Дмитрий Кирьянов

Mathcad 11

САМОУЧИТЕЛЬ

Символьные и численные методы вычислений

Оформление расчетов. Обработка ошибок

Программирование в Mathcad

Математическая статистика. Анализ данных

Справочник по компонентам и функциям

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

Эффективность и точность математических расчетов

Дмитрий Кирьянов

Самоучитель Mathcad 11

Санкт-Петербург
«БХВ-Петербург»
2003

УДК 681.3.06
ББК 32.973.26-018.2
К43

Кирьянов Д. **В.**

К43 Самоучитель Mathcad И. — СПб.: БХВ-Петербург, 2003. - 560 с: ил.

ISBN 5-94157-348.0

В книге автор попытался совместить две цели. Первая — последовательно рассказывая об основах расчетов, интерфейсе пользователя и переходя от простого к сложному, дать возможность читателю самостоятельно освоить Mathcad. Таким образом, книга может использоваться как самоучитель, позволяющий "с нуля" освоить ключевые возможности этой вычислительной системы. Вторая цель — ИЗЛОЖИТЬ материал, делая акцент на решении конкретных математических проблем. Поэтому, приступая к той или иной задаче, открывайте соответствующую главу книги и используйте ее как справочник. Изложение материала начинается с краткого определения математических понятий и терминов, при этом предполагается, что читатель имеет базовые математические знания.

Для начинающих пользователей, студентов, программистов и научных работников

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Анатолий Адаменко*
Зав. редакцией	*Григорий Добин*
Редактор	*Анатолий Хрипов*
Компьютерная верстка	*Екатерина Трубникова*
Корректор	*Анна Брезман*
Дизайн обложки	*Игорь Цырульников*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 07.07.03.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 44,50.
Доп. тираж 5000 экз. Заказ № 984.
"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

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

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН
199034, Санкт-Петербург, 9 линия, 12.

ISBN 5-94157-348.0

Содержание

Елене посвящаю эту книгу

Введение

Эта книга — о самом популярном из компьютерных математических пакетов Mathcad 11 компании MathSoft. С его помощью можно решать самые разные математические задачи и оформлять результаты расчетов на высоком профессиональном уровне.

Создавая эту книгу, я попытался совместить две цели. Первая — последовательно рассказывая об основах расчетов, интерфейсе пользователя и переходя от простого к сложному, дать возможность читателю самостоятельно освоить Mathcad. Книга может использоваться как самоучитель, позволяющий "с нуля" освоить ключевые возможности этой вычислительной системы. Вторая цель — изложить материал, делая акцент на решении конкретных математических проблем. Приступая к той или иной задаче, открывайте соответствующую главу книги и используйте ее как справочник. Причем я старался начинать рассказ с краткого определения математических понятий и терминов, конечно, предполагая, что читатель имеет базовые математические знания.

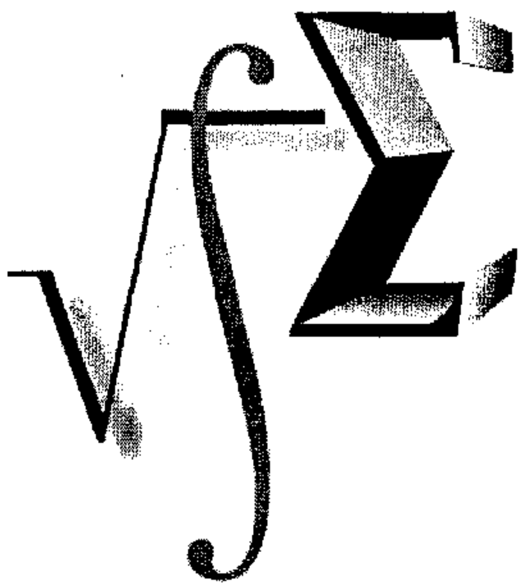
Книга разбита на четыре части. В первой даны основные сведения о Mathcad и приемы работы с его математическим редактором, во второй и третьей частях рассматриваются решения практических задач математики, снабженные примерами, которые представлены листингами. В четвертой части приводятся сведения, касающиеся профессионального оформления расчетов в Mathcad 11 и методы эффективной работы для опытных пользователей.

Хочется сделать еще несколько замечаний по строению книги. Все листинги автономны и работают вне каких-либо дополнительных модулей. В листингах умышленно, чтобы не загромождать их, нет текстовых полей, — они содержат только расчеты по формулам. Все комментарии к ним находятся в тексте. Почти все графики вынесены в рисунки, причем, если они являются продолжением листингов, это помечено в подрисуночной подписи. Обозначенные звездочкой разделы содержат информацию, относящуюся, в основном, к особенностям численных алгоритмов или полезным советам и программным решениям самого автора. Эти разделы при первом знакомстве с Mathcad могут быть пропущены.

Что же такое система Mathcad? Следует хорошо представлять себе, что в состав Mathcad 11 входят несколько интегрированных между собой компонентов:

- мощный текстовый редактор, позволяющий вводить, редактировать и форматировать как текст, так и математические выражения;
- вычислительный процессор, умеющий проводить расчеты по введенным формулам, используя встроенные численные методы;
- символьный процессор, являющийся, фактически, системой искусственного интеллекта;
- огромное хранилище справочной информации, как математической, так и инженерной, оформленной в виде библиотеки интерактивных электронных книг.

Обо всех перечисленных возможностях я попытался в доступной форме рассказать в этой книге. Дополнительную информацию читатель может получить в Интернете на сервере производителя Mathcad **http://www.Mathcad.com,** дистрибьютора Mathcad в России **http://www.Mathcad.ru** и на личной странице автора **http://www.kirianov.orc.ru.**



ЧАСТЬ I

ОБЩИЕ СВЕДЕНИЯ

ГЛАВА 1

Начинаем работу

В данной главе рассмотрено назначение приложения Mathcad 11 и, в целях знакомства с его основными возможностями, приведены базовые приемы его использования *(см. разд. 1.1—1.2).* Если вы уже имели дело с прежними версиями, начиная с Mathcad 7, и у вас неплохие навыки работы с его редактором, то можете смело пропустить эту главу. В ней основное внимание уделено главным компонентам интерфейса Mathcad 11, который интуитивен и похож на другие программы Windows *(см. разд. 1.3),* а также эффективному использованию справочной системы Mathcad *(см. разд. 1.4).*

1.1. Назначение Mathcad

Mathcad является математическим редактором, позволяющим проводить разнообразные научные и инженерные расчеты, начиная от элементарной арифметики и заканчивая сложными реализациями численных методов. Пользователи Mathcad — это студенты, ученые, инженеры, разнообразные технические специалисты. Благодаря простоте применения, наглядности математических действий, обширной библиотеке встроенных функций и численных методов, возможности символьных вычислений, а также превосходному аппарату представления результатов (графики самых разных типов, мощных средств подготовки печатных документов и Web-страниц), Mathcad стал наиболее популярным математическим приложением.

Mathcad 11, в отличие от большинства других современных математических приложений, построен в соответствии с принципом WYSIWYG ("What You See Is What You Get" — "что Вы видите, то и получите"). Поэтому он очень прост в использовании, в частности, из-за отсутствия необходимости сначала писать программу, реализующую те или иные математические расчеты, а потом запускать ее на исполнение. Вместо этого достаточно просто вводить математические выражения с помощью встроенного редактора формул, причем в виде, максимально приближенном к общепринятому, и тут же получать результат. Кроме того, можно изготовить на принтере печатную копию документа или создать страницу в Интернете именно в том виде, который этот документ имеет на экране компьютера при работе с Mathcad. Создатели Mathcad сделали все возможное, чтобы пользователь, не обладающий специальными знаниями в программировании (а таких большинство среди ученых и инженеров), мог в полной мере приобщиться к достижениям современной вычислительной науки и компьютерных технологий. Для эффективной работы с редактором Mathcad достаточно базовых навыков пользователя. С другой стороны, профессиональные программисты (к которым относит себя и автор этих строк) могут извлечь из Mathcad намного больше, создавая различные программные решения, существенно расширяющие возможности, непосредственно заложенные в Mathcad.

В соответствии с проблемами реальной жизни, математикам приходится решать одну или несколько из следующих задач:

- ввод на компьютере разнообразных математических выражений (для дальнейших расчетов или создания документов, презентаций, Web-страниц);
- проведение математических расчетов;
- подготовка графиков с результатами расчетов;
- ввод исходных данных и вывод результатов в текстовые файлы или файлы с базами данных в других форматах;
- подготовка отчетов работы в виде печатных документов;
- подготовка Web-страниц и публикация результатов в Интернете;
- получение различной справочной информации из области математики.

Со всеми этими (а также некоторыми другими) задачами с успехом справляется Mathcad:

- математические выражения и текст вводятся с помощью формульного редактора Mathcad, который по возможностям и простоте использования не уступает, к примеру, редактору формул, встроенному в Microsoft Word;
- математические расчеты производятся немедленно, в соответствии с введенными формулами;
- графики различных типов (по выбору пользователя) с богатыми возможностями форматирования вставляются непосредственно в документы;
- возможен ввод и вывод данных в файлы различных форматов;
- документы могут быть распечатаны непосредственно в Mathcad в том виде, который пользователь видит на экране компьютера, или сохранены в формате RTF для последующего редактирования в более мощных текстовых редакторах (например Microsoft Word);

- возможно полноценное сохранение документов Mathcad 11 в формате Web-страниц (генерация вспомогательных графических файлов происходит автоматически);
- имеется опция объединения разрабатываемых Вами документов в электронные книги, которые, с одной стороны, позволяют в удобном виде хранить математическую информацию, а с другой — являются полноценными Mathcad-программами, способными осуществлять расчеты;
- символьные вычисления позволяют осуществлять аналитические преобразования, а также мгновенно получать разнообразную справочную математическую информацию

Таким образом, следует хорошо представлять себе, что в состав Mathcad входят несколько интегрированных между собой компонентов — это мощный текстовый редактор для ввода и редактирования как текста, так и формул, вычислительный процессор — для проведения расчетов согласно введенным формулам и символьный процессор, являющийся, по сути, системой искусственного интеллекта. Сочетание этих компонентов создает удобную вычислительную среду для разнообразных математических расчетов и. одновременно, документирования результатов работы.

1.2. Знакомство с Mathcad

В данном разделе, несколько забегая вперед, покажем, как быстро начать работу с Mathcad, научиться вводить математические выражения и получать первые результаты расчетов.

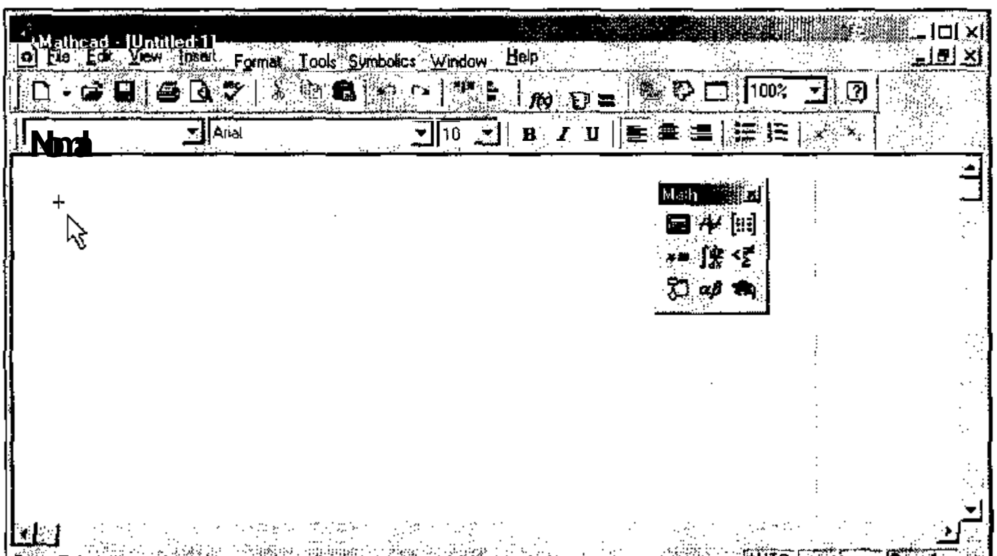


Рис. 1.1. Окно Mathcad 11 с новым документом

После того как Mathcad 11 установлен на компьютере и запушен на исполнение, появляется основное окно приложения, показанное на рис. 1.1. Оно имеет ту же структуру, что и большинство приложений Windows. Сверху вниз располагаются заголовок окна, строка меню, панели инструментов (стандартная и форматирования) и *рабочий лист* или *рабочая область* документа (worksheet). Новый документ создается автоматически при запуске Mathcad. В самой нижней части окна находится строка состояния. Не забывая о сходстве редактора Mathcad с обычными текстовыми редакторами, вы интуитивно поймете назначение большинства кнопок на панелях инструментов.

Помимо элементов управления, характерных для типичного текстового редактора, Mathcad снабжен дополнительными средствами для ввода и редактирования математических символов, одним из которых является панель инструментов **Math** (Математика) (рис. 1.1). С помощью этой, а также ряда вспомогательных наборных панелей, удобно осуществлять ввод уравнений.

Для того чтобы выполнить простые расчеты по формулам, проделайте следующее:

- определите место в документе, где должно появиться выражение, щелкнув мышью в соответствующей точке документа;
- введите левую часть выражения;
- введите знак равенства <=>.

Оставим пока разговор о более надежных способах ввода математических символов и приведем пример простейших расчетов. Для вычисления синуса какого-нибудь числа достаточно ввести с клавиатуры выражение типа `sin(1/4)=`. После того как будет нажата клавиша со знаком равенства, с правой стороны выражения, как по мановению волшебной палочки, появится результат (листинг 1.1).

Листинг 1.1. Расчет простого выражения

$$\sin\left(\frac{1}{4}\right) = 0.247$$

Примечание

Здесь и далее во всей книге в листинги вынесено содержание рабочей области документа Mathcad вместе с полученными результатами вычислений.

Подобным образом можно проводить и более сложные и громоздкие вычисления, пользуясь при этом всем арсеналом специальных функций, которые встроены в Mathcad. Легче всего вводить их имена с клавиатуры, как в примере с вычислением синуса, но, чтобы избежать возможных ошибок в их написании, лучше выбрать другой путь. Чтобы ввести встроенную функцию в выражение:

1. Определите место в выражении, куда следует вставить функцию.
2. Нажмите кнопку с надписью *f(x)* на стандартной панели инструментов (на нее указывает курсор на рис. 1.2).
3. В списке **Function Category** (Категория функции) появившегося диалогового окна **Insert Function** (Вставить функцию) выберите категорию, к которой принадлежит функция, — в нашем случае это категория **Trigonometric** (Тригонометрические).
4. В списке **Function Name** (Имя функции) выберите имя встроенной функции, под которым она фигурирует в Mathcad **(sin).** В случае затруднения с выбором ориентируйтесь на подсказку, появляющуюся при выборе функции в нижнем текстовом поле диалогового окна **Insert Function.**
5. Нажмите кнопку **OK** — функция появится в документе.
6. Заполните недостающие аргументы введенной функции (в нашем случае это 1/4).

Результатом будет введение выражения из листинга 1.1, для получения значения которого осталось лишь ввести знак равенства.

Примечание

Большинство численных методов, запрограммированных в Mathcad, реализовано в виде встроенных функций. Пролистайте на досуге списки в диалоговом окне **Insert Function** (Вставить функцию), чтобы представлять себе, какие специальные функции и численные методы можно использовать в расчетах (им полностью посвящена *часть III*, а подробный перечень встроенных функций приведен в *приложении 3).*

Конечно, не всякий символ можно ввести с клавиатуры. Например, неочевидно, как вставить в документ знак интеграла или дифференцирования. Для этого в Mathcad имеются специальные панели инструментов, очень похожие на средства формульного редактора Microsoft Word. Как уже было отмечено ранее, одна из них — панель инструментов **Math** — показана на рис. 1.1. Она содержит инструменты для вставки в документы математических объектов (операторов, графиков, элементов программ и т. п.). Эта панель показана более крупным планом на рис. 1.3 уже на фоне редактируемого документа.

