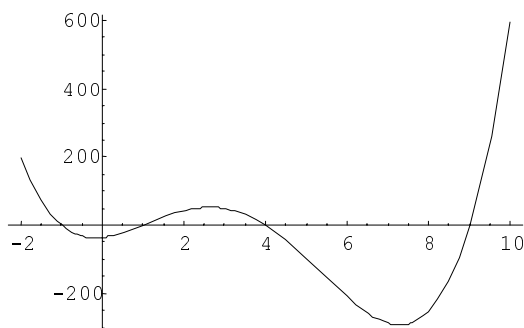
Из рис. 7.7 видно, что координатами точек минимума и максимума являются: $[-0.17, -37]$, $[2.63, 51.64]$, $[7.29, -293.36]$. Следует иметь в виду, что ординату максимума необходимо умножить на -1 .

Этот метод по сравнению с классическим имеет то преимущество, что откликом здесь является координата минимума (x и y) и вычисление значений y не требуется. Его недостаток в том, что он не позволяет вычислить одновременно координаты всех экстремальных точек.


```
f=x^4-13 x^3+35 x^2+13 x-36
```

```
Plot[f,{x,-2,10}]
```

```
-36+13 x+35 x^2-13 x^3+x^4
```



```
FindMinimum[f,{x,-0.5}]
```

```
{-37.1338,{x->-0.169441}}
```

```
FindMinimum[-f,{x,2}]
```

```
{-51.6363,{x->2.63205}}
```

```
FindMinimum[f,{x,7}]
```

```
{-293.358,{x->7.28739}}
```

Рис. 7.7. Отыскание экстремумов с помощью встроенной функции

Кроме функции `FindMinimum[f(x), {x, x0}` система Mathematica имеет следующие четыре функции, позволяющие отыскать локальный минимум аналитической функции:

- ☐ `NMaximize[f, x]` — ищет единственный локальный максимум функции $f(x)$;
- ☐ `NMinimize[f, x]` — ищет единственный локальный минимум функции $f(x)$;
- ☐ `NMaximize[{f, Zxy},{x, y, ...}]` — ищет локальный максимум функции $f(x)$, определенный условием Z_{xy} ;
- ☐ `NMinimize[{f, Zxy},{x, y, ...}]` — ищет локальный минимум функции $f(x)$, определенный условием Z_{xy} .

Технология определения локального максимума (минимума) с помощью приведенных функций практически не отличается от технологии определения локального минимума функцией `FindMinimum[f(x), {x, x0}`.

Приведем примеры определения экстремальных точек с помощью этих функций.

Пример 7.8

Необходимо найти координаты максимума (минимума) аналитических функций: $xe^{-x} + 1$, $3^x - 9x + 3$.

Изначально построим график функции и определим, имеет ли функция экстремальную точку и какая она (максимум или минимум). По виду графика выберем встроенную функцию. Решение выполним в последовательности приведенных функций.

Технология решения задачи и ее результаты показаны на рис. 7.8.

Из рис. 7.8 видно, что в результате решения получены координаты экстремальных точек, соответствующие графикам. При этом функции `NMaximize[f, x]` и `NMinimize[f, x]` не требуют задания аргумента вблизи локального максимума (минимума), как это было в случае использования функции `FindMinimum[f(x), {x, x0}]`.

Рассмотрим теперь случай, когда аналитическая функция имеет несколько экстремумов.

Пример 7.9

Необходимо определить координаты экстремальных точек следующих аналитических функций:

$$\ln(5 - 3x) + x^2 - 3,$$

$$x^5 - 5x^4 - 4x^3 + 31x^2 + 2x - 30.$$

Решение приведено на рис 7.9.

Обсудим результаты решения задач.

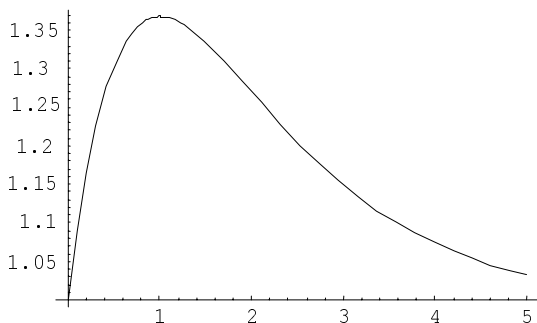
При определении с помощью функций `NMaximize[f, x]`, и `NMinimize[f, x]` координат экстремальных точек аналитической функции, имеющей много максимумов и минимумов, возникает ряд проблем.

В нашем примере функции имеют несколько экстремальных точек: первая функция содержит один максимум и один минимум, вторая — два максимума и два минимума. Решая первую задачу, система определила координаты минимума и не нашла координат максимума (решение абсурдно). Во второй функции система нашла только один максимум и один минимум, находящиеся слева у графика. Определить координаты остальных экстремумов не удастся. В этом существенный недостаток функций `NMaximize[f, x]` и `NMinimize[f, x]`.

Существуют функции `NMaximize[{f, Zxy...}, {x, y, ...}]`, `NMinimize[{f, Zxy...}, {x, y, ...}]`. Эти функции находят координаты экстремальных точек аналитических функций многих аргументов при ограничениях $Z_{xy, \dots}$.

```
f=x Exp[-x]+1
```

```
Plot[f,{x,0,5}]
```

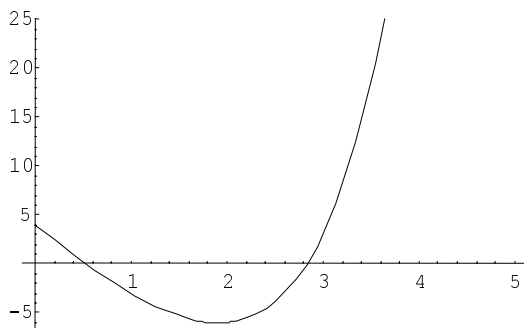
$$1 + e^{-x} x$$


```
NMaximize[f,x]
```

```
{1.36788, {x->1.}}
```

```
f1=3^x-9 x+3
```

```
Plot[f1,{x,0,5}]
```

$$3 + 3^x - 9x$$


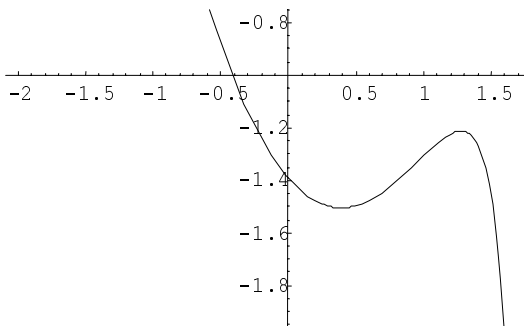
```
NMinimize[f1,x]
```

```
{-6.03739, {x->1.91439}}
```

Рис. 7.8. Отыскание локального максимума и минимума аналитических функций с помощью функций `NMaximize[f, x]` и `NMinimize[f, x]`

```
f2=Log[5-3 x]+x^2-3
```

```
Plot[f2,{x,-2,3}]
```

$$-3 + x^2 + \text{Log}[5 - 3x]$$


```
NMaximize[f2,x]
```

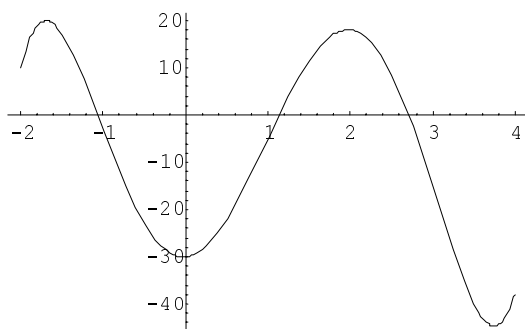
```
{9.65572578174896 x 10919, {x -> -9.82635526619558 x 10459}}
```

```
NMinimize[f2,x]
```

```
{-1.50504, {x->0.392375}}
```

```
f3=x^5-5 x^4-4 x^3+31 x^2+2 x-30
```

```
Plot[f3,{x,-2,4}]
```

$$-30 + 2x + 31x^2 - 4x^3 - 5x^4 + x^5$$


```
NMaximize[f3,x]
```

```
{19.8942, {x->-1.68837}}
```

```
NMinimize[f3,x]
```

```
{-30.0321, {x->-0.0320697}}
```

Рис. 7.9. Определение координат экстремальных точек аналитических функций примера 7.9

Приведем примеры определения координат функций многих переменных при ограничениях на их аргументы.

Пример 7.10

Необходимо определить координаты точек максимума и минимума следующих функций:

$$\sin x - e^{x+y}, \quad x + y < 1;$$

$$2^x - 4(x+y), \quad x + y < 5;$$

$$4(x+y+z) - 2^{x+y}, \quad x + y + z < 7.$$

У первой и второй функций определим координаты минимума, у третьей — максимума.

Решение приведено на рис. 7.10.

```

Minimize [{ Sin[x]-Exp[x+y], x+y<1 }, {x,y}]
{-3.71828, {x→-1.5708, y→2.5708}}

Minimize [{ 2^x-4 (x+y), x+y<5 }, {x,y}]
{-20., {x→-50.3938, y→55.3938}}

Maximize [{ 4 (x+y+z)-2^ (x+y), x+y+z<7 }, {x,y,z}]
{28., {x→-1.13548, y→-49.229, z→57.3645}}

```

Рис. 7.10. Определение максимумов и минимумов функций многих переменных

7.4. Отыскание глобального максимума (минимума) аналитической функции

Задача поиска глобального максимума (минимума) является задачей математического программирования. Среди этих задач наиболее часто приходится решать задачи линейного программирования.

Задача линейного программирования формулируется следующим образом.

Задана следующая линейная функция независимых переменных $x_1, x_2, \dots, x_n$:

$$a_1x_1 + a_2x_2 + \dots + a_nx_n,$$

где a_i — числа, называемые коэффициентами линейной функции, $i = 1, 2, \dots, n$.

Известно также несколько уравнений и неравенств, независимыми переменными которых являются $x_1, x_2, \dots, x_n$.

Таковыми уравнениями (неравенствами) могут быть:

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b \quad (a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b), \quad x_i \geq 0, \quad i = 1, 2, \dots, n$$

и ряд других.

Среди множества вариантов, удовлетворяющих условиям ограничения, необходимо найти совокупность переменных $x_1, x_2, \dots, x_n$, при которых линейная функция получает максимальное (минимальное) значение.

Функция $a_1x_1 + a_2x_2 + \dots + a_nx_n$ в теории математического программирования называется *целевой функцией*, а совокупность равенств и неравенств — *ограничениями*.

Сформулированная задача линейного программирования встречается на практике очень часто при решении оптимизационных задач в технике, управлении, экономике.

Система Mathematica имеет встроенные функции решения задач математического программирования. Эти функции имеют вид:

```
ConstrainedMax[f, {Q}, {x1, x2, ..., xn}]
```

```
ConstrainedMin[f, {Q}, {x1, x2, ..., xn}]
```

```
LinearPrograming[c, m, b]
```

В этих функциях приняты следующие обозначения:

- ☐ f — целевая функция;
- ☐ Q — вектор ограничений;
- ☐ x_i — искомые независимые переменные;
- ☐ c — минимизируемая величина;
- ☐ m — ограничения;
- ☐ b — величина ограничений.

Функция `ConstrainedMax[f, {Q}, {x1, x2, ..., xn}]` ищет глобальный максимум, т. е. такие значения x_i , при которых выполняются все ограничения Q , а целевая функция имеет максимальное значение.

Функция `ConstrainedMin[f, {Q}, {x1, x2, ..., xn}]` ищет глобальный минимум, т. е. такие значения x_i , при которых выполняются все ограничения Q , а целевая функция имеет минимальное значение.

Функция `LinearPrograming[c, m, b]` ищет вектор x . c при условиях $m.x \geq b$, $x \geq 0$.

Решение такой сложной задачи, как отыскание глобального максимума (минимума), в системе Mathematica предельно просто:

1. Ввод функции `ConstrainedMax[f, {Q}, {x1, x2, ..., xn}]`.
2. Получение решения путем нажатия комбинации клавиш <Shift>+<Enter>.

Покажем технологию отыскания глобального максимума и минимума на конкретных примерах.

Пример 7.11

Существует три технологии выпуска некоторой продукции. Для ее изготовления необходимо иметь три вида сырья. Каждая из технологий требует определенного количества сырья данного вида, которое имеется в ограниченных количествах. Объем выпускаемого продукта зависит от технологии изготовления и известен для каждой из технологий.

Необходимо найти такую технологию, при которой объем выпускаемой продукции максимален.

Исходные данные производства приведены в табл. 7.1.

Таблица 7.1. Исходные данные производства

Способ производства	Потребность в сырье (усл. ед.)			Выпуск продукции
	Первого вида	Второго вида	Третьего вида	
1	2	3	2	30
2	1	2	3	22
3	2	1	1	24

Технология производства допускает смену способа производства в течение рабочего дня.

7.4.1. Математическая формулировка задачи

Обозначим x_1 , x_2 , x_3 — доли рабочего времени, затрачиваемого на производство заданного объема готового продукта при условии обеспечения сырьем, соответственно, при первой, второй и третьей технологиях.

Ограничениями при решении этой задачи являются:

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0;$$

$$x_1 + x_2 + x_3 \leq 1;$$

$$2x_1 + x_2 + 2x_3 \leq 2;$$

$$3x_1 + 2x_2 + x_3 \leq 2;$$

$$2x_1 + 3x_2 + x_3 \leq 2.$$

Обсудим принятые ограничения.

Первое ограничение очевидно: доли рабочего времени не могут быть отрицательными.

Второе допущение означает, что суммарное рабочее время не должно превышать полного рабочего дня.

Третье, четвертое и пятое допущения означают, что расход сырья не должен превышать лимита (двойка для каждого сырья и технологии).

При любой технологии будет выпущено продукта

$$30x_1 + 22x_2 + 24x_3.$$

Эта функция и является целевой.

В результате решения задачи нужно определить значения x_1 , x_2 , x_3 и максимальное значение выпущенного продукта, т. е. определить оптимальную технологию производства.

В нашем случае функция глобального максимума будет иметь вид:

```
ConstrainedMax[30 x1+22 x2+24 x3, { x1>=0, x2>=0, x3>=0,
2 x1+ x2+ 2 x3<=2, 3 x1+2 x2+x3,<=2, 2 x1+3 x2+x3,<=2},
{x1, x2, x3}]
```

Решение задачи приведено на рис. 7.11.

```
ConstrainedMax[30*x1+22*x2+24*x3,
{x1>=0,x2>=0,x3>=0,
x1+x2+x3<=1,
2 *x1+x2+2*x3<=2,
3*x1+2*x2+x3<=2,
2*x1+3*x2+x3<=2},
{x1,x2,x3}]
{27, {x1 -> 1/2, x2 -> 0, x3 -> 1/2}}
```

Рис. 7.11. Решение задачи глобального максимума

Из рис. 7.11 видно, что предприятием за смену будет выпущено 27 единиц продукции по первой и третьей технологиям. Выпуска продукции по второй технологии не должно быть.

Классической задачей линейного программирования является транспортная задача. Приведем и мы такой пример.

Пример 7.12

Имеется четыре склада товаров и три их потребителя. Известны также количество товаров на каждом складе, потребности каждого потребителя, а также стоимость доставки товара до каждого потребителя.

Необходимо составить оптимальный план перевозок, при котором суммарная стоимость перевозок будет минимальной.

Исходные данные задачи приведены в табл. 7.2.

Таблица 7.2. Данные о перевозках товаров со складов до потребителей

	Π_1	Π_2	Π_3	Z_c
C_1	2	4	1	9
C_2	4	1	2	7
C_3	2	3	2	7
C_4	1	2	3	7
Z_n	12	10	8	

В таблице обозначено:

□ C_1, C_2, C_3, C_4 — склады;

□ Π_1, Π_2, Π_3 — потребители;

□ Z_c — количество товаров на складе;

□ Z_n — количество товаров, необходимых потребителю.

Цифры в таблице означают стоимость перевозок в условных единицах со склада C_i до потребителя Π_j , $i = 1, 2, 3, 4, j = 1, 2, 3$.

В данном случае целевой будет следующая функция:

$$2n_{11} + 4n_{12} + n_{13} + 4n_{21} + n_{22} + 2n_{23} + 2n_{31} + 3n_{32} + 2n_{33} + n_{41} + 2n_{42} + 3n_{43},$$

где n_{ij} — количество товара, планируемого для перевозки со склада C_i потребителю Π_j .

Сформулируем ограничения Q .

Так как числа n_{ij} не могут быть отрицательными, то

$$n_{11} \geq 0, n_{21} \geq 0, \dots, n_{42} \geq 0, n_{43} \geq 0.$$

Суммарное количество продуктов на складах и потребности потребителей можно представить следующими уравнениями:

$$n_{11} + n_{12} + n_{13} = 9 ;$$

$$n_{21} + n_{22} + n_{23} = 7 ;$$

$$n_{31} + n_{32} + n_{33} = 7 ;$$

$$n_{41} + n_{42} + n_{43} = 7 ;$$

$$n_{11} + n_{21} + n_{31} + n_{41} = 12 ;$$

$$n_{12} + n_{22} + n_{32} + n_{42} = 10 ;$$

$$n_{13} + n_{23} + n_{33} + n_{43} = 8 .$$

Совокупность неравенств и этих уравнений образуют ограничения Q .

Задача состоит в том, чтобы найти такие значения n_{ij} целевой функции, при которых суммарные затраты на перевозки были бы минимальными. Воспользуемся функцией `ConstrainedMin[f, {Q}, {x1, x2, ..., xn}]`.

Решение задачи приведено на рис. 7.12.

```
ConstrainedMin[
  2*n11+4*n12+1*n13+4*n21+1*n22+2*n23+
  2*n31+3*n32+2*n33+1*n41+2*n42+3*n43 ,
  {n11>=0,n12>=0,n13>=0,
   n21>=0,n22>=0,n23>=0,
   n31>=0,n32>=0,n33>=0,
   n41>=0,n42>=0,n43>=0,
   n11+n12+n13==9,n21+n22+n23==7,
   n31+n32+n33==7,n41+n42+n43==7,
   n11+n21+n31+n41==12,
   n12+n22+n32+n42==10,
   n13+n23+n33+n43==8},
  {n11,n12,n13,n21,n22,n23,n31,n32,n33,
   n41,n42,n43}]
{41,{n11→1,n12→0,n13→8,n21→0,n22→7,n23→0,
  n31→4,n32→3,n33→0,n41→7,n42→0,n43→0}}
```

Рис. 7.12. Решение транспортной задачи

Из рис. 7.12 видно, что все товары со складов доставлены потребителям. Однако далеко не все потребители получили товары по причине высокой стоимости перевозок.

7.5. Примеры на решение оптимизационных задач

Определение максимального и минимального значений чисел:

1. $\left\{ \frac{3}{7}, \frac{4}{9}, \frac{1}{3}, \frac{7}{22}, \frac{37}{121}, \frac{11}{27}, \frac{2}{7} \right\}.$
2. $\left\{ e^{1.2}, 5 \sin 2, \ln 123, \sinh 1.2, \frac{\cos 1}{\sin 1}, \frac{2}{3} \right\}.$

Определение координат экстремальных точек функций:

3. $1.5x^5 + 17x - 21.$
4. $x^4 + 2x^3 - 1.$
5. $x^4 - 4x^3 - 10x^2 + 3x - 4.$
6. $2 \sin(\ln x).$
7. $e^{-2x}(x+2) + 6/x - x.$
8. $\ln(4-2x) + x^3 - 1.$
9. $2^x(3^x - 9x + 1) - 4x3^x + 36x^2 - 4x.$
10. $e^x(x-3) - 3x^2 + 10.5x - 4.5.$
11. $(x^4 + 2x^3 - 1) \cdot (x - \ln(7-4x)).$
12. $x^x.$
13. $x^{2x}(\ln x - 2).$
14. $(x^x + x^{2x})/(x+1).$
15. $x^x + x^{-5x}.$
16. $x \ln x + \sin x.$
17. $x \ln x \cdot (\sin x + \cos x).$
18. $1/x + x + 1/x^2 + x^2 + 1/x^3 + x^3.$
19. $e^x + e^{-x} + e^{2x} + e^{-2x} + e^{3x} + e^{-3x}.$

Поиск глобального минимума:

20.

	Π_1	Π_2	Π_3	Z_c
C_1	2	4	1	12
C_2	3	1	2	7
C_3	3	2	3	10
C_4	1	2	3	6
Z_n	12	14	8	

21.

	Π_1	Π_2	Π_3	Z_c
C_1	4	2	1	19
C_2	4	1	2	8
C_3	2	3	2	7
C_4	1	3	2	13
Z_n	14	16	8	

22.

	Π_1	Π_2	Π_3	Z_c
C_1	1	4	2	12
C_2	4	2	1	9
C_3	2	3	2	7
C_4	2	3	2	7
Z_n	12	12	8	

23.

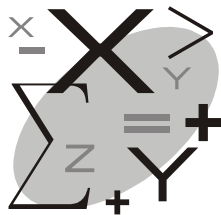
	Π_1	Π_2	Π_3	Z_c
C_1	2	1	4	15
C_2	1	4	2	17
C_3	2	3	2	12
C_4	3	2	2	10
Z_n	12	10	8	

Примечание

Задачи 20—23 — транспортные на поиск глобального минимума перевозок.

Постановка задач аналогична задаче примера 7.12.

ГЛАВА 8



Решение алгебраических и трансцендентных уравнений

Уравнения с одним неизвестным могут быть алгебраическими и трансцендентными. *Трансцендентным* называется такое уравнение, в состав которого входят функции. Например, уравнение $ax^5 - bx + 1 = 0$ является алгебраическим, а $e^{-ax} + 2 \ln x - 1 = 0$ — трансцендентным.

Система Mathematica позволяет решать такие уравнения в аналитическом и численном виде, определять вещественные и комплексные корни.

В системе реализованы многие численные методы. Это позволяет исследовать их достоинства и недостатки и выбрать наиболее рациональный метод для решения данной конкретной задачи. Имеются оригинальные способы проверки достоверности определения корней уравнения.

В данной главе излагаются алгоритмы и компьютерные технологии решения уравнений. Большое число примеров позволяет пользователю усвоить технологию и приобрести опыт решения уравнений с помощью системы Mathematica.

8.1. Решение уравнений в аналитическом виде

8.1.1. Функция *Solve*

Решение уравнений в аналитическом виде в системе Mathematica осуществляется с помощью встроенной функции `solve`, которая имеет вид:

`Solve[f, x]`

где:

- ☐ `f` — уравнение, записываемое в произвольном виде;
- ☐ `x` — искомое неизвестное.

Примеры записи уравнений:

$$ax^2+bx-1==0,$$

$$ax^2+bx==1,$$

$$ax^2== -bx+1.$$

Технология решения уравнений с помощью функции `Solve[f, x]` проста и состоит в выполнении следующих процедур:

1. Ввод уравнения с присвоением ему уникального имени.
2. Ввод функции `Solve[f, x]`, где `f` — имя уравнения.
3. Получение решения путем одновременного нажатия комбинации клавиш `<Shift>+<Enter>`.

Уравнение можно не вводить, а записать его в функцию `Solve` вместо `f`.

Рассмотрим технологию определения корней уравнения в аналитическом виде на примерах.

Пример 8.1

Необходимо определить корни следующих уравнений:

$$x^5 + bx^4 - a^4x - a^4b = 0,$$

$$2^x - 2(a+b) = 0,$$

$$\sin ax + \cos ax = 0,$$

$$2x^2 + 3x - 1 = 0,$$

$$\frac{64}{2541} - \frac{1784}{7623}x + \frac{50}{693}x^2 + \frac{27037}{7623}x^3 + \frac{73229}{15246}x^4 - \frac{208399}{30492}x^5 + \frac{20035}{2772}x^6 + x^7 = 0.$$

Решение приведено на рис. 8.1.

Из рис. 8.1 видно, что все задачи системой решены. В первых трех задачах получены корни в аналитическом виде. В предпоследнем довольно сложном примере значения корней получены в виде точных чисел, представленных в рациональной форме. Если же необходимо получить решение в виде вещественных чисел, то следует воспользоваться командой `N[%]`. Это показано в последней задаче примера 8.1.

Следует иметь в виду, что решение в аналитическом виде не всегда представляется системой в наиболее коротком виде. Для его оптимизации целесообразно применять команды упрощения выражений, например, такие как `Simplify`, `FullSimplify`, `Expand`. Пример использования команды `FullSimplify` показан при решении уравнения $2^x - 2(a+b) = 0$.

Функция Solve имеет большое количество опций. Вывести на экран их можно командой Options[Solve]:

```
Options[Solve]
{InverseFunctions -> Automatic,
 MakeRules -> False, Method -> 3,
 Mode -> Genari, Sort -> True,
 VerifySolutions -> Automatic,
 WorkingPrecision -> ∞}
```

```
F1 := x^5 + b x^4 - a^4 x - a^4 b == 0
Solve[F1, x]
{{x -> -a}, {x -> -I a}, {x -> I a}, {x -> a}, {x -> b}}
F2 := 2^x - 2 (a + b) == 0
Solve[F2, x]
{{x ->  $\frac{\text{Log}[2 (a + b)]}{\text{Log}[2]}$ }}
FullSimplify[%]
{{x ->  $1 + \frac{\text{Log}[a + b]}{\text{Log}[2]}$ }}
F3 := Sin[a x] + Cos[a x] == 0
Solve[F3, x]
{{x ->  $-\frac{\pi}{4 a}$ }, {x ->  $\frac{3 \pi}{4 a}$ }}
Solve $\left[\frac{64}{2541} - \frac{1784 x}{7623} + \frac{50 x^2}{693} + \frac{27037 x^3}{7623} - \frac{73229 x^4}{15246} - \frac{208399 x^5}{30492} + \frac{20035 x^6}{2772} + x^7 == 0, x\right]$ 
{{x -> -8}, {x ->  $-\frac{3}{4}$ }, {x ->  $-\frac{2}{7}$ }, {x ->  $\frac{2}{11}$ },
 {x ->  $\frac{2}{11}$ }, {x ->  $\frac{4}{9}$ }, {x -> 1}}
Solve $[2 x^2 + 3 x - 1 == 0, x]$ 
{{x ->  $\frac{1}{4} (-3 - \sqrt{17})$ }, {x ->  $\frac{1}{4} (-3 + \sqrt{17})$ }}
N[%]
{{x -> -1.78078}, {x -> 0.280776}}
```

Рис. 8.1. Определение корней уравнений примера 8.1

Приведем примеры использования опций функции `Solve`.

Необходимо вычислить тремя методами корни уравнения $\sin ax + \cos ax = 0$, вычислить корни уравнения $2^x - 4x + 1 = 0$ с числом цифр промежуточных вычислений, равным ∞ , 20, 40, 10.

В данном случае используются опции соответственно `Method->` и `WorkingPrecision->`. Решения приведены на рис. 8.2.

```
F := Sin[a x] + Cos[a x] == 0
Solve[F, x]

{{x -> - $\frac{\pi}{4a}$ }, {x ->  $\frac{3\pi}{4a}$ }}

Solve[F, x, Method -> 1]

{{2 Cos[a x]^2 -> 1, Sin[a x] -> -Cos[a x],
  a x -> ArcCos[Cos[a x]]}, {2 Cos[a x]^2 -> 1,
  Sin[a x] -> -Cos[a x], a x -> ArcCos[Cos[a x]]}}

Solve[F, x, Method -> 2]

{{2 Cos[a x]^2 -> 1, Sin[a x] -> -Cos[a x],
  a x -> ArcCos[Cos[a x]]}, {2 Cos[a x]^2 -> 1,
  Sin[a x] -> -Cos[a x], a x -> ArcCos[Cos[a x]]}}

Solve[F, x, Method -> 3]

{{x -> - $\frac{\pi}{4a}$ }, {x ->  $\frac{3\pi}{4a}$ }}

F := 2^x - 4 x + 1 == 0
Solve[F, x, WorkingPrecision -> Infinity]

{{x ->  $\frac{\text{Log}[2] - 4 \text{ProductLog}\left[-\frac{\text{Log}[2]}{2^{2^{3/4}}}\right]}{4 \text{Log}[2]}$ },
  {x ->  $\frac{\text{Log}[2] - 4 \text{ProductLog}\left[-1, -\frac{\text{Log}[2]}{2^{2^{3/4}}}\right]}{4 \text{Log}[2]}$ }}

Solve[F, x, WorkingPrecision -> 20]

{{x -> 0.63942780657650467952}, {x -> 3.8466563011830440018}}

Solve[F, x, WorkingPrecision -> 40]

{{x -> 0.6394278065765046795213353349791139499796695294581061'40},
  {x -> 3.8466563011830440018333384267424616525908444516827777'40}}

Solve[F, x, WorkingPrecision -> 10]

{{x -> 0.639428}, {x -> 3.84666}}
```

Рис. 8.2. Решение задач с применением опций функции `Solve`

Из рис. 8.2 видны действия опций функции `Solve`. Опция `Method` дает возможность решать уравнение тремя методами. Опции `Method->1` и `Method->2` не позволили функции `Solve` найти корни уравнения $\sin ax + \cos ax = 0$. Они только представили исходное уравнение в иной форме. Только опция `Method->3` позволила определить корни уравнения в аналитическом виде, при этом решение то же, что и без опции.

Опция `WorkingPrecision->n` устанавливает число цифр ответа, в нашем случае 20, 40, 10. При $n=\infty$ (Infinity) выдает аналитическое решение с применением специальных функций, а при $n=10$ выдает решение с шестью цифрами.

Опции функции `Solve` можно использовать также при решении систем уравнений. Некоторые из них будут рассмотрены в главе 9.

8.1.2. Функция *Roots*

Эта функция предназначена для определения корней полинома, она имеет вид:

`Roots[f, x]`

где:

- ☐ f — полином, корни которого необходимо найти (может быть представлен в виде уравнения);
- ☐ x — аргумент полинома f .

Откликом функции `Roots[f, x]` являются вещественные и комплексные корни уравнения $f(x)=0$. При этом решение может быть получено в аналитическом и численном виде. Решение в аналитическом виде в общем случае может быть получено для полинома не выше четвертой степени. Решение уравнения $f(x)=0$ не существует, если $f(x)$ — полином пятой и выше степени. Однако если полином может быть разложен на множители, то функция `Roots[f, x]` найдет все корни соответствующего уравнения.

Решения уравнений в указанных случаях покажем на примерах.

Пример 8.2

Необходимо решить следующие уравнения:

$$ax^2 + bx + c = 0,$$

$$ax^4 + bx^2 + c = 0,$$

$$ax^3 + b = 0,$$

$$ax^5 + bx^4 + cx + d = 0,$$

$$(a+x)(b+x)(x+ac)(x+1)(x-1).$$

Roots[a x^2 + b x + c == 0, x]

$$x == \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad || \quad x == \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Roots[a x^4 + b x^2 + c == 0, x]

$$x == \sqrt{-\frac{b}{2a} - \frac{\sqrt{b^2 - 4ac}}{2a}} \quad || \quad x == -\sqrt{-\frac{b}{2a} - \frac{\sqrt{b^2 - 4ac}}{2a}} \\ \left| \quad x == \sqrt{-\frac{b}{2a} + \frac{\sqrt{b^2 - 4ac}}{2a}} \quad || \quad x == -\sqrt{-\frac{b}{2a} + \frac{\sqrt{b^2 - 4ac}}{2a}} \right|$$

Roots[a x^3 + b == 0, x]

$$x == -\frac{(-1)^{2/3} b^{1/3}}{a^{1/3}} \quad || \quad x == \frac{(-1)^{1/3} b^{1/3}}{a^{1/3}} \quad || \quad x == -\frac{b^{1/3}}{a^{1/3}}$$

Roots[a x^5 + b x^4 + c x + d == 0, x]

$$x == \text{Root}[d + c \#1 + b \#1^4 + a \#1^5 \&, 1] \quad || \quad x == \text{Root}[d + c \#1 + b \#1^4 + a \#1^5 \&, 2] \quad | \\ | \quad x == \text{Root}[d + c \#1 + b \#1^4 + a \#1^5 \&, 3] \quad || \quad x == \text{Root}[d + c \#1 + b \#1^4 + a \#1^5 \&, 4] \quad | \\ | \quad x == \text{Root}[d + c \#1 + b \#1^4 + a \#1^5 \&, 5]$$

$$(a + x) (b + x) (x + ac) (x + 1) (x - 1)$$

$$(-1 + x) (1 + x) (a + x) (ac + x) (b + x)$$

Expand[Out [23]]

$$-aacb - aacx - abx - acbx - ax^2 - acx^2 - bx^2 + \\ aacbx^2 - x^3 + aacx^3 + abx^3 + acbx^3 + ax^4 + acx^4 + bx^4 + x^5$$

Roots[Out [24] == 0, x]

$$x == -b \quad || \quad x == -ac \quad || \quad x == -a \quad || \quad x == -1 \quad || \quad x == 1$$

Рис. 8.3. Решение примера 8.2

Решение примера приведено на рис. 8.3.

Из рис. 8.3 видно, что в первых трех примерах корни найдены в аналитическом виде. В четвертом примере представлен полином пятой степени, поэтому его корни не найдены. В последнем примере приведен полином пятой степени, при этом решение получено в аналитическом виде. Это объясняется тем, что многочлен $f(x)$ может быть разложен на множители, как и показано на рис. 8.3. На рис. 8.3 приведен многочлен пятой степени, разложенный на множители. С помощью функции `Expand` раскрыты скобки многочлена, и найдены все его корни с помощью функции `Roots[Out[24], x]`. Здесь `Out[24]` — номер строки, в которой находится полином в раскрытом виде.

Roots[$x^2 + 13/28x - 3/14 == 0, x$]

$$x == \frac{2}{7} \quad || \quad x == -\frac{3}{4}$$

Roots[$2x^7 + 3x^6 - 12x^2 + 7x + 5 == 0, x$]

x == **Root**[$5 + 7 \#1 - 12 \#1^2 + 3 \#1^6 + 2 \#1^7 \&, 1$] |
 | **x** == **Root**[$5 + 7 \#1 - 12 \#1^2 + 3 \#1^6 + 2 \#1^7 \&, 2$] |
 | **x** == **Root**[$5 + 7 \#1 - 12 \#1^2 + 3 \#1^6 + 2 \#1^7 \&, 3$] |
 | **x** == **Root**[$5 + 7 \#1 - 12 \#1^2 + 3 \#1^6 + 2 \#1^7 \&, 4$] |
 | **x** == **Root**[$5 + 7 \#1 - 12 \#1^2 + 3 \#1^6 + 2 \#1^7 \&, 5$] |
 | **x** == **Root**[$5 + 7 \#1 - 12 \#1^2 + 3 \#1^6 + 2 \#1^7 \&, 6$] ||
x == **Root**[$5 + 7 \#1 - 12 \#1^2 + 3 \#1^6 + 2 \#1^7 \&, 7$]

N[%]

x == -0.4173393384413579` |
 | **x** == -1.6147152813957173` - 0.7294064649822042`**i** |
 | **x** == -1.6147152813957173` + 0.7294064649822042`**i** |
 | **x** == 0.1147195547572578` - 1.3716602453531235`**i** |
 | **x** == 0.1147195547572578` + 1.3716602453531235`**i** |
 | **x** == 0.9586653958591383` - 0.2968262106007846`**i** |
 | **x** == 0.9586653958591383` + 0.2968262106007846`**i**

Roots[$x^{10} - 1 == 0, x$]

x == 1 || **x** == $(-1)^{1/5}$ || **x** == $(-1)^{2/5}$ |
 | **x** == $(-1)^{3/5}$ || **x** == $(-1)^{4/5}$ || **x** == -1 |
 | **x** == $(-1)^{1/5}$ || **x** == $(-1)^{2/5}$ |
 | **x** == $(-1)^{3/5}$ || **x** == $(-1)^{4/5}$

N[%]

x == 1. | | **x** == 0.8090169943749475` + 0.5877852522924731`**i** |
 | **x** == 0.3090169943749474` + 0.9510565162951535`**i** | |
x == -0.30901699437494734` + 0.9510565162951535`**i** | |
x == -0.8090169943749473` + 0.5877852522924731`**i** |
 | **x** == -1. | |
 | **x** == -0.8090169943749475` - 0.5877852522924731`**i** | |
x == -0.3090169943749474` - 0.9510565162951535`**i** | |
x == 0.30901699437494734` - 0.9510565162951535`**i** | |
x == 0.8090169943749473` - 0.5877852522924731`**i**

Рис. 8.4. Решение уравнений примера 8.3

Если коэффициенты полинома заданы в виде чисел, то функция `Roots[F, x]` выдает решение в виде точного или приближенного значения корней. Точное значение корней представляется числами в рациональной форме, приближенное — в форме вещественных чисел. Приведем пример.