Панель содержит девять кнопок, нажатие каждой из которых приводит, в свою очередь, к появлению на экране еще одной панели инструментов. С помощью этих девяти дополнительных панелей можно вставлять в документы Mathcad разнообразные объекты. На рис. 1.3, как легко увидеть, на панели **Math** в нажатом состоянии находятся две первые сверху слева кнопки (над левой из них находится указатель мыши). Поэтому на экране присутствуют еще две панели — **Calculator** (Калькулятор) и **Graph** (График). Легко догадаться, какие объекты вставляются при нажатии кнопок на этих панелях.

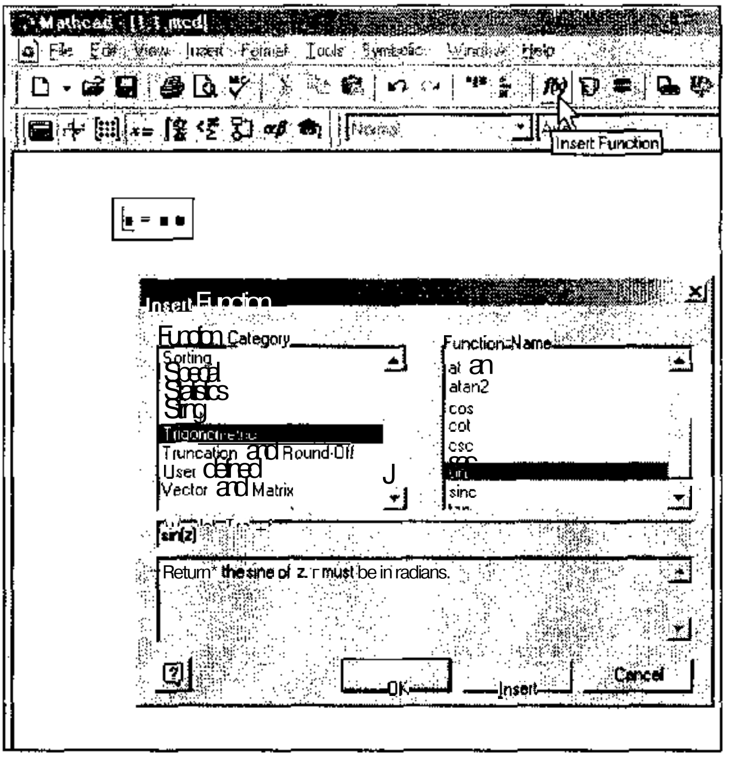


Рис. 1.2. Вставка встроенной функции

Примечание

Подробнее о назначении этих и других наборных панелей инструментов рассказано ниже *(см. разд. 1.3).*

К примеру, можно ввести выражение из листинга 1.1 исключительно с помощью панели **Calculator.** Для этого нужно сначала нажать кнопку **sin** (самую первую сверху). Результат данного действия показан на рис. 1.3 (выражение в рамке). Теперь остается лишь набрать выражение 1/4 внутри скобок (в *местозаполнителе*, обозначаемом черным прямоугольником). Для этого нажмите последовательно кнопки 1, — и 4 на панели **Calculator** и затем, на ней же, кнопку =, чтобы получить ответ (разумеется, тот же самый, что и в предыдущей строке документа).

Как видите, вставлять в документы математические символы можно поразному, как и во многих других приложениях Windows. В зависимости от опыта работы с Mathcad и привычек работы на компьютере, пользователь может выбрать любой из них.

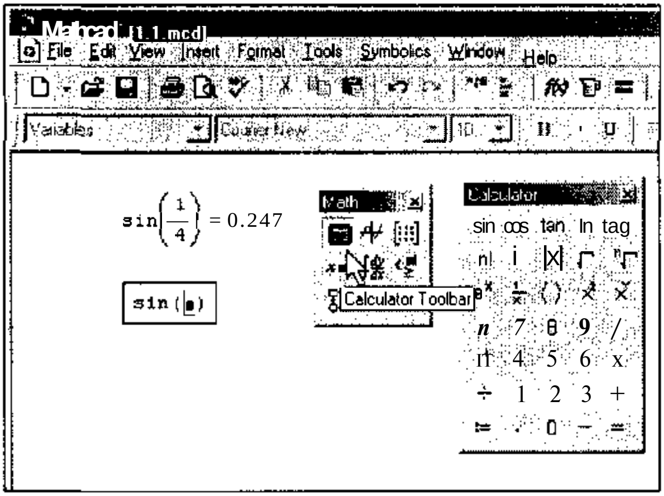


Рис. 1.3. Использование панели инструментов **Math**

Совет

Если вы только начинаете осваивать редактор Mathcad, настоятельно рекомендую, где это только возможно, вводить формулы, пользуясь наборными панелями инструментов и описанной процедурой вставки функций с помощью диалога **Insert Function** (Вставить функцию). Это позволит избежать многих возможных ошибок.

Описанные действия демонстрируют использование Mathcad в качестве обычного калькулятора с расширенным набором функций. Для математика же интерес представляет, как минимум, возможность задания переменных и операций с функциями пользователя. Нет ничего проще — в Mathcad эти действия, как и большинство других, реализованы по принципу "как принято в математике, так и вводится". Поэтому приведем соответствующие примеры (листинги 1.2 и 1.3), не теряя времени на комментарии *(если у вас возникнут проблемы с пониманием листингов, обратитесь за разъяснением к соответствующим разделам этой главы)*. Обратите внимание только на оператор присваивания, который применяется для задания значений переменным в первой строке листинга 1.2. Его, как и все остальные символы, можно ввести с помощью панели **Calculator.** Присваивание обозначается символом ":=" , чтобы подчеркнуть его отличие от операции вычисления.

Листинг 1.2. Использование переменных в расчетах

$x := 1.2 \qquad y := 55 \qquad z := 4$

$$\frac{\left(x^2 \cdot 250\right)}{\sqrt[5]{y}} \cdot \ln(z \cdot \text{я}) = 408.814$$

Листинг 1.3. Определение функции пользователя и расчет ее значения в точке x=1

a := 2

$$f(x) := x^{a} - \frac{2}{|x-5|}$$

f (1) = 0.5

В последнем листинге определяется функция f(x). Ее график показан на рис. 1.4. Чтобы построить его, следует нажать на панели Graph кнопку с нужным типом графика (на нее на рисунке наведен указатель мыши) и в появившейся заготовке графика определить значения, которые будут отложены по осям. В нашем случае потребовалось ввести x в местозаполнитель возле оси x и f(x) — возле оси Y.

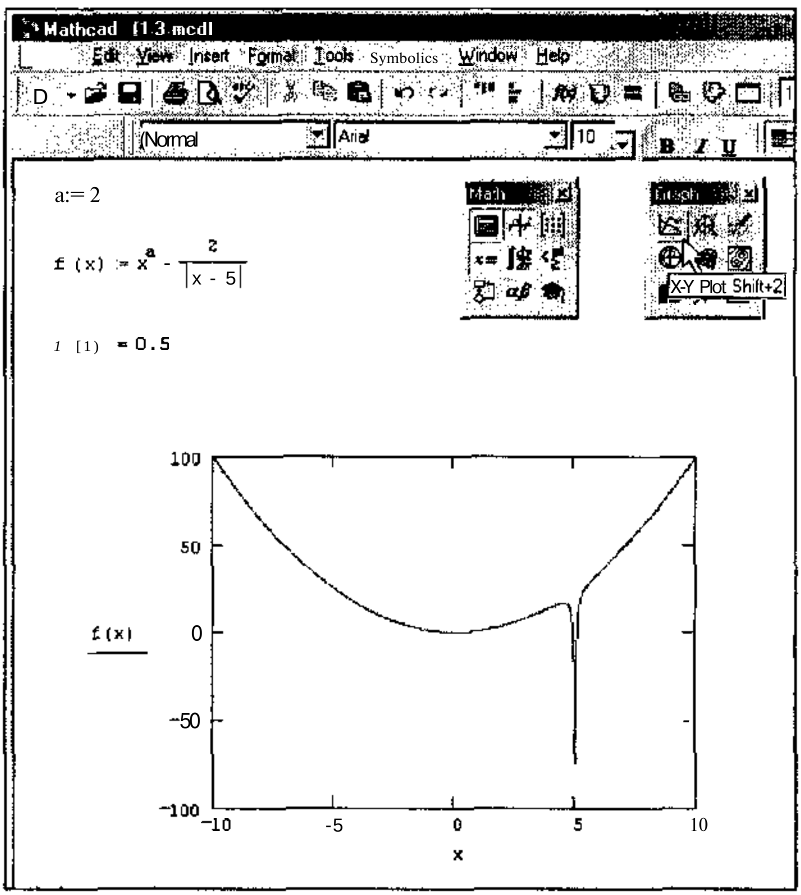


Рис. 1.4. Построение графика функции (листинг 1.3)

Примечание

Сравните содержание листинга 1.3 и рис. 1.4. Такой стиль подачи материала будет сохранен во всей книге. Листинги представляют собой фрагменты рабочих областей документа, которые работают без какого-либо дополнительного кода (если это не оговорено особо). Можно ввести содержание любого листинга в новый (пустой) документ, и он будет работать точно так же, как в книге. Чтобы не загромождать листинги, графики выведены в отдельные рисунки. В отличие от рис. 1.4, в следующих рисунках код листингов не дублируется, а если имеется ссылка на листинг в подрисуночной надписи, то это подразумевает, что данный график может быть вставлен в документ после упомянутого листинга.

Одной из самых впечатляющих возможностей Mathcad являются символьные вычисления, позволяющие решить многие задачи аналитически. Фактически, по мнению автора, Mathcad "знает" математику, по крайней мере, на уровне неплохого ученого. Умелое использование интеллекта символьного процессора Mathcad избавит вас от огромного количества рутинных вычислений, к примеру, интегралов и производных (листинг 1.4). Обратите внимание на традиционную форму написания выражений, единственная особенность заключается в необходимости применения символа символьных вычислений → вместо знака равенства. Его, кстати, можно ввести в редакторе Mathcad с любой из панелей **Evaluation** (Выражения) или **Symbolic** (Символика), а символы интегрирования и дифференцирования — с панели **Calculus** (Вычисления).

Листинг 1.4. Символьные вычисления

$$\int \frac{\ln(a \cdot x)}{x^{b}}\,dx \rightarrow \left[\frac{-(b \cdot \ln(a) - \ln(a) + 1)}{\left(-2 \cdot b + b^{2} + 1\right)} \cdot x - \frac{1}{(b-1)} \cdot x \cdot \ln(x)\right] \cdot x^{-t}$$

$$\frac{d}{dx}\frac{\left(x^{2} \cdot 250\right)}{\ }\ln(z \cdot \pi) \rightarrow 500\frac{x}{y^{\frac{1}{5}}}\ln(z \cdot \pi)$$

В этом разделе была рассмотрена лишь небольшая часть вычислительных возможностей системы Mathcad. Тем не менее, несколько приведенных здесь примеров дают неплохое представление о его назначении. Возможно даже, что преждевременно рассказав о простоте, с которой можно проводить математические расчеты, автор потерял некоторую часть самых нетерпеливых читателей, которые уже перешли к решению своих задач. Им я хочу посоветовать воспользоваться в качестве справочника второй и третьей частями книги, а для наилучшего оформления результатов — четвертой частью. Ниже, в этой и последующих главах данной части основы Mathcad освещены более подробно.

1.3. Интерфейс пользователя

В Mathcad интерфейс пользователя интуитивен и сходен с другими приложениями Windows. Его составные части:

- верхнее меню, или строка меню (menu bar);
- панели инструментов (toolbars) **Standard** (Стандартная), **Formatting** (Форматирование), **Resources** (Ресурсы) и **Controls** (Элементы управления);
- панель инструментов **Math** и доступные через нее дополнительные математические панели инструментов;
- рабочая область (worksheet);
- строка состояния (status line или status bar);
- всплывающие, или контекстные, меню (pop-up menus или context menus);
- диалоговые окна или диалоги (dialogs);

Большинство команд можно выполнить как с помощью меню (верхнего или контекстного), так и панелей инструментов или клавиатуры.

1.3.1. Меню

Строка меню располагается в самой верхней части окна Mathcad. Она содержит девять заголовков, щелчок мышью на каждом из которых приводит к появлению соответствующего меню с перечнем команд:

- **File** (Файл) — команды, связанные с созданием, открытием, сохранением, пересылкой по электронной почте и распечаткой на принтере файлов с документами;
- **Edit** (Правка) — команды, относящиеся к правке текста (копирование, вставка, удаление фрагментов и т. п.);
- **View** (Вид) — команды, управляющие внешним видом документа в окне редактора Mathcad, а также команды, создающие файлы анимации;
- **Insert** (Вставка) — команды вставки различных объектов в документы;
- **Format** (Формат) — команды форматирования текста, формул и графиков;
- **Tools** (Инструменты) — команды управления вычислительным процессом и дополнительными возможностями;
- **Symbolics** (Символика) — команды символьных вычислений;
- **Window** (Окно) — команды управления расположением окон с различными документами на экране;
- **Help** (Справка) — команды вызова справочной информации, сведений о версии программы, а также доступа к ресурсам и электронным книгам.

Примечание

Состав каждого меню, снабженный описанием действий каждого пункта, вы можете отыскать в *приложении 3.*

Чтобы выбрать нужную команду, щелкните мышью на содержащем ее меню и повторно на соответствующем элементе меню. Некоторые команды находятся не в самих меню, а в подменю, как это показано на рис. 1.5. Чтобы выполнить такую команду, например команду вызова на экран панели инструментов Symbolic, наведите указатель мыши на пункт Toolbars (Панели инструментов) выпадающего меню View (Вид) и выберите в появившемся подменю пункт Symbolic.

Примечание

Далее в книге, говоря о совершении того или иного действия с помощью меню, последовательность выбора пунктов меню будем приводить сокращенно, разделяя их косыми чертами. Например, рассмотренная команда обозначается как **View / Toolbars / Symbolic.**

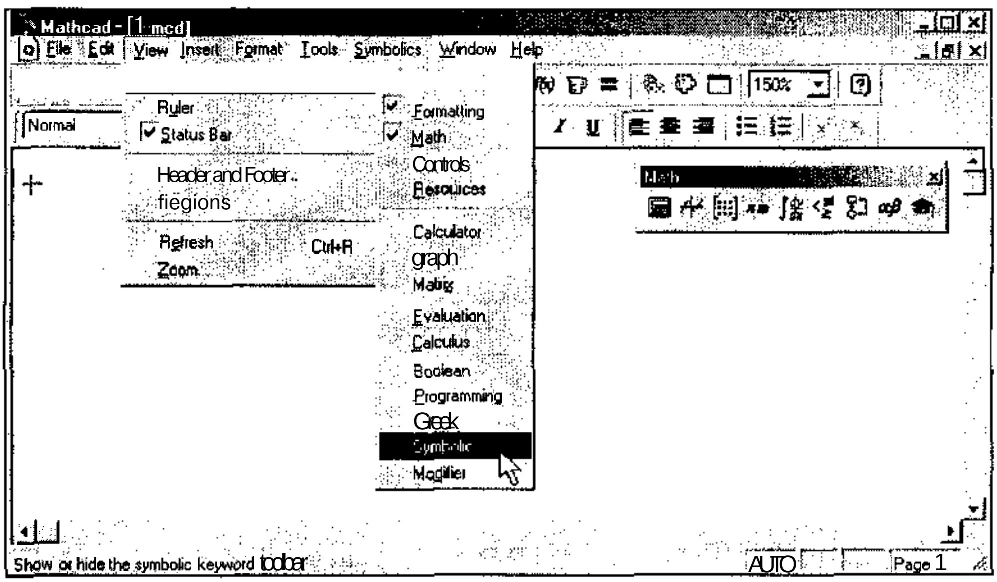


Рис. 1.5. Работа с меню

Обратите внимание, что пункты меню, которые содержат подменю, снабжены стрелками (как пункт Toolbars на рис. 1.5). Кроме того, некоторые пункты меню имеют (или не имеют) флажки проверки, указывающие на включение (или выключение) соответствующей опции в текущий момент. Так, на рис. 1.5 флажки проверки выставлены в пунктах Status **Bar** (Строка состояния) и имен трех панелей инструментов, что говорит о наличии в данный момент на экране строки состояния и трех панелей. Флажки же в пунктах **Ruler** (Линейка), **Regions** (Регионы) и имен математических панелей инструментов отсутствуют, т. е. в данный момент эти опции выключены.

Назначение пунктов меню, на которые наведен указатель мыши, появляется в виде подсказки слева на строке состояния (в нижней части окна Mathcad). На рис. 1.5 указатель наведен на пункт **Symbolic,** поэтому подсказка гласит "Show or hide the symbolic keyword toolbar" (Показать или скрыть панель символики).

Помимо верхнего меню, схожие функции выполняют всплывающие меню (рис. 1.6). Они появляются, как и в большинстве других приложений Windows, при нажатии в каком-либо месте документа правой кнопки мыши. При этом состав данных меню зависит от места их вызова, поэтому их еще называют контекстными. Mathcad сам "догадывается", в зависимости от контекста, какие операции могут потребоваться в текущий момент, и помещает в меню соответствующие команды. Поэтому использовать контекстное меню зачастую проще, чем верхнее, т. к. не надо вспоминать, где конкретно в верхнем меню находится нужный пункт. Как и верхнее меню, контекстное также может иметь подменю (на рис. 1.6 показан участок документа с примером изменения отображения знака умножения в формуле; примечательно, что эту операцию в Mathcad можно осуществить только при помощи контекстного меню).

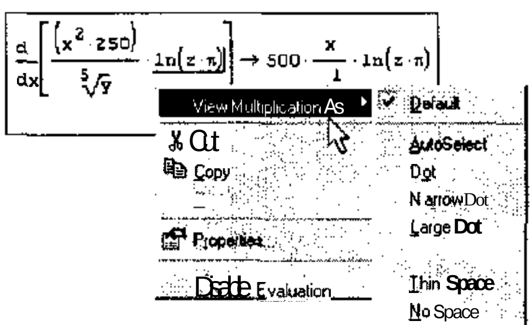


Рис. 1.6. Контекстное меню

1.3.2. Панели инструментов

Панели инструментов служат для быстрого (за один щелчок мыши) выполнения наиболее часто применяемых команд. Все действия, которые можно выполнить с помощью панелей инструментов, доступны и через верхнее меню. На рис. 1.7 изображено окно Mathcad, содержащее четыре основные панели инструментов, расположенные непосредственно под строкой меню. Кнопки в панелях сгруппированы по сходному действию команд:

- **Standard** — служит для выполнения большинства операций, таких, как действия с файлами, редакторская правка, вставка объектов и доступ к справочным системам;

- **Formatting** — для форматирования (изменения типа и размера шрифта, выравнивания и т. п.) текста и формул;
- **Math** — для вставки математических символов и операторов в документы;
- **Resources** — для вызова ресурсов Mathcad (примеров, справок и т.п.).

Группы кнопок на панелях инструментов разграничены по смыслу вертикальными линиями — *разделителями.* При наведении указателя мыши на любую из кнопок рядом с кнопкой появляется *всплывающая подсказка* — короткий текст, поясняющий назначение кнопки (см. рис. 1.3 и 1.4). Наряду со всплывающей подсказкой более развернутое объяснение готовящейся операции можно отыскать на строке состояния.

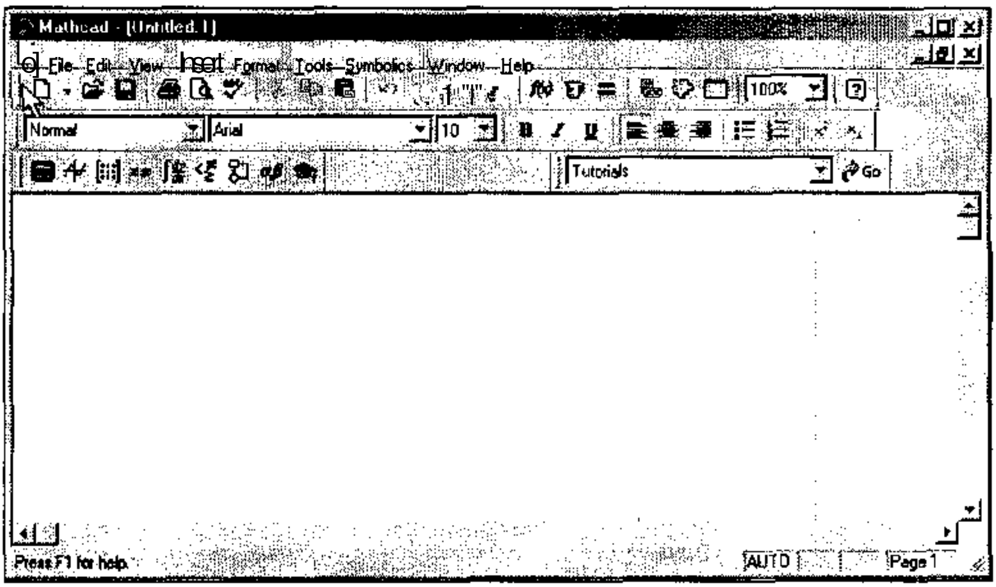


Рис. 1.7. Основные панели инструментов

Панель **Math** предназначена для вызова на экран еще девяти панелей (рис. 1.8), с помощью которых, собственно, и происходит вставка математических операций в документы. В прежних версиях Mathcad эти математические панели инструментов назывались *палитрами* (palettes) или *наборными панелями.* Чтобы показать какую-либо из них, нужно нажать соответствующую кнопку на панели **Math** (см. рис. 1.3). Перечислим назначение математических панелей:

- **Calculator** — служит для вставки основных математических операций;
- **Graph** (График) — для вставки графиков;
- **Matrix** (Матрица) — для вставки матриц и матричных операторов;

- **Evaluation** (Выражения) — для вставки операторов управления вычислениями;
- **Calculus** (Вычисления) — для вставки операторов интегрирования, дифференцирования, суммирования;
- Boolean (Булевы операторы) — для вставки логических (булевых) операторов;
- **Programming** (Программирование) — для программирования средствами Mathcad;
- **Greek** (Греческие символы) — для вставки греческих символов;
- **Symbolic** — для вставки символьных операторов.

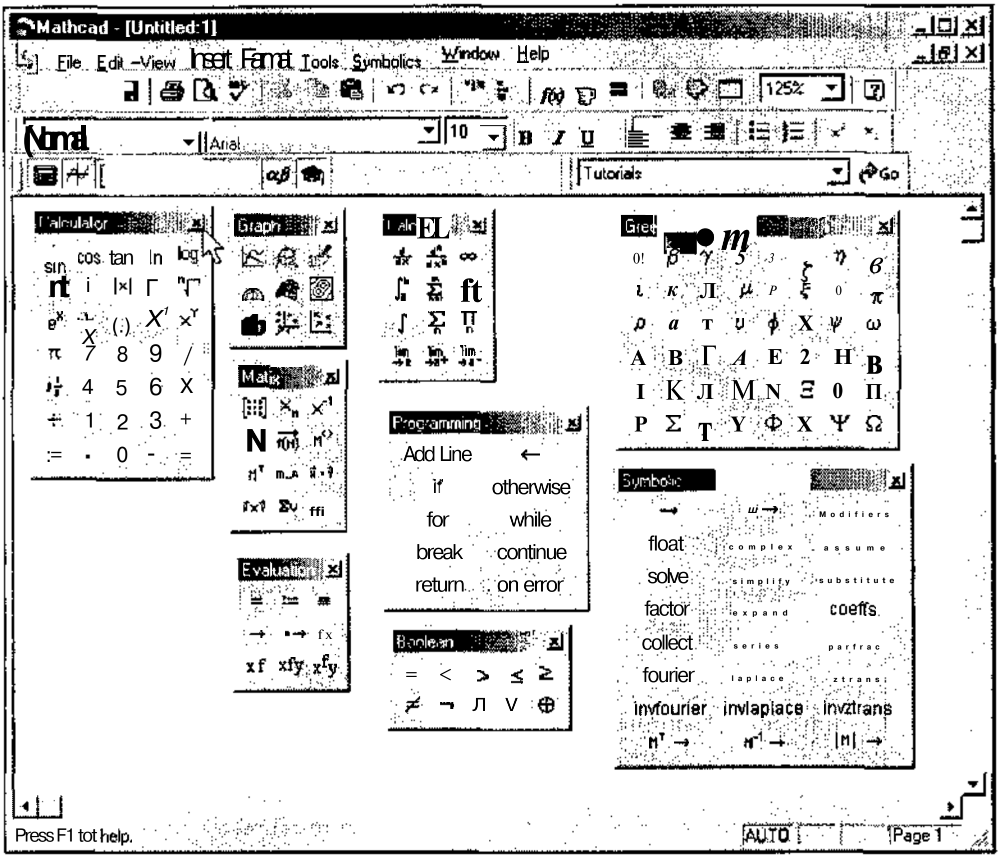


Рис. 1.8. Математические панели инструментов

При наведении указателя мыши на многие из кнопок математических панелей появляется всплывающая подсказка, содержащая еще и сочетание *"горячих клавиш"*, нажатие которых приведет к эквивалентному действию.

1.3.3. Настройка панели инструментов

В Mathcad, подобно другим программам Windows, пользователь может настроить внешний вид панелей инструментов наиболее оптимальным для него образом.

Вы можете:

- показывать или скрывать панели;
- перемещать панели в любое место экрана и изменять их форму;
- делать панели плавающими, и наоборот;
- настраивать основные панели, т. е. определять набор их кнопок.

Присутствие панелей на экране

Вызвать любую панель на экран или скрыть ее можно с помощью меню **View / Toolbars,** выбирая в открывающемся подменю имя нужной панели (см. рис. 1.5). Убрать любую панель с экрана можно еще и посредством контекстного меню, которое вызывается щелчком правой кнопкой мыши в любом месте панели (например на любой кнопке). В контекстном меню следует выбрать пункт **Hide** (Скрыть). Кроме того, если панель *плавающая,* т. е. не прикреплена к основному окну (как, например, все панели на рис. 1.8), то ее можно отключить кнопкой закрытия (на рис. 1.8 указатель мыши наведен на эту кнопку панели **Calculator).**

Математические панели, в отличие от основных, можно вызвать или скрыть нажатием соответствующей кнопки панели **Math.** Присутствие или отсутствие математических панелей показано в виде нажатой (или отжатой) соответствующей кнопки (см. рис. 1.3, 1.4 или 1.8).

Создание плавающих панелей

Чтобы открепить любую из панелей от границ окна Mathcad:

1. Поместите указатель мыши над первым (см. рис. 1.7) или последним разделителем панели (первый разделитель имеет характерный объемный вид, а последний — обычный).
2. Нажмите и удерживайте левую кнопку мыши — вы увидите характерный профиль очертаний панели.
3. Не отпуская кнопку, перетащите панель (для чего переместите указатель мыши в любое место экрана, ориентируясь на перемещение профиля панели).
4. Отпустите кнопку мыши — панель станет плавающей и переместится туда, где находился ее профиль.

Результат перетаскивания основных панелей показан на рис. 1.9. Обратите внимание, что у плавающих панелей инструментов появляется заголовок с названием панели. Чтобы снова прикрепить панель к окну, перетащите ее за этот заголовок к границе окна. При подведении панели на некоторое расстояние к границе можно увидеть, что панель "притягивается" ею. Следует отпустить в этот момент кнопку мыши, и панель перестанет быть плавающей. Можно прикреплять панели не только к строке меню в верхней части окна, но и к любой его границе.

Примечание

Большинство математических панелей могут быть только плавающими.

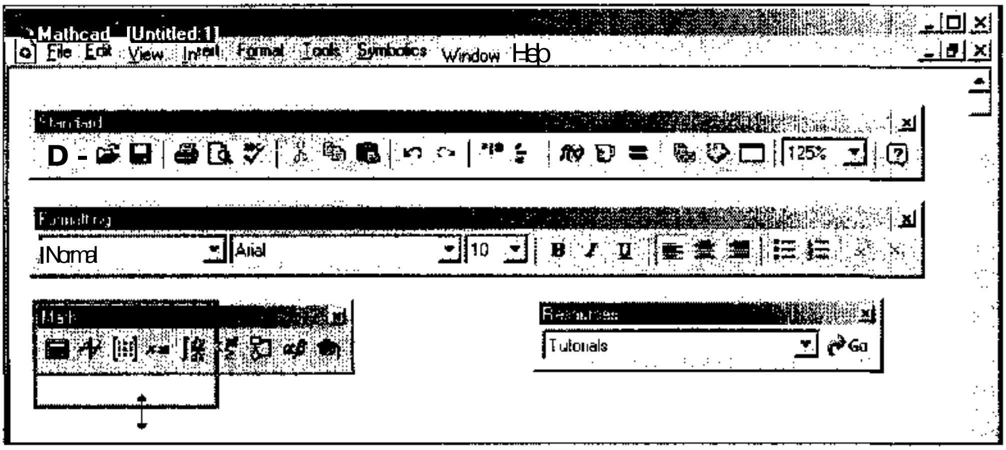


Рис. 1.9. Изменение расположения и размера панелей инструментов

Имеется также и более простой способ открепить панель от границ окна Mathcad. Для этого просто щелкните дважды на ее первом или последнем разделителе. Чтобы прикрепить панель к окну, достаточно двойного щелчка на ее заголовке.

Перемещение панелей по экрану

Чтобы перемещать панели инструментов по экрану, необходимо предварительно сделать их плавающими. Плавающую панель легко поместить в любое место экрана ее перетаскиванием.

Примечание

Переместить панель в любое место экрана можно, даже если основное окно Mathcad не развернуто во весь экран, а занимает только его часть.

Прикрепленные панели перемещайте вдоль фаниц окна, буксируя их мышью за крайние разделители. Таким способом удобно располагать несколько панелей инструментов в одну вертикальную или горизонтальную строку.

Чтобы изменить форму плавающей панели, т. е. выстроить кнопки на ней в другое соотношение рядов и столбцов, поместите указатель мыши на границу панели и, когда он приобретет характерную форму (рис. 1.9), нажмите левую кнопку мыши и перетащите при нажатой кнопке указатель, ориентируясь на изменяющуюся форму контура панели. Когда вы отпустите кнопку мыши, размеры панели изменятся.

Настройка состава основных панелей

Настройка означает изменение количества и состава кнопок на любой из трех основных панелей (Standard, Formatting и Mathematics). Она, например, полезна, если требуется убрать редко используемые кнопки, чтобы не загромождать экран (в особенности, если его невысокое разрешение вынуждает пользователя экономить место). Для изменения состава кнопок на панели вызовите щелчком правой кнопкой мыши в любом ее месте (но не на заголовке) контекстное меню и выберите в нем пункт Customize (Настроить). Появится диалоговое окно Customize Toolbar (Настройка панели инструментов), в котором имеются два списка — в левом перечислены отсутствующие кнопки, а в правом — кнопки, которые присутствуют в данный момент на панели (рис. 1.10).