Пример 8.3

Необходимо определить корни следующих уравнений:

$$x^2 + 13/28 x - 3/14 = 0,$$

$$2x^7 + 3x^6 - 12x^2 + 7x + 5 = 0,$$

$$x^{10} - 1 = 0.$$

Решения приведены на рис. 8.4.

Из рис. 8.4 видно, что функция определила точное значение корней первого уравнения, не смогла найти в явном виде корни второго уравнения и нашла корни третьего уравнения, но в виде, неудобном для пользователя: отсутствуют корни в комплексной форме. Для получения решения второго и третьего уравнений пришлось использовать команду `N[%]`.

Функция `Roots[f, x]` пытается решать также трансцендентные уравнения. При этом решения может не быть, а может быть ошибочным. Такие примеры показаны на рис. 8.5.

```
Roots[a^x + b == 0, x]
Roots[a^x + b == 0, x]
Roots[2^x - 4 x + 1 == 0, x]
x == 1/2
Roots[2^x - 4 x == 0, x]
x == 1/4
Roots[3^x - 9 x == 0, x]
x == 1/9
```

Рис. 8.5. Примеры ошибочных решений

Из рис. 8.5 видно, что первое уравнение не решено, а все остальные решения ошибочны.

8.2. Численные методы решения алгебраических и трансцендентных уравнений

Существует большое количество численных методов решения алгебраических и трансцендентных уравнений.

Алгоритм любого из этих методов является совокупностью условий выбора начального приближения, расчетных соотношений и признака окончания вычислительного процесса.

Алгоритмы некоторых наиболее популярных методов решения алгебраических и трансцендентных уравнений описаны в [16, 17, 18]. В настоящей главе излагаются лишь компьютерные технологии численных методов решения алгебраических и трансцендентных уравнений в системе Mathematica

Система Mathematica имеет много встроенных функций решения алгебраических и трансцендентных уравнений в численном виде. Основными из них являются: `NSolve`, `NRoots`, `FindRoot`. Рассмотрим подробно эти функции и приведем примеры.

8.2.1. Функция `NSolve`

Функция `NSolve` имеет вид:

`NSolve[f, x]`

где:

- `f` — решаемое уравнение;
- `x` — искомое неизвестное.

Откликом этой функции являются корни уравнения $f(x)=0$. Корнями могут быть вещественные и комплексные числа, представленные в естественной форме.

Функция `NSolve[f, x]` может решать все уравнения, которые решает функция `solve`. Ее отличие лишь в форме представления ответов.

Покажем действия этой функции на примерах, сравнивая ее ответы с ответами функции `Solve[f, x]`.

Пример 8.4

Необходимо определить корни следующих уравнений:

$$x^3 - 9/4 x^2 - 3/4 x + 5/16 = 0,$$

$$2^x - 4x + 1 = 0,$$

Solve[$x^3 - 2.25x^2 - 0.75x + 0.3125 == 0, x]$

{ $\{x \rightarrow -0.5\}, \{x \rightarrow 0.25\}, \{x \rightarrow 2.5\}$ }

Solve[$x^3 - 9/4x^2 - 3/4x + 5/16 == 0, x]$

{ $\{x \rightarrow -\frac{1}{2}\}, \{x \rightarrow \frac{1}{4}\}, \{x \rightarrow \frac{5}{2}\}$ }

NSolve[$x^3 - 9/4x^2 - 3/4x + 5/16 == 0, x]$

{ $\{x \rightarrow -0.5\}, \{x \rightarrow 0.25\}, \{x \rightarrow 2.5\}$ }

Solve[$2^x - 4x + 1 == 0, x]$

{ $\{x \rightarrow \frac{\text{Log}[2] - 4 \text{ProductLog}[-\frac{1 + \sqrt{2}}{2 \cdot 2^{3/4}}]}{4 \text{Log}[2]}\}, \{x \rightarrow \frac{\text{Log}[2] - 4 \text{ProductLog}[-1, -\frac{1 + \sqrt{2}}{2 \cdot 2^{3/4}}]}{4 \text{Log}[2]}\}$ }

NSolve[$2^x - 4x + 1 == 0, x]$

{ $\{x \rightarrow 0.639428\}, \{x \rightarrow 3.84666\}$ }

Solve[$E^{(-2x)} + 3/x - 1, x]$

Solve[$-1 + e^{-2x} + \frac{3}{x}, x]$

NSolve[$E^{(-2x)} + 3/x - 1 == 0, x]$

NSolve[$-1 + e^{-2x} + \frac{3}{x} == 0, x]$

Solve[$ax^3 - 1 == 0, x]$

{ $\{x \rightarrow \frac{1}{a^{1/3}}\}, \{x \rightarrow -\frac{(-1)^{1/3}}{a^{1/3}}\}, \{x \rightarrow \frac{(-1)^{2/3}}{a^{1/3}}\}$ }

NSolve[$ax^3 - 1 == 0, x]$

{ $\{x \rightarrow -\frac{0.5 + 0.8660254037844387 \sqrt{3} i}{a^{1/3}}\}, \{x \rightarrow -\frac{0.5 - 0.8660254037844387 \sqrt{3} i}{a^{1/3}}\}, \{x \rightarrow \frac{1}{a^{1/3}}\}$ }

Solve[$2x^5 - 3.2x^3 + 7.3x - 14 == 0, x]$

{ $\{x \rightarrow -1.43088700021456 - 0.9042760720039539 \sqrt{3} i\},$
 $\{x \rightarrow -1.43088700021456 + 0.9042760720039539 \sqrt{3} i\},$
 $\{x \rightarrow 0.6995754953748784 - 1.0881835477460546 \sqrt{3} i\},$
 $\{x \rightarrow 0.6995754953748784 + 1.0881835477460546 \sqrt{3} i\},$
 $\{x \rightarrow 1.4610264092931553\}$ }

NSolve[$2x^5 - 3.2x^3 + 7.3x - 14 == 0, x]$

{ $\{x \rightarrow -1.43088700021456 - 0.9042760720039539 \sqrt{3} i\},$
 $\{x \rightarrow -1.43088700021456 + 0.9042760720039539 \sqrt{3} i\},$
 $\{x \rightarrow 0.6995754953748784 - 1.0881835477460546 \sqrt{3} i\},$
 $\{x \rightarrow 0.6995754953748784 + 1.0881835477460546 \sqrt{3} i\},$
 $\{x \rightarrow 1.4610264092931553\}$ }

Рис. 8.6. Решение примера 8.4

$$e^{-2x} + 3/x - 1 = 0,$$

$$ax^3 - 1 = 0,$$

$$2x^5 - 3.2x^3 + 7.3x - 14 = 0.$$

Решение примера приведено на рис. 8.6.

Из рис. 8.6 видно, что первое уравнение решено с помощью функции `Solve[f, x]` с представлением корней в нормальной форме. Функция `NSolve[f, x]` их выдала в естественной форме. Отметим, что если уравнение содержит коэффициенты в естественной форме, то и корни будут определены в виде чисел, представленных в естественной форме, и не будут отличаться от решения, полученного с помощью функции `NSolve[f, x]`.

8.2.2. Функция *NRroots*

Функция `NRroots` имеет вид:

`NRroots[f, x]`

где:

□ `f` — решаемое уравнение;

□ `x` — искомое неизвестное.

Откликом этой функции являются корни полинома $f(x) = 0$, представленные в естественной форме. Приведем примеры.

Пример 8.5

Найти корни следующих уравнений:

$$x^2 + 13/28x - 3/14 = 0,$$

$$2x^7 + 3x^6 - 12x^2 + 5 = 0,$$

$$x^{10} - 1 = 0,$$

$$2^x - 4x + 1 = 0.$$

Решения приведены на рис. 8.7.

Из рис. 8.7 видно, что вещественные и комплексные корни найдены в численном виде с представлением чисел в естественной форме. Попытка решить трансцендентное уравнение $2^x - 4x + 1 = 0$ к успеху не привела — решение не получено.


```

NRoots[x^2 + 13 / 28 x - 3 / 14 == 0, x]

x == -0.75 || x == 0.285714

NRoots[2 x^7 + 3 x^6 - 12 x^2 + 7 x + 5 == 0, x]

x == -1.6147152813957173` - 0.7294064649822042` ħ |
| x == -1.6147152813957173` + 0.7294064649822042` ħ |
| x == -0.4173393384413579` |
| x == 0.1147195547572578` - 1.3716602453531235` ħ |
| x == 0.1147195547572578` + 1.3716602453531235` ħ |
| x == 0.9586653958591383` - 0.2968262106007846` ħ |
| x == 0.9586653958591383` + 0.2968262106007846` ħ

NRoots[x^10 - 1 == 0, x]

x == -0.999999999999999` |
| x == -0.8090169943749475` - 0.5877852522924731` ħ |
| x == -0.8090169943749475` + 0.5877852522924731` ħ |
| x == -0.3090169943749474` - 0.9510565162951535` ħ |
| x == -0.3090169943749474` + 0.9510565162951535` ħ |
| x == 0.3090169943749473` - 0.9510565162951535` ħ |
| x == 0.3090169943749473` + 0.9510565162951535` ħ |
| x == 0.8090169943749476` - 0.5877852522924732` ħ |
| x == 0.8090169943749476` + 0.5877852522924732` ħ || x == 1.`

NRoots[2^x - 4 x + 1 == 0, x]

NRoots[1 + 2^x - 4 x == 0, x]

```

Рис. 8.7. Решение примера 8.5

8.2.3. Функция *FindRoot*

Функция `FindRoot` имеет вид:

`FindRoot[f, {x, x0}]`

где:

- `f` — уравнение;
- `x` — искомое неизвестное (корень (корни) уравнения);
- `x0` — начальное приближение.

Функция `FindRoot[f, {x, x0}]` находит корень уравнения $f(x) = 0$ из области значений x , близких к x_0 .

Рекомендуется следующая технология определения корней алгебраических и трансцендентных уравнений с помощью функции `FindRoot[f, {x, x0}]`:

1. Определение области изоляции искомого корня и выбор значения приближения x_0 .
2. Ввод уравнения $f(x) = 0$ с присвоением ему уникального имени.
3. Ввод функции `FindRoot[f, {x, x0}]` с выбранным значением x_0 .
4. Получение решения нажатием комбинации клавиш <Shift>+<Enter>.

Приведем примеры.

Пример 8.6

Определить корни уравнения $3^x - 9x + 1 = 0$, выбрав значения приближений из графика функции $f(x) = 3^x - 9x + 1$.

Решение приведено на рис. 8.8 и 8.9.

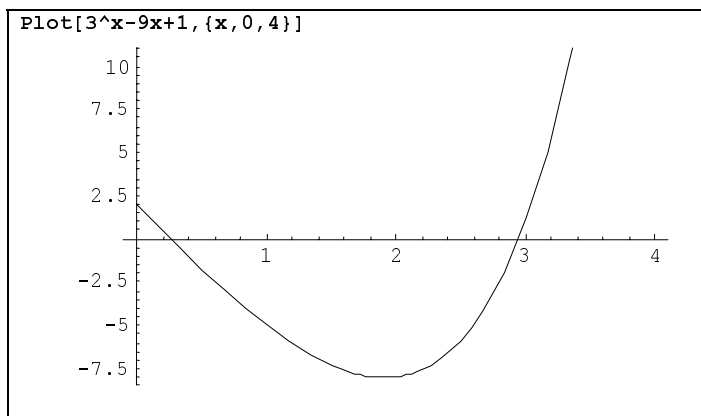


Рис. 8.8. График функции $f(x) = 3^x - 9x + 1$

Из рис. 8.8 видно, что уравнение имеет два корня, областями изоляции которых могут быть: $x_0 \in [0, 1]$ и $x_0 \in [2.5, 3]$.

Решение уравнения приведено на рис. 8.9.

При выборе начального приближения нужно быть внимательным, особенно в тех случаях, когда уравнение содержит несколько корней. Может оказаться, что при установленном пользователем приближении x_0 будет определен не тот корень, который ожидался. В нашем примере незначительное изменение x_0 привело к иному решению: при $x_0 = 1.9$ функция `FindRoot` нашла корень $x_0 = 0.258755$, а уже при $x_0 = 2$ — $x = 2.94964$.

Заметим, что область изоляции корня может быть очень широкой. В нашем примере решение получено при $x_0 = -50$. Однако при $x_0 = 20$ система выдала несуществующее решение: $x = 6.37646$. Чтобы избежать подобных ошибок, следует выбирать x_0 из узкой области изоляции корня.

```
F := 3^x - 9x + 1 == 0
FindRoot[F, {x, 0}]
{x → 0.258755}
FindRoot[F, {x, 3}]
{x → 2.94964}
FindRoot[F, {x, 1.9}]
{x → 0.258755}
FindRoot[F, {x, 2}]
{x → 2.94964}
FindRoot[F, {x, -50}]
{x → 0.258755}
FindRoot[F, {x, 20}]
{x → 6.37646}
```

Рис. 8.9. Решение уравнения $3^x - 9x + 1 = 0$ при различных значениях x_0

8.3. Интервальные методы решения алгебраических и трансцендентных уравнений

Интервальными называются такие методы, которые требуют для их реализации задания области изоляции корня.

Для этих целей в подпакете `IntervalRoots` пакета расширения `NumericaiMath` имеются следующие функции:

```
IntervalBisection[f, x, int, eps]
IntervalSecand[f, x, int, eps]
IntervalNewton[f, x, int, eps]
```

В функциях приняты следующие обозначения:

- f — решаемое уравнение;
- x — искомое неизвестное;

- `int` — интервал изоляции корня;
- `eps` — погрешность вычисления корня.

Рассмотрим подробно эти функции и приведем примеры решения уравнений.

- `IntervalBisection[f, x, int, eps]` — находит корень уравнения $f(x)=0$ методом дихотомии (половинного деления) в интервале изоляции $[x_1, x_2]$ с погрешностью, не превышающей `eps`.
- `IntervalSecand[f, x, int, eps]` — находит методом хорд (секущей) корень уравнения $f(x)=0$ в интервале изоляции $[x_1, x_2]$ с погрешностью, не превышающей `eps`.
- `IntervalNewton[f, x, int, eps]` — находит методом касательной (Ньютона) корень уравнения $f(x)=0$ в интервале изоляции $[x_1, x_2]$ с погрешностью, не превышающей `eps`.

Технология определения корней с помощью перечисленных функций следующая:

1. Определение числа вещественных корней и областей их изоляции.
2. Обращение к пакету расширения
`<NumericMath`IntervalRoots``
3. Выбор вида функции `Interval`.
4. Ввод функции `Interval`.
5. Получение решения путем нажатия комбинации клавиш `<Shift>+<Enter>`.

Пример 8.7

Необходимо определить корни уравнения $2^x - 4x + 1 = 0$ с помощью всех функций интервальных методов.

Определим число вещественных корней и области их изоляции методом визуализации вычислений.

На рис. 8.10 приведена функция $y = 2^x - 4x + 1$.

Из графика видно, что функция дважды пересекает ось x , т. е. уравнение имеет два вещественных корня. Области их изоляции могут быть: $[0, 1]$ и $[3, 4]$.

Решим теперь уравнение в соответствии с описанной ранее технологией.

Решение приведено на рис. 8.11.

Проанализируем полученные результаты.

В первых строках приведено решение уравнения с помощью функции `solve`.

Решение получено в аналитическом виде и является точным. Погрешность могла возникнуть только при преобразовании его в численные значения с помощью команды `N (%)`. С этим решением будем сравнивать все остальные.

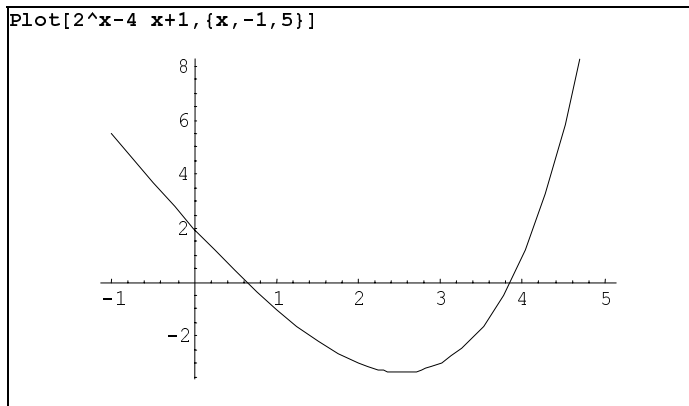


Рис. 8.10. График функции $y = 2^x - 4x + 1$

```

F := 2^x - 4x + 1
Solve[F == 0, x]
{{x ->  $\frac{\text{Log}[2] - 4 \text{ProductLog}[-\frac{1 \circ \text{Log}[2]}{2 \cdot 2^{3/4}}]}{4 \text{Log}[2]}$ }, {x ->  $\frac{\text{Log}[2] - 4 \text{ProductLog}[-1, -\frac{1 \circ \text{Log}[2]}{2 \cdot 2^{3/4}}]}{4 \text{Log}[2]}$ }}
N[%]
{{x -> 0.639428}, {x -> 3.84666}}
<< NumericalMath`IntervalRoots`
IntervalBisection[F, x, Interval[{0., 5.}], .001]
Interval[{0.625, 0.664063}, {3.78906, 3.90625}]
IntervalSecant[F, x, Interval[{0., 5.}], .001]
Interval[{0.639415, 0.639433}, {3.84665, 3.84674}]
IntervalNewton[F, x, Interval[{0., 5.}], .001]
Interval[{0.639302, 0.639504}, {3.84666, 3.84666}]

```

Рис. 8.11. Решение уравнения $2^x - 4x + 1 = 0$ с помощью интервальных методов

Обратим внимание на то, что при решении уравнения интервальными методами ответом является интервал значений x , внутри которого есть точное значение корня. Чем уже этот интервал, тем точнее решение. Если интервал равен нулю, то ответ получен с наперед заданной точностью.

Анализируя ответы, видим, что корень из области изоляции $[0, 1]$, точное значение которого 0.639428, получен с наиболее высокой точностью методом хорд, а

корень из области [3, 4] — методом касательных. Метод касательных выдал решение, совпадающее с точным. Метод дихотомии оказался наименее точным.

Следует иметь в виду, что эти выводы имеют частный характер. При решении другой задачи они могут быть совсем иными, например, метод дихотомии окажется наилучшим.

8.4. Определение корней уравнений с применением интерполяции

Сущность метода состоит в следующем. Функция $f(x)$, соответствующая уравнению $f(x) = 0$, интерполируется в области изоляции корня. Образуется новая функция $f_1(x)$, корень которой и является решением. Повышение точности результата и сокращение времени решения достигается за счет уменьшения объема вычислений.

Метод реализуется встроенной функцией `InterpolateRoot`, имеющей вид:

`InterpolateRoot[f, {x, a, b}]`

где:

□ f — функция исходного уравнения $f(x) = 0$;

□ x — искомое неизвестное;

□ a, b — область изоляции корня.

Технология решения уравнения:

1. Обращение к пакету расширения
`<NumericalMath`InterpolateRoot``
2. Определение области изоляции корня a, b .
3. Ввод функции `InterpolateRoot[f, {x, a, b}]`.
4. Получение решения путем нажатия комбинации клавиш `<Shift>+<Enter>`.

Приведем примеры.

Пример 8.8

Необходимо решить следующие уравнения:

$$x + \ln x - 0.5 = 0,$$

$$2^x - 4x + 1 = 0,$$

$$e^{2x} + 3x^3 - 125 = 0$$

Решение приведено на рис. 8.12.

```

F := x + Log[x] - 0.5
FindRoot[F, {x, 0.5}]

{x → 0.766249}

<< NumericalMath`InterpolateRoot`
InterpolateRoot[F, {x, 0.5, 1}]

{x → 0.766248608161750245789587}

F1 := 2^x - 4x + 1
FindRoot[F1, {x, 3}]

{x → 3.84666}

InterpolateRoot[F1, {x, 3, 4}]

{x → 3.846656301183044001833}

F3 := E^(2x) + 3x^3 - 125
FindRoot[F3, {x, 2, 4}]

{x → 2.25358}

InterpolateRoot[F3, {x, 2, 4}]

{x → 2.25358345632778330991826294}

```

Рис. 8.12. Решение уравнения примера 8.8

Функция `InterpolateRoot` имеет следующие опции:

```

WorkingPrecision->n
AccuracyGoal-> n
ShowProgress->True (False)

```

В опциях `n` — число, определяющее точность задания и вычисления корня.

Пример 8.9

Необходимо определить корень уравнения $3^x - 9x + 2 = 0$ с помощью функции `InterpolateRoot[f, {x, a, b}]` и ее опций, если область изоляции корня $[2, 4]$.

Сравнить ответы с решением, полученным с помощью функции `FindRoot`.

Решение задачи приведено на рис. 8.13.

Из сравнения результатов расчетов видно, что опции позволяют повысить точность определения корня. Однако при практических расчетах все полученные решения можно считать равноценными. Определить корень с точностью 5 знаков после запятой, как это получено с помощью функции `FindRoot`, вполне достаточно для инженерных расчетов.

```

F := 3^x - 9x + 2
FindRoot[F, {x, 2, 3}]

{x → 2.89481}

<< NumericalMath`InterpolateRoot`
InterpolateRoot[F, {x, 2, 3}]

{x → 2.89480716495696698997583970}

InterpolateRoot[F, {x, 2, 3}, WorkingPrecision → 100]

{x → 2.894807164956966989975839695261704056817754
559010590966348482729317807579171851788964393
71615562486736592910368}

InterpolateRoot[F, {x, 2, 3}, AccuracyGoal → 100]

{x → 2.89480716495696698997583969526170405681775
455901059096634848272931780757917185178896439
371615562486736592910368127628258920084572639252}

InterpolateRoot[F, {x, 2, 3}, ShowProgress → False]

{x → 2.89480716495696698997583970}

InterpolateRoot[F, {x, 2, 3}, ShowProgress → True]

{0, 2.7777777777777778}

{21, 0, -0.2054124447437385647}

{1, 2.92282971783602406378648624443516912680595}

{21, 21, 0.0754656401588680053386508133521}

{6, 2.894807164725532644987185831168619188141458208540}

{28, 21, -2.31136140988381275619094875794 × 10-1}

{14, 2.894807164956966989565635956349036261269223288772307440352313}

{40, 21, -4.10195181001028677235300737598 × 10-1}

{27, 2.894807164956966989975839695261704062977011589194922968760752}

{x → 2.89480716495696698997583970}

```

Рис. 8.13. Решение уравнения с помощью опций функции InterpolateRoot

8.5. Проверка достоверности решения уравнений

Эффективным способом проверки достоверности решения алгебраических и трансцендентных уравнений является подстановка вычисленных значений корней в исходное уравнение.

Компьютерная технология решения уравнения с проверкой достоверности ответа состоит в выполнении следующих операций:

1. Ввод решаемого уравнения с присвоением ему уникального имени, например f , путем использования символа "=" (равно).
2. Вывод на экран уравнения путем одновременного нажатия комбинации клавиш <Shift>+<Enter>.
3. Ввод соответствующей встроенной функции решения уравнения либо с присвоением ей уникального имени, либо без присвоения.
4. Решение уравнения путем нажатия комбинации клавиш <Shift>+<Enter>.
5. Ввод знака проверки решения f/z , где z — имя встроенной функции или строка вывода решения вида `Out[]`.
6. Проверка решения путем нажатия комбинации клавиш <Shift>+<Enter>.

Результат проверки может иметь различный вид, зависящий от вида самого решения. Все возможные случаи увидим на примерах.

Пример 8.10

Необходимо решить следующие уравнения:

$$ab^x - c = 0,$$

$$2x^4 - 3x^3 + 2x^2 - 1 = 0,$$

$$x^7 - 1 = 0,$$

$$3^x - 9x + 1 = 0,$$

$$2^x - 4x + 1 = 0,$$

проверив достоверность ответов.

Решение уравнений приведено на рис. 8.14.

Из рис. 8.14 видно, что решение уравнений с помощью функций `solve` и `NSolve` выполнено верно: при подстановке корней в исходное уравнение ответом является ноль для случая каждого корня. Правда, при решении уравнения $2x^4 - 3x^3 + 2x^2 - 1 = 0$ пришлось обратиться к функции `simplify` для упрощения результатов проверки.

Проверка решения уравнения $3^x - 9x + 1 = 0$, полученного с помощью функции `FindRoot`, не дала положительного ответа: он не равен нулю. Правда, мы можем догадываться, что решение верно, т. к. разница слишком мала и составляет примерно 1.5×10^{-10} .

Анализ решения алгебраических и трансцендентных уравнений и проверки результатов вычисления корней показывает, что проверка дает всегда правильный результат, если решение точно. Это имеет место при аналитических методах решения уравнений. Если решение уравнения ищется в численном виде и является приближенным, то результат проверки может быть не тривиальным или его не будет вовсе.

```

f1 = a b ^ x - c
ab^x - c

z1 = Solve[f1 == 0, x]
{{x ->  $\frac{\text{Log}\left[\frac{c}{a}\right]}{\text{Log}[b]}$ }}

f1 /. z1
{0}

f2 = 2 x ^ 4 - 3 x ^ 3 + 2 x ^ 2 - 1
-1 + 2 x^2 - 3 x^3 + 2 x^4

z2 = Solve[f2 == 0, x]
{{x ->  $-\frac{1}{2}$ }, {x -> 1}, {x ->  $(-1)^{1/3}$ }, {x ->  $-(-1)^{2/3}$ }}

f2 /. z2
{0, 0, 2 - 2 (-1)^{1/3} + 2 (-1)^{2/3}, 2 - 2 (-1)^{1/3} + 2 (-1)^{2/3}}

Simplify[%]
{0, 0, 0, 0}

f3 = x ^ 7 - 1
-1 + x^7

z3 = Solve[f3 == 0, x]
{{x -> 1}, {x ->  $-(-1)^{1/7}$ }, {x ->  $(-1)^{2/7}$ }, {x ->  $-(-1)^{3/7}$ },
 {x ->  $(-1)^{4/7}$ }, {x ->  $-(-1)^{5/7}$ }, {x ->  $(-1)^{6/7}$ }}

f3 /. z3
{0, 0, 0, 0, 0, 0, 0}

f4 = 3 ^ x - 9 x + 1
1 + 3^x - 9 x

z4 = FindRoot[f4, {x, 3}]
{x -> 2.94964}

f4 /. z4
1.49321 x 10^-10

f5 = 2 ^ x - 4 x + 1
1 + 2^x - 4 x

z5 = NSolve[f5 == 0]
{{x -> 0.639428}, {x -> 3.84666}}

f5 /. z5
{0., 0.}

```

Рис. 8.14. Решение уравнений примера 8.11

Покажем это на примерах.

Пример 8.11

Необходимо решить уравнение $x^5 - ax^3 + 2x^2 - 2a = 0$ и проверить его результаты. Решение получено с помощью функции `Roots` и приведено на рис. 8.15.

Из рис. 8.15 видно, что проверка не дала желаемого результата, хотя решение верно, в чем мы убедились, подставляя каждое значение корня в исходное уравнение и получая всегда ноль.

```

s1 = x^5 - a x^3 + 2 x^2 - 2 a
-2 a + 2 x^2 - a x^3 + x^5
r1 = Roots[s1 == 0, x]
x == (-1)^(2/3) 2^(1/3) || x == (-2)^(1/3) || x == -2^(1/3) || x == sqrt[a] || x == -sqrt[a]
s1 /. r1
-2 a + 2 x^2 - a x^3 + x^5 /. x == (-1)^(2/3) 2^(1/3) || x == (-2)^(1/3) || x == -2^(1/3) || x == sqrt[a] || x == -sqrt[a]
x = Sqrt[a]
s1
sqrt[a]
0
x = -Sqrt[a]
s1
-sqrt[a]
0
x = -2^(1/3)
s1
-2^(1/3)
0
x = (-2)^(1/3)
s1
(-2)^(1/3)
0
x = -(-1)^(2/3) 2^(1/3) 2^(1/3)
s1
-(-1)^(2/3) 2^(1/3)
0

```

Рис. 8.15. Решение уравнения примера 8.11

Имеют место случаи, когда результаты проверки зависят от вида встроенной функции решения уравнения. Покажем это на следующем примере.

Пример 8.12

Необходимо определить корни следующего уравнения:

$$x^5 + 4x^4 + 5x^3 - 7x^2 + x + 1 = 0.$$

Решение приведено на рис. 8.16.

Из рис. 8.16 видно, что проверка решения, полученного с помощью функции `NSolve`, дала ответы, позволяющие считать решение верным. Функция `NRroots` выдала те же ответы, однако осуществить проверку стандартным методом не удалось.

```
f = x^5 + 4 x^4 + 5 x^3 - 7 x^2 + x + 1
1 + x - 7 x^2 + 5 x^3 + 4 x^4 + x^5
z = NSolve[f == 0, x]
{{x -> -2.412273378741147` - 1.7628166835194017` I},
 {x -> -2.412273378741147` + 1.7628166835194017` I},
 {x -> -0.29426055789544936`},
 {x -> 0.5594036576888717` - 0.2603215017839416` I},
 {x -> 0.5594036576888717` + 0.2603215017839416` I}}
f /. z
{-5.684341886080802`*^-14 + 2.1316282072803006`*^-14 I,
 -5.684341886080802`*^-14 - 2.1316282072803006`*^-14 I,
 -6.331740687315346`*^-17,
 -6.938893903907228`*^-17 + 3.608224830031759`*^-16 I,
 -6.938893903907228`*^-17 - 3.608224830031759`*^-16 I}
z1 = NRroots[f == 0, x]
x == -2.412273378741147` - 1.7628166835194017` I |
 | x == -2.412273378741147` + 1.7628166835194017` I |
 | x == -0.29426055789544936` |
 | x == 0.5594036576888717` - 0.2603215017839416` I |
 | x == 0.5594036576888717` + 0.2603215017839416` I |
f /. z1
1 + x - 7 x^2 + 5 x^3 + 4 x^4 + x^5 /. x == -2.412273378741147` - 1.7628166835194017` I |
 | x == -2.412273378741147` + 1.7628166835194017` I |
 | x == -0.29426055789544936` |
 | x == 0.5594036576888717` - 0.2603215017839416` I |
 | x == 0.5594036576888717` + 0.2603215017839416` I
```

Рис. 8.16. Решение уравнения примера 8.12

Количество решений может быть равно, может быть меньше или больше числа неизвестных. В зависимости от этого система уравнений классифицируется

следующим образом. Система называется *совместной*, если она имеет хотя бы одно решение, в противном случае она называется *несовместной*. Совместная система может иметь единственное решение или бесконечное их число. Если система имеет бесконечное число решений, ее называют неопределенной. Рассмотрим примеры таких систем.

Система уравнений

$$x_1 + 2x_2 - x_3 = 1$$

$$2x_1 + x_2 + x_3 = 2$$

$$3x_1 + 2x_2 + 3x_3 = 5$$

имеет единственное решение: $x_1 = 0$, $x_2 = 1$, $x_3 = 1$, т. е. она является совместной и определенной.

Система

$$x_1 + 2x_2 - x_3 = 1$$

$$2x_1 + 4x_2 - 2x_3 = 2$$

$$3x_1 + 2x_2 + 3x_3 = 5$$

имеет бесконечное число решений, которые удовлетворяют следующим равенствам: $x_1 + 2x_3 = 2$, $2x_2 - 3x_3 = -1$, т. е. система является совместной и неопределенной. Определены лишь условия решения. В этой системе первое и второе уравнения идентичны.

Система

$$x_1 + 2x_2 - x_3 = 1$$

$$2x_1 + 4x_2 - 2x_3 = 2$$

$$3x_1 + 6x_2 - 3x_3 = 3$$

не имеет ни одного решения и является несовместной. Действительно, в этой системе будет три одинаковых уравнения после деления второго уравнения на 2, а третьего на 3.

Существует большое число различных методов решения систем линейных уравнений. Все они могут быть разделены на две группы: точные методы и методы последовательных приближений. Следует при этом иметь в виду, что точными методами являются только аналитические методы. Если с помощью этих методов решать систему уравнений с числовыми коэффициентами, то точных решений можно не получить за счет ошибок вычислений, связанных с конечной памятью компьютера.

Аналитические методы

Наиболее популярным из точных методов решения линейных алгебраических уравнений является метод Гаусса. Метод Гаусса изучается в математике, и нет надобности здесь его излагать. Напомним только теорему Крамера.

Теорема Крамера

Если определитель $|\mathbf{A}|$ матрицы коэффициентов системы n линейных уравнений с n неизвестными отличен от нуля, то система имеет решение, и притом единственное.

Метод Гаусса может привести к существенным ошибкам при определении неизвестных $x_1, x_2, \dots, x_n$ в случае плохо обусловленных систем. *Плохо обусловленной* называется такая система, у которой модуль определителя матрицы коэффициентов мал по сравнению с какой-либо из норм матрицы. Нормой матрицы может быть максимальная из сумм модулей коэффициентов строк или столбцов. Плохо обусловленные системы чувствительны к ошибкам округления, которые неизбежны при компьютерных методах реализации алгоритма Гаусса.

Методы итераций

Рассмотрим решение линейных алгебраических уравнений методом простой итерации.

Разрешим исходную систему уравнений (9.1) относительно неизвестных:

$$\begin{aligned} x_1 &= \alpha_{1,2}x_2 + \alpha_{1,3}x_3 + \dots + \alpha_{1,n}x_n + \alpha_{1,n+1} \\ x_2 &= \alpha_{2,1}x_1 + \alpha_{2,3}x_3 + \dots + \alpha_{2,n}x_n + \alpha_{2,n+1} \\ &\dots\dots\dots \\ x_n &= \alpha_{n,2}x_1 + \alpha_{n,2}x_2 + \dots + \alpha_{n,n-1}x_{n-1} + \alpha_{1,n+1} \end{aligned} \quad (9.2)$$

В системе приняты обозначения: $\alpha_{i,k} = \frac{\alpha_{i,k}}{\alpha_{i,i}}, i = 1, 2, \dots, n, k = 1, 2, \dots, n, n+1$.

Запишем систему уравнений (9.2) в следующем виде:

$$\begin{aligned} x_1 &= \varphi_1(x_1, x_2, \dots, x_n) \\ x_2 &= \varphi_2(x_1, x_2, \dots, x_n) \\ &\dots\dots\dots \\ x_n &= \varphi_n(x_1, x_2, \dots, x_n) \end{aligned} \quad (9.3)$$

Пусть $x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}$ являются начальными приближениями. Тогда, подставляя их в систему уравнений (9.3), получим:

$$\begin{aligned} x_1^{(1)} &= \varphi_1(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}) \\ x_2^{(1)} &= \varphi_2(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}) \\ &\dots\dots\dots \\ x_n^{(1)} &= \varphi_n(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}) \end{aligned} \quad (9.4)$$

Принимая $x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}$ в качестве первых приближений и подставляя их в исходное уравнение, получим вторые приближения. Повторяя вычисления, можно получить значения неизвестных на любой v -й итерации.

При компьютерной реализации итерационного метода возникают следующие вопросы:

- ☐ Как выбрать начальные приближения $x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}$?
- ☐ Каковы условия сходимости итерационного процесса?
- ☐ На какой итерации закончить вычисления?

Ответы на эти вопросы совместно с расчетными соотношениями и будут алгоритмом решения систем линейных уравнений методом итерации. Ответим на поставленные вопросы для случая линейных систем алгебраических уравнений.