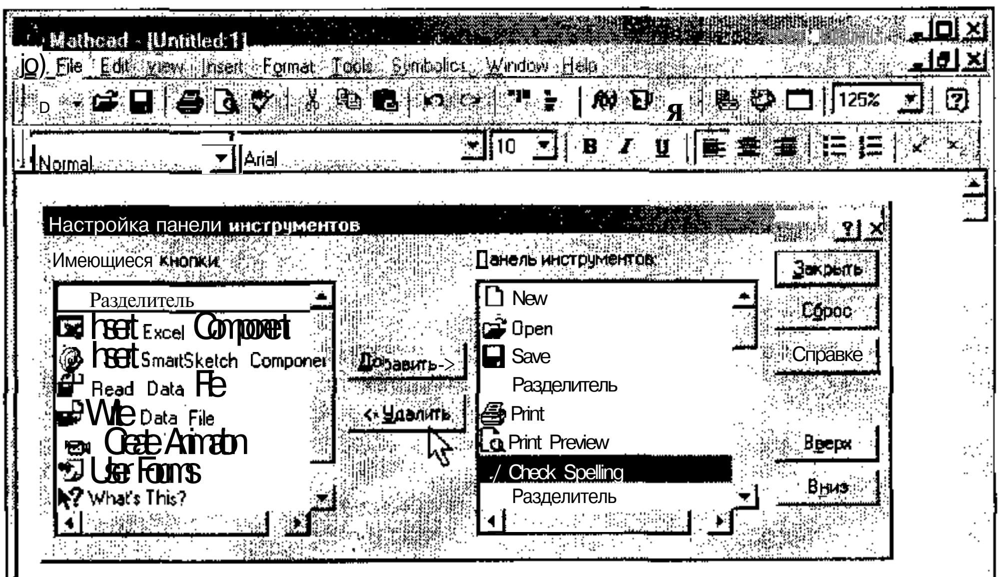


Рис. 1.10. Настройка состава панели инструментов

Чтобы убрать кнопку (или разделитель кнопок) с панели инструментов, выделите ее имя в правом списке и затем нажмите кнопку Remove (Удалить) в диалоговом окне (на нее наведен указатель мыши на рис. МО). Чтобы добавить новую кнопку, выделите ее имя в левом списке и нажмите Add (Добавить). Для изменения порядка расположения на панели той или иной кнопки выделите ее в правом списке и перемещайте в нужную сторону, нажимая кнопки Move Up (Вверх) или Move **Down** (Вниз).

Подтвердить сделанную настройку панели можно нажатием кнопки Close (Закрыть) или кнопки закрытия диалогового окна, а вернуться к прежнему составу панели — с помощью кнопки Reset (Сброс).

1.3.4. Рабочая область

Большую часть окна Mathcad занимает *рабочая область,* в которую пользователь вводит математические выражения, текстовые поля и элементы программирования. Важно уметь настроить рабочую область для работы, чтобы хорошо ориентироваться в документе.

Курсор ввода

На некоторых рисунках этой главы (см., например, рис. 1.7) виден *курсор* ввода в виде небольшого крестика (на дисплее он имеет красный цвет). С его помощью отмечается незаполненное место в документе, куда в текущий момент можно вводить формулы или текст. Чтобы переместить курсор, достаточно щелкнуть указателем мыши в требуемом месте, либо передвинуть его клавишами-стрелками. Если выполнить щелчок в области формулы или начать ввод выражения на пустом месте, вместо курсора появятся линии редактирования, отмечающие место в формуле или тексте, редактируемым в данный момент (см. рис. 1.3 и 1.4).

Примечание

Применение курсора ввода и приемов редактирования документов будет подробно рассмотрено в *главе 2.*

Внешний вид документа

Документ Mathcad строится по принципу размещения формул и текста в рабочей области, которая изначально является подобием чистого листа. Чтобы показать или скрыть расположение регионов с математическими выражениями, текстом или графиками, имеется возможность включить опцию показа границ регионов. Делается это с помощью главного меню View / Regions (Вид / Регионы). Если эта опция включена, документ выглядит так, как показано на рис. 1.11 (см. рис. 1.4 для сравнения).

Присмотревшись к рис. 1.11 и некоторым другим рисункам этой главы, вы обнаружите в правой части рабочей области вертикальную линию раздела страниц. Если документ большой, то в некотором месте будет наблюдаться и прерывистая горизонтальная линия раздела страниц. Эти линии показывают, каким образом будет осуществлено разбиение на страницы при распечатке документа на принтере. Изменить параметры страницы можно с помощью команды File / Page Setup (Файл / Параметры страницы).

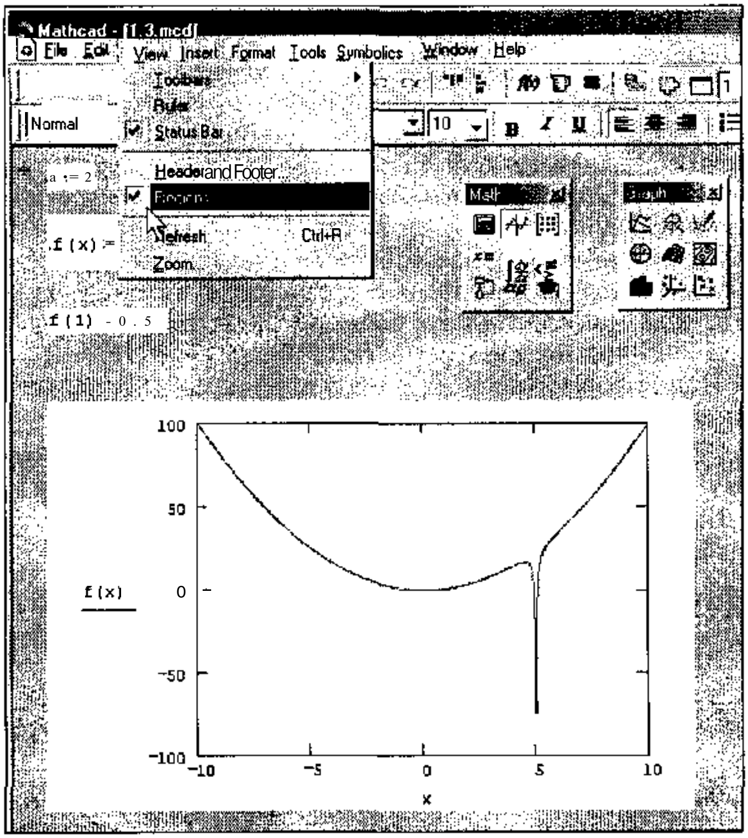


Рис. 1.11. Вид документа с выделенными границами регионов

Ориентироваться в размещении объектов на странице документа помогает горизонтальная линейка, расположенная под панелями инструментов в верхней части окна Mathcad (рис. 1.12). Линейку можно вызвать на экран с помощью команды View / Ruler (Вид / Линейка).

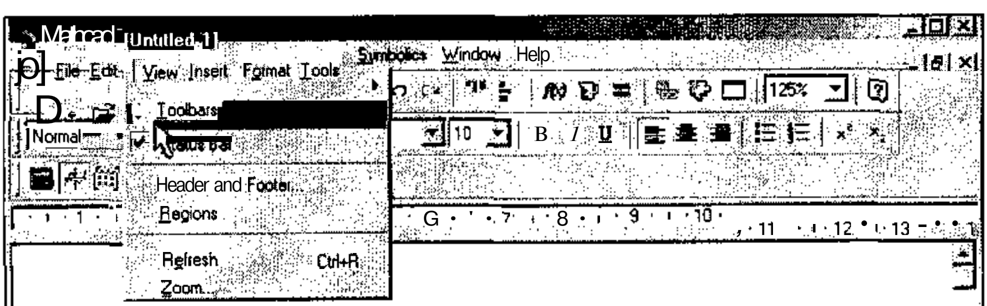


Рис. 1.12. Вызов линейки

Перемещение по документу

Просматривать документ вверх-вниз и вправо-влево удобно с помощью вертикальной и горизонтальной *полос прокрутки*, перемещая их бегунки (в этом случае обеспечивается плавное перемещение вдоль документа) или щелкая мышью с одной из двух сторон бегунка (при этом перемещение по документу будет скачкообразным). Также для перемещения курсора по документу можно использовать клавиши листания страниц <PgUp> и <PgDn>. Обратите внимание, что во всех перечисленных случаях положение курсора не меняется, а просто просматривается содержание документа. Кроме того, если документ имеет большой размер, просматривать его содержимое удобно при помощи меню Edit / Go to Page (Правка / Перейти к странице). При выборе этого пункта откроется диалог, позволяющий перейти к странице с заданным номером.

Для того чтобы двигаться по документу вверх-вниз и вправо-влево, перемещая курсор, следует нажимать на соответствующие клавиши управления курсором. Попадая в область регионов с формулами и текстом, курсор превращается в две *линии ввода* — вертикальную и горизонтальную синего цвета.

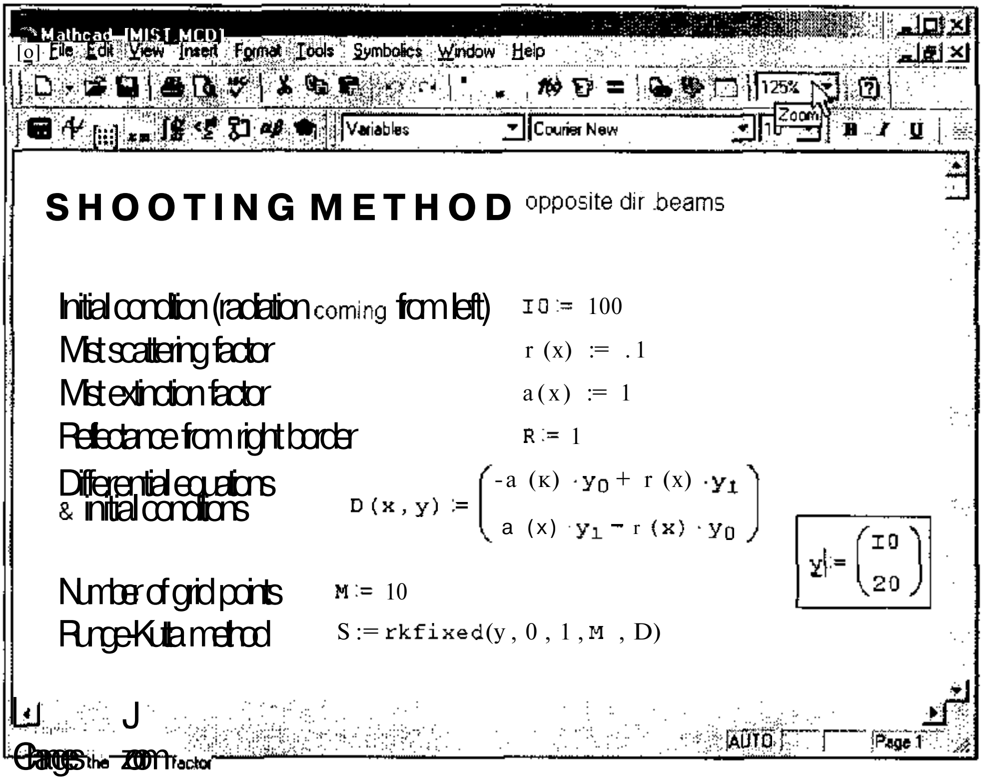


Рис. 1.13. Изменение масштаба отображения документа

При дальнейшем перемещении курсора внутри региона линии ввода смещаются на один символ в соответствующую сторону. При выходе за пределы региона курсор снова становится курсором ввода в виде красного крестика.

Конечно, переместить курсор можно и щелчком мыши в соответствующем месте. Если щелкнуть на пустом месте, то в нем появится курсор ввода, а если в пределах региона — то линии ввода.

Изменение масштаба

Изменение масштаба документа не влияет на его содержание, а просто определяет размер букв и графики, отображаемых на экране.

Для того чтобы изменить масштаб изображения, войдите в соответствующее поле на панели инструментов Standard, которое отмечено указателем мыши на рис. 1.13. Щелчок мыши на этом поле приводит к появлению списка возможных масштабов от 25 до 200%. Значение 100% соответствует размеру страницы документа, который получится при его распечатке. Сравните рис. 1.13 и 1.14, на которых один и тот же документ представлен при разном увеличении.

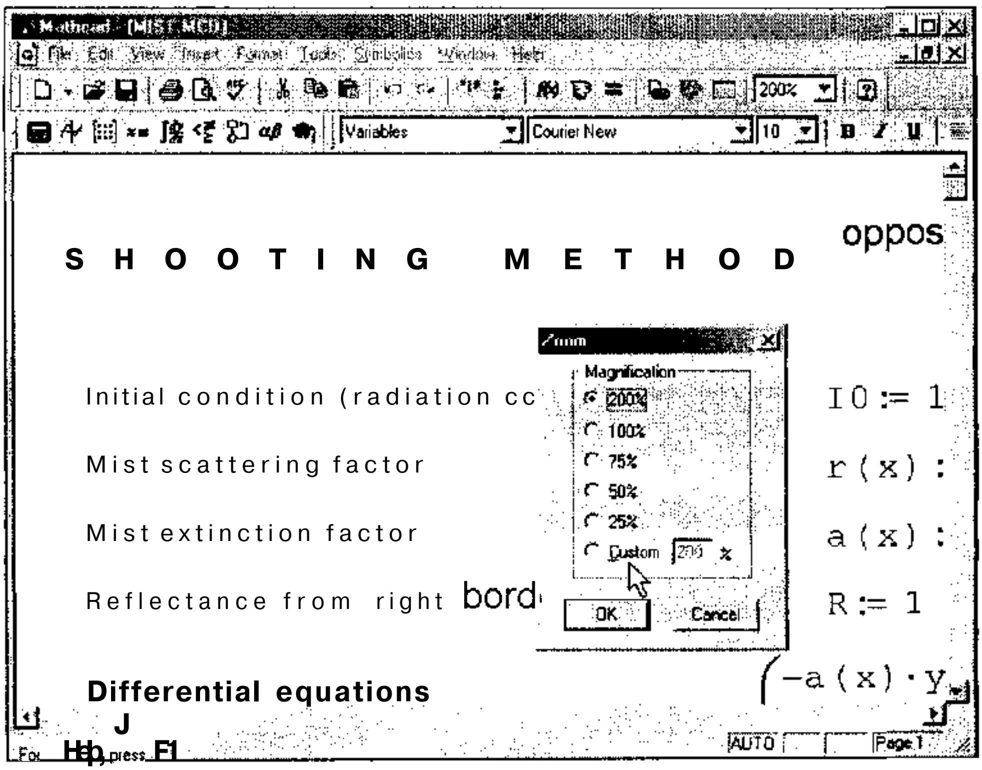


Рис. 1.14. Диалоговое окно выбора масштаба

Чтобы выбрать другое значение масштаба отображения документа, необходимо выполнить команду **View / Zoom** (Вид / Масштаб). В этом случае появляется диалоговое окно **Zoom** (Масштаб) управления масштабом (рис. 1.14), в котором можно выбрать один из переключателей с желаемым значением масштаба. Для задания значения вручную выберите переключатель **Custom** (Настройка) и в открывшемся текстовом поле введите нужное число (в процентах от реального масштаба страницы). Для подтверждения проделанных изменений нажмите кнопку ОК.

Многооконный режим редактирования

Все предыдущие рисунки были примерами одного документа, развернутого во все пространство окна Mathcad. Однако допускается одновременно держать на экране и редактировать сразу несколько документов. Их можно расположить на экране в любом порядке. Для этого, открыв меню **Window** (Окно), следует выбрать в нем один из пунктов **Cascade** (Расположить каскадом), **Tile Horizontal** (Горизонтальная мозаика), **Tile Vertical** (Вертикальная мозаика). В результате все окна будут расположены на экране в пределах окна либо каскадом друг за другом, либо вертикально или горизонтально так, чтобы они не перекрывались (рис. 1.15—1.17).