Выбор начальных приближений

Если области, в которых находятся неизвестные $x_i, i = 1, 2, \dots, n$, известны, то начальные значения выбираются произвольно из этой области. Если таковые неизвестны, то за начальные приближения можно взять свободные члены

$$\frac{b_1}{a_{11}}, \frac{b_2}{a_{22}}, \dots, \frac{b_n}{a_{nn}}.$$

Условия сходимости итерационного процесса

Условием сходимости итерационного процесса является: сумма абсолютных значений отношений коэффициентов в каждом уравнении системы к диагональному должна быть меньше единицы.

Обеспечить сходимость итерационного процесса можно путем преобразования исходной системы к эквивалентной. Эти преобразования можно выполнить путем перестановки уравнений, операций сложения и вычисления уравнений, умножения на постоянный коэффициент.

Рассмотрим пример обеспечения сходимости итераций. Необходимо решить методом итераций следующую систему уравнений:

$$\begin{aligned} 2x_1 + 3x_2 + x_3 &= 1 \\ -7x_1 - 2x_2 + 4x_3 &= 6 \\ 8x_1 + x_2 - 3x_3 &= 12 \end{aligned}$$

Эта система имеет решение: $x_1 = 5, x_2 = -\frac{11}{2}, x_3 = \frac{15}{2}$.

Однако решать эту систему уравнений методом итераций опасно, т. к. здесь не обеспечены условия сходимости итераций. Действительно, в первом уравнении

$$\left| \frac{3}{2} \right| + \left| \frac{1}{2} \right| > 1, \text{ во втором } \left| \frac{7}{2} \right| + \left| \frac{4}{2} \right| > 1, \text{ в третьем } \left| \frac{8}{3} \right| + \left| \frac{1}{3} \right| > 1.$$

Для обеспечения условий сходимости преобразуем исходную систему уравнений. Второе уравнение поставим в первую строку. Тогда $\left|\frac{2}{7}\right| + \left|\frac{4}{7}\right| < 1$. Заметим, что первое уравнение можно заменить также третьим.

Умножим первое уравнение на 4 и сложим его со вторым. Тогда получим $x_1 + 10x_2 + 8x_3 = 4$. Теперь $\left|\frac{1}{10}\right| + \left|\frac{8}{10}\right| < 1$.

Это уравнение можно сделать вторым.

Для получения третьего уравнения выполним следующие преобразования: сложим все уравнения, полученное уравнение умножим на 2 и сложим со вторым.

Тогда получим: $-x_1 + 2x_2 + 8x_3 = 44$. Теперь $\left|\frac{1}{8}\right| + \left|\frac{2}{8}\right| < 1$, условие сходимости

итераций выполнено. В результате всех этих операций получена следующая эквивалентная система уравнений:

$$\begin{aligned} -7x_1 - 2x_2 + 4x_3 &= 6 \\ x_1 + 10x_2 + 8x_3 &= 10 \\ -x_1 + 2x_2 + 8x_3 &= 44 \end{aligned}$$

Теперь условия сходимости итераций выполнены полностью. Решение системы уравнений итерационным методом возможно при начальных условиях

$$x_1^{(0)} = -\frac{6}{7}, \quad x_2^{(0)} = 1, \quad x_3^{(0)} = \frac{44}{8} = 5.5.$$

Признак окончания вычислений

Признаком окончания итерационного процесса из условий точности можно в первом приближении считать условие: $|x_k^{(v+1)} - x_k^{(v)}| \leq \varepsilon$, где $x_k^{(v+1)}$, $x_k^{(v)}$ — значение k -го неизвестного соответственно на $(v+1)$ и (v) итерациях, ε — допустимая погрешность вычисления неизвестных.

Существуют два способа решения уравнений методом итераций: метод простой итерации и метод Зейделя. Пользователь не имеет возможности выбирать метод итераций, т. к. функции и команды, реализующие метод итераций в системе Mathematica, не различают этих методов.

Алгоритмы метода итерации

Алгоритм метода простой итерации состоит из совокупности условий выбора начальных приближений, расчетных соотношений и признака окончания вычислений.

Условие выбора начальных приближений:

$$x_1^0 = \frac{b_1}{a_{11}}, \quad x_2^0 = \frac{b_2}{a_{22}}, \quad \dots, \quad x_n^0 = \frac{b_n}{a_{nn}}.$$

Расчетные соотношения:

$$\begin{aligned} x_1^{v+1} &= \varphi_1(x_1^v, x_2^v, \dots, x_n^v) \\ x_2^{v+1} &= \varphi_2(x_1^v, x_2^v, \dots, x_n^v) \\ &\dots\dots\dots \\ x_n^{v+1} &= \varphi_n(x_1^v, x_2^v, \dots, x_n^v) \end{aligned}$$

Признак окончания вычислений:

$$\left| x_k^{v+1} - x_k^v \right| \leq \varepsilon.$$

В алгоритме простой итерации все значения неизвестных на шаге $v+1$ вычисляются по их значениям на предыдущем шаге v . В алгоритме Зейделя результаты вычисления x_1 на шаге $v+1$ используются для вычисления $x_2, x_3, \dots, x_n$ на этом же шаге; результаты вычисления x_2 на шаге $v+1$ используются для вычисления неизвестных $x_3, x_4, \dots, x_n$ на этом же шаге и т. д. Расчетные соотношения имеют вид:

$$\begin{aligned} x_1^{v+1} &= \varphi_1(x_1^v, x_2^v, \dots, x_n^v) \\ x_2^{v+1} &= \varphi_2(x_1^{v+1}, x_2^v, \dots, x_n^v) \\ &\dots\dots\dots \\ x_n^{v+1} &= \varphi_n(x_1^{v+1}, x_2^{v+1}, \dots, x_{n-1}^{v+1}, x_n^v) \end{aligned}$$

Очевидно, что алгоритм Зейделя позволяет получить решение с большей точностью, чем алгоритм простой итерации при том же числе итераций.

Сравнительная оценка точных и итерационных методов

В методе простой итерации на одну итерацию необходимо выполнить приблизительно $2n^2$ арифметических операций типа сложения и умножения, в то время как по методу Гаусса для решения системы n уравнений необходимо выполнить $\frac{2}{3}n^3$ операций. Тогда очевидно, что метод итераций более целесообразный

для случая, когда возможно получить решение задачи не более чем за $\frac{1}{3}n$ итераций, т. е. он выгоден при решении уравнений больших размерностей.

Логическая схема итерационных методов очень проста, поэтому компьютерные программы короче, чем в методе Гаусса.

Итерационные методы позволяют распараллеливать алгоритм, что дает возможность эффективно решать уравнения на многопроцессорных компьютерах.

Недостатки итерационных методов в том, что они требуют от пользователя проверки условий сходимости итераций, и если они не выполняются, то преобразовывать исходные уравнения к виду, когда обеспечивается сходимость итерационного процесса. Кроме этого, итерационные методы требуют выбора начальных приближений. Все это существенно усложняет компьютерные технологии решения уравнений.

9.1.2. Решение систем нелинейных алгебраических уравнений

Решение систем нелинейных алгебраических уравнений аналитическими методами возможно лишь в редких случаях. Здесь общих методов и алгоритмов не существует. Нелинейные уравнения решаются обычно численными методами. Наиболее популярными из них являются методы Ньютона, итераций, наискорейшего спуска.

Все эти методы являются итерационными, реализующими метод последовательных приближений.

Алгоритмом численных методов является совокупность правил выбора начальных приближений, расчетных соотношений (формул) и признака окончания вычислений.

Методы Ньютона, итераций и наискорейшего спуска отличаются лишь формулами вычисления приближений.

В системах символьной математики наиболее популярными являются методы Ньютона и итераций. Рассмотрим вкратце эти методы.

Определение начальных приближений

Нелинейные алгебраические уравнения могут иметь большое число решений. Поэтому выбор начальных приближений, от которых и зависит вид решения, должен осуществляться весьма тщательно. Диапазон, из которого выбираются приближения, должен быть достаточно узким.

Существуют два основных способа выбора начальных приближений: графический и табличный.

Графический способ

Этот способ легко реализовать в случае системы двух уравнений.

Пусть имеется система уравнений:

$$f_1(x_1, x_2) = 0$$

$$f_2(x_1, x_2) = 0.$$

Каждое из этих уравнений на плоскости представляет собой линию. Точки их пересечения являются решениями системы уравнений. Их и можно взять за начальные приближения для уточнения решений любым из численных методов.

Использование графического метода для определения начальных приближений в случае систем уравнений высокого порядка затруднительно, но возможно.

Табличный способ

По этому способу функция $f_1(x_1, x_2)$ представляется в виде таблицы. Табличные значения подставляются в $f_2(x_1, x_2) = 0$. Определяются значения x_1, x_2 , при которых функция $f_2(x_1, x_2)$ меняет знак. Эти значения и принимаются за начальные приближения.

Графики и таблицы легко создаются с помощью компьютерных технологий, в частности в среде системы Mathematica.

Метод Ньютона

Метод является итерационным. На каждом шаге по известному предыдущему приближению x_{i-1} определяется следующее более точное значение

$$x_i = x_{i-1} + h_i,$$

где h_i — поправка к предыдущему решению.

Для определения поправки нелинейная система уравнений заменяется некоторой линейной системой, решения которой и являются поправками $h_i, i = 1, 2, \dots, n$.

Таким образом, для определения поправок необходимо решать систему линейных уравнений столько раз, сколько требуется итераций для обеспечения необходимой точности определения неизвестных.

Метод Ньютона является наиболее быстрым по скорости сходимости итераций. Весьма часто за несколько шагов можно получить решение с высокой точностью.

Однако этот метод наиболее трудоемкий, т. к. он требует решения системы линейных уравнений на каждом шаге итерационного процесса.

Метод является достаточно критичным к выбору начальных значений неизвестных. Если начальные значения выбраны неудачно, то можно получить решение не то, что нужно, или вовсе его не получить (расходящийся процесс).

Итерационные методы

Реализация итерационных методов при решении систем нелинейных уравнений не отличается от методов итераций для случая линейных систем. Здесь только возникают большие трудности приведения систем уравнений к виду:

$$x_i = f_i(x_1, x_2, \dots, x_n)$$

и особенно к виду, обеспечивающему сходимость итераций.

Как и в случае линейных систем, здесь применяется метод простой итерации и метод Зейделя.

Достоинством метода является простота алгоритма, что и является основной причиной широкого его использования в системах компьютерной алгебры. Недостаток метода — большая трудность обеспечения сходимости итераций.

Признак окончания итераций

Во всех численных методах признаком окончания итераций является условие:

$$h_i^{(k)} < \varepsilon, \quad i = 1, 2, \dots, n,$$

т. е. поправки всех неизвестных на k -м шаге должны быть меньше требуемой точности.

Рассмотрим технологии решения систем линейных и нелинейных уравнений в среде системы Mathematica.

9.2. Компьютерные технологии решения уравнений в системе Mathematica

Система Mathematica имеет богатые возможности решения систем алгебраических уравнений. Встроенные функции позволяют решать системы линейных и нелинейных уравнений в аналитическом и численном виде. Дают возможность достаточно эффективно и оригинально проверять достоверность результатов. В процессе функционирования система выдает комментарии, позволяющие пользователю принимать решения в отношении полученных ответов.

Основными функциями решения систем уравнений являются: `Solve[F,X]`, `Solve[F,X,Y]`, `N[Solve[F,X]]`, `FindRoot[F,X]`.

Рассмотрим эти функции, опишем технологию их реализации, приведем примеры и задачи для самостоятельного решения.

9.2.1. Функция `Solve[F,X]`

Функция `Solve[F,X]` позволяет решать системы линейных и нелинейных уравнений в аналитическом виде. Она представляется следующим образом:

`Solve[{f1, f2, ...}, {x1, x2, ...}]`

где:

- f_i — i -е уравнение, представленное в произвольном виде;
- x_i — i -е неизвестное.

Уравнения $f_1, f_2, \dots$ могут также представляться через объединительный знак $\&\&$.

Примеры представления функции `Solve[F, X]`:

```
Solve[{a*x^2+y==b, x+2*y==a+b}, {x,y}]
```

```
Solve[a*x^2+y==b&& x+2*y==a+b, {x,y}]
```

При вводе уравнений знак умножения (*) можно заменить нажатием клавиши <Пробел>.

При решении практических задач бывает удобно, а в ряде случаев даже целесообразно, вводить уравнения отдельно от функции `Solve`, присваивая им имена, которые затем вводить в функцию `Solve` вместо уравнений. Например:

```
f1=a*x^2+y==b; f2=x+2*y==a+b
```

```
Solve[{f1, f2}, {x,y}]
```

или

```
Solve[f1&&f2, {x,y}]
```

Такое представление упрощает проверку достоверности решения системы уравнений.

Из математики известно, что система линейных уравнений любого порядка имеет решение, и притом единственное. Поэтому в данном случае функция `Solve[F, X]` ограничений на порядок уравнений не имеет.

Решение систем линейных алгебраических уравнений

Технология решения уравнений:

1. Ввод уравнений с присвоением каждому из них, с помощью знака присвоения (=), уникального имени.
2. Ввод функции `Solve[{f1, f2, ...}, {x, y, ...}]` или `Solve[f1&&f2&&... {x, y, ...}]`.
3. Получение решения путем одновременного нажатия комбинации клавиш <Shift>+<Enter>.
4. Проверка достоверности решения системы уравнений.

Приведем примеры решения линейных алгебраических уравнений в аналитическом виде.

Пример 9.1

Необходимо решить следующие системы линейных уравнений:

$$x_1 + ax_2 + bx_3 = y_1$$

$$3x_1 - 4x_2 + 2x_3 = 1$$

$$x_1 + 7x_2 - x_3 = 3.5$$

$$(2+a)x_1 + x_2 + cx_3 = y_2$$

$$x_1 + 7x_2 - 2x_3 = -4$$

$$-1.6x_1 + 3.7x_2 = 12$$

$$ax_1 + bx_2 + cx_3 = y_3$$

$$2x_1 + 7x_2 + 3x_3 = 3$$

$$x_1 + 2x_2 + 5x_3 = 7.5$$

```

f1 = x1 + a * x2 + b * x3 == y1
x1 + a x2 + b x3 == y1
f2 = (2 + a) * x1 + x2 + c * x3 == y2
(2 + a) x1 + x2 + c x3 == y2
f3 = a * x1 + b * x2 + c * x3 == y3
a x1 + b x2 + c x3 == y3
Solve[{f1, f2, f3}, {x1, x2, x3}]


$$\left\{ \left\{ x1 \rightarrow -\frac{-c y1 + b c y1 - b^2 y2 + a c y2 + b y3 - a c y3}{-a b + 2 b^2 + a b^2 + c - 2 a c - b c}, \right. \right.$$


$$x2 \rightarrow -\frac{2 c y1 + a b y2 - c y2 - 2 b y3 - a b y3 + c y3}{-a b + 2 b^2 + a b^2 + c - 2 a c - b c},$$


$$x3 \rightarrow -\frac{a y1 - 2 b y1 - a b y1 - a^2 y2 + b y2 - y3 + 2 a y3 + a^2 y3}{-a b + 2 b^2 + a b^2 + c - 2 a c - b c} \left. \right\}$$

f3 = 3 x1 - 4 x2 + 2 x3 == 1
3 x1 - 4 x2 + 2 x3 == 1
f4 = x1 + 7 x2 - 2 x3 == -4
x1 + 7 x2 - 2 x3 == -4
f5 = 2 x1 + 7 x2 + 3 x3 == 3
2 x1 + 7 x2 + 3 x3 == 3
Solve[f3 && f4 && f5, {x1, x2, x3}]

$$\left\{ \left\{ x1 \rightarrow -\frac{87}{119}, x2 \rightarrow -\frac{3}{119}, x3 \rightarrow \frac{184}{119} \right\} \right\}$$

Solve[{x1 + 7 x2 - x3 == 3.5, -1.6 x1 + 3.7 x2 == 12,
x1 + 2 x2 + 5 x3 == 7.5}, {x1, x2, x3}]
{ {x1 → -4.31818, x2 → 1.37592, x3 → 1.81327} }

```

Рис. 9.1. Решение систем уравнений примера 9.1

Решение уравнений по изложенной технологии приведено на рис. 9.1.

Из рис. 9.1 видно, что система решила первую систему уравнений в аналитическом виде, вторую — в виде точных значений неизвестных, представленных в рациональной форме. Третья система уравнений также решена, но решение представлено в виде вещественных чисел. Это объясняется тем, что система уравнений точного решения не имеет. Таким образом, функция `Solve` может решать системы уравнений также в численном виде.

Из примера видно, что решение представляется в виде подстановок: $x_1 \rightarrow$, $x_2 \rightarrow$, $x_3 \rightarrow$. Это не дает возможности проверить достоверность решения, а также использовать значения x_1 , x_2 , x_3 в дальнейших расчетах.

Для проверки достоверности решения пользователь должен представить неизвестные в явном виде с присвоением им имени: $x_1 = -4.31818$, $x_2 = 1.37592$, $x_3 = 1.81327$, затем использовать их по назначению.

Получить решение в явном виде можно с помощью выражения вида $\{x_1, x_2, x_3\}/.$, которое ставится перед функцией Solve: $\{x_1, x_2, x_3\}/. \text{Solve}[\{f_1, f_2, f_3\}, \{x_1, x_2, x_3\}]$. После нажатия комбинации клавиш <Shift>+<Enter> получаем решение в явном виде. Теперь x_1, x_2, x_3 можно использовать по назначению, в том числе и для проверки достоверности решения системы уравнений.

Пример 9.2

Необходимо решить следующие системы алгебраических уравнений:

$$ax_1 + bx_2 + x_3 = s_1$$

$$(2+a)x_1 + x_2 + cx_3 = s_2$$

$$x_1 + x_2 + x_3 = s_3$$

$$2x_1 - x_2 + 4x_3 = 1$$

$$x_1 + 7x_2 - 2x_3 = -4$$

$$7x_1 + x_2 + x_3 = 3$$

и проверить достоверность решения.

Решения систем уравнений приведены на рис. 9.2.

```
f1 = a x1 + b x2 + x3 == s1
a x1 + b x2 + x3 == s1
f2 = (2 + a) x1 + x2 + c x3 == s2
(2 + a) x1 + x2 + c x3 == s2
f3 = x1 + x2 + x3 == s3
x1 + x2 + x3 == s3
z = Solve[{f1, f2, f3}, {x1, x2, x3}]
{{x1 -> -
  s1 - c s1 + s2 - b s2 - s3 + b c s3
  -1 - 2 a + 2 b + a b + a c - b c
  -2 s1 - a s1 + c s1 - s2 + a s2 + 2 s3 + a s3 - a c s3
  x2 -> -
  -1 - 2 a + 2 b + a b + a c - b c
  s1 + a s1 - a s2 + b s2 + a s3 - 2 b s3 - a b s3
  x3 -> -
  -1 - 2 a + 2 b + a b + a c - b c
  {f1, f2, f3} /. z]
{{
  s1 + a s1 - a s2 + b s2 + a s3 - 2 b s3 - a b s3
  -1 - 2 a + 2 b + a b + a c - b c
  a (s1 - c s1 + s2 - b s2 - s3 + b c s3)
  == s1,
  -1 - 2 a + 2 b + a b + a c - b c
  c (s1 + a s1 - a s2 + b s2 + a s3 - 2 b s3 - a b s3)
  -2 s1 - a s1 + c s1 - s2 + a s2 + 2 s3 + a s3 - a c s3
  -1 - 2 a + 2 b + a b + a c - b c
  (2 + a) (s1 - c s1 + s2 - b s2 - s3 + b c s3)
  -1 - 2 a + 2 b + a b + a c - b c
  == s2,
```

Рис. 9.2. (Часть 1 из 2) Решение задач примера 9.2


```

-  $\frac{s1 + as1 - as2 + bs2 + as3 - 2bs3 - abs3}{-1 - 2a + 2b + ab + ac - bc}$  -  $\frac{-2s1 - as1 + cs1 - s2 + as2 + 2s3 + as3 - acs3}{-1 - 2a + 2b + ab + ac - bc}$  -
 $\frac{s1 - cs1 + s2 - bs2 - s3 + bcs3}{-1 - 2a + 2b + ab + ac - bc} == s3\}}\}$ 
Simplify[%]
{{True, True, True}}

f1 /. z

$$\left\{ -\frac{s1 + as1 - as2 + bs2 + as3 - 2bs3 - abs3}{-1 - 2a + 2b + ab + ac - bc} - \frac{b(-2s1 - as1 + cs1 - s2 + as2 + 2s3 + as3 - acs3)}{-1 - 2a + 2b + ab + ac - bc}, \right.$$


$$\left. \frac{a(s1 - cs1 + s2 - bs2 - s3 + bcs3)}{-1 - 2a + 2b + ab + ac - bc} == s1 \right\}$$

Simplify[%]
{True}

f1 = 2 x1 - x2 + 4 x3 == 1
2 x1 - x2 + 4 x3 == 1
f2 = x1 + 7 x2 - 2 x3 == -4
x1 + 7 x2 - 2 x3 == -4
f3 = 7 x1 + x2 + x3 == 3
7 x1 + x2 + x3 == 3
z = Solve[f1 && f2 && f3, {x1, x2, x3}]
 $\left\{ \left\{ x1 \rightarrow \frac{89}{159}, x2 \rightarrow -\frac{113}{159}, x3 \rightarrow -\frac{11}{53} \right\} \right\}$ 
{f1, f2, f3} /. z
{{True, True, True}}
f1 = 2 x1 - x2 + 4 x3 == 1
2 x1 - x2 + 4 x3 == 1
f2 = x1 + 7 x2 - 2 x3 == -4
x1 + 7 x2 - 2 x3 == -4
f3 = 7 x1 + x2 + x3 == 3
7 x1 + x2 + x3 == 3
{x1, x2, x3} /. Solve[f1 && f2 && f3, {x1, x2, x3}]
 $\left\{ \left\{ \frac{89}{159}, -\frac{113}{159}, -\frac{11}{53} \right\} \right\}$ 
{f1, f2, f3}
{True, True, True}

```

Рис. 9.2. (Часть 2 из 2)

На рис. 9.2 показаны разные формы представления функции `Solve` и способы проверки достоверности решения систем уравнения.

Решение систем нелинейных уравнений в символьном виде

Технология решения систем нелинейных уравнений та же, что и линейных. Покажем ее на примерах.

Пример 9.3

Необходимо решить следующие системы нелинейных уравнений:

$$x + ay = b$$

$$xy = a$$

$$x + by^2 = a + b$$

$$x^2 + y^2 = ab$$

и проверить достоверность решения.

Решение приведено на рис. 9.3.

Из рис. 9.3 видно, что все решения получены верно.

```
f1 = x + a y == b
```

```
x + a y == b
```

```
f2 = x + b y ^ 2 == a + b
```

```
x + b y ^ 2 == a + b
```

```
w = Solve[{f1, f2}, {x, y}]
```

$$\left\{ \left\{ x \rightarrow \frac{-a^2 + 2 b^2 - a^{3/2} \sqrt{a + 4 b}}{2 b}, y \rightarrow \frac{a + \sqrt{a} \sqrt{a + 4 b}}{2 b} \right\}, \right.$$

$$\left. \left\{ x \rightarrow \frac{-a^2 + 2 b^2 + a^{3/2} \sqrt{a + 4 b}}{2 b}, y \rightarrow \frac{a - \sqrt{a} \sqrt{a + 4 b}}{2 b} \right\} \right\}$$

$$\left\{ \left\{ x \rightarrow \frac{-a^2 + 2 b^2 - a^{3/2} \sqrt{a + 4 b}}{2 b}, y \rightarrow \frac{a + \sqrt{a} \sqrt{a + 4 b}}{2 b} \right\}, \right.$$

$$\left. \left\{ x \rightarrow \frac{-a^2 + 2 b^2 + a^{3/2} \sqrt{a + 4 b}}{2 b}, y \rightarrow \frac{a - \sqrt{a} \sqrt{a + 4 b}}{2 b} \right\} \right\}$$

```
{f1, f2} /. w
```

$$\left\{ \left\{ \frac{a (a + \sqrt{a} \sqrt{a + 4 b})}{2 b} + \frac{-a^2 + 2 b^2 - a^{3/2} \sqrt{a + 4 b}}{2 b} == b, \right. \right.$$

$$\left. \frac{(a + \sqrt{a} \sqrt{a + 4 b})^2}{4 b} + \frac{-a^2 + 2 b^2 - a^{3/2} \sqrt{a + 4 b}}{2 b} == a + b \right\},$$

$$\left\{ \left\{ \frac{a (a - \sqrt{a} \sqrt{a + 4 b})}{2 b} + \frac{-a^2 + 2 b^2 + a^{3/2} \sqrt{a + 4 b}}{2 b} == b, \right. \right.$$

Рис. 9.3. (Часть 1 из 2) Решение примера 9.3

$$\left. \frac{(a - \sqrt{a} \sqrt{a+4b})^2}{4b} + \frac{-a^2 + 2b^2 + a^{3/2} \sqrt{a+4b}}{2b} == a + b \right\}$$

Simplify[%]

{(True, True), (True, True)}

f3 = x y == a

xy == a

f4 = x^2 + y^2 == a b

x^2 + y^2 == a b

r = Solve[{f3, f4}, {x, y}]

$$\left\{ \left\{ x \rightarrow \frac{1}{2} \left\{ -b \sqrt{\frac{ab}{2} - \frac{1}{2} a \sqrt{-4 + b^2}} - \sqrt{-4 + b^2} \sqrt{\frac{ab}{2} - \frac{1}{2} a \sqrt{-4 + b^2}} \right\}, \right. \right.$$

$$\left. y \rightarrow -\sqrt{\frac{ab}{2} - \frac{1}{2} a \sqrt{-4 + b^2}} \right\},$$

$$\left\{ x \rightarrow \frac{1}{2} \left\{ b \sqrt{\frac{ab}{2} - \frac{1}{2} a \sqrt{-4 + b^2}} + \sqrt{-4 + b^2} \sqrt{\frac{ab}{2} - \frac{1}{2} a \sqrt{-4 + b^2}} \right\}, \right.$$

$$\left. y \rightarrow \sqrt{\frac{ab}{2} - \frac{1}{2} a \sqrt{-4 + b^2}} \right\},$$

$$\left\{ x \rightarrow \frac{1}{2} \left\{ b \sqrt{\frac{ab}{2} + \frac{1}{2} a \sqrt{-4 + b^2}} - \sqrt{-4 + b^2} \sqrt{\frac{ab}{2} + \frac{1}{2} a \sqrt{-4 + b^2}} \right\}, \right.$$

$$\left. y \rightarrow \sqrt{\frac{ab}{2} + \frac{1}{2} a \sqrt{-4 + b^2}} \right\},$$

$$\left\{ x \rightarrow \frac{1}{2} \left\{ -b \sqrt{\frac{ab}{2} + \frac{1}{2} a \sqrt{-4 + b^2}} + \sqrt{-4 + b^2} \sqrt{\frac{ab}{2} + \frac{1}{2} a \sqrt{-4 + b^2}} \right\}, \right.$$

$$\left. y \rightarrow -\sqrt{\frac{ab}{2} + \frac{1}{2} a \sqrt{-4 + b^2}} \right\} \}$$

Simplify[{f3, f4} /. r]

{(True, True), (True, True), (True, True), (True, True)}

Рис. 9.3. (Часть 2 из 2)

9.2.2. Функция **Solve[F,X,Y]**

Функция **Solve[F,X,Y]**, так же, как и функция **Solve[F,X]**, позволяет решать системы линейных и нелинейных уравнений в аналитическом виде, но только с ограничением: решения осуществляются по переменным **X** и исключаются по переменным **Y**. Например, функция **Solve[{x+2*y-a==3, 2*x+y^2+b==7}, x, y]** определит **x** и исключит из решения **y**.

Пример 9.4

Необходимо решить следующие системы уравнений:

$$\begin{array}{ll} a \cdot x^2 + b \cdot x \cdot y = 1 & 7 \ a + 3 \ b + c = 1 \\ x \cdot y + c = 7 & -5 \ a + 12 \ b = 3 \\ a + b + 2c = -7 & c = -7 \end{array}$$

Первую систему решить относительно x , вторую — относительно a , затем относительно a и b . Решения получить в явном виде.

Решение приведено на рис. 9.4.

```

f1 = a x^2 + b x y == 1
a x^2 + b x y == 1
f2 = x y + c == 7
c + x y == 7
z = Solve[{f1, f2}, x, y]
{{x -> - $\frac{\sqrt{1 - 7b + bc}}{\sqrt{a}}$ }, {x ->  $\frac{\sqrt{1 - 7b + bc}}{\sqrt{a}}$ }}
x /. z
{- $\frac{\sqrt{1 - 7b + bc}}{\sqrt{a}}$ ,  $\frac{\sqrt{1 - 7b + bc}}{\sqrt{a}}$ }
f3 = 7 a + 3 b + c == 1
7 a + 3 b + c == 1
f4 = -5 a + 12 b == 3
-5 a + 12 b == 3
f5 = a + b + 2 c == -7
a + b + 2 c == -7
s = Solve[{f3, f4, f5}, {a, b, c}]
{{a ->  $\frac{93}{181}$ , b ->  $\frac{84}{181}$ , c ->  $-\frac{722}{181}$ }}
s1 = Solve[{f3, f4, f5}, a, {b, c}]
{{a ->  $\frac{93}{181}$ }}
a /. s1
{ $\frac{93}{181}$ }
s2 = Solve[{f3, f4, f5}, {a, b}, c]
{{a ->  $\frac{93}{181}$ , b ->  $\frac{84}{181}$ }}
{a, b} /. s2
{{ $\frac{93}{181}$ ,  $\frac{84}{181}$ }}

```

Рис. 9.4. Решение систем уравнений примера 9.4

9.2.3. Функция **NSolve[F,X]**

Функция **NSolve[F,X]** позволяет решать системы линейных и нелинейных уравнений в численном виде. Она представляется следующим образом:

NSolve[{f₁, f₂, ...}, {x₁, x₂, ...}]

где:

- ☐ **f_i** — *i*-е уравнение, представленное в произвольном виде;
- ☐ **x_i** — *i*-е неизвестное.

```

f1 = 2 x1 + 7 x2 - x3 == 5
2 x1 + 7 x2 - x3 == 5
f2 = x1 - 2 x2 + 5 x3 == 2
x1 - 2 x2 + 5 x3 == 2
f3 = 4 x1 + x2 + 3 x3 == -7
4 x1 + x2 + 3 x3 == -7
s = NSolve[{f1, f2, f3}, {x1, x2, x3}]
{{x1 -> -3.75, x2 -> 2.06818, x3 -> 1.97727}}
{f1, f2, f3} /. s
{{True, True, True}}
f1 = 2 y + 3 x^2 == 5
3 x^2 + 2 y == 5
f2 = x + 7 y^2 == 7.5
x + 7 y^2 == 7.5
r = NSolve[{f1, f2}, {x, y}]
{{x -> 1.5110628030619`, y -> -0.9249661921959316`},
 {x -> -0.9661774325244309`, y -> 1.0997517533207495`},
 {x -> 1.0123543818272729`, y -> 0.962707908392686`},
 {x -> -1.5572397523647399`, y -> -1.1374934695175007`}}
{f1, f2} /. r
{{True, True}, {True, True}, {True, True}, {True, True}}
f3 = Sin[x1] - x2 == 1.3
-x2 + Sin[x1] == 1.3
f4 = Cos[x2] - x1 == -0.82
-x1 + Cos[x2] == -0.82
r1 = NSolve[{f3, f4}, {x1, x2}]
NSolve[{-x2 + Sin[x1] == 1.3, -x1 + Cos[x2] == -0.82}, {x1, x2}]

```

Рис. 9.5. Решение систем уравнений примера 9.5

Уравнения $f_1, f_2, \dots$ могут также представляться через объединительный знак $\&\&$.

Технология решения систем уравнений с помощью функции `NSolve[F, X]` практически не отличается от технологии решения с помощью функции `Solve[F, X]`. Приведем примеры.

Пример 9.5

Требуется решить следующие системы уравнений:

$$2x_1 + 7x_2 - x_3 = 5$$

$$2y + 3x^2 = 5$$

$$\sin x_1 - x_2 = 1.3$$

$$x_1 - 2x_2 + 5x_3 = 2$$

$$x + 7y^2 = 7.5$$

$$\cos x_2 - x_1 = -0.82$$

$$4x_1 + x_2 + 3x_3 = -7$$

Решения приведены на рис. 9.5.

Из рис. 9.5 видно, что функция `NSolve[F, X]` не решила последнюю систему уравнений. Это объясняется тем, что система состоит из трансцендентных уравнений. Для ее решения необходимы методы, нереализованные в функции `NSolve[F, X]`.

9.2.4. Опции функции `Solve`

Опции функции `Solve` выводятся командой `Options[Solve]`:

Options[Solve]

```
{InverseFunctions → Automatic, MakeRules → False, Method → 3,  
  Mode → Generic, Sort → True, VerifySolutions → Automatic,  
  WorkingPrecision → ∞}
```

В этом списке значения опций указаны по умолчанию. Опции при практических расчетах применяются редко.

Приведем только один пример.

Пример 9.6

Необходимо решить с использованием опции `Method` следующую систему уравнений:

$$xy + z = a$$

$$x + y + z = b$$

$$y + z = c$$

Решение приведено на рис. 9.6.

Из рис. 9.6 видно, что лишь метод 3 выдал решение в явном виде.

```

Solve[{x y + z == a, x + y + z == b, y + z == c}, {x, y, z}, Method -> 1]
{{(1 - b + c) y -> -a + c, x -> b - c, z -> c - y}}
Solve[{x y + z == a, x + y + z == b, y + z == c}, {x, y, z}, Method -> 2]
{{(1 - b + c) y -> -a + c, x -> b - c, z -> c - y}}
Solve[{x y + z == a, x + y + z == b, y + z == c}, {x, y, z}, Method -> 3]
{{z -> (a - b c + c^2) / (1 - b + c), x -> b - c, y -> (-a + c) / (1 - b + c)}}

```

Рис. 9.6. Решение системы уравнений с помощью опции Method

9.2.5. Функция FindRoot[F, {X, x₀}]

Функция FindRoot[F, {X, x₀}] решает системы линейных и нелинейных уравнений численными методами итераций. Для ее реализации необходимо знать начальные приближения неизвестных. Функция имеет вид:

FindRoot[{f₁, f₂, ...}, {x₁, x₁₀}, {x₂, x₂₀}, ...]

где:

- ☐ f_i — *i*-е уравнение, представленное в произвольном виде;
- ☐ x_i — *i*-е неизвестное;
- ☐ x_{i0} — начальное приближение *i*-го неизвестного.

Уравнения f₁, f₂, ... могут также представляться через объединительный знак &&.

Технология решения систем уравнений с помощью функции FindRoot[F, {X, x₀}] существенно отличается от технологии решения уравнений с помощью функции NSolve[F, X]. Отличие состоит в необходимости определения начальных приближений. Методы выбора начальных приближений рассмотрены нами в разд. 9.1.

Приведем примеры решения систем уравнений функцией FindRoot[F, {X, x₀}].

Пример 9.7

Необходимо решить следующие системы уравнений:

$$\sin x_1 - x_2 = 1.3$$

$$\cos x_2 - x_1 = -0.82$$

$$x_{10} = 1.8$$

$$x_{20} = -0.35$$

$$\operatorname{tg}(y_1 y_2 + 0.2) = y_1^2$$

$$0.5 y_1^2 + 2 y_2^2 = 1$$

$$y_{10} = 0.9$$

$$y_{20} = 0.5$$

Проверить достоверность решения.