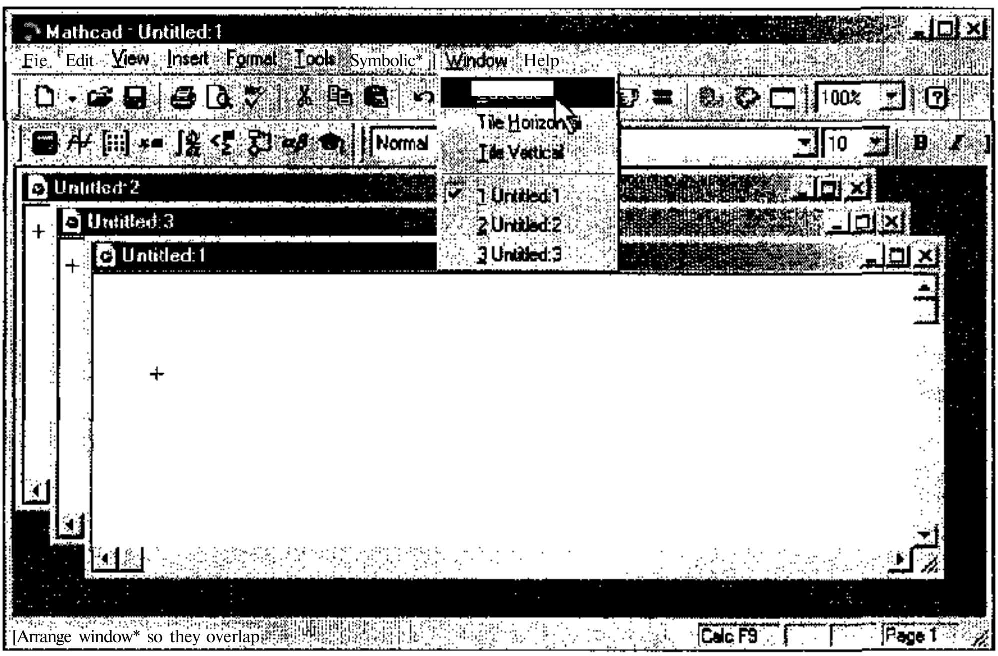


Рис. 1.15. Расположение документов каскадом

Обратите внимание, что в окне каждого документа расположен свой курсор (курсор ввода или линии редактирования, в зависимости от места в документе). Кроме того, для каждого документа легко включить либо отключить линейку, задать свой масштаб (как это сделано на рис. 1.17) или установить опцию отображения границ регионов. В каждый момент времени допускается редактирование только одного документа. Заголовок окна активного документа выделен более ярким цветом. Окно документа активизируется либо щелчком мыши в его пределах, либо выбором его имени в выпадающем меню Window (Окно). Имена открытых документов расположены в нижней части меню Window, а имя активного документа отмечено флажком.

Расположив несколько документов на экране, можно менять положение и размер каждого из них, перетаскивая их окна за заголовок и перетаскивая линии их границ.

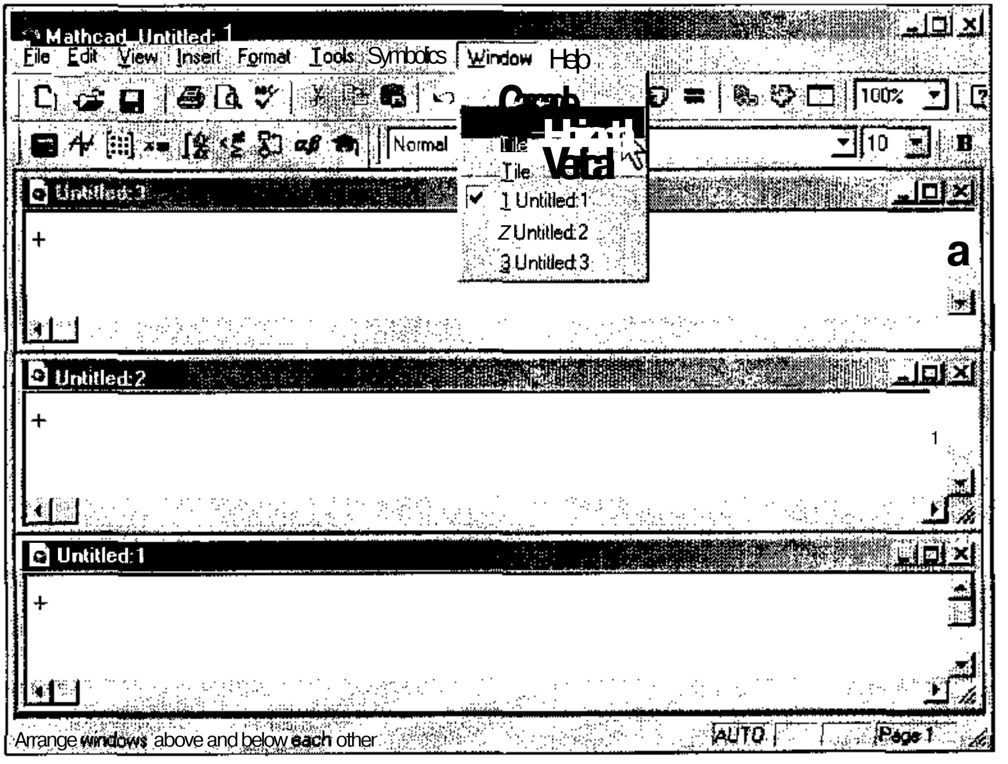


Рис. 1.16. Расположение документов по горизонтали

Примечание

В многооконном режиме удобно копировать объекты из одного документа в другой с помощью техники перетаскивания (Drag-and-Drop). Для этого достаточно ухватить объект указателем, используя левую кнопку мыши, за его границу и буксировать его в окно другого документа, не отпуская кнопку мыши (см. рис. 1.17).

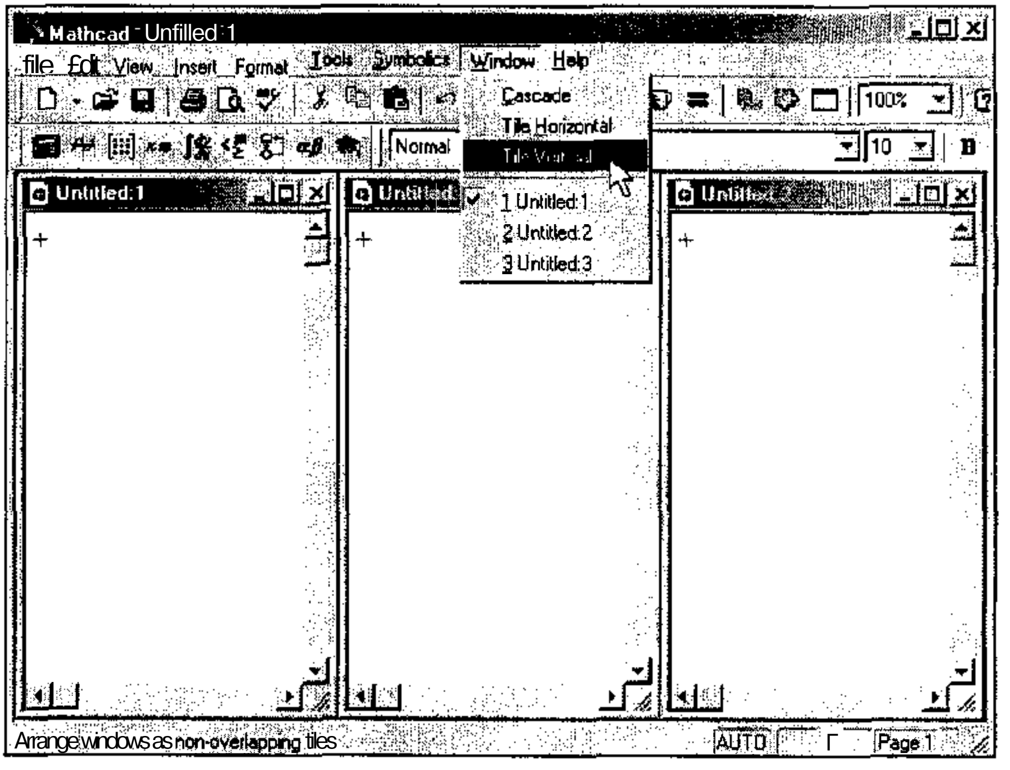


Рис. 1.17. Расположение документов по вертикали

В многооконном режиме любой документ можно закрыть или развернуть во весь экран с помощью кнопок управления окном в его правом верхнем углу. Когда окно документа развернуто, кнопки управления его окном помещаются в область верхнего меню, а переход между различными документами осуществляется только через меню **Window** (Окно).

1.3.5. Строка состояния

В нижней части окна Mathcad, под горизонтальной полосой прокрутки, на большинстве рисунков этой главы видна *строка (линия) состояния.* На ней отображается самая основная информация о режиме редактирования (рис. 1.18), разграниченная разделителями (слева направо):

- контекстно-зависимая подсказка о готовящемся действии;
- режим вычислений: автоматический **(AUTO)** или задаваемый вручную **(Calc F9);**
- текущий режим раскладки клавиатуры **CAP**;

- ❒ текущий режим раскладки клавиатуры **NUM**;
- ❒ номер страницы, на которой находится курсор.

Рис. 1.18. Строка состояния

Чтобы показать или скрыть строку состояния, выполните команду **View / Status Bar** (Вид / Строка состояния).

1.4. Справочная информация

Вместе с Mathcad поставляется несколько источников справочной информации, доступ к которым осуществляется через меню **Help** (Справка).

D Справочные системы по вопросам использования Mathcad:

- **Mathcad Help** (Справка) — система *справки* или *технической поддержки;*
- **What's This** (Что это такое?) — контекстно-зависимая интерактивная справка;
- **Developer's Reference** (Справка для разработчиков) — дополнительные главы справки для разработчиков собственных самостоятельных приложений на языке Mathcad;
- **Author's Reference** (Справка для авторов) — дополнительные главы справки для пользователей, разрабатывающих собственные электронные книги Mathcad.

❒ Ресурсы Mathcad — дополнительные материалы, организованные в специфическом формате электронных книг Mathcad с решением множества математических примеров:

- **Tutorials** (Учебники) — библиотека электронных книг Mathcad с примерами, которые построены в форме обучающих курсов;
- **QuickSheets** (Быстрые шпаргалки) — большое число документов Mathcad, которые удобно использовать в качестве шаблона для собственных расчетов;
- **Reference Tables** (Справочный стол) — физические и инженерные таблицы, включающие перечни фундаментальных констант, единиц измерения величин, сводку разнообразных параметров веществ и т. п.;
- **E-Books** (Электронные книги) — доступ к существующим библиотекам документов пользователя, примерам, а также встроенным электронным книгам, посвященным расширениям Mathcad.

Примечание

Техника создания электронных книг и возможности Ресурсов Mathcad рассматриваются в последней части этой книги. Там же рассмотрены дополнительные пути получения справочной **информации** — через Интернет, от сообщества пользователей Mathcad и его разработчиков.

Кроме поименованных, меню **Help** (Справка) содержит следующие пункты:

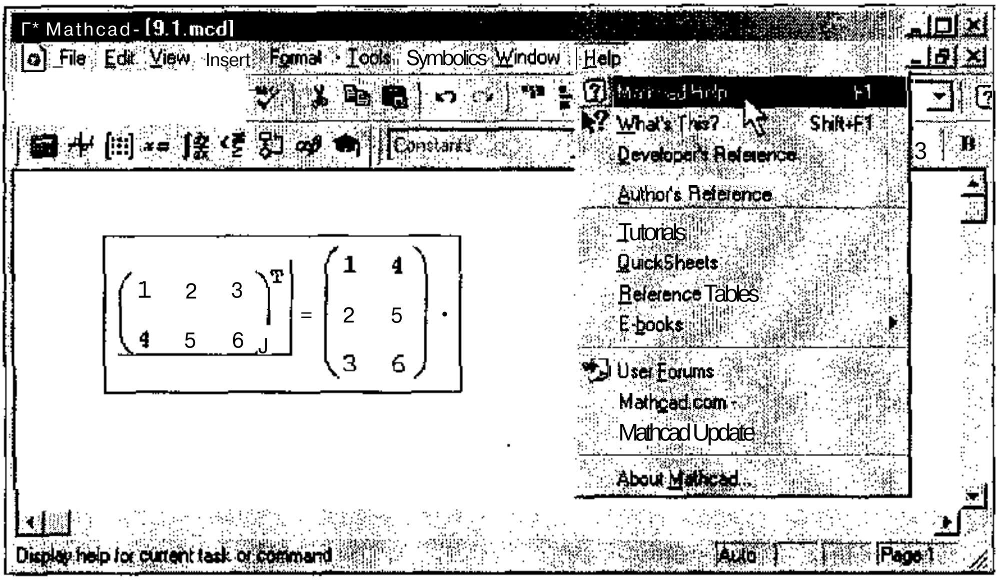


Рис. 1.19. Меню Help

Если в какой-либо момент работы с Mathcad вам потребовалась помощь, выберите **Help/ Mathcad Help,** либо нажмите клавишу <F1>, либо кнопку **Help** со знаком вопроса на стандартной панели инструментов.

Справка в Mathcad является контекстно-зависимой, т. е. ее содержание определяется тем, на каком месте документа она вызвана. Например, на рис. 1.19 курсор (линии ввода) указывает в редактируемом документе на оператор транспонирования матрицы. Поэтому вызов справки приведет к загрузке окна **Mathcad Help** (Справка Mathcad), открытого на месте описания операции транспонирования матриц (рис. 1.20).

Окно справочной системы Mathcad построено в характерном для Windows стиле и состоит из двух частей. Слева отображается содержание статей (вкладка **Contents),** а справа — их текст. Левая часть окна может быть временно скрыта нажатием кнопки **Hide** на панели инструментов в верхней части окна. Вновь вызвать ее на экран всегда можно кнопкой **Show** (Показать), заменяющей кнопку **Hide** в режиме скрытия содержания.

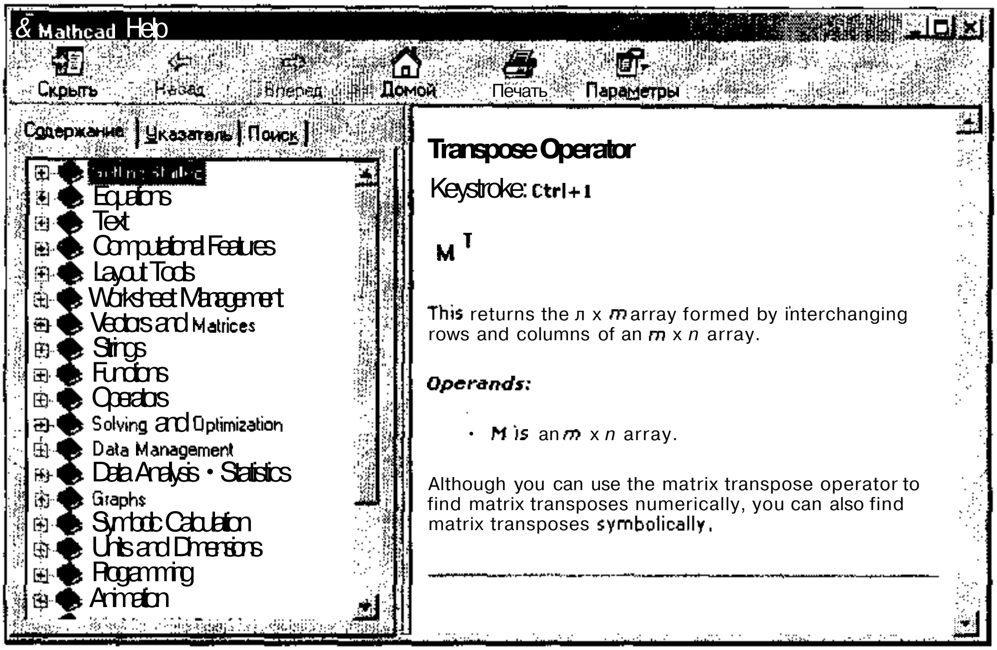


Рис. 1.20. Окно справочной системы Mathcad

Справочная система построена по принципу гиперссылок, находящихся в тексте статей и обеспечивающих переход от одной статьи к другой. Текст статьи загружается в правую часть окна. Для возвращения на однажды просмотренные страницы предусмотрены кнопки навигации **Back** (Назад), **Forward (Вперед)** и **Home** (Домой).

В тексте статей часто встречаются кнопки **QuickSheet Example** (Пример из шпаргалок), вызывающие образец из Ресурсов Mathcad, связанный с содержанием справочной статьи. Нажатие такой кнопки приведет к появлению окна **Mathcad Resource** с примером расчетов, относящихся к теме раздела справочной системы.

В левой части окна на вкладке Contents (Содержание) изначально перечислены только основные главы справочной системы, снабженные значком в виде закрытой книжки. Подзаголовки раскрываются с помощью двойного щелчка на названии нужной главы. При этом значок меняется на раскрытую книжку, а подзаголовки выводятся в сопровождении значков в виде вопросительного знака. Щелчок на любом из подзаголовков выводит соответствующую статью справки справа.

Статей гораздо больше, чем подзаголовков в содержании, для вывода большинства из них на экран потребуется один-два перехода по гиперссылкам. Кроме того, довольно мощное средство поиска статей на сходную тему — это кнопка Related Topics (Близкие статьи). Нажатие этой кнопки раскрывает диалоговое окно Topics Found (Найденные статьи) с перечнем справочных статей близкой тематики (рис. 1.21). Чтобы перейти к какой-либо статье, выберите соответствующий элемент из списка и нажмите кнопку Display (Показать) или просто дважды щелкните на элементе списка.

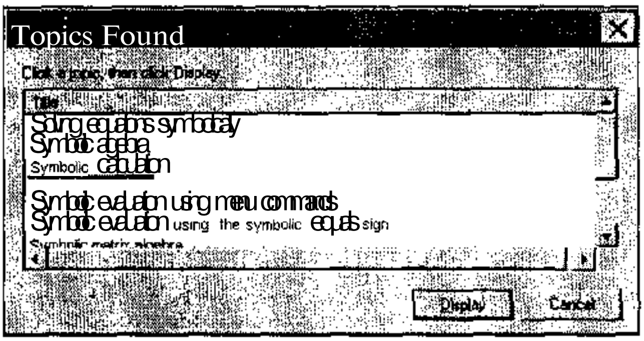


Рис. 1.21. Поиск близких по смыслу справочных статей

Примечание

Некоторые гиперссылки приводят не к переходу на другую страницу справочной системы, а выводят окно с термином (рис. 1.22). Для того чтобы после ознакомления убрать это окно, достаточно щелкнуть в его пределах мышью.

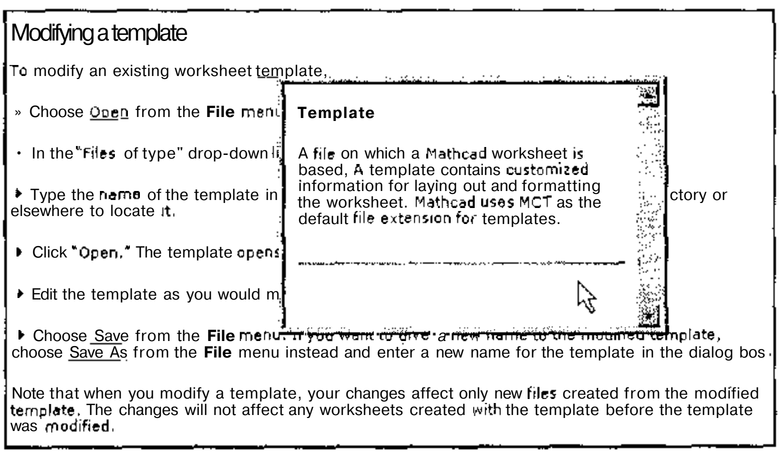


Рис. 1.22. Окно с термином на фоне справочной статьи

В левой части окна Mathcad Help (Справка Mathcad) отображается содержимое одной из трех закладок:

- Contents (Содержание) — вывод названий статей в рассмотренном смысловом порядке по главам и подзаголовкам;

- **Index** (Указатель) — перечень названий справочных статей в алфавитном порядке (рис. 1.23);
- **Search** (Поиск) — поиск статей справки по ключевым словам или фразам (рис. 1.24).

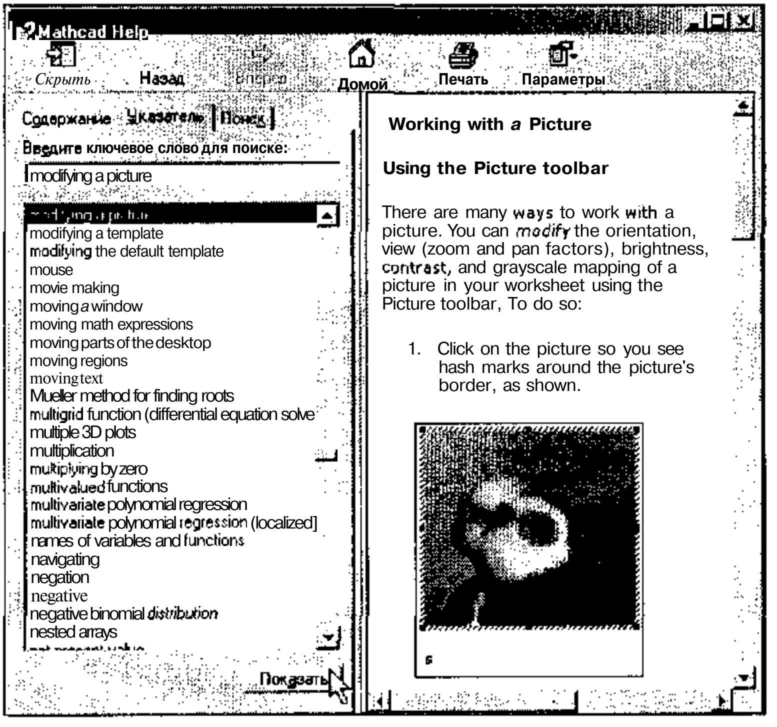


Рис. 1.23. Указатель справочных статей

На вкладке **Index** следует либо выбрать искомую статью из нижнего списка, либо ввести несколько первых букв в текстовое поле **Type in the keyword to find** (Введите ключевое слово для поиска). Для перехода к содержимому справки требуется дважды щелкнуть на элементе списка. После этого либо статья появится справа, либо будет выведено уже известное нам диалоговое окно **Topics Found** (Найденные статьи), позволяющее выбрать справочную статью из списка.

Если название искомой статьи известно точно, перейдите на вкладку **Index,** если же требуется отыскать справку по ключевым словам, выполните поиск по ключевому слову (рис. 1.24):

- перейдите в окне **Mathcad Help** на закладку **Search;**
- введите ключевые слова в текстовом поле сверху (через пробел, если слов несколько), например (для поиска статей о квадратных матрицах) "matrix square";

- нажмите кнопку **List Topics** (список статей);
- выберите из появившегося списка статью и нажмите кнопку **Display** (Показать);
- статья появится справа, причем искомые ключевые слова будут в ней выделены обращением цвета.

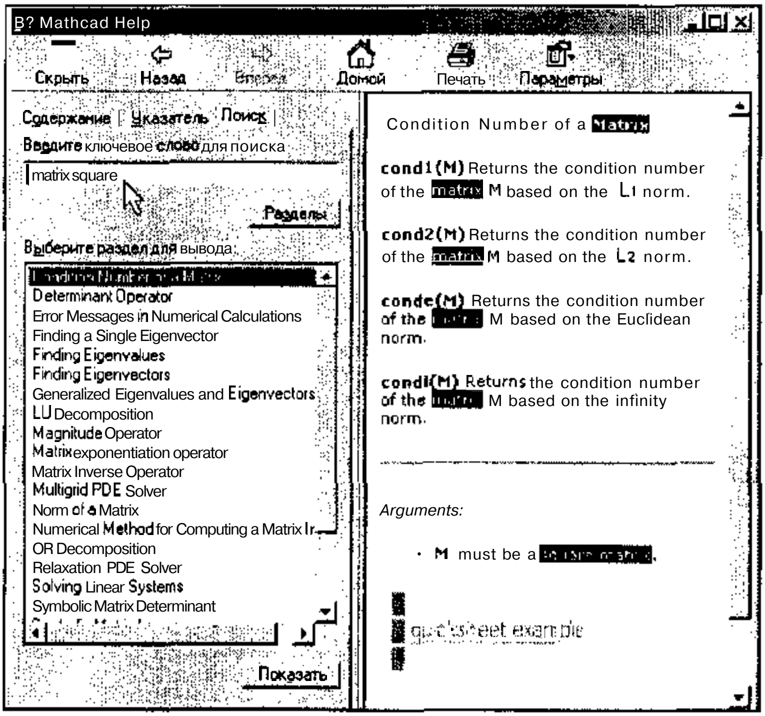


Рис. 1.24» Поиск справочных статей по ключевым словам

В заключение отметим, что как справочная система, так и Ресурсы Mathcad представляют собой не просто статьи и примеры с описанием его возможностей. Они могут быть названы полноправными учебными пособиями по нескольким курсам высшей математики (в случае Ресурсов, к тому же, еще и интерактивными). Там освещены основные определения, математический смысл многих операций и алгоритмы численных методов. Причем, на взгляд автора, некоторые из тем объяснены лучше, чем где бы то ни было. Если Вы в достаточной степени владеете английским, обязательно ознакомьтесь с Ресурсами Mathcad.

ГЛАВА 2

Редактирование документов

В данной главе рассматриваются основные приемы редактирования документов Mathcad. Первый раздел посвящен созданию новых документов и сохранению расчетов в файлах *(см. разд. 2.1)*. В трех следующих рассмотрены способы редактирования формул *(см. разд. 2.2)*, текста *(см. разд. 2.3)* и правки частей документа Mathcad *(см. разд. 2.4)*. В заключение главы приводятся основные сведения по распечатке документов *(см. разд. 2.5)*, рассылке их по электронной почте *(см. разд. 2.6)* .

2.1. Работа с документами

В Mathcad все расчеты организуются на рабочих областях, или "листах" (worksheets), изначально пустых, на которые можно добавлять формулы и текст. Здесь и далее будем называть рабочий лист *документом* Mathcad. Это не совсем точно передает смысл английского термина "worksheet", зато более привычно с точки зрения терминологии Windows-приложений. Каждый документ представляет собой независимую серию математических расчетов и сохраняется в отдельном файле. Документ является одновременно и листингом Mathcad-программы, и результатом исполнения этой программы, и отчетом, пригодным для распечатки на принтере или публикации в Web.

2.1.1. Управление документами

Если Mathcad запускается из главного меню Windows (с помощью кнопки **Пуск** в углу экрана), например **Start /Programs /MathSoft Apps /Mathcad 11** (Пуск/ Программы /Приложения MathSoft /Mathcad 11), то окно Mathcad появляется с открытым в нем новым пустым безымянным документом, условно называемым **Untitled: 1**.

Для того чтобы создать новый пустой документ, уже работая в Mathcad, следует выполнить одно из трех эквивалентных действий:

- нажатие одновременно клавиш <Ctrl>+<N>;
- нажатие кнопки **New** (Создать) на панели инструментов;
- щелкнув на команде верхнего меню **File** / **New** (Файл / Создать).

В Mathcad 11 кнопка **New** на стандартной панели состоит из двух частей (рис. 2.1).

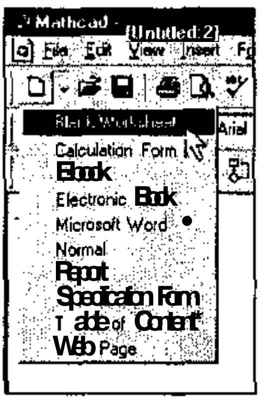


Рис. 2.1. Кнопка New

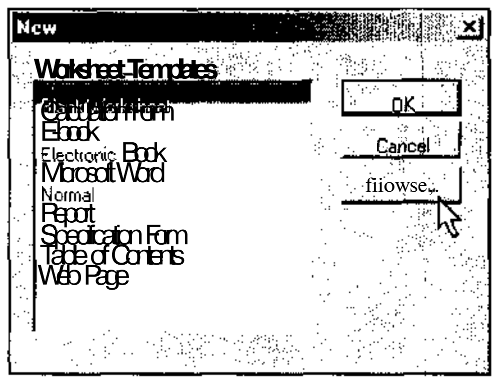


Рис. 2.2. Диалоговое окно New

В результате любого из проделанных действий в окне Mathcad появляется пустой документ с условным названием **Untitled:2**, или **Untitled:3** или т. д., в зависимости от того, какой по счету новый документ создается. *(Сохранение документа описывается в разд. 2.1.3.)*

2.1.2. Создание документа на основе шаблона

Поработав с Mathcad некоторое время, вы, скорее всего, часто будете создавать новые документы не с чистого листа, а на основе разработанных ранее. Для этого имеются два пути:

- открыть ранее созданный документ и сохранить его под другим именем;
- использовать шаблоны.

Шаблон (template) — это заготовка нового документа с введенными формулами, графиками, текстом, включая разметку, форматирование, выбор по умолчанию режима вычислений и т. д. В предыдущем разделе было рассмотрено создание нового документа на основе пустого шаблона. Пример документа, созданного на основе шаблона **Web Page** (Web-страница), показан на рис. 2.3. Новый документ, имеющий дизайн и характерную разметку Web-страницы, следует использовать, например, для представления расчетов в формате HTML в сети Интернет.

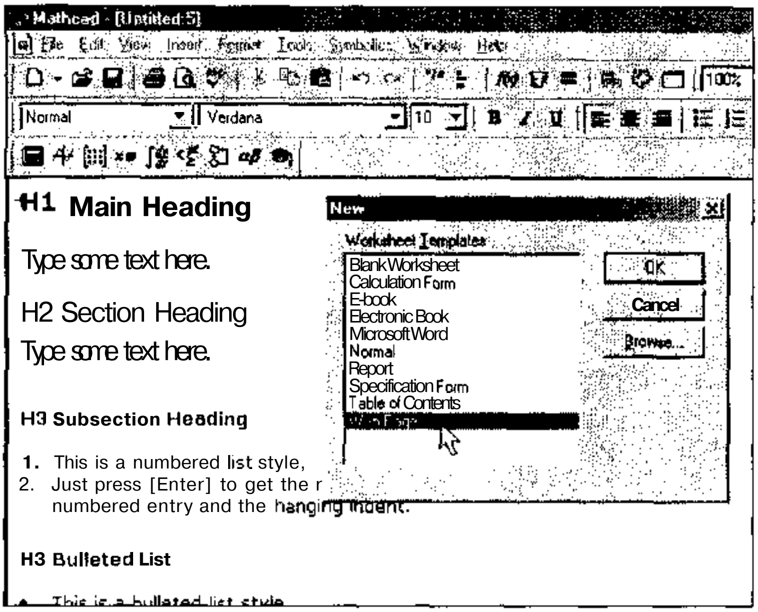


Рис. 2.3. Новый документ на основе шаблона Web-страницы

Для выбора другого шаблона, который имеется на Вашем компьютере в виде файла соответствующего формата, нажмите в диалоговом окне **New** (Создать), показанном на рис. 2.2, кнопку **Browse** (Обзор). В появившемся диалоге **Browse** (Обзор) найдите местоположение нужного файла с шаблоном Mathcad. Эти файлы имеют расширение .mct (Math Cad Template). Выберите желаемый шаблон в списке файлов и нажмите кнопку **Открыть** (Open).