Решение систем уравнений приведено на рис. 9.7.

```

f1 = Sin[x1] - x2 == 1.3
-x2 + Sin[x1] == 1.3

f2 = Cos[x2] - x1 == -0.82
-x1 + Cos[x2] == -0.82

r = FindRoot[{f1, f2}, {x1, 1.8}, {x2, -0.35}]
{x1 → 1.76935, x2 → -0.319646}

{f1, f2} /. r
{False, False}

x1 := 1.76935; x2 := -0.319646
Sin[x1] - x2
1.3

Cos[x2] - x1
-0.820003

{x1, x2} /. r
{1.76935, -0.319646}

f3 = Tan[y1 y2 + 0.2] == y1^2
Tan[0.2 + y1 y2] == y1^2

f4 = 0.5 y1^2 + 2 y2^2 == 1
0.5 y1^2 + 2 y2^2 == 1

z = FindRoot[{f3, f4}, {y1, 0.9}, {y2, 0.5}]
{y1 → 0.910994, y2 → 0.540854}

{y1, y2} /. z
{0.910994, 0.540854}

y1 := 0.910994; y2 := 0.540854
Tan[y1 y2 + 0.2]
0.82991

y1^2
0.82991

0.5 y1^2 + 2 y2^2
1.

```

Рис. 9.7. Решение систем уравнений примера 9.7

9.2.6. Функция *Eliminate*[F,x]

Функция `Eliminate[F,x]` предназначена для сокращения числа уравнения системы путем исключения заданных переменных x . Она имеет вид:

`Eliminate[{f1, f2, ...}, x]`

где:

□ f_i — i -е уравнение, представленное в произвольном виде;

□ x — неизвестное, подлежащее исключению.

Уравнения $f_1, f_2, \dots$ могут также представляться через объединительный знак `&&`.

Эта функция осуществляет преобразование исходной системы уравнений так, что число уравнений и переменных сокращается. Предельным является одно уравнение с одним неизвестным. Например, пусть дана система уравнений:

$$2x_1 + 3x_2 = 1$$

$$x_1 - 2x_2 = 2$$

Решить эту систему уравнений можно методом подстановки. Находим x_2 из второго уравнения и подставляем в первое. Получаем после преобразования уравнение $x_1 = 2 - 3/4 x_1$, которое решается элементарно. Преобразование уравнений функцией `Eliminate[F,x]` может быть не только методом подстановки, но также любым другим методом, известным в алгебре. Наличие этой функции — еще одно доказательство высокой интеллектуальности системы `Mathematica`.

Приведем пример, на котором объясним смысл этой функции.

Пример 9.8

Дана следующая система уравнений:

$$x^2 + ay^2 - z = 2$$

$$x + y - bz = 7$$

$$2x + 3y + 9z = 1$$

Необходимо получить эквивалентную систему, состоящую из двух уравнений с двумя неизвестными, исключив z , и одно уравнение с неизвестным x , исключив y и z .

Решение приведено на рис. 9.8.

Из рис. 9.8 видно, что при исключении из системы уравнений z программа определила z из какого-то одного уравнения и подставила его в два других. Образовалась система из двух уравнений с двумя неизвестными: x и y . При исключении двух переменных y и z получено одно уравнение с неизвестным x . С целью упрощения этого выражения нами использовалась функция `FullSimplify`.

```

f1 = x^2 + a y^2 - z == 2
 $x^2 + a y^2 - z = 2$ 
f2 = x + y - b z == 7
 $x + y - b z = 7$ 
f3 = 2 x + 3 y + 9 z == 1
 $2 x + 3 y + 9 z = 1$ 
Eliminate[{f1, f2, f3}, z]
 $9 a y^2 == 19 - 2 x - 9 x^2 - 3 y \&\& b (-1 + 2 x + 3 y) == 63 - 9 x - 9 y$ 
Eliminate[{f1, f2, f3}, {y, z}]
 $b^2 (-18 + a - 4 a x + 9 x^2 + 4 a x^2) + b (-48 + 126 a - 3 x - 270 a x + 54 x^2 + 36 a x^2) ==$ 
 $-18 - 3969 a + 9 x + 1134 a x - 81 x^2 - 81 a x^2$ 
FullSimplify[Out[5]]
 $a (9 (-7 + x) + b (-1 + 2 x))^2 + 3 (3 + b) (2 + x (-1 + 9 x) + 3 b (-2 + x^2)) == 0$ 

```

Рис. 9.8. Преобразование системы уравнений

Достоверность решения легко проверить, сравнивая результаты, полученные с помощью функции `Eliminate` и другими методами. Покажем это на примере.

Пример 9.9

Дана следующая система уравнений:

$$25x^2 + 12y - 7z = 5$$

$$12x + 25y^2 + 3z = 7$$

$$x + y + 10z^2 = 14$$

Необходимо определить неизвестные с помощью функции `NSolve` и методом подстановки с использованием функции `Eliminate`. Решение приведено на рис. 9.9. Определение значений x , y , z из выражений, полученных методом подстановки, выполнено с помощью функции `NRoots`. Из рис. 9.9 видно, что решения совпадают для всех неизвестных.

Функция `Eliminate` дает возможность определить области неизвестных x , y , z путем построения графиков функций. На рис. 9.10 показан график функции $w = f(x)$ примера 9.9. Из графика видно, что координаты точек пересечения функции с осью x совпадают со значениями x , полученными в примере 9.9.

Функция `Eliminate` может быть весьма полезной для определения начальных значений неизвестных при решении системы нелинейных уравнений с помощью функции `FindRoot`.

```

f1 = 25 x^2 + 12 y - 7 z == 5

$$25x^2 + 12y - 7z = 5$$

f2 = 12 x + 25 y^2 + 3 z == 7

$$12x + 25y^2 + 3z = 7$$

x + y + 10 z^2 == 14

$$x + y + 10z^2 = 14$$

w = NSolve[{f1, f2, f3}, {x, y, z}]
{{x → -0.382156401766938`, y → 0.6022236130606954`, z → 0.8396816060020971`},
 {x → 0.5265369748220278`, y → -0.17169375052956706`, z → -0.018470765712354995`},
 {x → -0.8674761977115225`, y → -0.7885509839907304`, z → 0.6214660045731777`},
 {x → 0.3730956246564325`, y → 0.268021121459602`, z → 0.24232315513708044`}}
Eliminate[{f1, f2, f3}, {y, z}]

$$-422x - 4770x^2 + 2800x^3 + 8000x^4 == -521$$

NRoots[Out[18], x]
x == -0.867476 || x == -0.382156 || x == 0.373096 || x == 0.526537
Eliminate[{f1, f2, f3}, {x, z}]

$$37170y - 538850y^2 + 90000y^3 + 1000000y^4 == -21853$$

NRoots[Out[23], y]
y == -0.788551 || y == -0.171694 || y == 0.268021 || y == 0.602224
Eliminate[{f1, f2, f3}, {x, y}]

$$-7057520z + 54044100z^2 - 107840000z^3 + 64000000z^4 == 149483$$

NRoots[Out[26], z]
z == -0.0184708 || z == 0.242323 || z == 0.621466 || z == 0.839682

```

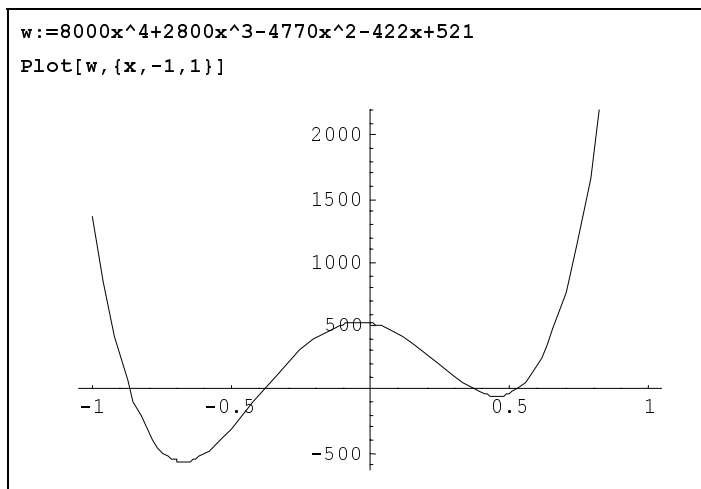
Рис. 9.9. Решение задачи примера 9.9

9.2.7. Матричные методы решения систем линейных уравнений

Систему линейных алгебраических уравнений можно представить в следующем виде: $A \cdot X = B$, где A — матрица коэффициентов, X — вектор неизвестных, B — вектор свободных членов (правых частей) системы уравнений.

Технология решения уравнений в системе Mathematica проста и состоит в следующем:

1. Ввод матрицы коэффициентов с присвоением ей имени, например A .
2. Ввод вектора неизвестных с именем X .

Рис. 9.10. График функции $w = f(x)$

3. Ввод вектора свободных членов с именем В.
4. Образование выражения $z = A \cdot X == B$.
5. Ввод функции `Solve[z, x]`.
6. Получение решения одновременным нажатием клавиш <Shift>+<Enter>.

Ввод данных можно упростить, если вместо образования выражения $A \cdot X == B$ представить его в виде произведения матрицы A на вектор X и приравнять к вектору B .

Матричные методы позволяют решать системы линейных уравнений с вещественными и комплексными переменными.

Покажем технологию решения системы линейных уравнений матричным методом на примерах.

Пример 9.10

Решить матричным методом следующую систему уравнений:

$$\begin{aligned} 2x_1 + 3x_2 - 7x_3 &= 1 \\ -3x_1 + x_2 + 5x_3 &= 7.5 \\ 5x_1 + 3x_3 &= 2.5 \end{aligned}$$

Решение приведено на рис. 9.11.

```

A = {{2, 3, -7}, {-3, 1, 5}, {5, 0, 3}}
{{2, 3, -7}, {-3, 1, 5}, {5, 0, 3}}
X = {x1, x2, x3}
{x1, x2, x3}
B = {1, 7.5, 2.5}
{1, 7.5, 2.5}
Z = A.X == B
{2 x1 + 3 x2 - 7 x3, -3 x1 + x2 + 5 x3, 5 x1 + 3 x3} == {1, 7.5, 2.5}
Solve[%]
{{x1 -> -0.0664336, x2 -> 2.58042, x3 -> 0.944056}}
{{2, 3, -7}, {-3, 1, 5}, {5, 0, 3}}. {x1, x2, x3} == {1, 7.5, 2.5}
{2 x1 + 3 x2 - 7 x3, -3 x1 + x2 + 5 x3, 5 x1 + 3 x3} == {1, 7.5, 2.5}
Solve[%]
{{x1 -> -0.0664336, x2 -> 2.58042, x3 -> 0.944056}}

```

Рис. 9.11. Решение системы уравнений матричным методом

Матричный метод позволяет решать системы линейных алгебраических уравнений в аналитическом виде. Покажем эту возможность на примере.

Пример 9.11

Необходимо решить матричным методом следующую систему уравнений:

$$a_1x_1 + a_2x_2 + a_3x_3 = b_1$$

$$b_1x_1 + b_2x_2 + b_3x_3 = b_2$$

$$c_1x_1 + c_2x_2 + c_3x_3 = b_3$$

Решение приведено на рис. 9.12.

Кроме приведенного существуют в системе Mathematica следующие два матричных способа решения систем алгебраических уравнений.

□ **Способ 1.** Определение вектора неизвестных x по формуле: $x = A^{-1}B$.

При этом операция умножения записывается функцией `Dot`, а операция инвертирования матрицы A — функцией `Inverse`. Тогда решение записывается в следующем виде:

```
X:=Dot[Inverse[A], B]
```

□ **Способ 2.** Применение функции `LinearSolve`.

Функция `LinearSolve` записывается в следующем виде:

```
X:=LinearSolve[A, B]
```

```

A = {{a1, a2, a3}, {b1, b2, b3}, {c1, c2, c3}}
{{a1, a2, a3}, {b1, b2, b3}, {c1, c2, c3}}
X = {x1, x2, x3}
{x1, x2, x3}
B = {b1, b2, b3}
{b1, b2, b3}
Z = A.X == B
{a1 x1 + a2 x2 + a3 x3, b1 x1 + b2 x2 + b3 x3, c1 x1 + c2 x2 + c3 x3} == {b1, b2, b3}
Solve[Z, {x1, x2, x3}]
{{x1 -> - $\frac{-a_3 b_2 b_3 + a_2 b_3^2 + a_3 b_2 c_2 - b_1 b_3 c_2 - a_2 b_2 c_3 + b_1 b_2 c_3}{a_3 b_2 c_1 - a_2 b_3 c_1 - a_3 b_1 c_2 + a_1 b_3 c_2 + a_2 b_1 c_3 - a_1 b_2 c_3}$ ,
  x2 -> - $\frac{a_3 b_1 b_3 - a_1 b_3^2 - a_3 b_2 c_1 + b_1 b_3 c_1 - b_1^2 c_3 + a_1 b_2 c_3}{a_3 b_2 c_1 - a_2 b_3 c_1 - a_3 b_1 c_2 + a_1 b_3 c_2 + a_2 b_1 c_3 - a_1 b_2 c_3}$ ,
  x3 -> - $\frac{-a_2 b_1 b_3 + a_1 b_2 b_3 + a_2 b_2 c_1 - b_1 b_2 c_1 + b_1^2 c_2 - a_1 b_2 c_2}{a_3 b_2 c_1 - a_2 b_3 c_1 - a_3 b_1 c_2 + a_1 b_3 c_2 + a_2 b_1 c_3 - a_1 b_2 c_3}$ }}

```

Рис. 9.12. Решение системы уравнений в аналитическом виде

Приведем пример решения системы линейных уравнений двумя способами.

Пример 9.12

Необходимо решить двумя матричными методами следующую систему линейных уравнений:

$$\begin{aligned}
 ax_1 + 2x_2 - 3x_3 &= b_1 \\
 -7x_1 + cx_2 + x_3 &= b_2 \\
 x_1 + x_2 + dx_3 &= b_3
 \end{aligned}$$

Решение задачи приведено на рис. 9.13.

9.2.8. Особые случаи решения систем уравнений

Система уравнений может иметь число решений, равное числу неизвестных, такая система называется *совместной*. Если число уравнений бесконечно большое, то систему называют *совместной и неопределенной*. Если же система не имеет ни одного решения, то ее называют несовместной. Рассмотрим примеры таких систем.

$$A = \{\{a, 2, -3\}, \{-7, c, 1\}, \{1, 1, d\}\}$$

$$\{\{a, 2, -3\}, \{-7, c, 1\}, \{1, 1, d\}\}$$

$$B = \{b1, b2, b3\}$$

$$\{b1, b2, b3\}$$

$$X = \text{Dot}[\text{Inverse}[A], B]$$

$$\left\{ \frac{b3(2+3c)}{23-a+3c+14d+acd} + \frac{b2(-3-2d)}{23-a+3c+14d+acd} + \frac{b1(-1+c d)}{23-a+3c+14d+acd}, \right. \\ \left. \frac{(21-a)b3}{23-a+3c+14d+acd} + \frac{b1(1+7d)}{23-a+3c+14d+acd} + \frac{b2(3+ad)}{23-a+3c+14d+acd}, \right. \\ \left. \frac{(2-a)b2}{23-a+3c+14d+acd} + \frac{b1(-7-c)}{23-a+3c+14d+acd} + \frac{b3(14+ac)}{23-a+3c+14d+acd} \right\}$$

$$\text{Simplify}[\%]$$

$$\left\{ \frac{b3(2+3c) - b2(3+2d) + b1(-1+cd)}{23+3c+14d+a(-1+cd)}, \frac{b1+3b2+21b3-ab3+7b1d+ab2d}{23-a+3c+14d+acd}, \right. \\ \left. \frac{-(-2+a)b2 - b1(7+c) + b3(14+ac)}{23+3c+14d+a(-1+cd)} \right\}$$

$$X = \text{LinearSolve}[A, B]$$

$$\left\{ \frac{-b1-3b2+2b3+3b3c-2b2d+b1cd}{23-a+3c+14d+acd}, \frac{b1+3b2+21b3-ab3+7b1d+ab2d}{23-a+3c+14d+acd}, \right. \\ \left. \frac{-7b1+2b2-ab2+14b3-b1c+ab3c}{23-a+3c+14d+acd} \right\}$$

Рис. 9.13. Решение системы линейных уравнений двумя матричными способами

Пример 9.13

Требуется решить две следующие системы уравнений:

$$x_1 + 2x_2 - x_3 = 1$$

$$2x_1 + 4x_2 - 2x_3 = 2$$

$$3x_1 + 2x_2 + 3x_3 = 5$$

$$x_1 + 2x_2 - x_3 = 1$$

$$2x_1 + 4x_2 - 2x_3 = 2$$

$$3x_1 + 6x_2 - 3x_3 = 3$$

Решение приведено на рис. 9.14.

Из рис. 9.14 видно, что первая система уравнений имеет решение: $x_1 = 2 - 2x_3$, $x_2 = -1/2 + 3/2 x_3$, т. е. имеет бесконечное число решений (при любом значении x_3). Система совместна, но не определена.

Вторая система не имеет ни одного решения — она несовместна.

Отметим, что в обоих случаях главный определитель системы равен нулю.

```

Solve[{x1 + 2 x2 - x3 == 1, 2 x1 + 4 x2 - 2 x3 == 2, 3 x1 + 2 x2 + 3 x3 == 5}, {x1, x2, x3}]

{{x1 -> 2 - 2 x3, x2 -> -1/2 + 3 x3/2}}

Solve[{x1 + 2 x2 - x3 == 1, 2 x1 + 4 x2 - 2 x3 == 2, 3 x1 + 6 x2 - 3 x3}, {x1, x2, x3}]

Solve[{x1 + 2 x2 - x3 == 1, 2 x1 + 4 x2 - 2 x3 == 2, 3 x1 + 6 x2 - 3 x3}, {x1, x2, x3}]

Det[{{1, 2, -1}, {2, 4, -2}, {3, 2, 3}}]

0

Det[{{1, 2, -1}, {2, 4, -2}, {3, 6, -3}}]

0

```

Рис. 9.14. Решение систем уравнений примера 9.13

9.3. Примеры для самостоятельного решения систем уравнений

Приведенные далее системы уравнений для самостоятельного решения позволят учащемуся более глубоко усвоить методы решения алгебраических уравнений. Их также можно использовать для самоконтроля и оценочного контроля знаний

9.3.1. Варианты систем линейных алгебраических уравнений

Решить приведенные в табл. 9.1 системы уравнений матричным и другими методами, изложенными в настоящей главе. Сравнить результаты этих методов, указав их достоинства и недостатки.

9.3.2. Варианты систем нелинейных алгебраических уравнений

В уравнениях с четными номерами (табл. 9.2), а также № 17 и 19, начальные приближения должны быть определены пользователем. Так как система уравнений второго порядка, то области начальных приближений легко найти графическим способом.

Таблица 9.1

№ п/п	Система уравнений	№ п/п	Система уравнений
1	$1.5x_1 - 0.8x_2 + 4.25x_3 = 5.1$ $1.2x_1 + 7.18x_2 - 3.2x_3 = 4.2$ $0.5x_1 - 1.5x_2 + 7.1x_3 = -1.2$	2	$6.7x_1 - 0.6x_2 + 0.83x_3 = 6.8$ $0.8x_1 + 1.1x_2 + 7.2x_3 = 5.2$ $1.2x_1 + 5.4x_2 - 0.54x_3 = -3.2$
3	$-1.32x_1 + 2.15x_2 + 7.6x_3 = -1.4$ $2.62x_1 + 6.1x_2 - 4.12x_3 = 5.6$ $8.3x_1 - 2.84x_2 - 1.5x_3 = -6.5$	4	$0.51x_1 - 10x_2 - 3.62x_3 = -2.05$ $3.09x_1 + 1.23x_2 - 4.64x_3 = -5.6$ $3.2x_1 - 2.31x_2 - 8.4x_3 = 6.1$
5	$7.12x_1 - 6.66x_2 + 2.6x_3 = -3.1$ $-1.7x_1 + 6.5x_2 - 0.87x_3 = 2.85$ $0.65x_1 + 0.87x_2 - 8.7x_3 = 5.56$	6	$6.4x_1 - 0.73x_2 + 2.1x_3 = 3.8$ $-1.07x_1 + 3.8x_2 - 1.5x_3 = -1.2$ $2.7x_1 - 3.1x_2 + 4.2x_3 = -7.5$
7	$9.21x_1 - 1.84x_2 + 0.7x_3 = -3.2$ $-6.17x_1 + 8.5x_2 - 2.87x_3 = -3.75$ $0.7x_1 + 0.87x_2 - 8.7x_3 = 2.64$	8	$4.3x_1 - 1.2x_2 + 10.3x_3 = 4.2$ $0.21x_1 + 6.2x_2 + 3.54x_3 = 5.1$ $-0.31x_1 - 0.52x_2 + 3.6x_3 = -2.1$
9	$6.9x_1 + 2.3x_2 + 1.21x_3 = 3.1$ $x_1 + 2.3x_2 - 3.4x_3 = -2.3$ $0.21x_1 - 0.43x_2 + 6.3x_3 = 3.6$	10	$12.4x_1 - 0.56x_2 + 4.2x_3 = 6.3$ $-0.65x_1 + 4.4x_2 + 1.5x_3 = 1.5$ $1.5x_1 + 2.1x_2 - 2.8x_3 = 1.7$

Таблица 9.1 (окончание)

№ п/п	Система уравнений	№ п/п	Система уравнений
11	$1.2x_1 - 1.06x_2 - 6.7x_3 = 2.12$ $4.2x_1 - 6.3x_2 - 0.9x_3 = -1.1$ $0.6x_1 + 6.8x_2 + 0.82x_3 = 0.83$	12	$9.7x_1 + 0.35x_2 - 1.84x_3 = 2.15$ $4.64x_1 - 7.1x_2 - 4.3x_3 = 1.5$ $0.32x_1 + 0.348x_2 - 3.3x_3 = -3.1$
13	$6.5x_1 - 2.34x_2 + 1.4x_3 = 2.8$ $0.5x_1 + 7.3x_2 - 2.4x_3 = -3.8$ $8.6x_1 + 0.34x_2 - 6.4x_3 = 0.64$	14	$2.8x_1 + 4.3x_2 - 3.7x_3 = 5.1$ $-0.45x_1 - 8.24x_2 + 4.8x_3 = 5.4$ $0.54x_1 + 2.3x_2 + 3.7x_3 = 1.54$
15	$6x_1 + 0.13x_2 - 0.67x_3 = 1.9$ $3.8x_1 + 1.25x_2 - 4.3x_3 = 6.4$ $0.38x_1 - 0.64x_2 + 3.2x_3 = 5.4$	16	$1.5x_1 - 2.6x_2 + 7x_3 = -11.2$ $6.6x_1 + 1.3x_2 - 1.24x_3 = 5.3$ $0.85x_1 - 8.4x_2 + 4.7x_3 = 1.6$
17	$6.2x_1 - 0.52x_2 + 2.3x_3 = -1.8$ $-4.2x_1 + 3.4x_2 - 0.5x_3 = 0.7$ $0.2x_1 + 0.8x_2 + 3.6x_3 = 3.2$	18	$0.63x_1 - 0.54x_2 + 1.7x_3 = 3.6$ $0.65x_1 + 4.4x_2 + 0.15x_3 = 2.3$ $1.5x_1 + 0.2x_2 + 4.1x_3 = 2.8$
19	$8.4x_1 - 0.25x_2 + 3.1x_3 = -5.7$ $-0.3x_1 + 6.1x_2 - 1.54x_3 = 3.3$ $-6.8x_1 + 1.2x_2 - 7x_3 = 4.5$	20	$12x_1 + 4.2x_2 - 0.8x_3 = -5.4$ $-4.1x_1 + 2.2x_2 - 0.16x_3 = 1.6$ $-1.6x_1 - 4.3x_2 + 8.4x_3 = 12.2$

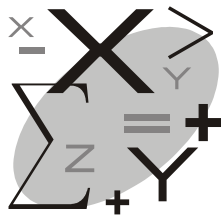
Таблица 9.2

№ п/п	Система уравнений	Начальные приближения	№ п/п	Система уравнений
1	$\sin(x_1 + x_2) - 1.2x_1 = 0.1$ $x_1^2 + x_2^2 = 1$	$x_1 = 0.74$ $x_2 = 0.67$	2	$\sin(x_2 + 1) - x_1 = 1.2$ $2x_2 + \cos x_1 = 2$
3	$\operatorname{tg}(x_1 x_2 + 0.2) = x_1^2$ $0.6x_1^2 + 2x_2^2 = 1$	$x_1 = 0.88$ $x_2 = 0.52$	4	$\cos(x_2 - 1) + x_1 = 0.5$ $x_2 - \cos x_1 = 3$
5	$\sin x_1 + 2 \sin x_2 = 1$ $2 \sin 3x_1 + 3 \sin 3x_2 = 0.3$	$x_1 = 1.08$ $x_2 = 0.06$	6	$x_1^2 x_2 - x_2 - 9 = 0$ $x_1 x_2 - x_1^2 + 10 = 0$
7	$\operatorname{tg}(x_1 - x_2) - 4x_1 = 0$ $x_1^2 + 2x_2^2 = 1$	$x_1 = -0.5$ $x_2 = 0.6$	8	$x_1^2 + x_2^2 - x_1 x_2 = 0$ $\sin(x_1 + x_2) - 2.4x + 3.2 = 0$
9	$x_1^4 + x_2^2 - 3 = 0$ $x_1^3 + x_2^3 - 4 = 0$	$x_1 = 0.95$ $x_2 = 1.4$	10	$2x_1 x_2^2 - 4x_2 - 7.5 = 0$ $x_1^2 - 3x_1 x_2 + 4.5 = 0$
11	$\sin x_1 - x_2 = 1.3$ $\cos x_2 - x_1 = -0.82$	$x_1 = 1.8$ $x_2 = -0.35$	12	$\sin x + 2 \cos x - 0.8 = 0$ $x_1 x_2^2 + 3x_1 - 4.5 = 0$
13	$x_1^3 + x_2^3 - 6x_1 + 3 = 0$ $x_1^3 - x_2^3 - 6x_2 = 2$	$x_1 = 0.52$ $x_2 = -0.37$	14	$x_1^2 x_2^2 - x_2^2 - 18.75 = 0$ $x_1 + x_2^2 - 8.25 = 0$

Таблица 9.2 (окончание)

№ п/п	Система уравнений	Начальные приближения	№ п/п	Система уравнений
15	$x_2 + e^{x_1 - x_2} = 0$ $x_1 + e^{x_1 + x_2} = 0$	$x_1 = -0.7$ $x_2 = -0.35$	16	$\operatorname{tg}(x_1 - x_2) - 4x_1 = 0$ $x_1^2 + 3x_2^3 - 4 = 0$
17	$x_1^2 x_2 - 8x_1 + 5.5 = 0$ $x_1 x_2 + 3x_2 - 10 = 0$	—	18	$x_1^2 \sin x_2 + x_2^2 \sin x_1 + 1 = 0$ $2x_1 + e^{(x_1 + x_2)} - 4 = 0$
19	$e^{x_1} + 2x_2 \ln x_1 - 3 = 0$ $x_1^2 x_2 - 3x_1 + 5.4 = 0$	—	20	$\sin x_1 + 3.5 \sin x_2 - 1 = 0$ $2 \sin 3x + 3 \sin 2x_2 - 0.4 = 0$

ГЛАВА 10



Решение дифференциальных уравнений

10.1. Методические замечания

В настоящей главе излагаются лишь компьютерные технологии решения дифференциальных уравнений в системе Mathematica.

Перед изучением компьютерных технологий весьма полезно вспомнить методы и алгоритмы решения дифференциальных уравнений, обратив внимание на следующие методы:

☐ аналитические:

- метод последовательного дифференцирования;
- метод неопределенных коэффициентов;
- метод последовательных приближений;

☐ численные:

- Эйлера;
- усовершенствованный метод Эйлера;
- Эйлера—Коши с итерационной обработкой результатов;
- Рунге—Кутты.

Эти методы достаточно подробно изложены в [16, 17, 18].

10.2. Решение дифференциальных уравнений в среде Mathematica

Система Mathematica позволяет решать в аналитическом и численном виде линейные и нелинейные дифференциальные уравнения и системы. Решение можно получить в общем и частном виде.

Решение дифференциальных уравнений и систем осуществляется с помощью встроенных функций. Система также имеет внешнюю функцию `RungeKutta` из пакета расширения `startup`, реализующую один из наиболее эффективных численных методов — метод Рунге—Кутты.

В данной главе приводятся встроенные функции системы, технологии их реализации, а также примеры решения дифференциальных уравнений и систем различного вида.

10.2.1. Аналитические методы

Аналитические методы решения дифференциальных уравнений в системе `Mathematica` реализуются с помощью следующих двух встроенных функций:

```
DSolve[f, y[x], x]
```

```
DSolve[{f1, f2, ...}, {y[x1], y[x2], ...}, {x1, x2, ...}]
```

Рассмотрим подробно эти функции и приведем примеры.

Функция `DSolve[f, y[x], x]` предназначена для решения дифференциального уравнения f относительно функции $y(x)$ с аргументом x . Функция дает общее решение уравнения с постоянными интегрирования, которые обозначаются `c[i]`.

Дифференциальное уравнение представляется в произвольном виде. Например, $y' = 2x^2 - 1$, $y' - 2x^2 = -1$ или $y' - 2x^2 + 1 = 0$. Функция позволяет решать дифференциальное уравнение любого порядка.

Компьютерная технология решения дифференциального уравнения:

1. Ввод уравнения f с присвоением ему уникального имени, например f .
2. Ввод встроенной функции `DSolve[f, y[x], x]`.
3. Получение решения нажатием комбинации клавиш `<Shift>+<Enter>`.

Можно уравнение f отдельно не вводить, а вписать его в функцию `DSolve`.

Приведем примеры.

Пример 10.1

Даны следующие дифференциальные уравнения:

$$y' = 2x^2 + 3x - 1,$$

$$y' = 2 \ln x + x - 2,$$

$$y' = 3e^{-2x} - x^2 + x - 1.$$

Необходимо определить $f(x)$.

Решение уравнений по описанной выше технологии приведено на рис. 10.1.

```

F = y' [x] == 2 x^2 + 3 x - 1
y' [x] == -1 + 3 x + 2 x^2
DSolve[F, y[x], x]
 $\left\{ \left\{ y[x] \rightarrow -x + \frac{3 x^2}{2} + \frac{2 x^3}{3} + C[1] \right\} \right\}$ 

DSolve[y' [x] == 2 Log[x] + x - 2, y[x], x]
 $\left\{ \left\{ y[x] \rightarrow -4 x + \frac{x^2}{2} + C[1] + 2 x \text{Log}[x] \right\} \right\}$ 

DSolve[y' [x] - 3 E^(-2 x) + x^2 - x + 1 == 0, y[x], x]
 $\left\{ \left\{ y[x] \rightarrow -\frac{3}{2} e^{-2 x} - x + \frac{x^2}{2} - \frac{x^3}{3} + C[1] \right\} \right\}$ 

```

Рис. 10.1. Решение уравнений примера 10.1

Из рис. 10.1 видно, что при решении первого уравнения ему присвоено имя `F` и осуществлен ввод вне функции `DSolve`. При решении второго уравнения последнее введено непосредственно в функцию `DSolve`. При решении третьего уравнения оно представлено в виде, отличном от первых двух.

Из примера видно, что программа нашла общее решение с произвольными постоянными, а решение сведено к простому интегрированию правой части уравнений.

Функция `DSolve` позволяет решать дифференциальные уравнения высокого порядка.

Пример 10.2

Необходимо решить следующие уравнения:

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

$$y'''(x) = y'(x) - 5y(x) + x^2 - 1.$$

Решение приведено на рис. 10.2.

Из примера 10.2 видно, что программа легко нашла решение в случае, когда неизвестной является функция, входящая в уравнение первой и третьей производной. При этом решение получено не в явном виде. Для его преобразования в явное применена функция `N[%]`.

Частное решение дифференциального уравнения

При решении практических задач задаются начальными условиями. Тогда постоянные интегрирования вычисляются программой, и решение не имеет

```
DSolve[y'''[x] == 3 x^2 - 2 x + 1, y[x], x]
```

$$\left\{ \left\{ y[x] \rightarrow \frac{x^3}{6} - \frac{x^4}{12} + \frac{x^5}{20} + C[1] + x C[2] + x^2 C[3] \right\} \right\}$$

```
F1 = y'''[x] == y'[x] - 5 y[x] + x^2 - 1
```

$$y^{(3)}[x] == -1 + x^2 - 5 y[x] + y'[x]$$

```
DSolve[F1, y[x], x]
```

$$\left\{ \left\{ y[x] \rightarrow \frac{1}{125} (-23 + 10 x + 25 x^2) + e^{\sqrt[5]{-5-5 i+1^3} x} C[1] + e^{\sqrt[5]{-5-5 i+1^3} x} C[2] + e^{\sqrt[5]{-5-5 i+1^3} x} C[3] \right\} \right\}$$

```
N[%]
```

$$\left\{ \left\{ y[x] \rightarrow 0.008 (-23. + 10. x + 25. x^2) + 2.718281828459045^{-1.9041608591349206} x C[1] + 2.718281828459045^{(0.9520804295674603 - 1.3112480440771224 i) x} C[2] + 2.718281828459045^{(0.9520804295674603 + 1.3112480440771224 i) x} C[3] \right\} \right\}$$

Рис. 10.2. Решение уравнений примера 10.2

коэффициентов $C[i]$. Для получения частного решения используется функция `DSolve`, которая представляется теперь в следующем виде:

```
DSolve[f(x, x0), y[x], x]
```

где:

- ☐ $f(x, x_0)$ — дифференциальное уравнение в совокупности с начальными условиями;
- ☐ $y(x)$ — искомая функция;
- ☐ x — независимая переменная.

Например, если дифференциальное уравнение имеет вид: $y''(x) = xy'(x) + 2.5y(x) - 3.5$, а начальными условиями являются $y(0)=1$, $y'(0)=0$, то $f(x, x_0) = \{y'[x] == x y'[x] + 2.5 y[x] - 3.5, y[0] == 1, y'(0) == 0\}$

Рассмотрим примеры частного решения дифференциальных уравнений с помощью функции `DSolve`.

Пример 10.3

Необходимо решить следующие уравнения:

- ☐ $y'(x) = 2 \ln x + x - 2$ при начальном условии: $y(0) = 1$;
- ☐ $y''(x) = 3y(x) - e^{2x} + x - 1$ при $y(0) = 1$, $y'(0) = 0$;
- ☐ $y'''(x) = -5y(x) - 1$ при $y(1) = y'(1) = y''(1) = 0$.

Решение приведено на рис. 10.3.


```
DSolve[{y'[x]==2 Log[x]+x-2,y[0]==1},y[x],x]
```

$$\left\{\left\{y[x] \rightarrow \frac{1}{2} (2 - 8x + x^2 + 4x \log[x])\right\}\right\}$$

```
DSolve[{y'[x]==3 y[x]-E^(2 x)+x-1,y[0]==1,y'[0]==0},y[x],x]
```

$$\left\{\left\{y[x] \rightarrow -\left(e^{-\sqrt{3}x} \left(15 - 7\sqrt{3} + 6e^{\sqrt{3}x} + 15e^{2\sqrt{3}x} + 7\sqrt{3}e^{2\sqrt{3}x} - 18e^{2x+\sqrt{3}x} - 6e^{\sqrt{3}x}x\right)\right) / \left(18(-2+\sqrt{3})(2+\sqrt{3})\right)\right\}\right\}$$

```
Simplify[%]
```

$$\left\{\left\{y[x] \rightarrow \frac{1}{18} e^{-\sqrt{3}x} \left(15 - 7\sqrt{3} + (15 + 7\sqrt{3})e^{2\sqrt{3}x} - 18e^{(2+\sqrt{3})x} - 6e^{\sqrt{3}x}(-1+x)\right)\right\}\right\}$$

```
DSolve[{y''[x]==-5 y[x]-1,y[1]==0,y'[1]==0,y''[1]==0},y[x],x]
```

$$\left\{\left\{y[x] \rightarrow \left(e^{-\frac{5^{1/3}}{2} - \frac{5^{1/3}}{2}x} \left(e^{\frac{3}{2} \frac{5^{1/3}}{2}} \cos\left[\frac{1}{2} \sqrt{3} 5^{1/3}\right]^2 - 3e^{\frac{5^{1/3}}{2} + \frac{5^{1/3}}{2}x} \cos\left[\frac{1}{2} \sqrt{3} 5^{1/3}\right]^2 + 2e^{\frac{3}{2} \frac{5^{1/3}}{2}x} \cos\left[\frac{1}{2} \sqrt{3} 5^{1/3}\right] \cos\left[\frac{1}{2} \sqrt{3} 5^{1/3}x\right] + e^{\frac{3}{2} \frac{5^{1/3}}{2}x} \sin\left[\frac{1}{2} \sqrt{3} 5^{1/3}\right]^2 - 3e^{\frac{5^{1/3}}{2} + \frac{5^{1/3}}{2}x} \sin\left[\frac{1}{2} \sqrt{3} 5^{1/3}\right]^2 + 2e^{\frac{3}{2} \frac{5^{1/3}}{2}x} \sin\left[\frac{1}{2} \sqrt{3} 5^{1/3}\right] \sin\left[\frac{1}{2} \sqrt{3} 5^{1/3}x\right]\right)\right) / \left(15 \left(\cos\left[\frac{1}{2} \sqrt{3} 5^{1/3}\right]^2 + \sin\left[\frac{1}{2} \sqrt{3} 5^{1/3}\right]^2\right)\right)\right\}\right\}$$

Рис. 10.3. Решение уравнений примера 10.3

Решение систем дифференциальных уравнений в аналитическом виде

Системы дифференциальных уравнений в аналитическом виде решаются также с помощью встроенной функции `DSolve`, которая в данном случае имеет вид:

```
DSolve[{f1, f2, ...}, {y1(x), y2(x), ...}, x]
```

где:

- f_i — i -е уравнение системы;
- $y_i[x]$ — i -е искоемое неизвестное;
- x — независимая переменная.