В результате этих действий будет создан новый документ с уже имеющимися элементами оформления и определенными установками.

Для того чтобы разработать шаблон самостоятельно, проделайте следующее:

1. Отредактируйте документ, вводя формулы, текст, графики, отформатируйте их и задайте прочие установки документа.
2. Выполните команду **File / Save As** (Файл / Сохранить как).
3. Найдите на диске папку \template (ее местонахождение может быть примерно таким: \Program Files \ Mathcad 11 \ template) и перейдите к ее содержимому (рис. 2.4).
4. Выберите в списке **Тип файла** (File type) элемент **Mathcad Template (*.mct)** (Шаблон Mathcad).
5. Введите имя шаблона в поле ввода имени файла (на рис. 2.4 он назван "user1").
6. Нажмите кнопку **Сохранить** (Save).

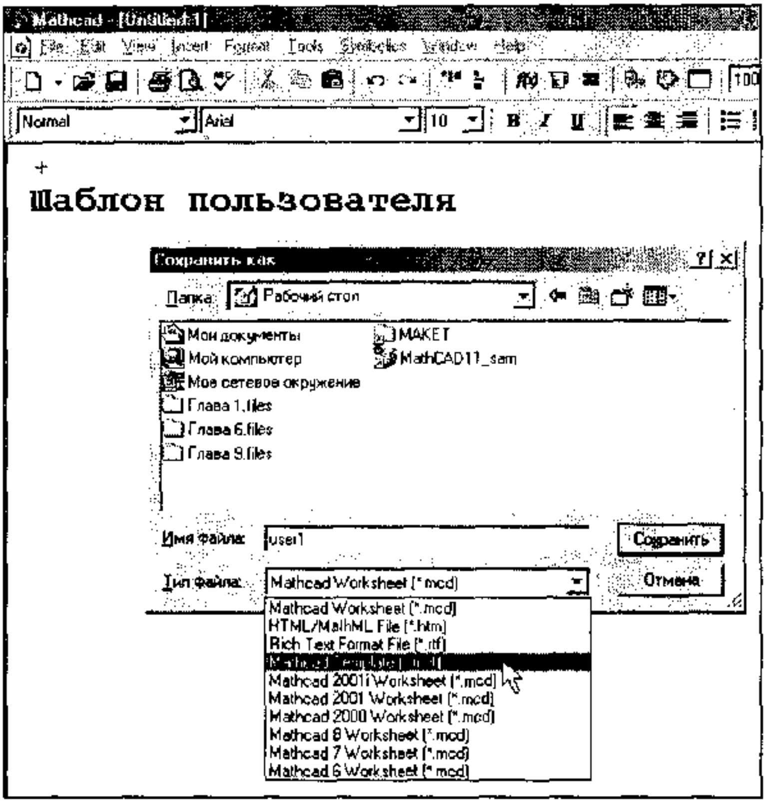


Рис. 2.4. Сохранение шаблона в файле

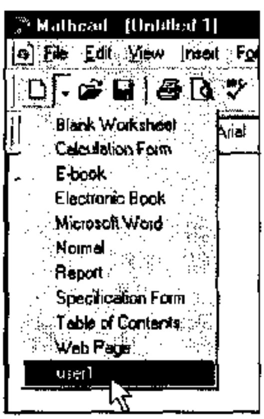


Рис. 2.5. Сохраненный шаблон в списке шаблонов

Таким образом, при создании нового документа Ваш шаблон будет доступен как через меню File/ New (Файл/ Создать), так и через кнопку New на панели инструментов (рис. 2.5).

Впоследствии шаблон можно редактировать, открывая его как обычный документ Mathcad, помня лишь о том, что искать его надо среди файлов с расширением .mct.

Примечание

Допускается как сохранение шаблона в любом другом месте на жестком диске, так и использование в качестве шаблона документа Mathcad в обычном формате MCD.

2.1.3. Сохранение документа

Для того чтобы сохранить документ в формате Mathcad, выберите **File / Save** (Файл / Сохранить), либо нажмите клавиши <Ctrl>+<S> или кнопку Save на стандартной панели инструментов. Если созданный документ сохраняется впервые, на экран будет выведено диалоговое окно **Сохранение** (Save), в котором потребуется определить его имя (рис. 2.6).

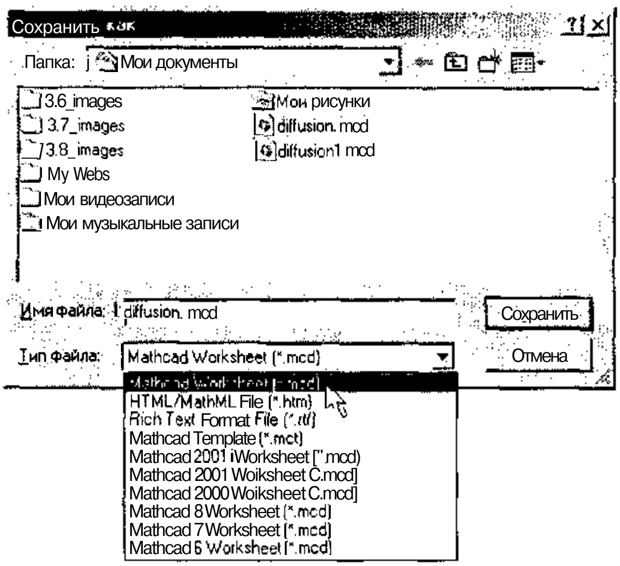


Рис. 2.6. Сохранение документа

Чтобы переименовать документ, сохраните его под другим именем с помощью команды **File / Save As** (Файл / Сохранить как).

Совет

Чаще создавайте резервные копии документов, чтобы сохранить результаты прежней работы.

На рис. 2.6 показан раскрывающийся список с возможными форматами сохраняемых файлов:

- **Mathcad 11 Worksheet (*.mcd)** — последний и наиболее мощный формат, используется по умолчанию;
- **HTML/MathML File (*.htm)** — формат Web-страницы. Начиная с версии Mathcad 11, все атрибуты документа Mathcad могут сохраняться в HTML-файле (с дополнительной XML-разметкой). С одной стороны, такие файлы могут просматриваться обычным браузером, а с другой — без ущерба для функциональности — открываться и редактироваться в Mathcad как обычные (*.mcd) документы.

Примечание

При сохранении в формате *.htm можно выбирать различные опции экспорта. *Подробнее об этом написано в последнем разделе данной главы.*

- **Mathcad Template (*.mct)** — формат шаблона *(см. предыдущий разд.)*;
- **Rich Text Format (*.rtf)** — сохраняйте файлы в этом формате только для дальнейшего редактирования в текстовых редакторах с целью создания отчетов. В частности, сохранив документ в RTF-файле, можно загрузить его в Microsoft Word или другой текстовый процессор, большинство из которых поддерживает этот формат;
- **Mathcad 6...2001i Worksheet (*.mcd)** — форматы прежних версий Mathcad. Если вы работаете одновременно с несколькими версиями Mathcad (например, разрабатываете с другими разработчиками общую задачу), то запасайте файлы в наиболее раннем формате из тех, с которыми приходится иметь дело. Однако помните, что возможности прежних версий (в частности, наборы встроенных функций) ограничены по сравнению с более поздними версиями Mathcad, поэтому некоторые из них будут недоступны.

2.1.4. Открытие существующего документа

Чтобы открыть существующий документ для редактирования, выполните команду **File / Open** (Файл / Открыть) или нажмите клавиши <Ctrl>+<0> (или кнопку **Open** на стандартной панели инструментов). В диалоговом окне **Open** выберите файл и нажмите кнопку OK.

Кроме того, открыть файл можно и в обозревателе Windows, щелкнув дважды на его имени с расширением .mcd.

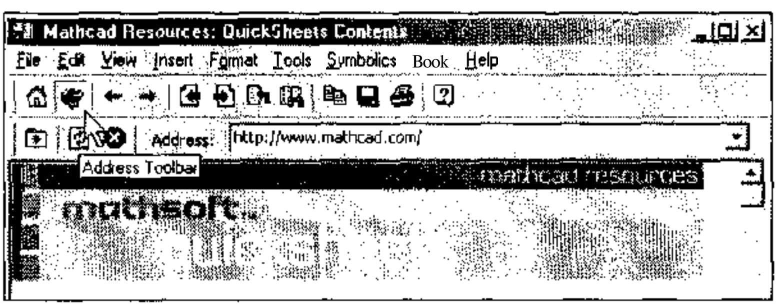


Рис. 2.7. Отображение панели адреса URL в окне Ресурсов Mathcad

Открыть документ Mathcad, находящийся в сети Интернет, можно мри помощи окна Ресурсов Mathcad:

1. Вызовите один из Ресурсов Mathcad, например, Быстрые шпаргалки **(Help / QuickSheets).**

2. Нажмите кнопку с изображением глобуса и двух стрелок на панели инструментов появившегося окна **Mathcad Resources** (рис. 2.7).
3. Введите URL-адрес страницы в сети Интернет, где находится документ Mathcad, например **http://www.mathsoft.com** — в поле для ввода адреса в окне.
4. Нажмите клавишу <Enter>.

2.1.5. Закрытие документа

Активный документ закрывается одним из способов:

- нажатием кнопки закрытия окна документа в его правой верхней части (рис. 2.8);

Рис. 2.8. Кнопки управления окном

- при помощи команды **File** / **Close** (Файл / Закрыть);
- нажатием клавиш <Ctrl>+<W>;
- при завершении сеанса работы с Mathcad: посредством либо команды **File** / **Exit** (Файл / Выход), либо кнопки управления окном, либо панели задач Windows, — будут закрыты все открытые документы, включая и неактивные.

Если внесенные изменения не были сохранены, Mathcad предложит сделать это, выводя на экран соответствующее диалоговое окно. Следует либо сохранить файл в окончательном виде, либо отказаться от изменений, либо вернуться к редактированию, нажав кнопку **Cancel** (Отмена).

2.2. Ввод и редактирование формул

Формульный редактор Mathcad позволяет быстро и эффективно вводить и изменять математические выражения. Тем не менее, некоторые аспекты его применения не совсем интуитивны, что связано с необходимостью избежать ошибок при расчетах по этим формулам. Поэтому не пожалейте немного времени на знакомство с особенностями формульного редактора, и впоследствии при реальной работе Вы сэкономите гораздо больше.

2.2.1. Элементы интерфейса

Перечислим еще раз элементы интерфейса редактора Mathcad *(со многими из них мы познакомились в разд. "Знакомство с Mathcad" гл. 1)*:

- указатель мыши (mouse pointer) — играет обычную для приложений Windows роль, следуя за движениями мыши;

- курсор — обязательно находится внутри документа в одном из трех видов:
  - курсор ввода (crosshair) — крестик красного цвета, который отмечает пустое место в документе, куда можно вводить текст или формулу;
  - линии ввода (editing lines) — горизонтальная (underline) и вертикальная (insertion line) линии синего цвета, выделяющие в тексте или формуле определенную часть;
  - линия ввода текста (text insertion point) — вертикальная линия, аналог линий ввода для текстовых областей.
- местозаполнители (placeholders) — появляются внутри незавершенных формул в местах, которые должны быть заполнены символом или оператором:
  - местозаполнитель символа — черный прямоугольник;
  - местозаполнитель оператора — черная прямоугольная рамка.

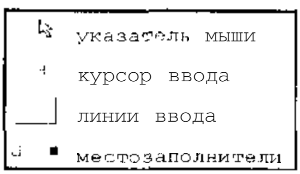


Рис. 2.9. Интерфейс редактирования

Курсоры и местозаполнители, относящиеся к редактированию формул, представлены на рис. 2.9.

2.2.2. Ввод формул

Ввести математическое выражение можно в любом пустом месте документа Mathcad. Для этого поместите курсор ввода в желаемое место документа, щелкнув в нем мышью, и просто начинайте вводить формулу, нажимая клавиши на клавиатуре. При этом в документе создается *математическая область* (math region), которая предназначена для хранения формул, интерпретируемых процессором Mathcad. Продемонстрируем последовательность действий на примере ввода выражения x^{5+x} (рис. 2.10):

1. Щелкните мышью, обозначив место ввода.
2. Нажмите клавишу <x> — в этом месте вместо курсора ввода появится регион с формулой, содержащей один символ x, причем он будет выделен линиями ввода.
3. Введите оператор возведения в степень, нажав клавишу <^>, либо выбрав кнопку возведения в степень на панели инструментов **Calculator** — в формуле появится местозаполнитель для введения значения степени, а линии ввода выделят этот местозаполнитель.
4. Последовательно введите остальные символы <5>, <+>, <x>.

Рис. 2.10. Пример ввода формулы (коллаж)

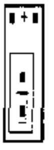


Рис. 2.11. Пример начала ввода операторов

Таким образом, поместить формулу в документ можно, просто начиная вводить символы, числа или операторы, например + или /. Во всех этих случаях на месте курсора ввода создается математическая область, иначе называемая *регионом,* с формулой, содержащей и линии ввода. В последнем случае, если пользователь начинает ввод формулы с оператора (рис. 2.11), в зависимости от его типа, автоматически появляются и местозаполнители, без заполнения которых формула не будет восприниматься процессором Mathcad.

2.2.3. Перемещение линий ввода внутри формул

Чтобы изменить формулу, щелкните на ней мышью, поместив таким образом в ее область линии ввода, и перейдите к месту, которое хотите исправить. Перемещайте линии ввода в пределах формулы одним из двух способов:

- щелкая в нужном месте мышью;
- нажимая на клавиатуре клавиши — со стрелками, пробел и <Ins>:
  - клавиши со стрелками имеют естественное назначение, переводя линии ввода вверх, вниз, влево или вправо;
  - клавиша <Ins> переводит вертикальную линию ввода с одного конца горизонтальной линии ввода на противоположный;
  - пробел предназначен для выделения различных частей формулы.

Если раз за разом нажимать клавишу пробела в формуле, пример ввода которой рассмотрен выше (см. рис. 2.10), то линии ввода будут циклически изменять свое положение, как это показано на рис. 2.12. Если в ситуации, показанной сверху на этом рисунке, нажать стрелку ←, то линии ввода переместятся влево (рис. 2.13). При нажатии пробела теперь линии ввода будут попеременно выделять одну из двух частей формулы.

Совет

Привыкнув к использованию пробела для перемещения внутри формул, можно существенно облегчить себе работу с Mathcad.

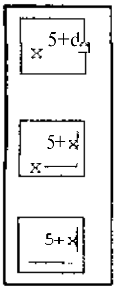


Рис. 2.12. Изменение положения линий ввода с помощью пробела (коллаж)

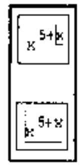


Рис. 2.13. Изменение положения линий ввода пробелом после сдвига стрелкой ← (коллаж)

Таким образом, комбинация клавиш со стрелками и пробела позволяет легко перемещаться внутри формул. Накопив некоторый опыт, Вы без труда освоите эту технику. Иногда поместить линии ввода в нужное место формулы с помощью указателя мыши непросто. Поэтому в Mathcad для этого лучше использовать клавиатуру.

2.2.4. Изменение формул

Редактируйте формулы в Mathcad так, как подсказывают Вам интуиция и опыт работы с другими текстовыми редакторами. Большинство операций правки формул реализованы естественным образом, однако некоторые из них несколько отличаются от общепринятых, что связано с особенностью Mathcad как вычислительной системы. Рассмотрим основные действия по изменению формул.