Приведем пример записи функции `DSolve` для случая решения систем дифференциальных уравнений. Пусть необходимо решить следующую систему уравнений:

$$\begin{aligned}x'(t) &= y(t) + z(t) - 1 \\y'(t) &= 2x(t) - 3z(t) + 2 \\z'(t) &= x(t) + y(t) + z(t).\end{aligned}$$

Тогда функция DSolve будет иметь вид:

```
DSolve[{x'[t]==y[t]+z[t]-1, y'[t]==2 x[t]-3 z[t]+2, x[t]+y[t]+z[t]},
{x[t], y[t], z[t]}, t]
```

Технология решения систем дифференциальных уравнений практически не отличается от технологии решения одиночных уравнений.

Приведем примеры решения системы уравнений по этой технологии.

Пример 10.4

Пусть необходимо решить следующую систему дифференциальных уравнений:

$$\begin{aligned}x'(t) &= y(t) + z(t) \\y'(t) &= x(t) + 3z(t) \\z'(t) &= x(t) + y(t).\end{aligned}$$

Решение приведено на рис. 10.4. Чтобы не повторяться в дальнейшем, при получении частного решения вектору уравнений присвоено имя f.

```
f = {x'[t] == y[t] + z[t], y'[t] == x[t] + 3 z[t], z'[t] == x[t] + y[t]}
{x'[t] == y[t] + z[t], y'[t] == x[t] + 3 z[t], z'[t] == x[t] + y[t]}
DSolve[f, {x[t], y[t], z[t]}, t]
{{x[t] -> 1/3 e^-t (2 + e^3 t) C[1] + 1/3 e^-t (-1 + e^3 t) C[2] + 1/3 e^-t (-1 + e^3 t) C[3],
 y[t] -> 1/3 e^-t (-1 + e^3 t) C[1] + 1/3 e^-t (2 + e^3 t) C[2] + 1/3 e^-t (-1 + e^3 t) C[3],
 z[t] -> 1/3 e^-t (-1 + e^3 t) C[1] + 1/3 e^-t (-1 + e^3 t) C[2] + 1/3 e^-t (2 + e^3 t) C[3]}}
```

Рис. 10.4. Решение задачи примера 10.4

Из рис. 10.4 видно, что решение получено в общем виде, т. к. начальные условия не были заданы.

Получим частное решение при следующих начальных условиях: $x(0)=1$, $y(0)=z(0)=0$. В этом случае функция DSolve представится так:

```
DSolve[{x'[t]==y[t]+z[t], y'[t]==x[t]+3 z[t], z'[t]==x[t]+y[t], x[0]==1,
y[0]==z[0]==0}, {x[t], y[t], z[t]}, t]
```

а решение на экране будет иметь вид:

```
DSolve[{f, x[0] == 1, y[0] == z[0] == 0}, {x[t], y[t], z[t]}, t]
{{x[t] -> 1/3 a^-t (2 + a^3 t), y[t] -> 1/3 a^-t (-1 + a^3 t), z[t] -> 1/3 a^-t (-1 + a^3 t)}}
```

Достоверность решения легко проверить методом подстановки полученных значений $x(t)$, $y(t)$, $z(t)$ в исходные уравнения системы.

Решим теперь практическую задачу.

Пусть система массового обслуживания (СМО) представляет собой объект с одним обслуживающим органом. Это может быть железнодорожная касса, банкомат, продовольственный ларек бизнесмена и т. п. Поток заявок на обслуживание (пассажиров, получателей жалования, покупателей продуктов питания) характеризуется интенсивностью потока λ , а обслуживание — интенсивностью μ . Интенсивности λ и μ имеют размерность 1/час.

Время поступления заявки на обслуживание и время обслуживания являются величинами случайными. Требуется определить вероятности состояний системы: $p_0(t)$ — вероятность того, что в момент времени t заявок на обслуживание в системе нет, $p_1(t)$ — вероятность того, что в момент времени t СМО занята обслуживанием заявки. Практическая важность решения очевидна.

Из теории массового обслуживания известно, что в данном случае функционирование СМО можно описать следующей системой дифференциальных уравнений:

$$\begin{aligned}\frac{dp_0(t)}{dt} &= -\lambda p_0(t) + \mu p_1(t), \\ \frac{dp_1(t)}{dt} &= \lambda p_0(t) - \mu p_1(t).\end{aligned}$$

Будем решать эту систему дифференциальных уравнений при следующих начальных условиях: $p_0(0) = 1$, $p_1(0) = 0$.

В нашем случае функция **DSolve** имеет вид:

```
DSolve[{p0'[t]==-λp0[t]+μp1[t], p1'[t]== λp0[t]-μp1[t], p0[0]==1, p1[0]==0},{p0[t], p1[t]}, t]
```

Решение приведено на рис. 10.5. Для удобства ввода и чтения λ , μ и t обозначены соответственно a , m и x . Вначале получено общее, а затем частное решение при установленных нами начальных условиях. Проверена достоверность решения и выполнен небольшой анализ результатов.

Из рис. 10.5 видно, что выражения для вероятностей получены в аналитическом виде, что позволяет получить их численные значения для любого t при произвольных значениях интенсивности потока заявок и интенсивности обслуживания.

```

DSolve[{p0'[x] == -a p0[x] + m p1[x],
  p1'[x] == a p0[x] - m p1[x]}, {p0[x], p1[x]}, x]
{{p0[x] -> (a e^(-a-m) x + m) C[1] - (-1 + e^(-a-m) x) m C[2],
  p1[x] -> - (a (-1 + e^(-a-m) x) C[1] + (a + e^(-a-m) x) m C[2]) / (a + m)}}
DSolve[{p0'[x] == -a p0[x] + m p1[x], p1'[x] == a p0[x] - m p1[x],
  p0[0] == 1, p1[0] == 0}, {p0[x], p1[x]}, x]
{{p0[x] -> (a e^(-a-m) x + m) / (a + m), p1[x] -> - (a (-1 + e^(-a-m) x)) / (a + m)}}
x := 0
Out[44]
{{p0[0] -> 1, p1[0] -> 0}}
x := Infinity
Out[44]
{{p0[Infinity] -> (a e^(-a-m) Infinity + m) / (a + m), p1[Infinity] -> - (a (-1 + e^(-a-m) Infinity)) / (a + m)}}
a := 2; m := 5; x := Infinity
Out[55]
{{p0[Infinity] -> 5/7, p1[Infinity] -> 2/7}}
Limit[Out[55]]
{{Limit[p0[Infinity] -> 5/7], Limit[p1[Infinity] -> 2/7]}}

```

Рис. 10.5. Решение задачи СМО

Дополнительно вычислены значения p_0 и p_1 при $x=0$ и $x=\infty$. Из рисунка видно, что при $x=0$ (присвоение $x:=0$) $p_0(0)=1$, $p_1(0)=0$. Подстановка осуществлена в аналитическое выражение, которое находится в строке Out[44] в виде ответа. Установившиеся значения $p_0(\infty)$ и $p_1(\infty)$ таким способом получить не удалось. Это объясняется тем, что значения символьных переменных a и m системе неизвестны, а при $a+m < 0$ решения нет ($p_0(\infty)=p_1(\infty)=\infty$). При $a > 0$ и $m > 0$ решение существует. Если, например, $a=2$, $m=5$, то при $t=\infty$ решение будет получено как путем подстановки, так и путем вычисления пределов выражений $p_0(t)$ и $p_1(t)$ при $t \rightarrow \infty$.

Опции функции DSolve

Функция DSolve имеет две следующие опции:

- ☐ MaxSteps — максимальное число шагов интегрирования;
- ☐ DSolveConstants — определяет постоянные интегрирования.

При практическом решении дифференциальных уравнений этими опциями в большинстве случаев пользоваться не приходится.

10.2.2. Численные методы решения дифференциальных уравнений

Численные методы решения дифференциальных уравнений в системе Mathematica реализуются с помощью следующих двух встроенных функций:

`NDSolve[f, y[x], {x, xmin, xmax}]`

`NDSolve[{f1, f2, ..., y1(x0), y2(x0), ...},
{y1[x], y2[x], ...}, {x, xmin, xmax}]`

где:

- ☐ f — дифференциальное уравнение и начальные условия;
- ☐ f_i — i -е уравнение системы дифференциальных уравнений;
- ☐ $y[x]$ — искомая функция;
- ☐ $y_i[x]$ — i -я искомая функция системы дифференциальных уравнений;
- ☐ $y_i(x_0)$ — i -е начальное условие;
- ☐ x_{min}, x_{max} — минимальное и максимальное значения независимой переменной;
- ☐ x — аргумент искомой функции.

Функции численного решения дифференциальных уравнений и систем имеют опцию `StartingStepSize`, которая определяет величину начального шага интегрирования.

Численные методы наиболее часто применяются в тех случаях, когда уравнение в аналитическом виде системой не решается или вовсе не имеет аналитического решения. Такими являются большинство нелинейных уравнений.

Далее подробно излагаются встроенные функции, способы их реализации и приводятся примеры.

Функция `NDSolve[f, y[x], {x, xmin, xmax}]`

Эта функция решает дифференциальное уравнение n -го порядка, вычисляя искомую функцию $y(x)$ в диапазоне независимой переменной x от x_{min} до x_{max} . Решение можно получить в виде таблицы или графика.

Технологию решения задачи покажем на примере.

Пример 10.5

Пусть необходимо решить следующее дифференциальное уравнение:

$$y'(x) - xy(x) = 1$$

при начальных условиях $y(0)=1$. Решение следует получить в табличном и графическом виде в диапазоне x от 0 до 5 с шагом 0.5.

Технология решения дифференциального уравнения с помощью функции `NDSolve` состоит из выполнения следующих операций:

1. Ввод функции `NDSolve`, которая в нашем примере имеет вид:
2. Получение решения путем одновременного нажатия клавиш `<Shift>+<Enter>`. Решение будет получено в виде сообщения без вывода самого решения на экран.
3. Ввод функции `Table` для получения решения в табличной форме. В нашем примере эта функция будет иметь вид:

```
Table[{x, y[x]/.Out[8]}, {x, 0, 5, 0.5}]
```

Здесь `Out[8]` — это строка 8, в которой находится решение уравнения. Откликом будет вектор, представленный в виде строки.

4. Ввод функции `TableForm[%]` для получения решения в форме таблицы. Откликом будет функция $y(x)$, представленная в виде таблицы.
5. Ввод функции `Plot[y[x]/.Out[8], {x, 0, 5}]`. Откликом будет решение в виде графика.

Решение задачи приведено на рис. 10.6. Вначале приведено решение уравнения в аналитическом виде.

Рассматриваемая функция позволяет решать дифференциальные уравнения высокого порядка. При этом технология решения остается прежней. Несколько иной будут только функция `NDSolve`.

Пример 10.6

Необходимо решить следующее дифференциальное уравнение:

$$y'''(x) - 3x^2 y''(x) - 2xy'(x) - y(x) = 1$$

при начальных условиях $y(1)=1$, $y'(1)=y''(1)=0$. Решение получить в диапазоне x от 0 до 3 с шагом 0.5 при представлении решения в табличном виде и в диапазоне от 0 до 2 — в графическом.

Решение приведено на рис. 10.7.

```
DSolve[{y'[x]==x* y[x]+1,y[0]==1},y[x],x]
```

$$\left\{\left\{y[x] \rightarrow \frac{1}{2} e^{\frac{x^2}{2}} \left(2 + \sqrt{2\pi} \operatorname{Erf}\left[\frac{x}{\sqrt{2}}\right]\right)\right\}\right\}$$

```
NSolve[{y'[x]==y[x]+1,y[0]==1},y[x],{x,0,5}]
```

```
{y[x]->InterpolatingFunction[{{0.,5.}},<>][x]}
```

```
Table[{x,y[x]/.Out[8]},{x,0,5,0.5}]
```

```
{{0.,{1.}}, {0.5,{2.29744}}, {1.,{4.43656}}, {1.5,{7.96337}}, {2.,{13.7781}}, {2.5,{23.365}}, {3.,{39.171}}, {3.5,{65.2308}}, {4.,{108.196}}, {4.5,{179.034}}, {5.,{295.826}}}
```

```
TableForm[%]
```

0	1.
0.5	2.29744
1.	4.43656
1.5	7.96337
2.	13.7781
2.5	23.365
3.	39.171
3.5	65.2308
4.	108.196
4.5	179.034
5.	295.826

```
Plot[y[x]/.Out[8],{x,0,5}]
```

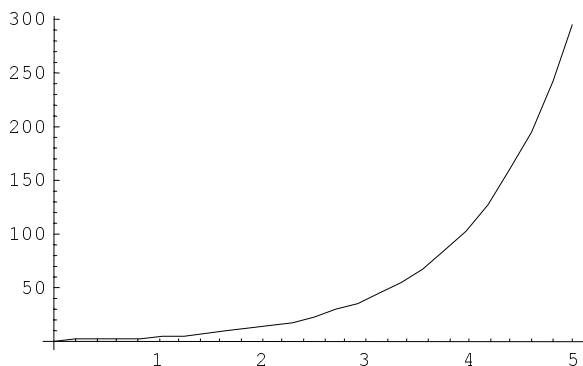


Рис. 10.6. Решение уравнения примера 10.5

```

NDSolve[{y'''[x]==3 x^2 y'[x]+2 x
y'[x]+y[x]+1,y[1]==1,y'[1]==y''[1]==0},y[x],{x,0,5}]
{y[x]→InterpolatingFunction[{{0.,5.}},<>][x]}
Table[{x,y[x]/.Out[1]},{x,0,3,0.5}]
{{0.,{0.7363584435178943`}}, {0.5.,{0.9662626674117316`}},
{1.,{1.`}}, {1.5.,{1.0834912468759925`}},
{2.,{6.94346300495399`}}, {2.5.,{5233.080815734498`}},
{3.,{2.368089715606595`*^8}}]
TableForm[%]
0      0.736358
0.5    0.966263
1.      1.
1.5    1.08349
2.      6.94346
2.5    5233.08
3.      2.36809×108
Plot[{y[x]/.Out[1]},{x,0,2}]

```

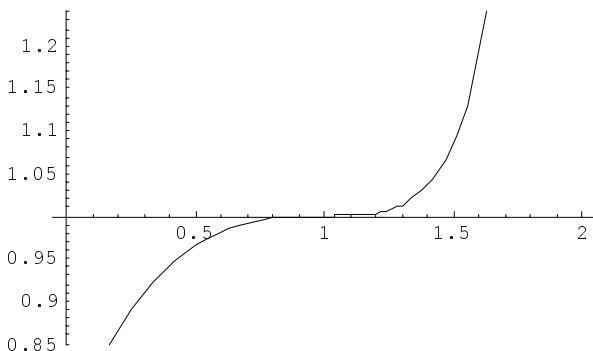


Рис. 10.7. Решение примера 10.6

Функция $NDSolve\{f_1, f_2, \dots, y_1(x_0), y_2(x_0), \dots, \{y_1[x], y_2[x], \dots\}, \{x, x_{min}, x_{max}\}\}$

Эта функция решает систему дифференциальных уравнений n -го порядка, вычисляя искомые функции $y_1(x)$, $y_2(x)$, ... в диапазоне независимой переменной x от x_{min} до x_{max} . Решение можно получить в виде таблицы или графиков.

Технологию решения задачи покажем на примерах.

Пример 10.7

Функционирование двухканальной системы массового обслуживания с отказами описывается следующей системой дифференциальных уравнений:

$$\begin{aligned}p_0'(t) &= -0.9p_0(t) + 2p_1(t) \\p_1'(t) &= 0.9p_0(t) - 2.9p_1(t) + 4p_2(t) \\p_2'(t) &= 0.9p_1(t) - 4p_2(t).\end{aligned}$$

Необходимо определить вероятности состояний системы $p_0(t)$, $p_1(t)$, $p_2(t)$ в течение 100 часов ее функционирования при следующих начальных условиях ее работы: $p_0(0)=1$, $p_1(0)=p_2(0)=0$. Решение получить в виде таблиц и графиков.

В данном случае функция `NDSolve` будет иметь вид:

```
NDSolve[{p0'[t]==-0.9 p0[t]+2p1[t], p1'[t]==0.9 p0[t]-2.9p1[t]+4p2[t],
p2'[t]==0.9p1[t]-4p2[t], p0[0]==1, p1[0]==p2[0]==0}, {p0[t], p1[t], p2[t]},
{t, 0, 100}]
```

Решение приведено на рис. 10.8. Таблица представлена в диапазоне t от 0 до 1 с шагом 0.1, а график — в диапазоне t от 0 до 3.

Откликом функции `NDSolve[{f1, f2, ..., y1[x0], y2[x0], ...}, {y1[x], y2[x], ...}, {x, xmin, xmax}]` в табличном и графическом виде может быть любая комбинация функций $y_i[x]$. Приведем пример.

Пример 10.8

Необходимо решить следующую систему дифференциальных уравнений:

$$\begin{aligned}x'(t) &= y(t)z(t)+1 \\y'(t) &= x(t)z(t)-1 \\z'(t) &= x(t)y(t)+1\end{aligned}$$

при начальных условиях $x(0)=1$, $y(0)=z(0)=0$.

Решение представить для функции $x(t)$ и совокупности функций $x(t)$, $y(t)$, $z(t)$ в виде таблицы в диапазоне t от 0 до 10 с шагом 1 и в виде графика в диапазоне t от 0 до 4.

Решение приведено на рис. 10.9.

Имеют место случаи, когда нелинейное уравнение высокого порядка не решается. В таких случаях иногда удается получить ответ, если преобразовать уравнение высокого порядка в систему уравнений и попытаться ее решить. Это преобразование можно осуществить методом введения новых переменных. Приведем пример.

Пример 10.9

Пусть необходимо решить следующее уравнение:

$$y'''(t) = y''(t) + 3y'(t) + 3y(t) + 1$$

при начальных условиях $y(0)=1$, $y'(0)=y''(0)=0$.

```

NDSolve[{p0'[t]==-0.9 p0[t]+2 p1[t], p1'[t]==0.9
p0[t]-2.9 p1[t]+4 p2[t], p2'[t]==0.9 p1[t]-4
p2[t], p0[0]==1, p1[0]==p2[0]==0}, {p0[t], p1[t], p2[t]},
{t, 0, 100}]

```

```

{{p0[t]→InterpolatingFunction[{{0., 100.}}, <>][t], p
1[t]→InterpolatingFunction[{{0., 100.}}, <>][t], p2[t]
→InterpolatingFunction[{{0., 100.}}, <>][t]}}

```

```
Table[{t, {p0[t], p1[t], p2[t]}/.Out[2]}, {t, 0, 1, 0.1}]
```

```

{{0, {{1., -2.38559×10-21, -6.03826×10-26}}},
{0.1, {{0.921667, 0.0751894, 0.00314313}}},
{0.2, {{0.862132, 0.127986, 0.00988277}}},
{0.3, {{0.81629, 0.166013, 0.0176963}}},
{0.4, {{0.780636, 0.194028, 0.0253356}}},
{0.5, {{0.752693, 0.215063, 0.0322435}}},
{0.6, {{0.730668, 0.231105, 0.038227}}},
{0.7, {{0.713235, 0.243489, 0.0432764}}},
{0.8, {{0.699393, 0.253141, 0.0474669}}},
{0.9, {{0.688377, 0.260716, 0.0509062}}},
{1., {{0.679598, 0.266694, 0.053708}}}}

```

```
TableForm[%]
```

0	1.	-2.38559×10 ⁻²¹	-6.03826×10 ⁻²⁶
0.1	0.921667	0.0751894	0.00314313
0.2	0.862132	0.127986	0.00988277
0.3	0.81629	0.166013	0.0176963
0.4	0.780636	0.194028	0.0253356
0.5	0.752693	0.215063	0.0322435
0.6	0.730668	0.231105	0.038227
0.7	0.713235	0.243489	0.0432764
0.8	0.699393	0.253141	0.0474669
0.9	0.688377	0.260716	0.0509062
1.	0.679598	0.266694	0.053708

```
Plot[{p0[t]/.Out[2]}, {p1[t]/.Out[2]}, {p2[t]/.Out[2]}, {t, 0, 3}]
```

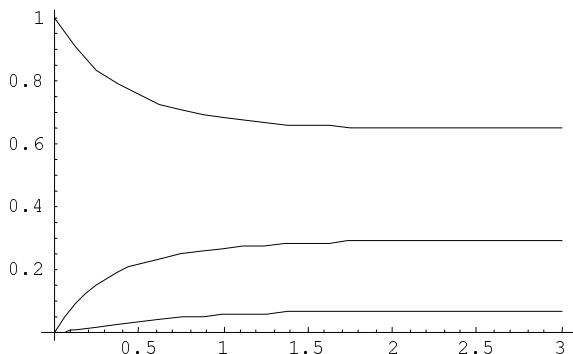


Рис. 10.8. Решение задачи примера 10.7

```

NDSolve[{x'[t]==y[t] z[t]+1,y'[t]==x[t] z[t]-
1,z'[t]==x[t]
y[t]+1,x[0]==1,y[0]==z[0]==0},{x[t],y[t],z[t]},{t,0,10}]

{{x[t]→InterpolatingFunction[{{0.,10.}},<>][t],y[t]
→InterpolatingFunction[{{0.,10.}},<>][t],z[t]→In
terpolatingFunction[{{0.,10.}},<>][t]}}

Table[{t,x[t]/.Out[1]},{t,0,10,1}]

{{0,{1.}}, {1,{1.87415}}, {2,{2.64865}}, {3,{3.51418}}, {4,
{4.44004}}, {5,{5.3943}}, {6,{6.3633}}, {7,{7.34092}}, {8,{
8.32399}}, {9,{9.31074}}, {10,{10.3001}}

TableForm[%]
0      1.
1      1.87415
2      2.64865
3      3.51418
4      4.44004
5      5.3943
6      6.3633
7      7.34092
8      8.32399
9      9.31074
10     10.3001

Table[{t,{x[t]/.Out[1]},{y[t]/.Out[1]},{z[t]/.Out[1]},{t,0,10
,1}]

{{0,{{1.`}},{{-1.9190590211230242`*^-21}},
{1.9190590211230242`*^-21}}, {1,{{1.8741514974656315`}},
{-0.49068522651298985`}}, {{0.49068522651298985`}}},
{2,{{2.6486484555017222`}}, {{-0.42547939415814945`}},
{{0.42547939415814945`}}}, {3,{{3.514178182379078`}},
{{-0.3110725739861868`}}, {{0.3110725739861868`}}},
{4,{{4.440044147411033`}}, {{-0.2379557654813793`}},
{{0.2379557654813793`}}}, {5,{{5.394299483307163`}},
{{-0.19223107394893138`}}, {{0.19223107394893138`}}},
{6,{{6.363304792409019`}}, {{-0.16123653543792826`}},
{{0.16123653543792826`}}}, {7,{{7.340917281618858`}},
{{-0.13884899977940118`}}, {{0.13884899977940118`}}},
{8,{{8.323988759679063`}}, {{-0.12192046649117368`}},
{{0.12192046649117368`}}}, {9,{{9.31073951868693`}},
{{-0.10867122095330521`}}, {{0.10867122095330521`}}},
{10,{{10.30008763937424`}}, {{-0.09801933978192177`}},
{{0.09801933978192177`}}})

```

Рис. 10.9. (Часть 1 из 3) Решение задачи примера 10.8

TableForm[%]

0	1.
1	1.87415
2	2.64865
3	3.51418
4	4.44004
5	5.3943
6	6.3633
7	7.34092
8	8.32399
9	9.31074
10	10.3001

Table[{t,{x[t]/.Out[1]},{y[t]/.Out[1]},{z[t]/.Out[1]},{t,0,10,1}]

```
{0,{{1.}},{{-1.9190590211230242`*^-21}},
{1.9190590211230242`*^-21}}, {1,{{1.8741514974656315`},
{-0.49068522651298985`}}, {{0.49068522651298985`}}},
{2,{{2.6486484555017222`}}, {{-0.42547939415814945`}},
{{0.42547939415814945`}}}, {3,{{3.514178182379078`}},
{{-0.3110725739861868`}}, {{0.3110725739861868`}}},
{4,{{4.440044147411033`}}, {{-0.2379557654813793`}},
{{0.2379557654813793`}}}, {5,{{5.394299483307163`}},
{{-0.19223107394893138`}}, {{0.19223107394893138`}}},
{6,{{6.363304792409019`}}, {{-0.16123653543792826`}},
{{0.16123653543792826`}}}, {7,{{7.340917281618858`}},
{{-0.1388489977940118`}}, {{0.1388489977940118`}}},
{8,{{8.323988759679063`}}, {{-0.12192046649117368`}},
{{0.12192046649117368`}}}, {9,{{9.31073951868693`}},
{{-0.10867122095330521`}}, {{0.10867122095330521`}}},
{10,{{10.30008763937424`}}, {{-0.09801933978192177`}},
{{0.09801933978192177`}}}
```

TableForm[%]

0	1.	-1.91906×10^{-21}	1.91906×10^{-21}
1	1.87415	-0.490685	0.490685
2	2.64865	-0.425479	0.425479
3	3.51418	-0.311073	0.311073
4	4.44004	-0.237956	0.237956
5	5.3943	-0.192231	0.192231
6	6.3633	-0.161237	0.161237
7	7.34092	-0.138849	0.138849
8	8.32399	-0.12192	0.12192
9	9.31074	-0.108671	0.108671
10	10.3001	-0.0980193	0.0980193

Plot[{x[t]/.Out[1]},{y[t]/.Out[1]},{z[t]/.Out[1]},{t,0,4}]

Рис. 10.9. (Часть 2 из 3)

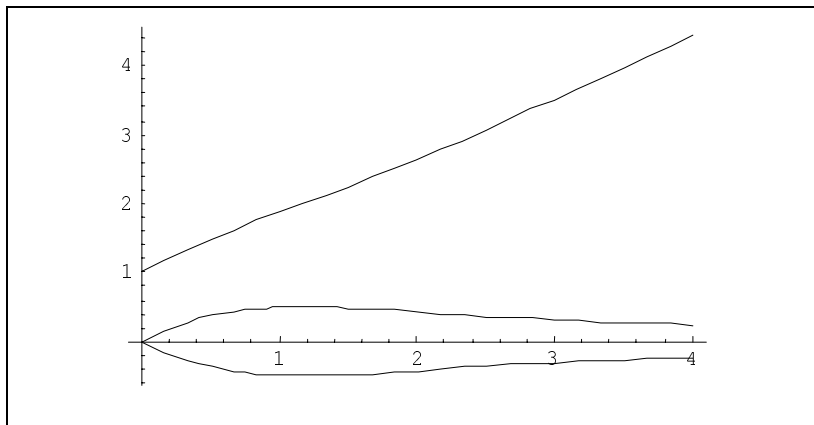


Рис. 10.9. (Часть 3 из 3)

Решение нужно иметь в диапазоне t от 0 до 10. Ответ следует получить в табличной форме в диапазоне t от 0 до 2 с шагом 0.4. График $y(t)$ представить в диапазоне t от 0 до 2.

Преобразуем уравнение в систему уравнений методом введения переменных. Обозначим: $y'(t) = z(t)$, $z'(t) = w(t)$. Тогда система уравнений будет иметь вид:

$$y'(t) = z(t)$$

$$z'(t) = w(t)$$

$$w'(t) = w(t) + 3z(t) + 3y(t) + 1.$$

Решение примера приведено на рис. 10.10.

Из рис. 10.10 видно, что в данном случае решение уравнения имеется, и оно совпадает с решением эквивалентной системы уравнений. Следует иметь в виду, что представление уравнения высокого порядка в виде системы уравнений позволяет получить в табличном и графическом виде функции $z(t)$ и $w(t)$, т. е. производные $y'(t)$, $y''(t)$.

Система Mathematica, как и любая другая система компьютерной алгебры, не является идеальной в отношении решения дифференциальных уравнений. Получаемое решение редко когда совпадает с ответом, имеющимся в математических справочниках. Не редки случаи, когда встроенная функция не дает решения или оно ошибочно, хотя уравнение достаточно простое. Приведем такие примеры.

```

NDSolve[{y'[t]==y'[t]+3 y'[t]+3
y[t]+1,y[0]==1,y'[0]==y'[0]==0},y[t],{t,0,10}]

{{y[t]→InterpolatingFunction[{{0.,10.}},<>][t]}}

Table[{t,y[t]/.Out[17]},{t,0,2,0.4}]

{{0,{1.}},{0.4,{1.04856}},{0.8,{1.47345}},{1.2,{3.10894
}}, {1.6,{8.18663}},{2.,{22.947}}}}

TableForm[%]
0      1.
0.4    1.04856
0.8    1.47345
1.2    3.10894
1.6    8.18663
2.     22.947

NDSolve[{y'[t]==z[t],z'[t]==w[t],w'[t]==w[t]+3 z[t]+3
y[t]+1,y[0]==1,z[0]==w[0]==0},{y[t],z[t],w[t]},{t,0,10}]

{{y[t]→InterpolatingFunction[{{0.,10.}},<>][t],z[t]→Int
erpolatingFunction[{{0.,10.}},<>][t],w[t]→Interpolating
Function[{{0.,10.}},<>][t]}}

Table[{t,y[t]/.Out[20]},{t,0,2,0.4}]

{{0,{1.}},{0.4,{1.04856}},{0.8,{1.47345}},{1.2,{3.1
0894}},{1.6,{8.18662}},{2.,{22.947}}}}

TableForm[%]
0      1.
0.4    1.04856
0.8    1.47345
1.2    3.10894
1.6    8.18662
2.     22.947

Plot[y[t]/.Out[20],{t,0,2}]

```

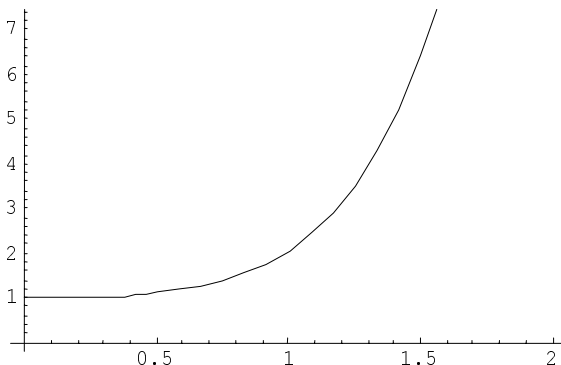


Рис. 10.10. Решение задачи примера 10.9

Пример 10.10

Пусть необходимо решить следующие уравнения и системы уравнений:

$$y''(x) = -y(x) + \operatorname{tg} x;$$

$$y''(x) = -y(x) + 4x \sin x;$$

$$x'(t) + y'(t) - tx(t) = t$$

$$x'(t) + y'(t) + y(t) = t(t+2);$$

$$3z^2(x)z'(x) - 2x = 0.$$

```
DSolve[y' '[x]+y[x]==Tan[x],y[x],x]
```

```
{{y[x] -> -2 ArcTan[Tan[x/2]] Cos[x] + C[1] Cos[x] + C[2] Sin[x]}}
```

```
DSolve[y' '[x]+y[x]==4 x Sin[x],y[x],x]
```

```
{{y[x] -> C[1] Cos[x] + C[2] Sin[x] + 1/2 (-2 x^2 Cos[x] +  
Cos[x] Cos[2 x] - 2 x Cos[2 x] Sin[x] +  
2 x Cos[x] Sin[2 x] + Sin[x] Sin[2 x])}}
```

```
Simplify[%]
```

```
{{y[x] -> (1/2 - x^2 + C[1]) Cos[x] + (x + C[2]) Sin[x]}}
```

```
DSolve[{x'[t]+y'[t]-t x[t]==t,x'[t]+y'[t]+y[t]==t  
(t+2)},
```

```
x[0]==1,y[0]==0},{x[t],y[t]},t]
```

```
DSolve[{-t x[t] + x'[t] + y'[t] == t, y[t] + x'[t] + y'[t] == t (2 + t),
```

```
x[0] == 1, y[0] == 0}, {x[t], y[t]}, t]
```

```
NDSolve[{x'[t]+y'[t]-t x[t]==t,x'[t]+y'[t]+y[t]==t  
(t+2)},
```

```
x[0]==1,y[0]==0},{x[t],y[t]}, {t,-1,1}]
```

```
NDSolve[{-t x[t] + x'[t] + y'[t] == t, y[t] + x'[t] + y'[t] == t (2 + t),
```

```
x[0] == 1, y[0] == 0}, {x[t], y[t]}, {t, -1, 1}]
```

```
DSolve[3 z[x]^2 z'[x]-2 x==0,z[x],x]
```

```
{{z[x] -> (x^2 + 3 C[1])^(1/3)}, {z[x] -> -(-1)^(1/3) (x^2 + 3 C[1])^(1/3)}, {z[x] -> (-1)^(2/3) (x^2 + 3 C[1])^(1/3)}}
```

Рис. 10.11. Решение задач примера 10.10

Систему уравнений решить аналитическим и численным методами при начальных условиях $x(0) = 1$, $y(0) = 0$ в диапазоне x от -1 до 1 .

Решение уравнений приведено на рис. 10.11.

Прокомментируем полученные решения.

Решения первого и второго уравнений являются верными, но не совпадают со справочными данными, в которых приводятся следующие ответы:

$$y(x) = c_1 \cos x + c_2 \sin x - \cos x \ln(\operatorname{tg}(\pi/4 + x/2)),$$

$$y(x) = c_1 \cos x + c_2 \sin x + x(\sin x - x \cos x).$$

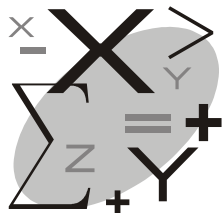
Полученные решения можно существенно упростить с помощью функции `simplify[%]`.

Решение системы уравнений не получено ни аналитическим, ни численным методами, хотя такое решение существует.

Интересным является ответ решения третьего уравнения — три равноценных результата.

При решении дифференциальных уравнений численными методами могут возникать недопустимо большие ошибки за счет методических ошибок и ошибок выбора шага интегрирования. Всегда необходимо помнить, что при компьютерных технологиях решения дифференциальных уравнений необходима проверка достоверности полученных результатов.

ГЛАВА 11



Компьютерные технологии вычисления интегралов

В системе Mathematica реализованы аналитические и численные методы вычисления интегралов.

Вычисляются неопределенные и определенные, кратные и несобственные интегралы, интегралы многих переменных. При этом переменными интегрирования могут быть числа, символьные переменные и даже функции. С помощью одной встроенной функции можно одновременно вычислить интеграл от многих функций.

Рассмотрим технологию вычисления интегралов для всех перечисленных случаев.

11.1. Аналитические методы вычисления интегралов

Интеграл в аналитическом виде вычисляется с помощью следующих встроенных функций:

- `Integrate[f(x), x]` — вычисляет неопределенный интеграл функции $f(x)$ по аргументу x . Откликом является первообразная $F(x)$ в аналитическом виде;
- `Integrate[f(x), x, {x, xn, xk}]` — вычисляет определенный интеграл функции $f(x)$ по переменной x с нижним x_n и верхним x_k пределами интегрирования. Пределами интегрирования могут быть символьные переменные, числа и даже функции. Откликом является выражение интеграла в аналитическом виде;
- `Integrate[f(x, y, ...), {x, xn, xk}, {y, yn, yk, ...}]` — вычисляет определенный интеграл многих переменных $x, y, \dots$ с пределами интегрирования $x_n, x_k, y_n, y_k, \dots$. Откликом является аналитическое выражение первообразной от многих переменных.

Технология вычисления интеграла исключительно проста и состоит в следующем:

1. Ввод с клавиатуры подынтегральной функции с присвоением ей уникального имени.
2. Ввод встроенной функции `Integrate` с требуемыми параметрами.
3. Получение решения нажатием клавиш `<Shift>+<Enter>` — на экране значение интеграла.

Примеры вычисления интегралов приведены на рис. 11.1. Вычисляются интегралы:

$$\int \frac{ax-1}{bx+1} dx, \quad \int_a^b \frac{2+x}{x} dx, \quad \int_a^b (1+2xy+4x^2y^2) dx dy.$$

```

f1 := (a * x - 1) / (b * x + 1)
Integrate[f1, x]

  a x      (-a - b) Log[1 + b x]
  --- + ----
   b      b2

f2 := (2 + x) / x
Integrate[f2, {x, a, b}]

-a + b - 2 Log[a] + 2 Log[b]

f3 := 1 + 2 * x * y + 4 * x2 * y2
Integrate[f3, {x, a, b}, {y, a, b}]

a2 +  $\frac{a^4}{2}$  +  $\frac{4 a^6}{9}$  - 2 a b + b2 - a2 b2 -  $\frac{8 a^3 b^3}{9}$  +  $\frac{b^4}{2}$  +  $\frac{4 b^6}{9}$ 

```

Рис. 11.1. Примеры вычисления интегралов в аналитическом виде

Подынтегральную функцию можно на экран не выводить, а представлять ее непосредственно во встроенной функции. Этот случай показан на рис. 11.2.

Из рис. 11.1 и 11.2 видно, что решения при различных способах ввода функции $f(x)$ совпадают.

В предыдущих примерах пределами интегрирования были символьные переменные a и b . На рис. 11.3 показано вычисление интегралов тех же функций в случае, когда пределами интегрирования являются числа: $x_n=1$, $x_k=5$, $y_n=0$, $y_k=10$.

Из рис. 11.3 видно, что в данном случае получены значения интегралов в аналитическом виде (в виде точных решений). Значения интегралов в естественной форме получены с помощью функции `N(%)`.

```
Integrate[(a x - 1) / (b x + 1), x]
```

$$\frac{a x}{b} + \frac{(-a-b) \operatorname{Log}[1+b x]}{b^2}$$

```
Integrate[(2 + x) / x, {x, a, b}]
```

$$-a + b - 2 \operatorname{Log}[a] + 2 \operatorname{Log}[b]$$

```
Integrate[1 + 2 x y + 4 x^2 y^2, {x, a, b}, {y, a, b}]
```

$$a^2 + \frac{a^4}{2} + \frac{4 a^6}{9} - 2 a b + b^2 - a^2 b^2 - \frac{8 a^3 b^3}{9} + \frac{b^4}{2} + \frac{4 b^6}{9}$$

Рис. 11.2. Вычисление интегралов при вводе $f(x)$ во встроенную функцию `Integrate`

```
Integrate[(2 + x) / x, {x, 1, 5}]
```

$$2 (2 + \operatorname{Log}[5])$$

```
N[%]
```

$$7.21888$$

```
Integrate[1 + 2 x y + 4 x^2 y^2, {x, 1, 5}, {y, 0, 10}]
```

$$\frac{507160}{9}$$

```
N[%]
```

$$56351.1$$

Рис. 11.3. Вычисление интегралов при численных значениях пределов интегрирования

На рис. 10.4 приведен случай вычисления интеграла, когда пределами интегрирования являются функции: $x_n = \operatorname{Log}[a]$, $x_k = \operatorname{Log}[b]$, $x_n = \operatorname{Sin}[0.5]$, $x_k = \operatorname{Cos}[0.5]$.

```
Integrate[(2 + x) / x, {x, Log[a], Log[b]}]
```

$$-\operatorname{Log}[a] + \operatorname{Log}[b] - 2 \operatorname{Log}[\operatorname{Log}[a]] + 2 \operatorname{Log}[\operatorname{Log}[b]]$$

```
Integrate[(2 + x) / x, {x, Sin[0.5], Cos[0.5]}]
```

$$1.60732$$

Рис. 11.4. Вычисление определенного интеграла с пределами интегрирования в виде функций

Из рис. 11.4 видно, что система Mathematica выдала значение определенного интеграла в аналитическом и численном виде. При этом ответ в численном виде

представлен в естественной форме. Это объясняется тем, что функции $\sin[0.5]$ и $\cos[0.5]$ точных значений не имеют.

11.2. Численные методы вычисления интегралов

Вычисление интеграла в численном виде необходимо в следующих случаях:

- ☐ первообразная не выражается через элементарные функции;
- ☐ подынтегральная функция задана в виде таблицы;
- ☐ аналитическое выражение первообразной слишком сложно.

В качестве примера на рис. 10.5 приведены вычисления неопределенных и определенных интегралов от функций: $y(x) = x^{\frac{1}{x}} e^x$ и $y(x) = x^{20} e^{-x}$. Пределы интегрирования видны из рис. 11.5.

```

Integrate[x^(1/x) * Exp[x], x]

$$\int e^x x^{\frac{1}{x}} dx$$

NIntegrate[x^(1/x) * Exp[x], {x, 1, 5}]
204.435
Integrate[x^20 Exp[-x], x]
Integrate[x^20 * Exp[-x], x]
Integrate[x^20 * Exp[-x], x]

$$e^{-x} \left( -2432902008176640000 - 2432902008176640000 x - \right.$$


$$1216451004088320000 x^2 - 405483668029440000 x^3 -$$


$$101370917007360000 x^4 - 20274183401472000 x^5 -$$


$$3379030566912000 x^6 - 482718652416000 x^7 -$$


$$60339831552000 x^8 - 6704425728000 x^9 - 670442572800 x^{10} -$$


$$60949324800 x^{11} - 5079110400 x^{12} - 390700800 x^{13} -$$


$$27907200 x^{14} - 1860480 x^{15} - 116280 x^{16} - 6840 x^{17}$$


$$- 380 x^{18} - 20 x^{19} - x^{20} \Big)$$

NIntegrate[x^20 Exp[-x], {x, 1, 2}]
14860.3

```

Рис. 11.5. Вычисление интегралов аналитическим и численным методом

Из рис. 11.5 видно, что первый интеграл оказался неберущимся, а второй — слишком сложным для дальнейших расчетов и трудно объяснимым. Между

тем, вычисления интегралов численными методами с помощью функции `NIntegrate` выполнено без каких-либо затруднений.

Существует ряд способов численного интегрирования. Во всех таких способах вычисление осуществляется по приближенным формулам, называемым *квадратурными*. Наиболее часто используются следующие квадратурные формулы:

- ☐ прямоугольников;
- ☐ трапеций;
- ☐ парабол;
- ☐ высшего порядка.

В настоящей главе приводятся лишь компьютерные технологии вычисления интегралов численными методами в системе Mathematica.

11.3. Технология вычисления интегралов численными методами

В системе Mathematica численное интегрирование осуществляется с помощью функции `NIntegrate`, которая имеет вид:

`NIntegrate(f(x), {x, xn, xk})`

где:

- ☐ $f(x)$ — подынтегральная функция ;
- ☐ x — аргумент подынтегральной функции;
- ☐ x_n, x_k — нижний и верхний пределы интегрирования.

Технология вычисления интеграла с помощью функции `NIntegrate` не отличается от технологии вычисления определенного интеграла в аналитическом виде. Пределами интегрирования по-прежнему могут быть числа и функции.

```
f := (x - 1) / (x + 1) * Exp[x] + Log[x^2]
NIntegrate[f, {x, 1, 10}]
17588.4
F[x] := NIntegrate[f, {x, 1, 10}]
F[x]
17588.4
NIntegrate[(x - 1) / (x + 1) Exp[x] + Log[x^2], {x, 1, 10}]
17588.4
NIntegrate[(x - 1) / (x + 1) Exp[x] + Log[x^2], {x, Log[2], Exp[1.2]}]
13.6735
```

Рис. 11.6. Пример вычисления интеграла численными методами

Примеры численного интегрирования приведены на рис. 11.6.

На рис. 10.6 показаны четыре способа вычисления определенного интеграла:

- ☐ подынтегральной функции присвоено имя `f`;
- ☐ имя присвоено встроенной функции `NIntegrate`;
- ☐ присвоение имени функций отсутствует;
- ☐ пределами интегрирования являются числа и функции $\ln 2$ и $e^{1.2}$.

11.4. Использование символа интеграла ($\int$)

Система Mathematica позволяет вычислять неопределенные и определенные интегралы аналитическими и численными методами, используя математический символ интеграла в одном из следующих видов:

$$\int f(x) dx, \quad \int_a^b f(x) dx$$

Первый применяется при вычислении неопределенного интеграла, второй — при вычислении определенного.

Технология вычисления интеграла состоит в следующем:

1. Вывод на экран пиктограммы интеграла с помощью меню **File | Palettes | BasicInput** (Файл | Палитры | Основной ввод).
2. Ввод в поля пиктограммы выражений подынтегральной функции, знака дифференцирования и пределов интегрирования.
3. Получение решения нажатием клавиш `<Shift>+<Enter>` — на экране значение интеграла.

Процедуры вычисления интегралов с помощью символа интеграла приведены на рис. 11.7.

Из рис. 11.7 видно, что при использовании пиктограмм Mathematica выдает решение в аналитическом виде. Для получения ответа в виде числа необходимо дополнительно использовать команду `N[%]` или перед пиктограммой ввести символ `N`.

11.5. Вычисление кратных интегралов

Вычисление кратных интегралов в системе Mathematica осуществляется путем многократного применения функции `Integrate` при аналитическом или

```


$$\int (a x - 1) / (b x + 1) dx$$


$$\frac{a x}{b} + \frac{(-a - b) \operatorname{Log}[1 + b x]}{b^2}$$


$$\int_a^b (2 + x) / x dx$$


$$-a + b - 2 \operatorname{Log}[a] + 2 \operatorname{Log}[b]$$


$$\int_1^5 x^{-1} \operatorname{Exp}[x] dx$$


$$\int_1^5 e^x x^{-1/x} dx$$

N[%]
103.896
N[  $\int_1^{10} ((x - 1) / (x + 1) \operatorname{Exp}[x] + x + 1) dx$  ]
17618.8

$$\int_{\sin[0.5]}^{\cos[0.5]} (2 + x) / x dx$$

1.60732

$$\int_1^2 \operatorname{Exp}[-2 x] dx$$


$$-\frac{1}{2 e^4} + \frac{1}{2 e^2}$$

N[  $\int_1^2 \operatorname{Exp}[-2 x] dx$  ]
0.0585098

```

Рис. 11.7. Вычисление интегралов с использованием пиктограмм

`NIntegrate` при численном интегрировании. Вместо этих функций можно использовать их пиктограммы.

На рис. 11.8 приведены примеры вычисления кратных интегралов от функции $f(x) = \frac{x-1}{x+1}$. Рассмотрены случаи двукратного неопределенного интеграла с применением встроенной функции `Integrate` и ее пиктограммы, трехкратного определенного интеграла при различных вариантах пределов интегрирования (рис. 11.8):

- ☐ символьных переменных от a до b ;
- ☐ численных значениях пределов интегрирования от 0 до 2;
- ☐ пределов интегрирования, заданных функциями: $\ln 2$ и $e^{1/2}$.

```
Integrate[Integrate[(x - 1) / (x + 1), x], {x, a, b}]
```

$$\frac{1}{2} (-4 a - a^2 + 4 b + b^2 + 4 \operatorname{Log}[1 + a] + 4 a \operatorname{Log}[1 + a] - 4 \operatorname{Log}[1 + b] - 4 b \operatorname{Log}[1 + b])$$

$$\int_a^b \int (x - 1) / (x + 1) \, dx \, dx$$

$$\frac{1}{2} (-4 a - a^2 + 4 b + b^2 + 4 \operatorname{Log}[1 + a] + 4 a \operatorname{Log}[1 + a] - 4 \operatorname{Log}[1 + b] - 4 b \operatorname{Log}[1 + b])$$

$$\int_a^b \int \int (x - 1) / (x + 1) \, dx \, dx \, dx$$

$$\frac{1}{6} (-6 a - 9 a^2 - a^3 + 6 b + 9 b^2 + b^3 + 6 \operatorname{Log}[1 + a] + 12 a \operatorname{Log}[1 + a] + 6 a^2 \operatorname{Log}[1 + a] - 6 \operatorname{Log}[1 + b] - 12 b \operatorname{Log}[1 + b] - 6 b^2 \operatorname{Log}[1 + b])$$

$$\int_0^2 \int \int (x - 1) / (x + 1) \, dx \, dx \, dx$$

$$\frac{1}{3} (28 - 27 \operatorname{Log}[3])$$

```
N[%]
```

```
-0.554177
```

$$\int_{\operatorname{Log}[2]}^{\operatorname{Exp}[1.2]} \int \int (x - 1) / (x + 1) \, dx \, dx \, dx$$

```
-1.31499
```

Рис. 11.8. Вычисление кратных интегралов

11.6. Вычисление несобственных интегралов

Система Mathematica позволяет вычислять интегралы с бесконечными пределами. При этом используются те же функции и технологии, что и в случае вычисления интегралов с конечными пределами. Значение бесконечности обозначается либо символом ∞ , либо словом *Infinity*.

Примеры вычисления несобственных интегралов приведены на рис. 11.9.

Из рис. 11.9 можно сделать следующие выводы.

- Решение несобственного интеграла получаем в аналитическом виде. Для получения численного значения интеграла применяется команда `N[%]` или символ численного интегрирования `N`, который ставится перед символом интеграла ($\int$).
- Выражение первообразной функции в аналитическом виде может быть сложным. Для его упрощения следует воспользоваться функциями `Simplify`, `Expand`, `Factor` или их прототипами, как это показано на рис. 11.9 при вычислении интеграла от функции $\frac{x+1}{x^3+1}$.

- Если интеграл не имеет первообразной, то откликом будет исходное выражение интеграла.

$$\int_0^{\infty} \frac{\text{Sin}[x]}{x} dx$$

$$\frac{\pi}{2}$$

$$\int_0^{\infty} \frac{(\text{Sinh}[x] + \text{Cosh}[x])}{\text{Exp}[x^2]} dx$$

$$\frac{1}{2} e^{1/4} \sqrt{\pi} \left(1 + \text{Erf}\left[\frac{1}{2}\right] \right)$$

$$\mathbf{N[\%]}$$

$$1.73023$$

$$\int_0^{\infty} \frac{(x+1)}{(x^3+1)} dx$$

$$\frac{\pi + i \left(\text{Log}\left[3 - i \sqrt{3}\right] - \text{Log}\left[3 + i \sqrt{3}\right] \right)}{\sqrt{3}}$$

$$\mathbf{FullSimplify[\%]}$$

$$\frac{4\pi}{3\sqrt{3}}$$

$$\int_{-\infty}^{\infty} \frac{\text{Exp}[x]}{(a + \text{Exp}[x])^2} dx$$

$$\frac{1}{a}$$

$$\int_0^{\infty} \frac{(x+1)}{(2x+1)} dx$$

$$\int_0^{\infty} \frac{1+x}{1+2x} dx$$

$$\mathbf{NIntegrate[1/(2x^2+1), \{x, 0, Infinity\}]}$$

$$1.11072$$

Рис. 11.9. Вычисление несобственных интегралов

11.7. Табличное интегрирование

Подынтегральная функция $f(x)$ может быть задана в виде таблицы. Наиболее часто это случается при проведении экспериментальных исследований. В таких случаях вычисление интеграла можно осуществить по формулам прямоуголь-

ников, трапеций или парабол. Решение можно получить также путем интерполяции функции $f(x)$ с последующим ее интегрированием.

При использовании системы Mathematica эти методы применять нет смысла. Система имеет следующие встроенные функции, позволяющие выполнить табличное интегрирование:

```
ListIntegrate[{y1, y2, ..., yn}, h]
```

```
ListIntegrate[{y1, y2, ..., yn}, h, k]
```

```
ListIntegrate[{{x1, y1}, {x2, y2}, ..., {xn, yn}}, k]
```

В функциях приняты следующие обозначения:

□ y_i — значение функции $y = f(x)$ в x_i -м узле, $i = 1, 2, \dots, n$;

□ h — шаг изменения аргумента;

□ k — число точек каждого из подынтегралов.

В качестве примера вычислим значение интеграла функции $y = f(x)$, представленной табл. 11.1.

Таблица 11.1. Табличное представление функции $y = f(x)$

x	1	2	3	4	5	6	7	8
y	1	8	27	64	125	216	343	512

В данном случае шаг интегрирования постоянный и равен $h=1$.

Решение задачи приведено на рис. 11.10.

Из табл. 11.1 видно, что функция $y = f(x)$ имеет вид: $y = x^3$. На рис. 11.10 приведены процедуры вычисления интеграла в случае задания подынтегральной функции в виде таблицы и в виде аналитического выражения $y = x^3$. Результаты расчетов одинаковы. Табличное интегрирование оказалось точным.

На практике бывает иначе. Обычно функция $y = f(x)$ представляется в табличном виде как результат опыта, а это значит, что интеграл вычисляется с погрешностью, вызванной неточностью исходных данных. Таким образом, при табличном интегрировании возникает необходимость оценки погрешности вычисления интеграла. Приведем пример.

Пусть в результате эксперимента получены данные, приведенные в табл. 11.2.

Таблица 11.2. Табличное представление функции $y = f(x)$

x	1	2	3	4	5	6	7	8	9	10
y	4.5	12	20	35	46	68	90	100	130	160

```

f = {1, 8, 27, 64, 125, 216, 343, 512}
{1, 8, 27, 64, 125, 216, 343, 512}
ListIntegrate[f, 1]

$$\frac{4095}{4}$$

Integrate[x^3, {x, 1, 8}]

$$\frac{4095}{4}$$

ListIntegrate[{ {1,1}, {2,8}, {3,27},
    {4,64}, {5,125}, {6,216}, {7,343},
    {8,512} }]

$$\frac{4095}{4}$$


```

Рис. 11.10. Вычисление интеграла при табличном задании подынтегральной функции

Требуется:

- ☐ вычислить интеграл табличным численным интегрированием;
- ☐ решить задачу полиномиальной интерполяции;
- ☐ вычислить численное значение интеграла функции интерполяции $\phi(x)$;
- ☐ оценить погрешность численного интегрирования.

Для вычисления табличного интеграла необходимо обратиться к подпакету `ListIntegrate` пакета `NumericalMath`.

Решение приведено на рис. 11.11.

```

<<NumericalMath`ListIntegrate`
ListIntegrate[{4.5, 12, 20, 35, 46, 68, 90, 100, 130, 160}, 1]
582.458
Fit[{{1, 4.5}, {2, 12}, {3, 20}, {4, 35}, {5, 46}, {6, 68},
    {7, 90}, {8, 100}, {9, 130}, {10, 160}}, {a, x, x^2}, x]
-0.533333 + 3.71212 x + 1.21212 x^2
Integrate[Out[11], {x, 1, 10}]
582.586

```

Рис. 11.11. Решение примера вычисления интеграла с использованием интерполяции

Из рис. 11.11 видно, что значение интеграла, вычисленное численным табличным интегрированием (582.458) и интегрированием функции $\varphi(x)$ (582.586), практически совпадают. При этом функцией интерполяции является полином, второй степени $\varphi(x) = 1.212x^2 + 3.71x - 0.53$. Интерполяция выполнена с помощью встроенной функции `Fit`. Подробно решение задачи интерполяции изложено в главе 12.

Отметим, что максимальная относительная среднеквадратическая погрешность интерполяции велика и составляет 68.3%. Несмотря на это, погрешность вычисления интеграла практически равна нулю. Из этого можно сделать следующий важный вывод: интеграл, вычисленный по табличным данным, может иметь высокую точность, если даже ошибки эксперимента велики.

Имеют место случаи, когда табличное интегрирование просто необходимо. Это бывает тогда, когда первообразная не существует, а численное интегрирование не дает результата. В таких случаях значение интеграла может быть получено с помощью табличного интегрирования.

Технология вычисления интеграла табличным способом при этом состоит из выполнения следующих пунктов:

- ☐ определение области интегрирования на основании графического представления подынтегральной функции;
- ☐ табулирование подынтегральной функции;
- ☐ вычисление интеграла табличным способом.

Рассмотрим эту технологию на примере.

Пусть необходимо вычислить следующий интеграл:

$$\int_0^{\infty} \left(\frac{x}{1+x^2} \right)^x dx.$$

На рис. 11.12 показан график подынтегральной функции.

Из графика видно, что функция стремится к нулю и быстро затухает (область ее изменения ограничена диапазоном x от 0 до 4).

Вычислим значение неопределенного, несобственного интегралов, интеграла с ограниченными пределами от 0 до 10, используя встроенные функции `Integrate`, а также табличное интегрирование.

Решение приведено на рис. 11.13.

Из рис. 11.13 видно, что первообразная не существует (система повторяет выражение интеграла), интеграл с бесконечным пределом система не вычисляет. Зато табличное интегрирование позволило вычислить определенный интеграл, предварительно протабулировав подынтегральную функцию в диапазоне x от 0 до 10 с шагом 0.5.

```
f:=(x/(1+x^2))^x
Plot[f, {x, 0, 10}]
```

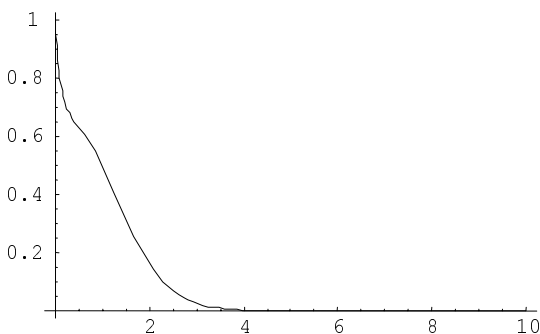


Рис. 11.12. График подынтегральной функции

```
f := (x / (1 + x^2))^x
```

$$\int f \, dx$$

$$\int \left(\frac{x}{1+x^2} \right)^x dx$$

$$\int_0^{\infty} f \, dx$$

$$\int_0^{\infty} \left(\frac{x}{1+x^2} \right)^x dx$$

```
Table[f, {x, 0.001, 10, 0.5}]
```

```
{0.9931160474278182`, 0.6322554979764589`, 0.499653296668
5779`, 0.313190687164853`, 0.15975752780524083`,
0.0696992354057739`, 0.026945941703968145`, 0.00945225921
0883605`, 0.003057965841851687`, 0.0009231525820754326`,
0.0002623411057651897`, 0.00007065756133293688`, 0.000018
134110187323895`, 4.454405730844279`*^-
6, 1.0510549523783843`*^-6, 2.389689228882619`*^-
7, 5.2490687366156434`*^-8, 1.1164471523185427`*^-
8, 2.303958305947073`*^-9, 4.6211987501374336`*^-10}
```

```
<< NumericalMath`ListIntegrate`
```

```
ListIntegrate[Out[7], 0.5, 10]
```

```
1.06924
```

$$\int_0^{10} f \, dx$$

$$\int_0^{10} \left(\frac{x}{1+x^2} \right)^x dx$$

Рис. 11.13. Вычисление интеграла табличным методом

При табулировании подынтегральной функции в диапазоне x от 0 до 10 система не выдает значение функции при $x=0$: она "не знает", что $0^0=1$. Поэтому табулирование выполнено в диапазоне x от 0.001 до 10. Это не повлияло на результат, т. к. $f(0.001) \approx f(0) = 1$.

В последних строках рис. 11.13 показан результат вычисления интеграла в ограниченном диапазоне x . Однако и в этом случае решение не получено.

11.8. Проверка правильности вычисления интеграла

Проверить достоверность вычисления интеграла можно многими способами. Основными из них являются следующие:

- ☐ вычисление производной первообразной функции;
- ☐ применение различных методов интегрирования;
- ☐ преобразование подынтегральной функции;
- ☐ использование численного интегрирования для проверки истинности первообразной;
- ☐ сравнение результатов вычисления интеграла с помощью нескольких универсальных программных средств символьной математики.

11.8.1. Вычисление производной первообразной функции

Производная первообразной функции является подынтегральной функцией. Это позволяет осуществлять проверку правильности результатов интегрирования аналитическими методами. Проверить правильность численного интегрирования этим методом невозможно.

На рис. 11.14 приведены результаты проверки вычисления следующих интегралов:

$$\begin{aligned} & \int (a \sin x + b \cos x) dx \\ & \int a e^{-bx} dx \\ & \int \frac{x-1}{x+1} dx \\ & \iint \frac{x-1}{x+1} dx \\ & \int (x e^x + \ln x + 1) dx \\ & \int (a + 2xy + 4x^2 y^2) dx dy. \end{aligned}$$

```

Integrate[(a Sin[x] + b Cos[x]) dx
-a Cos[x] + b Sin[x]
D[% , x]
b Cos[x] + a Sin[x]
Integrate[a Exp[-b x] dx
- a Exp[-b x] / b
D[% , x]
a Exp[-b x]
Integrate[(x - 1) / (x + 1) dx
x - 2 Log[1 + x]
D[% , x]
1 - 2 / (1 + x)
Integrate[(x - 1) / (x + 1) dx dx
2 x + x^2 / 2 - 2 Log[1 + x] - 2 x Log[1 + x]
D[Out[12] , x]
2 x + x^2 / 2 - 2 Log[1 + x] - 2 x Log[1 + x]
D[Out[17] , x]
1 + 2 / (1 + x)^2 + 2 x / (1 + x)^2 - 4 / (1 + x)
Integrate[(a + 2 x y + 4 x^2 y^2) dx dy
a x y + x^2 y^2 / 2 + 4 x^3 y^3 / 9
D[Out[1] , x]
a y + x y^2 + 4 x^2 y^3 / 3
D[Out[5] , y]
a + 2 x y + 4 x^2 y^2

```

Рис. 11.14. Проверка правильности результатов вычисления интегралов

По результатам проверки убеждаемся, что все интегралы вычислены верно. Однако подынтегральная функция после дифференцирования отличается от исходной подынтегральной функции. Пользователь должен упростить полученные производные до исходных выражений. В этом недостаток такого метода проверки. Правда, для этой цели можно использовать функции системы `Simplify` и `Expand`.

11.8.2. Применение различных методов интегрирования

Этот способ основан на сравнении результатов вычисления интеграла численным методом с помощью функции `NIntegrate[f(x), {x, a, b}]` и аналитическим, использующим формулу Ньютона:

$$\int_a^b f(x)dx = F(b) - F(a), \quad (11.1)$$

где $F(a)$, $F(b)$ — значения первообразных при нижнем и верхнем значениях пределов интегрирования.

Эта формула в системе Mathematica реализуется функцией `Integrate[f(x), {x, a, b}]`.

```
f := x Exp[x] + Log[x] - 1
NIntegrate[f, {x, 1, 10}]
198243.
Integrate[f, {x, 1, 10}]
-18 + 9 e10 + 10 Log[10]
N[%]
198243.
Integrate[f, x]
ex (-1 + x) - 2 x + x Log[x]
N[Out[5], x = 10]
198241.
N[Out[5], x = 1]
-2.
Out[6] - Out[8]
198243.
```

Рис. 11.15. Вычисление интеграла и проверка достоверности результата

Приведем пример вычисления интеграла и проверки достоверности результата.

Пусть необходимо вычислить определенный интеграл

$$\int_1^{10} (xe^x + \ln x - 1) dx$$

и проверить правильность ответа путем сравнения результатов двух методов интегрирования: точного и приближенного.

Процедуры решения задачи в системе Mathematica приведены на рис. 11.15.

Из рис. 11.15 видно, что интеграл вычислен правильно, результаты двух методов совпадают.

11.8.3. Сравнение результатов интегрирования различными системами символьной математики

Этот способ очевиден. Вычисление интеграла выполняется с помощью нескольких систем символьной математики. В результате сравнения результатов делается вывод о достоверности вычисления интеграла.

В качестве примера на рис. 11.16 и 11.17 приведены результаты вычисления с помощью систем Derive 5 и Mathematica трех следующих несобственных интегралов:

$$\int_0^{\infty} x^a e^{-x} dx, \int_0^{\infty} x^a e^{-ax} dx, \int \frac{ax}{\sinh ax} dx.$$

The screenshot shows the Derive 5 interface with the following content:

- #1: $a \in \text{Real } (0, \infty)$
- #2: $\int_0^{\infty} x^a \cdot e^{-x} dx$ $a!$
- #3: $\int_0^{\infty} x^a \cdot e^{-a \cdot x} dx$ $a^{-a} \cdot (a - 1)!$
- #4: $\int_0^{\infty} x^a \cdot e^{-a \cdot x} dx$ $a^{-a} \cdot (a - 1)!$
- #5: $\int_0^{\infty} x^a \cdot e^{-a \cdot x} dx$ $a^{-a} \cdot (a - 1)!$
- #6: $\int_0^{\infty} \frac{a \cdot x}{\sinh(a \cdot x)} dx$ $\frac{2}{\pi}$
- #7: $\int_0^{\infty} \frac{a \cdot x}{\sinh(a \cdot x)} dx$ $\frac{2}{\pi}$

Рис. 11.16. Вычисление интегралов системой Derive 5

$$\begin{aligned}
 & \int_0^{\infty} x^a \text{Exp}[-x] \, dx \\
 & \text{If} \left[\text{Re}[a] > -1, \text{Gamma}[1+a], \int_0^{\infty} e^{-x} x^a \, dx \right] \\
 & \int_0^{\infty} x^a \text{Exp}[-a x] \, dx \\
 & \text{If} \left[\text{Re}[a] > 0, a^{-a} \text{Gamma}[a], \int_0^{\infty} e^{-a x} x^a \, dx \right] \\
 & \int_0^{\infty} a x / \text{Sinh}[a x] \, dx \\
 & a \text{ If} \left[\text{Re}[a] > 0, \frac{\pi^2}{4 a^2}, \int_0^{\infty} x \text{Csch}[a x] \, dx \right]
 \end{aligned}$$

Рис. 11.17. Вычисление интегралов системой Mathematica

Анализируя результаты вычисления интегралов, можно сделать следующие выводы:

- ☐ результаты вычисления интегралов верны, так как ответы одинаковы, хотя и представлены в разных видах;
- ☐ ответы системы Derive 5 более естественны. Однако пользователь при этом должен знать, что интеграл вычисляется при условии $a > 0$. Система Mathematica это условие сообщает пользователю.

11.8.4. Особенности вычисления интегралов в системе Mathematica

Система Mathematica высоко интеллектуальна: она объясняет пользователю в необходимых случаях результаты вычислений.

Если интеграл расходящийся, то система повторяет выражение интеграла с комментариями.

Если интеграл не существует, то система лишь повторяет его выражение.

Если интеграл берется только при определенных условиях, то система выдает решение и указывает эти условия. Первообразная может содержать специальные функции.

Все эти случаи видны на рис. 11.18, где вычисляются следующие интегралы:

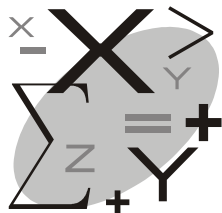
$$\int_0^{\infty} \frac{1 + \sin x}{1 + \cos x} dx, \quad \int_0^{\infty} \frac{\sin x}{\cos x} dx, \quad \int \left(\frac{1}{1-x} \right)^{\frac{1}{x}} dx, \quad \int_0^{\infty} x^a e^{-x} dx.$$

В настоящей главе описаны практически все встроенные функции системы Mathematica. Однако они имеют большое число опций [12], применяемых в особых случаях. Здесь они не приводятся, т. к. такие случаи вряд ли встретятся в студенческой практике.

$$\begin{aligned}
& \int_0^{\infty} (1 + \sin[x]) / (1 + \cos[x]) \, dx \\
& \int_0^{\infty} \frac{1 + \sin [x]}{1 + \cos [x]} \, dx \\
& \int_0^{\infty} \sin[x] / \cos[x] \, dx \\
& \int_0^{\infty} \tan[x] \, dx \\
& \int (1 / (1 - x))^{\frac{1}{x}} \, dx \\
& \int_0^{\infty} x^a \exp[-x] \, dx \\
& \text{If} \left[\text{Re}[a] > -1, \text{Gamma}[1 + a], \int_0^{\infty} e^{-x} x^a \, dx \right]
\end{aligned}$$

Рис. 11.18. Особенности вычисления интегралов системой Mathematica

ГЛАВА 12



Компьютерные технологии решения задач интерполяции

12.1. Виды и этапы компьютерных технологий интерполяции

Интерполяцией называется представление функции $y = f(x)$, заданной аналитически или в виде таблицы, функцией $y = \varphi(x)$, которая идентична исходной в некоторой области аргумента. Интерполяция совместно с теорией подобия и теорией размерностей является научной основой моделирования.

График и таблица не могут быть моделью объекта или явления. Только математическая функция (формула) может быть моделью изучаемого объекта или явления.

Но не только в моделировании используется интерполяция. Она необходима при планировании и статистической обработке эксперимента, при замене сложной аналитической функции более простой в нужном диапазоне изменения аргумента.

Основными этапами компьютерных технологий интерполяции являются:

- ☐ выбор вида функции интерполяции;
- ☐ определение коэффициентов функции интерполяции;
- ☐ определение адекватности функции интерполяции.

Реализация этих этапов определяется выбранным методом интерполяции, типом применения встроенной функции и формой выходного документа.

Система Mathematica позволяет реализовать различные методы и технологии интерполяции. Все они рассматриваются в данной главе.

Существуют два основных вида интерполяции: точная в узлах и приближенная в узлах.

Интерполяцией точной в узлах называется такая интерполяция, результатом которой является функция $y = \varphi(x)$, совпадающая в узлах интерполяции с функцией $y = f(x)$. Такая интерполяция показана на рис. 12.1.

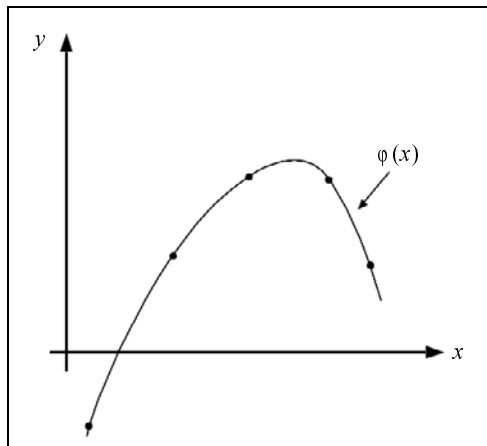


Рис. 12.1. Интерполяция точная в узлах

Интерполяция точная в узлах применяется главным образом в тех случаях, когда требуется сложную математическую функцию заменить более простой в узком диапазоне значений аргумента.

Ее можно применить также в том случае, когда требуется найти математическую модель объекта, характеристики которого получены экспериментально с высокой точностью.

Интерполяцией приближенной в узлах называется интерполяция, при которой значения функции $y = \varphi(x)$ не совпадают в узлах со значениями исходной функции $y = f(x)$. Такая интерполяция применяется для сглаживания неточностей исходных данных. В математике она называется *аппроксимацией*. Ее геометрический смысл показан на рис. 12.2.

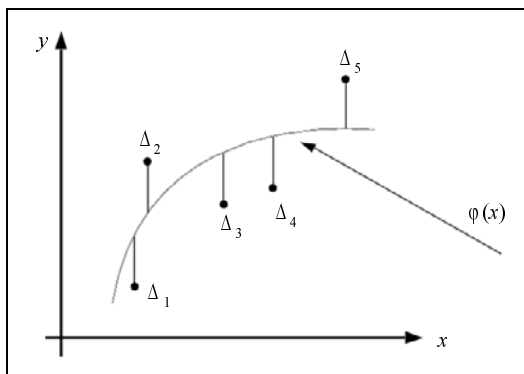


Рис. 12.2. Интерполяция приближенная в узлах

На рис. 12.2 приняты следующие обозначения:

- ☐ $\varphi(x)$ — эмпирическая функция;
- ☐ $\langle, \rangle$ (точка) — значения функции, полученные экспериментально;
- ☐ Δ_i — уклонение (невязка) — разность в i -м узле интерполяции между $\varphi(x_i)$ и y_i .

Функция интерполяции $y = \varphi(x)$ находится на основании критериев близости эмпирической функции $\varphi(x)$ и исходной, полученной экспериментальным или иным путем.

Таковыми критериями могут быть:

- ☐ алгебраическая сумма уклонений равна нулю:

$$\Delta_c = \sum_{i=1}^n \Delta_i = 0;$$

- ☐ сумма квадратов уклонений минимальна:

$$\Delta_c = \sum_{i=1}^n \Delta_i^2 = \min$$

- ☐ среднее значение уклонений минимально:

$$\Delta_c = \frac{\sum_{i=1}^n \Delta_i}{n} = \min$$

Компьютерные технологии решения задач интерполяции определяются видом критерия близости эмпирической и исходной функции, в частности методикой оценки погрешностей.

12.1.1. Выбор вида функции интерполяции

Выбор вида функции интерполяции является наиболее важным этапом интерполяции, т. к. функция $\varphi(x)$ определяет математическую модель изучаемого объекта или явления.

Существуют следующие, применяемые в практических расчетах, способы выбора вида функции $\varphi(x)$:

- ☐ графоаналитический;
- ☐ линеаризации нелинейных функций;
- ☐ метод табличных разностей;
- ☐ обращение к программам автоматизации выбора функции $\varphi(x)$.

Рассмотрим эти способы.

Способ 1. Графоаналитический

Функция $y = f(x)$ представляется в виде графика. График сравнивается с графиками типичных математических функций и на основании их сравнения выбирается подходящая. Ниже приводятся часто используемые функции и их графики.

□ Степенная функция $y = ax^b$.

График функции показан на рис. 12.3.

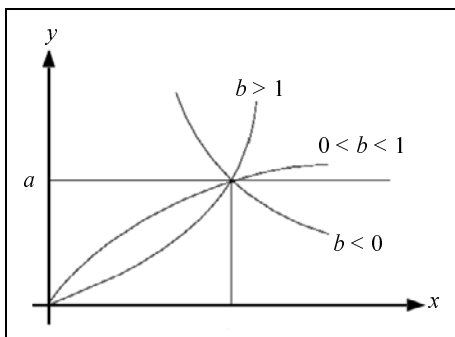


Рис. 12.3. Степенная функция $y = ax^b$

□ Логарифмическая функция $y = a + b \ln x$.

График функции показан на рис. 12.4.

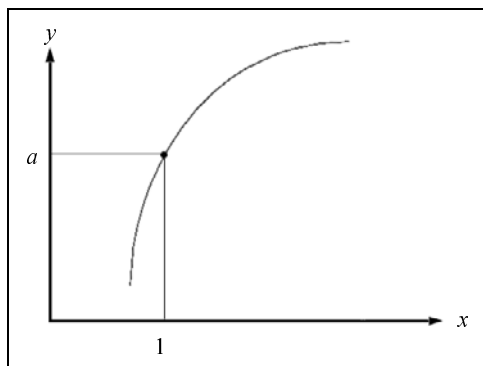


Рис. 12.4. Логарифмическая функция $y = a + b \ln x$

- Дробно-линейная функция $y = \frac{x}{a + bx}$.

График функции показан на рис. 12.5.

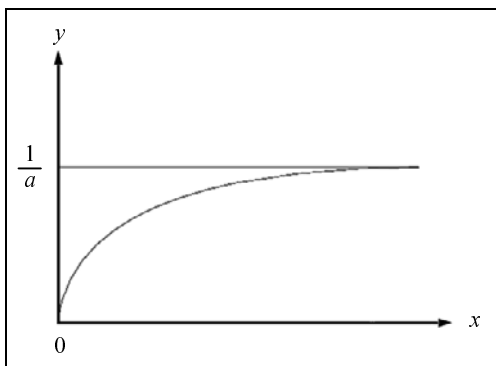


Рис. 12.5. Дробно-линейная функция $y = \frac{x}{a + bx}$

- Показательная функция $y = ab^x$.

График функции показан на рис. 12.6.

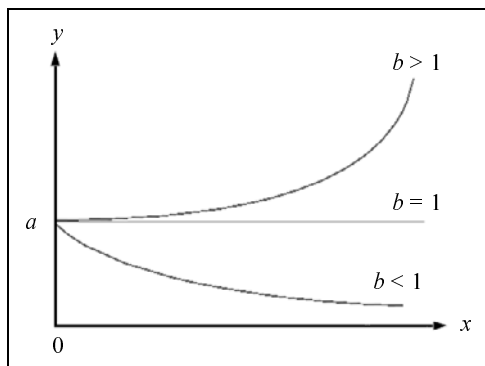


Рис. 12.6. Показательная функция $y = ab^x$

Способ 2. Линеаризация нелинейных функций

Линеаризация нелинейных функций осуществляется путем представления их в иной системе координат методом замены переменных x и y . Покажем способы линеаризации на примерах, рассмотренных выше функций.

□ Показательная функция $y = ab^x$.

Прологарифмируем показательную функцию: $\ln y = \ln a + x \ln b$. Заменой переменных $\ln y = Y$, $x = X$ получим линейную функцию $Y = \ln a + X \ln b$.

□ Степенная функция $y = ax^b$.

Представим функцию в виде: $\ln y = \ln a + b \ln x$. Тогда введением новых переменных $Y = \ln y$, $X = \ln x$ получим линейную функцию $Y = \ln a + bX$.

□ Логарифмическая функция $y = a + b \ln x$.

Выравнивание осуществляется заменой переменных $y = Y$, $\ln x = X$. После замены переменных получим: $Y = a + bX$.

□ Дробно-линейная функция $y = \frac{x}{a + bx}$.

Представим функцию в виде: $\frac{x}{y} = a + bx$. Введя новые переменные $Y = \frac{x}{y}$,

$X = x$, получим линейную функцию $Y = a + bX$.

Способ 3. Анализ табличных разностей

Этот способ позволяет выбрать степень многочлена при полиномиальной интерполяции. Если n -е табличные разности функции $y = f(x)$ одинаковы, то степень многочлена будет не выше n .

Рассмотрим этот способ на примере.

Пример 12.1

Предположим, что функция интерполяции является многочленом и представлена в виде табл. 12.1.

Таблица 12.1. Таблица функции $f(x)$

x	1	2	3	4	5	6	7
y	2.2	6.5	12.8	21.1	31.4	43.7	58

В табл. 12.2 приведены значения табличных разностей.

Из табл. 12.2 видно, что вторые табличные разности постоянны. Это значит, что интерполяционный полином должен быть не выше второй степени, т. е.

$$y = a_0 + a_1x + a_2x^2.$$

В результате решения задачи интерполяции получим:

$$y = x^2 + 1.3x = 0.1.$$

Таблица 12.2. Значения табличных разностей

x	1	2	3	4	5	6	7
y	2.2	6.5	12.8	21.1	31.4	43.7	58
$\Delta^{(1)}$		4.3	6.3	8.3	10.3	12.3	14.3
$\Delta^{(2)}$			2	2	2	2	

Способ 4. Использование специальных программ автоматизации интерполяции

Существуют универсальные программные средства, позволяющие выбрать функцию интерполяции, если исходная функция представлена в табличной форме. Такими программными средствами являются: SIMPLE FORMULA, TableCurve, Curve Expert.

Эти программы выдают более тысячи различных функций с указанием их погрешностей. Их описание и примеры использования приведены в [15].

12.1.2. Определение коэффициентов функции интерполяции

Коэффициенты функции $\varphi(x)$ определяются многими способами. Выбор способа зависит от вида функции $\varphi(x)$ (линейная, нелинейная, полиномиальная), требуемой точности интерполяции, возможностей используемой универсальной математической системы.

Система Mathematica имеет богатые возможности решать задачи интерполяции. В дальнейшем они нами рассматриваются и приводятся примеры их реализации.

12.1.3. Определение адекватности функции интерполяции

Адекватность полученного решения определяется величиной погрешности функции $\varphi(x)$. За критерий близости функций наиболее часто принимаются значения абсолютной ε и относительной δ среднеквадратических погрешностей, вычисляемых по формулам:

$$\varepsilon = \sqrt{\frac{\sum_{i=1}^n \Delta_i^2}{n}}, \quad \delta = \frac{\varepsilon}{y_{\min}} \times 100\%. \quad (12.1)$$

В формулах приняты обозначения:

- $\Delta_i = f(x_i) - \varphi(x_i)$ — разность между исходной функцией $f(x_i)$ и функцией интерполяции $\varphi(x_i)$;
- n — число аргументов функции $f(x)$;
- $y_{\min}$ — минимальное значение функции $f(x)$.

Решение адекватно, если $\delta \leq \delta_{\text{доп}}$, где $\delta_{\text{доп}}$ — допустимая погрешность.

Следует иметь в виду, что не всегда полученная в результате моделирования функция $y = \varphi(x)$ является моделью исследуемого объекта или явления, если даже выполняется условие $\delta \leq \delta_{\text{доп}}$. Это будет справедливо только в том случае, если функция $y = f(x)$ получена с высокой точностью.

12.2. Компьютерные технологии интерполяции в среде Mathematica

12.2.1. Интерполяция, точная в узлах

В среде Mathematica интерполяция, точная в узлах, может быть реализована следующими методами:

- универсальным;
- с помощью универсальных функций `InterpolatingPolynomial` и `Inpolation`.

Универсальный метод

Универсальный метод требует решения систем алгебраических уравнений, полученных на основании данных функции $y = f(x)$, представленной в виде таблицы или матрицы.

Технология этого способа состоит в выполнении следующих действий:

1. Составление системы уравнений (предполагается, что вид функции интерполяции известен).
2. Ввод функции решения системы уравнений.
3. Получение решения путем нажатия комбинации клавиш `<Shift>+<Enter>`.

Приведем примеры интерполяции универсальным методом.

Пример 12.2

Функция $y = f(x)$ задана в виде табл. 12.3.

Таблица 12.3. Функция $y = f(x)$ в табличной форме

x	1	2	3	4
y	6.2	4.1	1.9	0.6

Необходимо решить задачу интерполяции, точной в узлах, если функция $y = f(x)$ является полиномом. Так как число узлов $n = 4$, то полином должен быть степени не выше $n - 1$, т. е. $y = a_0 + a_1x + a_2x^2 + a_3x^3$.

Составим систему уравнений:

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

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

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

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

Решение получим с помощью функции $\text{Solve}[F, \{a_0, a_1, a_2, a_3\}]$, где F — исходная система уравнений.

Технология решения и ее результаты приведены на рис. 12.7.

```
F={a0+a1+a2+a3==6.2,
  a0+2 a1+2^2 a2+2^3 a3==4.1,
  a0+3 a1+3^2 a2+3^3 a3==1.9,
  a0+4 a1+4^2 a2+4^3 a3==0.6}
Solve[F,{a0, a1,a2,a3}]
{a0+a1+a2+a3==6.2,a0+2 a1+4 a2+8 a3==4.1,a0+3 a1+9
a2+27 a3==1.9,a0+4 a1+16 a2+64 a3==0.6}
{{a0->7.2,a1->-0.116667,a2->-1.05,a3->0.166667}}
```

Рис. 12.7. Решение примера 12.2

В результате полученного решения интерполяционная формула будет иметь вид:

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

Задачу интерполяции, точной в узлах, в данном случае можно решить с помощью матричного метода решения системы линейных уравнений.

Технология этого способа состоит в следующем:

1. Ввод функции $A \cdot X = B$, где A — матрица коэффициентов, X — матрица неизвестных, B — вектор правых частей системы уравнений.
2. Ввод функции `Solve[%]`.
3. Получение решения путем нажатия комбинации клавиш <Shift>+<Enter>.

Результаты решения приведены на рис. 12.8.

Из рис. 12.7 и 12.8 видно, что решения совпадают.

```
F1={{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}
Solve[%]
{{a0+a1+a2+a3,a0+2 a1+4 a2+8 a3,a0+3 a1+9 a2+27
a3,a0+4 a1+16 a2+64 a3}={6.2,4.1,1.9,0.6}}
{{a0→7.2,a1→-0.116667,a2→-1.05,a3→0.166667}}
```

Рис. 12.8. Решение системы уравнений матричным методом

Проверка достоверности решения задачи интерполяции

Для проверки правильности решения задачи выполним табулирование полученной формулы и результаты сравним с исходными данными. Напомним, что функция табуляции в Mathematica имеет вид:

`Table[f(x), {x, xн, xк, h}]`

где:

- ☐ $f(x)$ — 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, xн, xк, h}}`

Эта функция выдает решение в виде вектора-столбца.

Выполним табулирование полученной функции с помощью `Do{Print[f(x)], {x, xн, xк, h}}`.

Введем функцию:

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

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

```
6.2
4.1
1.90001
0.60002
```

Сравнивая полученный результат с исходными данными, убеждаемся в правильности решения задачи интерполяции.

В предыдущем примере число уравнений и число неизвестных одно и то же. При решении практических задач интерполяции ситуация другая. Почти всегда размеры таблицы исходных данных превосходят порядок алгебраических уравнений, т. е. число узлов интерполяции, а значит, и степень интерполяции меньше, чем размеры таблицы.

В таких случаях ограниченное число узлов интерполяции приходится выбирать из всего диапазона исходных данных. Это обычно приводит к погрешностям интерполяции.

Пример 12.3

Данные эксперимента представлены в табл. 12.4.

Таблица 12.4. Данные эксперимента

x	3	6	9	12	15	18	21	24	27	30
y	26	90	180	300	500	700	1000	1200	1500	2000

Известно, что функцией интерполяции может быть полином второй степени, т. е. $y = a_0 + a_1 x + a_2 x^2$. Необходимо решить задачу интерполяции и проверить правильность решения.

Решение приведено на рис. 12.9.

Сравнение результатов табулирования с исходными данными показывает, что погрешности интерполяции при некоторых значениях x существенны. Это видно также из графиков функции интерполяции и исходных данных, представленных на рис. 12.10.

Из рис. 12.10 видно, что не все точки исходных данных лежат на кривой. Правда, из графиков видно и то, что точки таблицы близко подходят к кривой, и можно сделать вывод, что погрешность интерполяции мала. А какая она в действительности?

```

F={a0+6 a1+6^2 a2==90,
  a0+18 a1+18^2 a2==700,
  a0+27 a1+27^2 a2==1500}
Solve[F,{a0,a1,a2}]
{a0+6 a1+36 a2==90,a0+18 a1+324 a2==700,a0+27 a1+729
a2==1500}
{{a0 -> -135/7, a1 -> 925/126, a2 -> 685/378}}
N[%]
{{a0->-19.2857,a1->7.34127,a2->1.81217}}

y=-135/7+925/126 x+685/378 x^2
Table[y,{x,3,30,3}]
-135/7 + 925 x/126 + 685 x^2/378
{400/21, 90, 1355/7, 6925/21, 3490/7, 700, 19615/21, 8405/7, 1500, 38470/21}
N[%]
{19.0476,90.,193.571,329.762,498.571,700.,934.048,1200.
71,1500.,1831.9}

```

Рис. 12.9. Результаты решения примера 12.3

Вычислим абсолютную и относительную среднеквадратические погрешности интерполяции, воспользовавшись формулами (12.1).

Создадим два вектора: вектор v_1 исходных данных и вектор v_2 функции интерполяции и выполним операции в соответствии с формулой (12.1). Вычисления показаны на рис. 12.11.

Примечание

Для выполнения операции суммирования элементов вектора z^2 необходимо его активизировать и нажать клавишу <Enter>. Полученный вектор отредактировать, заменив фигурные скобки квадратичными и записав в начале выражения функцию Plus. После нажатия комбинации клавиш <Shift>+<Enter> получим сумму квадратов разностей v_1 и v_2 .

Из рис. 12.11 видно, что максимальная относительная погрешность слишком велика и равна примерно 223%. Полученная функция интерполяции не может быть математической моделью изучаемого объекта или явления. Судя по графику, можно предположить, что основной причиной этого является неточность исходных данных.

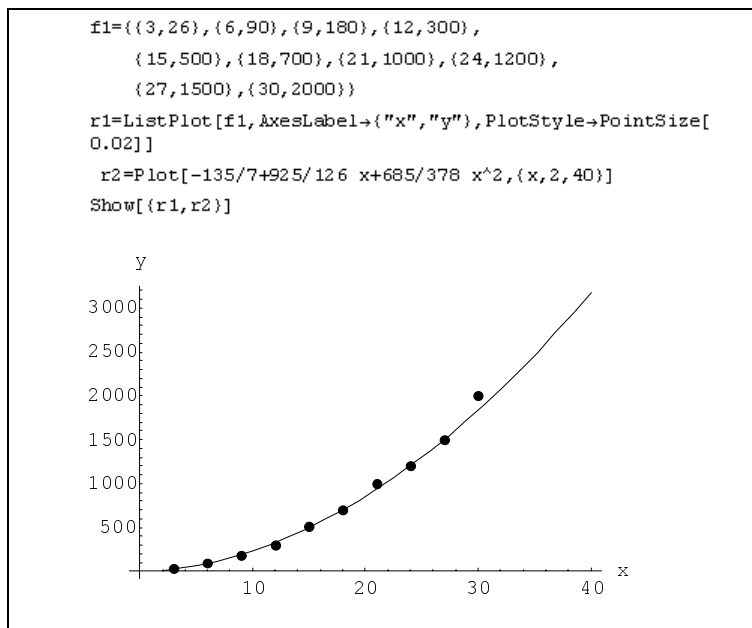


Рис. 12.10. Графики функций исходной и табуляции

Функция *InterpolatingPolynomial*

Интерполяция полиномами хорошо и достаточно просто реализуется в среде Mathematica функцией *InterpolatingPolynomial*. Функция имеет вид:

InterpolatingPolynomial[z, x]

где:

□ z — матрица исходных данных;

□ x — аргумент функции z .

Технология решения задачи очевидна. Приведем примеры.

Пример 12.4

Пусть функция $y = f(x)$ задана в виде табл. 12.5.

Необходимо выполнить полиномиальную интерполяцию с помощью функции *InterpolatingPolynomial*[z, x].


```

v1={26,90,180,300,500,700,1000,1200,1500,2000}
v2={19.0476,90.,193.571,329.762,498.571,700.,
    934.048,1200.71,1500.,1831.9}
z=v1-v2
{26,90,180,300,500,700,1000,1200,1500,2000}

{19.0476,90.,193.571,329.762,498.571,700.,934.048,1200.
71,1500.,1831.9}
{6.9524,0.,-13.571,-29.762,1.429,0.,65.952,-
0.71,0.,168.1}
z^2

{48.3359,0.,184.172,885.777,2.04204,0.,4349.67,0.5041,0
.,28257.6}

Plus[48.33586576000001`,0.`,184.17204099999995`,885.776
644`,2.0420409999999247`,0.`,4349.666303999999`,0.50410
00000000516`,0.`,28257.6099999999968`]
33728.1
Sqrt[33728.1/10]
58.0759
58.0759/26*100
223.369
58.0759/2000*100
2.9038

```

Рис. 12.11. Вычисление погрешностей интерполяции

Таблица 12.5. Табличное задание функции $y = f(x)$

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

Решение задачи сводится к реализации следующих процедур:

1. Ввод матрицы исходных данных

```
z = {{1,1},{2,8},{3,27},{4,64},{5,125}}
```

2. Ввод функции `InterpolatingPolynomial[z,x]`.

3. Нажатие комбинации клавиш <Shift>+<Enter> для получения решения.

Решение на экране будет иметь вид, показанный на рис. 12.12.

Из рис. 12.12 следует, что функцией интерполяции будет: $y = x^3$. Решение является точным. Для его получения пришлось упростить результат интерполяции, использовав функцию упрощения `simplify`.

```

z={ {1,1},{2,8},{3,27},{4,64},{5,125}}
InterpolatingPolynomial[z,x]
{{ {1,1},{2,8},{3,27},{4,64},{5,125}}
1+(-1+x) (7+(-2+x) (3+x))
Simplify[%]
x3

```

Рис. 12.12. Интерполяция с помощью функции InterpolatingPolynomial[z,x]

Функция InterpolatingPolynomial решает задачи интерполяции также в случае не равноотстоящих узлов.

Элементами вектора z может быть набор функций $f_1(x)$, $f_2(x)$, ... Тогда решение будет выдаваться в виде многочлена с сохранением исходного вида выражений $f_1(x)$, $f_2(x)$, ...

```

z={1,Sin[x],1/x,Exp[-x]}
z1=InterpolatingPolynomial[z,x]
{1, Sin[x], 1/x, e-x}
1+(-1+x) (-1+Sin[x]+(-2+x) (1/2 (1+1/x-2 Sin[x]) +
1/3 (-3+x) (-1/2 (1+1/x-2 Sin[x]) + 1/2 (e-x-2/x+Sin[x]))))
Table[z1,{x,1,4}]
N[%]
{1, Sin[2], 1+2 (-1+1/2 (4/3-2 Sin[3]) + Sin[3]),
3 (-1+Sin[4]+2 (1/2 (5/4-2 Sin[4]) +
1/3 (1/2 (-1/2+1/e4+Sin[4]) + 1/2 (-5/4+2 Sin[4]))))}
{1.,0.909297,0.333333,0.0183156}

```

Рис. 12.13. Решение задачи примера 12.5

Пример 12.5

Функция z задана в виде табл. 12.6.

В последней строке таблицы приведены значения элементов вектора z при заданных значениях аргумента x.

Таблица 12.6. Вид функций — элементов вектора z

x	1	2	3	4
y	1	$\sin x$	$1/x$	e^{-x}
$y(x)$	1	0.909297	0.333333	0.0183156

Процедуры решения задачи с помощью функции `InterpolatePolynomial` приведены на рис. 12.13.

Из рис. 12.13 видно, что решение совпадает с исходными данными. Формула интерполяции получена верно. Ее, конечно, можно сократить с помощью функций упрощения выражений.

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, x_k — диапазон аргументов функции интерполяции. Если теперь ввести значение аргумента из диапазона x_n — x_k , то откликом будет значение функции при заданном значении аргумента.

Технология решения задачи проста и состоит в выполнении следующих процедур:

1. Ввод матрицы исходных данных с присвоением ей уникального имени.
2. Ввод функции `Interpolation[data]`.
3. Получение решения путем нажатия комбинации клавиш <Shift>+<Enter>.

Матрице исходных данных имя можно не присваивать, а вводить ее непосредственно в функцию `Interpolation[data]`.

Рассмотрим технологию на примерах.

Пример 12.6

Функция $y = f(x)$ задана в виде табл. 12.7.

Необходимо определить значение функции при x , равном 5.8 и 18.5, и проверить достоверность полученных решений.

Решение задачи приведено на рис. 12.14.

Таблица 12.7. Табличное задание функции $y = f(x)$

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

```

f={{2,1},{3,2.5},{8,4.6},{12,3.2},{20,1.6}}
y=Interpolation[f]
      {{2,1},{3,2.5},{8,4.6},{12,3.2},{20,1.6}}
      InterpolatingFunction[{{2.,20.}},<>]
y[5.8]
      4.56372
y[18.5]
      1.18763
Table[y[{2,3,8,12,20}]]
      {1.,2.5,4.6,3.2,1.6}

```

Рис. 12.14. Решение примера 12.6

Из рис. 12.14 видно, что функция `Interpolation[data]` решает задачи интерполяции методом точным в узлах. Результатами табулирования являются точные значения функции в узлах интерполяции.

12.2.2. Интерполяция нелинейными функциями

Если интерполяционной является нелинейная функция, то для определения ее коэффициентов методами точными в узлах используются следующие два способа:

- ☐ создание и решение системы нелинейных уравнений;
- ☐ линеаризация нелинейной интерполяционной функции путем преобразования координат.

Рассмотрим оба способа.

Способ 1. Решение системы нелинейных уравнений

Технология интерполяции этим способом определяется видом функции интерполяции и способом решения системы нелинейных уравнений.

Покажем технологию на примерах.

Пример 12.7

Функция $y = f(x)$ представлена в виде табл. 12.8.

Таблица 12.8. Табличное представление функции $y = f(x)$

x	1	2	3	4	5	6	7	8
y	7.6	16	33	71	156	341	750	1650

Известно, что функцией интерполяции может быть функция $y = ab^x + c$. Необходимо определить неизвестные a , b , c и погрешность функции интерполяции.

```

f={a b+c==7.6, a b^4+c==71,a b^7+c==750}
FindRoot[f,{a,2.5},{b,3},{c,1.5}]
{a b + c == 7.6, a b^4 + c == 71, a b^7 + c == 750}
{a→2.96224,b→2.20425,c→1.0705}
y=2.96224*2.20425^x+1.0705
Table[y,{x,1,8}]
1.0705+2.96224 2.20425^x

{7.60002,15.4632,32.7956,71.0005,155.214,340.841,75
0.009,1651.92}
v1={7.5,16,33,71,156,341,750,1650}
v2={7.6,15.4632,32.7956,71,155.214,
340.841,750,1651.92}
(v1-v2)^2
{7.5,16,33,71,156,341,750,1650}

{7.6,15.4632,32.7956,71,155.214,340.841,750,1651.92
}

{0.01,0.288154,0.0417794,0,0.617796,0.025281,0,3.68
64}
Plus[0.0099999999999999929`,0.288154239999999945`,0.0
41779359999999988`,0,0.61779600000000021`,0.025280999
999997396`,0,3.68640000000002792`]

4.66941
Sqrt[Out[33]/8]
0.763987
Out[34]/7.6 100
10.0525

```

Рис. 12.15. Решение задачи примера 12.7

Выберем три следующие координаты функции: (1, 7.6), (4, 71), (7, 750) и составим систему уравнений:

$$ab^1 + c = 7.6$$

$$ab^4 + c = 71$$

$$ab^7 + c = 750$$

Решим эту систему нелинейных уравнений с помощью функции FindRoot при следующих начальных значениях неизвестных: $a=2.5$, $b=3$, $c=1.5$.

Решение приведено на рис. 12.15.

Из рис. 12.15 видно, что математической моделью объекта является следующая функция интерполяции:

$$y = 2.96 \cdot 2.2^x + 1.07$$

Абсолютная и относительная среднеквадратические погрешности вычислены в соответствии с формулой (12.1). Максимальная относительная погрешность равна 10%.

Способ 2. Линеаризация нелинейной функции

В системе Mathematica аппроксимация нелинейными функциями может быть сведена к решению линейных уравнений путем преобразования координат.

Выравнивание функций осуществляется путем их преобразования в линейную функцию методом замены переменных. Наиболее просто эти преобразования осуществляются, если нелинейные функции типа: степенная, логарифмическая, дробно-линейная, показательная.

Технология интерполяции методом линеаризации нелинейных функций состоит в выполнении следующих действий:

1. Приведение функции интерполяции в линейную форму.
2. Преобразование матрицы исходных данных в матрицу новых переменных.
3. Образование системы линейных уравнений.
4. Решение системы линейных уравнений.
5. Образование функции интерполяции.
6. Проверка адекватности полученной модели.

Рассмотрим методику на примере.

Пример 12.8

Функция $y = f(x)$, полученная в результате эксперимента, представлена в виде табл. 12.9.

Таблица 12.9. Экспериментальные данные исследуемого объекта

x	1	3	5	7	9	11	13	15
y	2.5	7.8	18.7	28.5	39	50	61.7	73.8

Необходимо найти математическую модель исследуемого объекта, если известно, что функцией интерполяции может быть показательная функция $y = ax^b$.

Решение задачи показано на рис. 12.16.

```

f1={1.,3.,5.,7.,9.,11.,13.,15.}
f2={2.5,7.8,18.7,28.5,39.,50.,61.7,73.8}
Log[f1]
{1.,3.,5.,7.,9.,11.,13.,15.}
{2.5,7.8,18.7,28.5,39.,50.,61.7,73.8}

{0.,1.09861,1.60944,1.94591,2.19722,2.3979,2.56495,
2.70805}
Log[f2]

{0.916291,2.05412,2.92852,3.3499,3.66356,3.91202,4.
12228,4.30136}
Solve[{2.054==A+1.09861 b,3.912==A+2.398 b},{A,b}]
{{A->0.483096,b->1.4299}}
a=Exp[0.483096]
1.62109
y=1.62 x^1.43
Table[{x,y},{x,1,15,2}]
1.62 x1.43
{{1,1.62},{3,7.79468},{5,16.1824},{7,26.1821},{9,37
.5044},{11,49.9697},{13,63.4533},{15,77.862}}

```

Рис. 12.16. Решение задачи примера 12.9

Прокомментируем рис. 12.16.

В первых двух строках рисунка находятся векторы исходных данных с именами **f1** и **f2**. За ними следуют те же векторы в логарифмическом масштабе. Для вычисления коэффициентов **a** и **b** линейной функции $\ln y = \ln a + b \ln x$ за исходные данные взяты координаты (3, 7), (11, 50), которые в логарифмическом масштабе имеют значения: (1.09861, 2.05412), (2.3979, 3.91202). Тогда система линейных уравнений имеет вид:

$$2.05412 = A * 1.098616^b$$

$$3.91202 = A * 2.3979^b$$

где $A = \ln a$.

Система решается с помощью функции `solve`. В результате решения получены следующие значения коэффициентов: $a=0.483096$, $b=1.4299$. Тогда

$$a = e^A = 1.62109$$

а функция интерполяции имеет вид:

$$y = 1.62 x^{1.43}$$

Коэффициенты a и b функции интерполяции округлены до двух значащих цифр после запятой (точки).

Адекватность модели доказана табулированием функции интерполяции с помощью функции `Table`.

Сравнение результатов табулирования с исходными данными позволяет сделать выводы, что задача решена верно (значения функций практически совпадают в узлах интерполяции $x=3$ и $x=11$), а функция интерполяции является математической моделью изучаемого объекта.

С целью сравнения результатов двух способов интерполяции нами решена задача интерполяции нелинейной функцией $y = ax^b$. Взяты те же узлы интерполяции $x=3$ и $x=11$ и решена система нелинейных уравнений с помощью функции `FindRoot` при начальных приближениях $a_0 = 2$, $b_0 = 1$.

Решение имеет вид:

$$\mathbf{f} = \{7.8 == a \cdot 3^b, 50 == a \cdot 11^b\}$$

$$\mathbf{FindRoot}[\mathbf{f}, \{\mathbf{a}, 2\}, \{\mathbf{b}, 1\}]$$

$$\{7.8 == 3^b a, 50 == 11^b a\}$$

$$\{a \rightarrow 1.62121, b \rightarrow 1.42994\}$$

Из решения видно, что коэффициенты функции интерполяции практически совпадают со случаем линеаризации уравнения.

12.3. 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:

$$\mathbf{Fit}[\{\{M\}\}, \{X\}, x]$$

где:

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

Технология интерполяции с помощью функции `Fit[{{M}}, {X}, x]` состоит в выполнении следующих действий:

1. Ввод матрицы исходных данных с присвоением ей уникального имени, например, M .
2. Ввод базисных переменных X .
3. Ввод функции `Fit[{{M}}, {X}, x]`.
4. Получение решения одновременным нажатием клавиш `<Shift>+<Enter>`.

Приведем примеры решения задач интерполяции по приведенной технологии.

Пример 12.9

Функция $y(x)$ задана в виде табл. 12.10.

Таблица 12.10. Табличное задание функции $y(x)$

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]`

Решение показано на рис. 12.17.

Примечание

В системах Mathematica 3, 4 свободный член базисных переменных кодировался символьными переменными, например, $X=\{a, x, x^2, x/(1+x), \text{Exp}[x]\}$.

Из рис. 12.17 видно, что коэффициент при базисной переменной `Exp[x]` мал. По-видимому, эта базисная переменная не оказывает существенного влияния на вид функции интерполяции. Проверим наше предположение, для чего решим задачу интерполяции, исключив базисную переменную `Exp[x]`.

Решение приведено на рис. 12.18.

```

M={ {1,3.5} , {3,6.7} , {4,4.2} , {7,2.8} , {10,1.2} }
X={ 1,x,x^2,x/(1+x) ,Exp [x] }
Fit [M,X,x]
  { {1,3.5} , {3,6.7} , {4,4.2} , {7,2.8} , {10,1.2} }
  { 1, x, x^2,  $\frac{x}{1+x}$ ,  $e^x$  }
  -33.1913 - 0.00090638  $e^x$  - 16.2137 x + 1.22525 x^2 +  $\frac{103.364 x}{1+x}$ 

```

Рис. 12.17. Решение задачи примера 12.9

```

M={ {1,3.5} , {3,6.7} , {4,4.2} , {7,2.8} , {10,1.2} }
X={ 1,x,x^2,x/(1+x) }
Fit [M,X,x]
  { {1,3.5} , {3,6.7} , {4,4.2} , {7,2.8} , {10,1.2} }
  { 1, x, x^2,  $\frac{x}{1+x}$  }
  -9.42867 - 3.57786 x + 0.166826 x^2 +  $\frac{32.8097 x}{1+x}$ 

```

Рис. 12.18. Решение задачи примера 12.9 с измененным базисом

Сравнение функций интерполяции показывает, что наше предположение было ошибочным: функции сильно отличаются друг от друга.

Пример 12.10

Зависимость высоты сосны от ее возраста, полученная в результате экспериментальных исследований, приведена в табл. 12.11. Необходимо найти математическую модель роста сосны и проверить адекватность модели.

Таблица 12.11. Зависимость высоты сосны h от ее возраста L

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

Предположим, что функция интерполяции представляет собой полином второй степени, т. е. $h=a+bL+cL^2$. Тогда базисными переменными будут: a , x , x^2 . Применим интерполяцию, приближенную в узлах, и функцию **Fit**.

Технология определения функции интерполяции и ее результаты приведены на рис. 12.19.

```

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}}
X={1,x,x^2}
Fit[M,X,x]
{{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},{1
30,36.3}}
{1, x, x^2}
10.0635+0.435307 x - 0.00182443 x^2
Table[Out[25],{x,20,130,10}]
{18.0398,21.4807,24.5567,27.2678,29.614,31.5953,33.
2117,34.4633,35.3499,35.8717,36.0286,35.8206}

```

Рис. 12.19. Решение задачи примера 12.10

```

f1={{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}}

{{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},{1
30,36.3}}

f2=10.0635+0.435307 x-0.00182443 x^2
10.0635+0.435307 x - 0.00182443 x^2

r1=ListPlot[f1,AxesLabel→{"h", "L"},PlotStyle→PointS
ize[0.02]]
r2=Plot[f2,{x,20,130}]
Show[{r1,r2}]

```

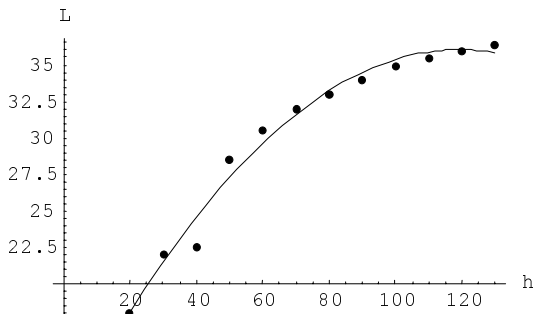


Рис. 12.20. Графическое представление функций примера 12.10

Табулирование функции высоты сосны от возраста показало, что, на первый взгляд, погрешность модели незначительна. Вычислим ее, предварительно представив функцию в виде графиков.

Графики исходной функции (в виде точек) и функции, полученной в результате интерполяции (сплошная кривая), приведены на рис. 12.20.

Из рис. 12.20 видно, что в некоторых узлах имеются погрешности интерполяции. Оценим эти погрешности, вычислив абсолютную и относительную сред-

```

v1={18,22,22.5,28.5,30.5,32,33,34,35,35.5,36,36.3}
{18,22,22.5,28.5,30.5,32,33,34,35,35.5,36,36.3}
f2=10.0635+0.435307 x-0.00182443 x^2
Table[f2,{x,20,130,10}]

{18.0399,21.4807,24.5567,27.2678,29.614,31.5953,33.
2117,34.4632,35.3499,35.8717,36.0285,35.8205}

v2={18.0399,21.4807,24.5567,27.2678,29.614,31.5953,
33.2117,34.4632,35.3499,35.8717,36.0285,35.8205}
(v1-v2)^2

{18.0399,21.4807,24.5567,27.2678,29.614,31.5953,33.
2117,34.4632,35.3499,35.8717,36.0285,35.8205}

{0.00159201,0.269672,4.23001,1.51832,0.784996,0.163
782,0.0448169,0.214554,0.12243,0.138161,0.00081225,
0.22992}

Plus[0.0015920099999999506`,0.269672490000000126`,4.
2300148899999997`,

1.5183168399999971`,0.7849959999999987`,0.163782089
9999986`,

0.0448168900000000185`,0.214554240000000045`,0.122430
00999999867`,

0.1381608899999978`,0.00081225000000000615`,0.229920
24999999472`]

7.71907
Sqrt[Out[58]/12]
0.802032
Out[59]/18 100
4.45573

```

Рис. 12.21. Вычисление погрешностей интерполяции

неквадратические погрешности. Для этого воспользуемся формулами (12.1). Решение представлено на рис. 12.21.

Из рис. 12.21 видно, что погрешность интерполяции не превышает 5%. Можно считать, что математическая модель роста сосны найдена и представляет собой полином второй степени.

12.4. Паде-аппроксимация

Аппроксимация Паде служит для интерполяции функции, заданной в аналитическом виде, дробно-рациональной функцией.

Встроенная функция Паде-аппроксимации находится в пакете расширения

`<<calculus\pade.m`

Функция Паде имеет вид:

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

где:

- ☐ $f(x)$ — 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.

Технология интерполяции очевидна и не требует пояснений. Приведем примеры.

Пример 12.11

Необходимо выполнить Паде-аппроксимацию функции $y = \sin x$ вблизи $x = 0$ при $n_1 = 3$ и $n_2 = 4$. Убедиться в достоверности аппроксимации путем построения графиков.

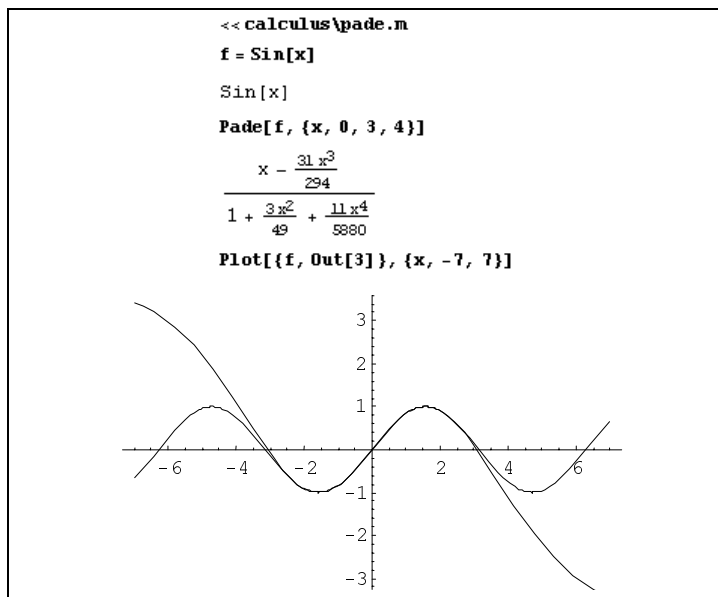
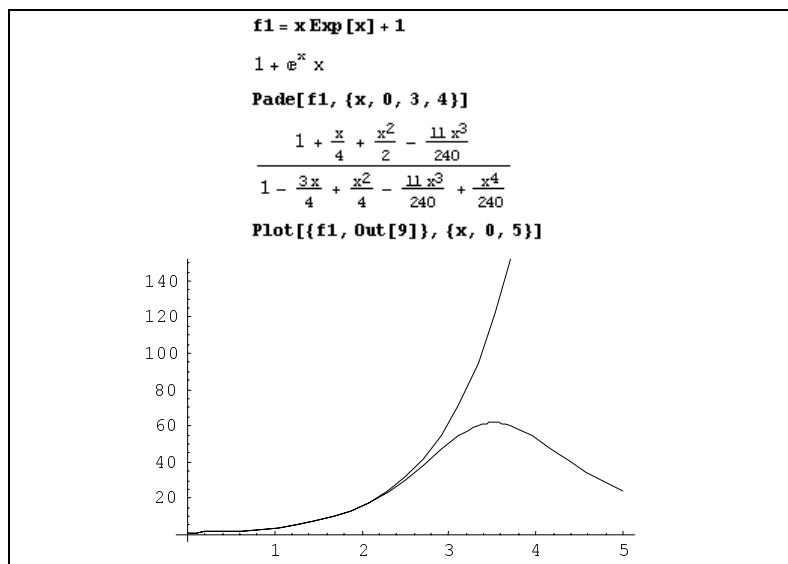
Процедуры аппроксимации в среде Mathematica имеют вид, приведенный на рис. 12.22.

Из рис. 12.22 видно, что аппроксимация выполнена верно. В диапазоне от $x = -3$ до $x = 3$ функции совпадают.

Аппроксимация Паде не имеет ограничений на вид исходной функции. Следующий пример это иллюстрирует.

Пример 12.12

Необходимо выполнить Паде-аппроксимацию функции $y = xe^x + 1$ вблизи $x = 0$ при $n_1 = 3$ и $n_2 = 4$. Убедиться в достоверности аппроксимации путем построения графиков.

Рис. 12.22. Паде-аппроксимация функции $y = \sin x$ Рис. 12.23. Паде-аппроксимация функции $y = xe^x + 1$

Процедура аппроксимации в среде Mathematica имеет вид, приведенный на рис. 12.23.

12.5. Функции аппроксимации в пакетах расширения

Система Mathematica содержит ряд функций аппроксимации, которые находятся в пакетах расширения `Statistics` и `NumericalMath`.

Рассмотрим возможности этих функций и приведем примеры.

12.5.1. Линейная аппроксимация

В подпакете `LinearRegression` находятся функции линейной аппроксимации, которые являются дополнениями к функции `Fit`, включенной в ядро системы.

Основными функциями подпакета являются:

- ☐ `Regress[F, {I, x, x^2}, x]` — решает задачу аппроксимации данных F , используя квадратичную модель вида: $y = a + bx + cx^2$;
- ☐ `Regress[F, {I, x1, x2, x1x2}, {x1, x2}]` — решает задачу двухпараметрической линейной аппроксимации данных F ;
- ☐ `Regress[F, {f1, f2, ... }, x]` — осуществляет линейную аппроксимацию данных F функциями $f_1, f_2, \dots$

Пример 12.13

Необходимо получить модель объекта, представленного следующей матрицей: $\{\{1, 1\}, \{2, 4\}, \{3, 9\}, \{4, 16\}, \{5, 25\}\}$. Предполагается, что функция аппроксимации имеет вид: $y = a + bx + cx^2$.

Сравнить результаты решения с результатами аппроксимации с помощью функции `Fit`.

В данном случае функция `Regress` будет иметь вид:

```
Regress[M, {1, x, x^2}, x]
```

Решение приведено на рис. 12.24.

Из рис. 12.24 следует, что решением является функция $y = x^2$. Свободный член и коэффициент при x практически равны нулю (столбик `Estimate`). Это подтверждается также результатом решения задачи аппроксимации с помощью функции `Fit`.

```

<<Statistics LinearRegression
M = {{1, 1}, {2, 4}, {3, 9}, {4, 16}, {5, 25}}
Regress[M, {1, x, x^2}, x]
{{1, 1}, {2, 4}, {3, 9}, {4, 16}, {5, 25}}
{ParameterTable → x
  x^2
  SE
  Estimate
  TStat
  PValue
  RSquared → 1., AdjustedRSquared → 1., EstimatedVariance → 9.54522 × 10-29,
  ANOVATable →
    Model
    Error
    Total
    SumOfSq
    DF
    MeanSq
    FRatio
    PValue
  Fit[M, {1, x, x^2}, x]
  9.5618 × 10-15 - 4.55191 × 10-15 x + 1. x^2
}

```

	SE	Estimate	TStat	PValue
1	2.09542×10^{-14}	3.16232×10^{-15}	0.150916	0.893889
x	1.59685×10^{-14}	-9.65581×10^{-16}	-0.0604678	0.957282
x ²	2.61113×10^{-15}	1.	3.82976×10^{14}	0.

	SumOfSq	DF	MeanSq	FRatio	PValue
Model	374.	2	187.	1.9591×10^{30}	0.
Error	1.90904×10^{-28}	2	9.54522×10^{-29}		
Total	374.	4			


```

Fit[M, {1, x, x^2}, x]
9.5618 × 10-15 - 4.55191 × 10-15 x + 1. x^2

```

Рис. 12.24. Решение примера 12.13

Отличие в решении очевидно — это полнота анализа результатов, которые, надеемся, ясны без объяснений.

12.5.2. Нелинейная аппроксимация

В подпакете `NonlinearFit` находятся следующие функции нелинейной аппроксимации:

`NonlinearFit[F, f, x, A]`

`NonlinearRegress[F, f, x, A]`

где:

- ☐ `F` — данные, представленные в виде списков, в частности в виде матрицы;
- ☐ `f` — вид (формула) функции аппроксимации;
- ☐ `x` — аргумент функции аппроксимации;
- ☐ `A` — параметры функции `f`, которые определяются в результате аппроксимации.

Функции нелинейной аппроксимации имеют следующие назначения:

- ☐ `NonlinearFin[F, f, x, A]` — выдает функцию аппроксимации, реализованную над данными `F` при известном виде функции аппроксимации `f(x)` с параметрами `A`;
- ☐ `NonlinearRegress[F, f, x, A]` — решает ту же задачу, что и функция `NonlinearFin[F, f, x, A]`, только с выдачей результата в виде диагностики решения, подобно тому, как это было в случае линейной аппроксимации с помощью функции `Regress`.

Вид функции аппроксимации $f(x)$ задается пользователем. Она может быть как линейной, так и нелинейной.

Приведем пример решения задачи нелинейной аппроксимации.

Пример 12.14

Дана функция, представленная в виде следующей матрицы:

$M = \{\{1, 2\}, \{2, 4\}, \{2, 12\}, \{4, 21\}, \{5, 32\}\}$

Известно, что функция аппроксимации нелинейна и имеет вид: $y = ab^x$.

Необходимо решить задачу аппроксимации и проверить достоверность решения путем построения графиков.

В данном случае $F=M$, $y = ab^x$, $A=a, b$. Тогда функция `NonlinearFit` будет иметь вид:

`NonlinearFit[M, a b^x, {x}, {a,b}]`

Решение приведено на рис. 12.25.

```

<<Statistics`NonlinearFit`
M={{1,2},{2,4},{3,12},{4,21},{5,32}}
s1=NonlinearFit[M,a b^x,{x},{a,b}]
{{1,2},{2,4},{3,12},{4,21},{5,32}}
2.05068 1.74287^x
r1=Plot[s1,{x,1,6}]
r2=ListPlot[M,PlotStyle->{PointSize[0.02]}]
Show[r1,r2]

```

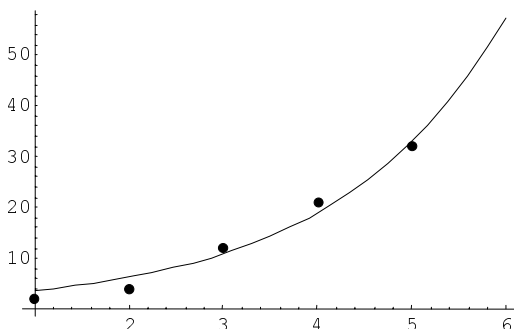


Рис. 12.25. Решение примера 12.14

Из рис. 12.25 видно, что полученная функция аппроксимации проходит вблизи координат точек исходных данных. Она может быть математической моделью изучаемого объекта.

12.5.3. Полиномиальная аппроксимация

В пакете NumericalMath имеется подпакет PolynomialFit, в котором находится функция полиномиальной аппроксимации, имеющая вид:

PolynomialFit[F, n]

где:

- F — данные, представленные в виде списков, например, в виде вектора или матрицы;
- n — степень возвращаемого полинома.

Если данные F представлены в виде вектора, то они являются значениями ординат функции $y = f(x)$. Абсциссы формируются автоматически с шагом, равным единице. Если данные представлены в виде матрицы, то элементы матрицы являются координатами точек функции $y = f(x)$.

Функция PolynomialFit[F, n] решает задачу полиномиальной аппроксимации по методу наименьших квадратов.

Приведем примеры полиномиальной аппроксимации.

Пример 12.15

Необходимо представить функцию, ординатами которой является вектор $V = (1, 3, 7, 13, 22, 34, 50, 75)$, полиномом четвертой степени. Адекватность модели установить с помощью сравнения графиков исходной функции и модели.

В данном случае функция полиномиальной регрессии будет иметь вид:

`PolynomialFit[V, 4]`

Решение приведено на рис. 12.26.

```
<< NumericalMath`PolynomialFit`
V = {1, 3, 7, 13, 22, 34, 50, 75}
S = PolynomialFit[V, 4]
{1, 3, 7, 13, 22, 34, 50, 75}
FittingPolynomial[<>, 4]
S[6]
33.6066
S[{2, 5, 7}]
{2.77435, 21.8512, 50.415}
Expand[S[x]]
3.33929 - 4.89295 x + 3.02746 x2 - 0.427399 x3 + 0.0331439 x4
r1 = Plot[S[x], {x, 1, 8}]
r2 = ListPlot[V, PlotStyle -> {PointSize[0.02]}]
Show[{r1, r2}]
```

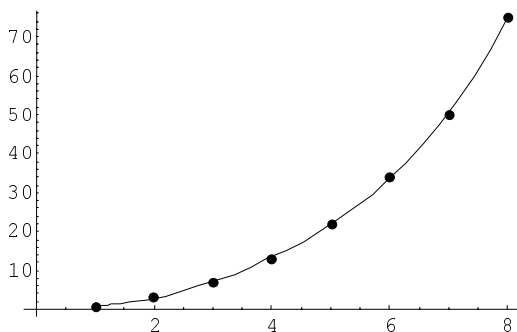


Рис. 12.26. Полиномиальная аппроксимация функции, заданной вектором

Пример 12.16

Необходимо найти математическую модель роста сосны по данным, приведенным в табл. 12.11. Предполагается, что моделью является полином второй степени, т. е. $n = 2$. Решение приведено на рис. 12.27.

```
<< NumericalMath`PolynomialFit`
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}}

{{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}}

L=PolynomialFit[M,2]
FittingPolynomial[<>, 2]
L[{20,70,120}]
{18.0398,31.5953,36.0286}
Expand[L[h]]
10.0635+0.435307 h - 0.00182443 h^2
```

Рис. 12.27. Математическая модель роста сосны

Сравнение полученных данных с данными примера 12.10 показывает, что решения полностью совпадают. Тогда что же нового дает функция `PolynomialFit[F, n]` по сравнению с функцией `Fit[{{M}}, {X}, x]`? В нашем примере разными являются лишь формы ответов.

12.5.4. Сплайн-интерполяция

В подпакете `SplineFit` пакета `NumericalMath` имеется функция, реализующая сплайн-интерполяцию. Интерполяция кубическими сплайнами обеспечивает высокую точность математической модели, в чем мы убедимся на примерах.

Функция имеет вид:

`SplineFit[F, type]`

где:

- ☐ `F` — данные, представленные в виде матрицы;
- ☐ `type` — тип аппроксимации, по умолчанию — это аппроксимация кубическими сплайнами (`Cube`). Допускается аппроксимация сплайнами `Bezier` и `CompositeBezier`.

Пример 12.17

Функция $y = f(x)$ представлена в виде следующей матрицы: $M = ((1, 13), (2, 7.4), (3, 2.2), (4, 4.4), (5, 9.5), (6, 16))$.

Необходимо получить математическую модель функции $y = f(x)$, используя кубическую сплайн-интерполяцию. Достоверность решения проверить графически.

Решение приведено на рис. 12.28.

```
<<NumericalMath`SplineFit`
M = {{1, 13}, {2, 7.4}, {3, 2.2}, {4, 4.4}, {5, 9.5}, {6, 16}}
{{1, 13}, {2, 7.4}, {3, 2.2}, {4, 4.4}, {5, 9.5}, {6, 16}}
Z1 = SplineFit[M, Cubic]
SplineFunction[Cubic, {0., 5.}, <>]
Z1[0]
{1, 13}
Z1[1]
Z1[5]
{6., 16.}
Z1[3.53]
{4.53, 6.92269}
r1 = ParametricPlot[Z1[x], {x, 0, 5}]
r2 = ListPlot[M, PlotStyle -> {PointSize[0.02]}]
Show[r1, r2]
```

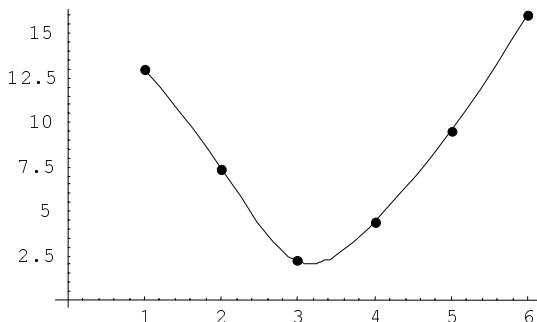


Рис. 12.28. Интерполяция кубическими сплайнами

Из рис. 12.28 видно, что решение получено не в явном виде, т. е. аналитическое выражение функции интерполяции отсутствует. Это является существенным недостатком функции `splineFit[F, type]`.

Из графика следует, что интерполяция выполнена с идеальной точностью.

При этом следует иметь в виду, что полученная математическая модель может быть моделью изучаемого явления только в том случае, если исходные данные удовлетворяют условиям точности. При сплайн-интерполяции сглаживания неточностей исходных данных нет.

12.6. Многопараметрическая интерполяция

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 этой теории свойства объекта изучаются по значениям откликов при воздействии на объект n факторов. Значения откликов определяются в процессе эксперимента. В результате эксперимента получают функции $y = f(x_1, x_2, \dots, x_n)$. 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 функции 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 [15].

Система Mathematica позволяет решать задачи интерполяции многих переменных.

Технология интерполяции функций многих переменных в системе Mathematica наиболее проста по сравнению с другими математическими системами. Она реализуется функцией `Fit`, имеющей вид:

`Fit[M, X, A]`

где:

- 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. Перечень аргументов A заключается в фигурные скобки, аргументы отделяются друг от друга запятыми.

Технологию интерполяции рассмотрим на примере интерполяции функции двух переменных.

Пример 12.18

Матрица исходных данных представлена табл. 12.12.

Таблица 12.12. Результаты экспериментальных исследований объекта

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
Q , м ³	36	42	58	71	85	97	112	132	153	187	201	221

В табл. 12.12 приняты следующие обозначения:

- L — возраст древостоя;
- H — средняя высота дерева;
- Q — запас древесины на участке.

Необходимо найти зависимость $Q = f(L, H)$.

В качестве функции регрессии возьмем следующую квадратичную функцию:

$$Q = f(L, H) = b_0 + b_1 L + b_2 L^2 + b_3 H + b_4 H^2 + b_5 LH.$$

Решение задачи приведено на рис. 12.29.

Результаты табулирования функции приведены в табл. 12.13.

Таблица 12.13. Сравнение исходных данных и результатов интерполяции

Q	36	42	58	71	85	97	112	132	153	187	201	221
Z	34.9	45.8	57	69	82	98	116	135.3	155.8	177.7	200.5	224

```

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} }
{ {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={1,x,x^2,y,y^2,x*y}
Z=Fit[M,X,{x,y}]

{1, x, x^2, y, y^2, x y}
-35.1074`-0.483089 x + 0.00179859 x^2 +
6.56929 y + 0.0654316 x y - 0.1940335 y^2
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
{20,30,40,50,60,70,80,90,100,110,120,130}
{18,22,22.5,28.5,30.5,32,33,34,35,35.5,36,36.3}
{34.886,45.8157,56.9148,68.0958,81.9858,97.9833,115.98,135.257,155.814,177.705,200.513,224.048}

```

Рис. 12.29. Решение задачи примера 12.18

Из табл. 12.13 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 $\delta_{\min} = \frac{\varepsilon}{221} \cdot 100\% = 1.7\%$.

Расчеты погрешностей не приводятся ввиду их очевидности.

Литература

1. Аладьев В., Шишаков М. Автоматизированное рабочее место математика. — М.: Лаборатория базовых знаний, 2000.
2. Аладьев В. З., Шишаков М. Л. Введение в среду пакета Mathematica 2.2. — М.: Филинь, 1997.
3. Бабенко К. Н. Основы численного анализа. — М.: Наука, Физматлит, 1986.
4. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы. — М.: Наука, Физматлит, 1987.
5. Воробьева Г. Н., Данилова А. Н. Практикум по вычислительной математике. — М.: Высшая школа, 1990.
6. Воробьев Е. М. Введение в систему "Mathematica". — М.: Финансы и статистика, 1998.
7. Градштейн И. С., Рыжик И. М. Таблицы интегралов, сумм, рядов и произведений // Издание 4. — М.: Государственное издательство физико-математической литературы, 1962.
8. Гутер Р. С., Резниковский П. Т. Программирование и вычислительная математика // Выпуск 2. — М.: Наука, 1971.
9. Демидович Б. П., Марон И. А. Основы вычислительной математики. — М.: Наука, 1970.
10. Дьяконов В. Mathematica 4 // Учебный курс. — СПб.: Питер, 2001.
11. Дьяконов В. П. Системы символьной математики Mathematica 2 и Mathematica 3. — М.: СК Пресс/PC Week, 1998.
12. Дьяконов В. П. Mathematica 3/4 с пакетами расширения. — М.: Нолидж, 2000.
13. Капустина Т. В. Компьютерная система Mathematica 3.0 для пользователей. — М.: Солон-Р, 1999.
14. Копченова Н. В., Марон И. А. Вычислительная математика в примерах и задачах. — М.: Наука, 1972.
15. Половко А. М., Бутусов П. Н. Интерполяция. Методы и компьютерные технологии их реализации. — СПб.: БХВ-Петербург, 2004.

16. Половко А. М. Derive для студента. — СПб.: БХВ-Петербург, 2005.
17. Половко А. М., Бутусов П. Н. MATLAB для студента. — СПб.: БХВ-Петербург, 2005.
18. Половко А. М., Ганичев И. В. Mathcad для студента. — СПб.: БХВ-Петербург, 2006.
19. Тюрин Ю. Н., Макаров А. Л. Анализ данных на компьютере. — М.: Финансы и статистика, 1995.
20. Форсайт Дж., Малькольм М., Моулер К. Машинные методы математических вычислений. — М.: Мир, 1980.

Предметный указатель

А

Аппроксимация 324, 343

- ◇ линейная 350
- ◇ Паде 348

Б

Бета-функция 185

В

Вектор 147

- ◇ выделение элемента 150
 - ◇ выявление структуры 154
 - ◇ генерация 148, 149
 - ◇ добавление элемента 156
 - ◇ изменение порядка элементов 156
 - ◇ комбинирование 156, 159
 - ◇ создание 160
 - ◇ удаление элемента 156
- Выражение 50
- ◇ преобразование 55
 - ◇ тригонометрическое 63

Г

Гамма-функция 174

График:

- ◇ в виде гистограммы 96
 - ◇ в логарифмическом масштабе 92
 - ◇ в полярной системе координат 94
 - ◇ выбор стиля 86
 - ◇ контурный 102
 - ◇ поверхности 102
 - ◇ точечный 81
- Графика:
- ◇ двумерная 71
 - ◇ трехмерная 101

Д

Диаграмма:

- ◇ круговая 99
- ◇ столбцовая 96

Дифференциальное уравнение 284

Достоверность решения задачи 74

И

- Именованная константа 39
- Интеграл 304
- Интерполяция 323
 - ◇ многопараметрическая 357
 - ◇ приближенная в узлах 324, 343
 - ◇ сплайнами 355
 - ◇ точная в узлах 323, 330

М

- Массив 147
- Матрица 147
 - ◇ выделение элемента 150
 - ◇ выявление структуры 154
 - ◇ генерация 149
 - ◇ добавление элемента 156
 - ◇ изменение порядка элементов 156
 - ◇ комбинирование 156, 159
 - ◇ создание 160
 - ◇ удаление элемента 156
- Матричное умножение 162
- Меню:
 - ◇ Cell 22
 - ◇ Edit 21
 - ◇ File 18
 - ◇ Find 30
 - ◇ Format 25
 - ◇ Help 31
 - ◇ Input 26
 - ◇ Kernel 28
 - ◇ Window 31
- Метод:
 - ◇ Гаусса 253
 - ◇ интервальный 242
 - ◇ итераций 254
 - ◇ матричный 274
 - ◇ Ньютона 259

О

- Область изоляции корня 73
- Оператор:
 - ◇ = 47
 - ◇ арифметический 35
- Операция, математическая над вектором, матрицей 162
- Оптимизационные задачи 207
- Ортогональный полином 168

П

- Переменная 47
 - ◇ символьная 146
- Подстановки 52
- Полином 197
- Правило Лопиталя 125
- Предел 124
- Преобразование
 - ◇ Z 203
 - ◇ Лапласа 198
 - обратное 200
 - ◇ Фурье 204
- Произведение 114
- Производная 138

Р

- Регрессионный анализ 357
- Регрессия 357
- Ряд:
 - ◇ Маклорена 130
 - ◇ Тейлора 130

С

- Система уравнений:
 - ◇ несовместная 277
 - ◇ совместная 277
- Система уравнений, линейных 252
 - ◇ матричная форма представления 274

Система уравнений, линейных

(прод.):

◇ несовместная 253

◇ плохо обусловленная 254

◇ решение 261

◇ совместная 253

Система уравнений, нелинейных
258

◇ решение 265

▫ графическим способом 258

▫ табличным способом 259

Список 146, 154

Сплайн-интерполяция 355

Сумма 107

Т

Табулирование функции 119

Тип данных 42, 142

Точка перегиба 213

Точность вычислений 67

У

Уравнение, трансцендентное 229

Ф

Факториал 37

Формула Ньютона 319

Функция:

◇ Abs 46, 166

◇ AddTo 41

◇ AiryAi 182

◇ AiryAiPrime 183

◇ AiryBi 182

◇ AiryBiPrime 183

◇ Append 156

◇ ArcCos 166

◇ ArcCosh 167

◇ ArcCot 167

◇ ArcCoth 167

◇ ArcCsc 167

◇ ArcCsch 167

◇ ArcSec 167

◇ ArcSech 167

◇ ArcSin 166

◇ ArcSinh 166

◇ ArcTan 167

◇ ArcTanh 167

◇ Arg 46

◇ Array 160

◇ BarChart 97

◇ BaseForm 143

◇ Bernoulli 197

◇ BernoulliB 197

◇ BesselI 179

◇ BesselJ 179

◇ BesselK 179

◇ BesselY 179

◇ Beta 186

◇ BetaRegularized 186

◇ Binomial 197

◇ Ceiling 37

◇ CentralMoment 187

◇ ChebyshevT 168

◇ ChebyshevU 168

◇ Collect 55, 60, 132

◇ Complement 159

◇ ComplexExpand 58

◇ Conjugate 46

◇ ConstrainedMax 221

◇ ConstrainedMin 221

◇ ContourPlot 102

◇ Cos 166

◇ Cosh 166

◇ CoshIntegral 173

◇ CosIntegral 173

◇ Cot 167

◇ Coth 167

◇ Count 154

◇ Csc 167

◇ Csch 167

◇ Cyclotomic 198

- ◊ D 138, 139
- ◊ Delete 156
- ◊ Det 164
- ◊ Dimensions 154
- ◊ Divide 36, 42
- ◊ Divisors 36
- ◊ DivisorSigma 36
- ◊ Dot 162, 276
- ◊ Drop 156
- ◊ DSolve 285, 287, 288
- ◊ Dt 139
- ◊ Eigensystem 164
- ◊ Eigenvalues 164
- ◊ Eigenvectors 164
- ◊ Eliminate 272
- ◊ EllipticE 194
- ◊ EllipticF 195
- ◊ EllipticK 194
- ◊ EllipticPi 195
- ◊ EllipticTheta 195
- ◊ EllipticThetaC 195
- ◊ EllipticThetaD 195
- ◊ EllipticThetaN 195
- ◊ Erf 187
- ◊ Erfc 188
- ◊ Erfi 188
- ◊ EulerE 198
- ◊ Exp 166
- ◊ Expand 55, 58
- ◊ ExpandAll 58
- ◊ ExpandDenominator 58
- ◊ ExpandNumerator 58
- ◊ ExpIntegralE 173
- ◊ ExpIntegralEi 173
- ◊ ExpToTrig 63
- ◊ ExtendedGCD 36
- ◊ Factor 55, 61
- ◊ Factorial 37
- ◊ Factorial2 37
- ◊ FactorialInteger 62
- ◊ FactorList 62
- ◊ FactorTerms 62
- ◊ FactorTermsList 62
- ◊ Fibonacci 198
- ◊ FindMinimum 215
- ◊ FindRoot 240, 270
- ◊ Fit 343, 357
- ◊ Flatten 157
- ◊ Floor 37
- ◊ FourierCosTransform 204
- ◊ FourierSinTransform 204
- ◊ FourierTransform 204
- ◊ FreeQ 154
- ◊ FresnelC 195
- ◊ FresnelS 195
- ◊ FullSimplify 55, 56
- ◊ FunctionExpand 59
- ◊ Gamma 176
- ◊ GammaRegularized 176
- ◊ GCD 36
- ◊ GegenbauerC 169
- ◊ GeneralizedBarChart 97
- ◊ GeometricMean 187
- ◊ HarmonicMean 187
- ◊ HermiteH 169
- ◊ IdentityMatrix 164
- ◊ Im 46
- ◊ Insert 156
- ◊ Integrate 304
- ◊ InterpolateRoot 245
- ◊ InterpolatingFunction 338
- ◊ InterpolatingPolynomial 335
- ◊ InterpolatingQuantile 187
- ◊ Interpolation 338
- ◊ Intersection 159
- ◊ IntervalBisection 243
- ◊ IntervalNewton 243
- ◊ IntervalSecand 243
- ◊ Inverse 164, 276
- ◊ InverseErf 188
- ◊ InverseErfC 188
- ◊ InverseFourierTransform 204

Функция (*prod.*):

- ◊ InverseLaplaceTransform 200
- ◊ InverseZTransform 203
- ◊ JacobiP 169
- ◊ JacobiZeta 195
- ◊ Join 159
- ◊ LaguerreL 169
- ◊ LaplaceTransform 200
- ◊ Last 156
- ◊ LCM 37
- ◊ LegendreP 168
- ◊ LegendreQ 169
- ◊ Length 154
- ◊ Limit 125
- ◊ LinearPrograming 221
- ◊ LinearSolve 164, 276
- ◊ List 148
- ◊ ListIntegrate 313
- ◊ ListPlot 81
- ◊ Log 166
- ◊ LogGamma 176
- ◊ LogIntegral 173
- ◊ LogLinearListPlot 92
- ◊ LogLinearPlot 92
- ◊ LogListPlot 92
- ◊ LogLogListPlot 92
- ◊ LogLogPlot 92
- ◊ LogPlot 92
- ◊ MantissaExponent 51
- ◊ MatrixForm 147, 152
- ◊ MatrixQ 154
- ◊ Max 208
- ◊ Mean 187
- ◊ MeanDeviation 187
- ◊ Median 187
- ◊ MedianDeviation 187
- ◊ MemberQ 154
- ◊ Min 208
- ◊ Mod 36
- ◊ MoebiusMu 198
- ◊ MultipleListPlot 83
- ◊ N 51, 145
- ◊ N[%] 40
- ◊ n^{N_n} 144
- ◊ NDSolve 292, 295
- ◊ NIntegrate 308
- ◊ NMaximize 216
- ◊ NMinimize 216
- ◊ NonlinearFit 352
- ◊ NonlinearRegress 352
- ◊ Normal 133
- ◊ NProduct 116
- ◊ NRoots 239
- ◊ NSolve 237, 268
- ◊ NSum 110
- ◊ Pade 348
- ◊ ParametricPlot3D 104
- ◊ Part 151
- ◊ PercentileBarChart 97
- ◊ PieChart 99
- ◊ Plot 71
- ◊ Plot3D 102
- ◊ Plus 35
- ◊ PolarPlot 94
- ◊ PolyGamma 176
- ◊ PolynomialFit 353
- ◊ Position 154
- ◊ PowerExpand 58
- ◊ Prepend 156
- ◊ Prime 38
- ◊ PrimePi 38
- ◊ Product 115
- ◊ Pseudoinverse 164
- ◊ Quantile 187
- ◊ Quotient 37
- ◊ Random 192
- ◊ Range 148
- ◊ Rationalize 67
- ◊ Re 46
- ◊ Regress 350
- ◊ Rest 156
- ◊ Reverse 158

- ◇ RootMeanSquare 187
 - ◇ Roots 233
 - ◇ RotateLeft 158
 - ◇ RotateRight 158
 - ◇ Round 37
 - ◇ Sec 167
 - ◇ Sech 167
 - ◇ SeedRandom 192
 - ◇ Series 130—132
 - ◇ Set 47
 - ◇ SetDelayed 47
 - ◇ Show 83
 - ◇ Sign 166
 - ◇ Simplify 55
 - ◇ Sin 166
 - ◇ Sinh 166
 - ◇ SinhIntegral 173
 - ◇ SinIntegral 172
 - ◇ Solve 229, 260, 266
 - ◇ Sort 157
 - ◇ SplineFit 355
 - ◇ Sqrt 166
 - ◇ Srewness 187
 - ◇ StackedBarChart 97
 - ◇ StandardDeviation 187
 - ◇ StirlingS1 198
 - ◇ StirlingS2 198
 - ◇ SubtractFrom 41
 - ◇ Sum 107, 110
 - ◇ Table 120, 149, 332
 - ◇ TableForm 54, 147, 152
 - ◇ Take 156
 - ◇ Tan 167
 - ◇ Tanh 167
 - ◇ TensorRank 154
 - ◇ Times 35
 - ◇ TimesBy 41
 - ◇ Tr 164
 - ◇ Transpose 158, 164
 - ◇ TrigExpand 55, 63
 - ◇ TrigFactor 63
 - ◇ TrigReduce 63
 - ◇ TrigToExp 63
 - ◇ Union 159
 - ◇ VariaceData 187
 - ◇ VectorQ 154
 - ◇ WeirstrassP 195
 - ◇ Zeta 198
 - ◇ ZTransform 203
 - ◇ арифметическая 35
 - ◇ Бесселя 179
 - ◇ интегральная 194
 - ◇ интегральная показательная 172
 - ◇ разложение в степенной ряд 129
 - ◇ статистических распределений 187
 - ◇ Эйри 182
- ## Ч
- Число:
- ◇ вещественное 44, 144
 - ◇ комплексное 45, 146
 - ◇ максимальное 207
 - ◇ минимальное 207
 - ◇ простое 38
 - ◇ рациональное 42
 - ◇ случайное 191
 - ◇ специальное 197
 - ◇ целое 42
- ## Э
- Экстремум 210
- Эллиптический интеграл 194