Вставка оператора

Операторы могут быть унарными (действующими на один операнд, как, например, оператор транспонирования матрицы или смены знака числа), так и бинарными (например + или /, действующими на два операнда). При вставке нового оператора в документ Mathcad определяет, сколько операндов ему требуется. Если в точке вставки оператор один или оба операнда отсутствуют, Mathcad автоматически помещает рядом с оператором один или два местозаполнителя.

Внимание!

То выражение в формуле, которое выделено линиями ввода в момент вставки оператора, становится его первым операндом.

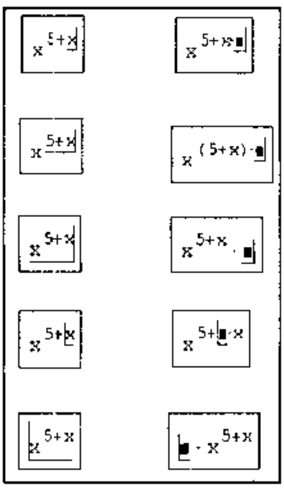
Последовательность вставки оператора в формулу такова:

1. Поместите линии ввода на часть формулы, которая должна стать первым операндом.
2. Введите оператор, нажав кнопку на панели инструментов или сочетание клавиш.

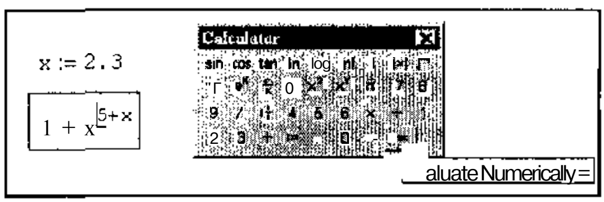
Примечание

Для того чтобы вставить оператор не после, а перед частью формулы, выделенной линиями ввода, нажмите перед его вводом клавишу <Ins>, которая передвинет вертикальную линию ввода вперед. Это важно, в частности, для вставки оператора отрицания.

На рис. 2.14 показано несколько примеров вставки оператора сложения в разные части формулы, создание которой мы подробно разбирали выше (см. рис. 2.10). В левой колонке рис. 2.14 приведены возможные размещения линий ввода в формуле, а в правой — результат вставки оператора сложения (т. е. нажатия клавиши <+>). Как видно, Mathcad сам расставляет, если это необходимо, скобки, чтобы часть формулы, отмеченная линиями ввода, стала первым слагаемым.



< **Рис. 2.14.** Вставка оператора в разные части формулы (коллаж)



▲ **Рис. 2.15.** Вставка оператора вывода

Некоторые операторы Mathcad вставит в правильное место независимо от положения линий ввода. Таков, например, оператор численного вывода =, который по смыслу выдает значение всей формулы в виде числа. На рис. 2.15 показан момент ввода этого оператора в формулу при помощи панели **Calculator** (Калькулятор), а в листинге 2.1 приведен результат его действия.

Листинг 2.1. Вычисление простого выражения

$x := 2.3$

$1 + x^{5+x} = 438.133$

Выделение части формулы

Чтобы выделить часть формулы в некоторой математической области (рис. 2.16):

1. Поместите ее между линиями ввода, пользуясь, при необходимости, клавишами-стрелками и пробелом.
2. Поместите указатель мыши на вертикальную линию ввода, нажмите и удерживайте левую кнопку мыши.
3. Удерживая кнопку мыши, протащите указатель мыши вдоль горизонтальной линии ввода, при этом часть формулы будет выделяться обращением цвета.
4. Отпустите кнопку мыши, когда будет выделена нужная часть формулы.

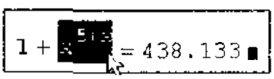


Рис. 2.16. Выделение части формулы

Примечание

Часть формулы можно выделить и без помощи мыши, нажимая клавиши со стрелками при удерживаемой клавише <Shift>. В этом случае вместо перемещения линий ввода происходит выделение соответствующей части формулы. Многие пользователи находят работу с клавиатурой при выделении части математических областей более удобной.

Удаление части формулы

Чтобы удалить часть формулы:

1. Выделите ее.
2. Нажмите клавишу <Del>.
3. Кроме того, можно удалить часть формулы, помещая ее перед вертикачьной линией ввода и нажимая клавишу <BackSpace>.

Примечание

Имеется еще один способ удаления части формулы: выделите ее нужную часть, затем нажмите комбинацию клавиш <Ctrl>+<X>, тем самым вырезая и помещая ее в буфер обмена. Этот способ удобен в случае, если требуется использовать фрагмент формулы в дальнейшем.

Вырезка, копирование и вставка части формулы

Для правки части формулы:

1. Выделите ее либо просто поместите между линиями ввода, пользуясь либо мышью, либо клавишами-стрелками и пробелом.

2. Воспользуйтесь либо верхним меню **Edit** (Правка), либо контекстным меню, либо кнопкой на панели инструментов, либо соответствующим сочетанием горячих клавиш (см. рис. 2.17 и 2.18):
   - **Cut** (Вырезать) или <Ctrl>+<X> — для вырезки части формулы в буфер;
   - **Copy** (Копировать) или <Ctrl>+<C> — для копирования в буфер;
   - **Paste** (Вставить) или <Ctrl>+<V> — для вставки из буфера предварительно помещенной туда части формулы.

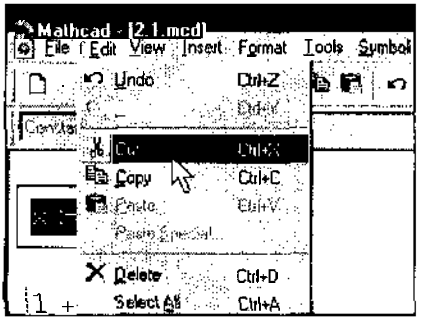


Рис. 2.17. Правка формул с помощью верхнего меню

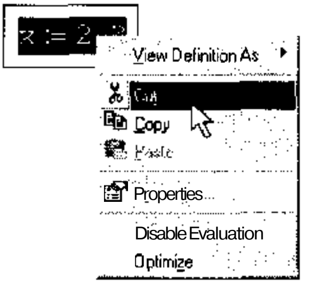


Рис. 2.18. Правка формул с помощью контекстного меню

Чтобы переместить (или скопировать) часть формулы из одной части документа в другую, вырежьте (скопируйте) ее в буфер обмена, перейдите к желаемому новому местоположению и вставьте ее туда из буфера.

Изменение чисел или имен переменных и функций

Для того чтобы в уже введенном математическом выражении изменить какое-нибудь число или имя (переменной или функции):

1. Щелкните мышью на имени переменной или функции, при необходимости передвиньте линии ввода, пользуясь либо мышью, либо клавишами-стрелками и пробелом.
2. Введите с клавиатуры другие числа или буквы, при необходимости удалите существующие символы, помещая их перед вертикальной линией ввода и нажимая клавишу <BackSpace>.

Примечание

Иногда бывает удобнее удалить старую часть формулы и в появившемся **местозаполнителе** ввести новое имя или число.

Изменение операторов

Для того чтобы удалить оператор, поместите его перед вертикальной линией ввода и нажмите клавишу <BackSpace>. В результате оператор либо исчезнет (а операнды слева и справа сольются в одно имя), либо {в сложных формулах) появится местозаполнитель оператора в виде черной рамки. При желании можно удалить и этот местозаполнитель повторным нажатием <BackSpace>.

2.2.5. Ввод символов, операторов и функций

Подведем некоторый итог. Математические выражения содержат, как правило, самые различные, в том числе специфичные символы, набор которых в Mathcad выполняется не так, как в большинстве текстовых процессоров. Для вставки символов в документы доступны следующие инструменты:

- большинство символов, например латинские буквы или цифры, для определения имен переменных и функций набираются на клавиатуре;
- греческие буквы легче всего вставляются с помощью панели инструментов **Greek** (Греческие символы) (рис. 2.19). Можно также ввести соответствующую латинскую букву и нажать клавиши <Ctrl>+<G> (после этого, например, из латинской буква "а" получается греческая a);
- операторы могут быть вставлены либо с различных математических панелей инструментов, либо соответствующим сочетанием клавиш. Например, наиболее часто употребляемые операторы (см. рис. 2.15) сгруппированы на панели **Calculator** (Калькулятор);
- имена функций вводятся либо с клавиатуры, либо, с большей надежностью, с помощью команды **Insert** / **Function** (Вставка / Функция) *(см. разд. "Знакомство с Mathcad" гл. 1).*
- Скобки могут быть вставлены нажатием соответствующих клавиш. Однако, для того чтобы выделить скобками уже введенную часть формулы, лучше поместить ее между линиями ввода и нажать клавишу <'> (апостроф).

Особенности применения операторов и функций будут подробно рассмотрены в гл. 3.

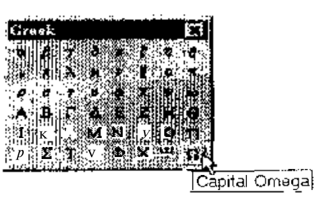


Рис. 2.19. Панель инструментов Greek

2.2.6. Управление отображением некоторых операторов

Некоторые операторы, например умножения или присваивания численного вывода, допускают различное представление в документах Mathcad. Сделано это, главным образом, для упрощения подготовки отчетов (в Mathcad-программе символ присваивания в виде := выглядит естественно, но в отчетной документации зачастую неприемлем).

Оператор умножения может иметь различный вид (рис. 2.20):

- Dot (Точка);
- Narrow Dot (Узкая точка);
- Large Dot (Большая точка);
- **x;**
- Thin Space (Тонкий пробел);
- No Space (Вместе).

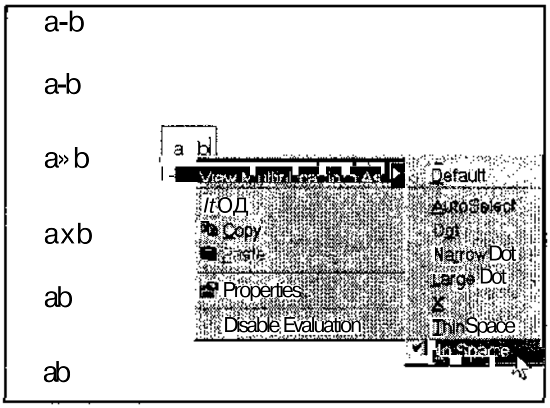


Рис. 2.20. Различный вид оператора умножения и его изменение

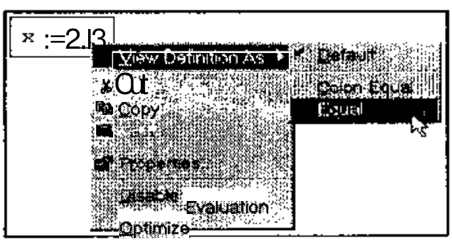


Рис. 2.21. Смена вида оператора присваивания

Оператор присваивания представляется либо знаками := (двоеточием и равенством — Colon Equal), либо просто знаком равенства (Equal), как показано на рис. 2.21.

Для того чтобы в документе поменять отображение указанных операторов:

1. Поместите указатель мыши на оператор и вызовите щелчком правой кнопкой мыши контекстное меню.
2. Наведите указатель мыши на его первый пункт.
3. В открывшемся подменю выберите нужный вид оператора, как показано на рис. 2.20 и 2.21.

Обратите внимание, что в режиме редактирования формулы, символ оператора временно меняется на представление по умолчанию, даже если выбран другой.

Совет

Изменять вид операторов следует только при серьезной необходимости. Помните, что непривычный вид оператора может ввести вас при дальнейшем разборе программ в заблуждение и вызвать ошибки.

Всегда можно вновь переложить ответственность за представление операторов на Mathcad, выбирая в контекстном меню пункт **Default** (По умолчанию). Кроме того, выбор пункта AutoSelect (Автоматический выбор) этого меню приводит к отображению оператора редактором Mathcad в зависимости от контекста.

Выбор представления по умолчанию для этих и некоторых других операторов производится на вкладке **Display** (Отображение) диалогового окна **Worksheet Options** (Опции документа) (рис. 2.22). Для его вызова выполните команду **Tools / Worksheet Options** (Сервис / Опции). Отображение по умолчанию задается с помощью соответствующих раскрывающихся списков, в частности списка **Multiplication** для оператора умножения или списка **Definition** для оператора присваивания.

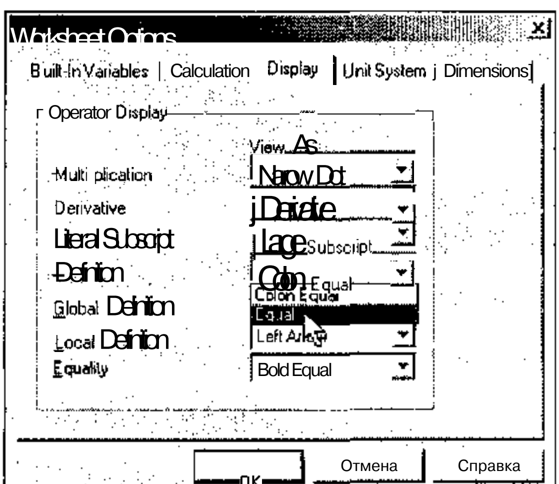


Рис. 2.22. Диалоговое окно Worksheet Options

2.3. Ввод и редактирование текста

Mathcad — это математический редактор. Основное его назначение заключается в редактировании математических формул и расчете по ним. Вместе с тем, наряду с формульным редактором Mathcad обладает довольно развитыми средствами по оформлению текста. Назначение текстовых областей в документах Mathcad для разных пользователей и разных задач может быть различным. Качественно стоит различать подход к тексту, используемому:

- просто в качестве комментариев;
- как элемент оформления документов для создания качественных отчетов в печатной и электронной формах.

Рассмотрим в этом разделе основные приемы работы с текстом, а разбор расширенных возможностей текстового процессора Mathcad отложим до последней части этой книги *(см. гл. 16).*

2.3.1. Ввод текста

Текстовую область (или, по-другому, *регион с текстом* — text region) можно разместить в любом незанятом месте документа Mathcad. Однако, когда пользователь помещает курсор ввода в пустое место документа и просто начинает вводить символы, Mathcad по умолчанию интерпретирует их как начало формулы. Чтобы до начала ввода указать программе, что требуется создать не формульный, а текстовый регион, достаточно, перед тем как ввести первый символ, нажать клавишу <">. В результате на месте курсора ввода появляется новый текстовый регион, который имеет характерное выделение (рис. 2.23). Курсор принимает при этом вид вертикальной линии красного цвета, которая называется *линией ввода текста* и аналогична по назначению линиям ввода в формулах.

Рис. 2.23. Вновь созданный текстовый регион

Примечание

Создать текстовый регион можно и эквивалентным способом, с помощью команды Insert / Text Region (Вставка / Текстовая область).

Теперь можно просто вводить любой текст в текстовый регион, причем очередной символ будет вставлен в позицию, обозначенную линией ввода текста.

2.3.2. Редактирование текста

Чтобы изменить какой-либо текст внутри документа:

1. Щелкните мышью на области текста — она приобретет характерный вид (рис. 2.24).
2. При необходимости переместите линию ввода текста внутри текстовой области к символам, которые собираетесь изменить, щелкая мышью в

нужном месте текста или нажимая клавиши со стрелками и клавиши <Home> и <End>.

3. Отредактируйте текст.

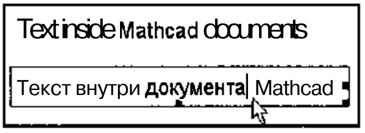


Рис. 2.24. Текстовые области (нижняя — в процессе редактирования)

Для редактирования текста применяются те же средства, что и для редактирования формул:

- выделение части текста протаскиванием указателя мыши или нажатием клавиш-стрелок при удерживаемой клавише <Shift>;
- вырезка, копирование и вставка части текста либо сочетанием горячих клавиш <Ctrl>+<X>, <Ctrl>+<C>, <Ctrl>+<V>, соответственно, либо при помощи меню **Edit**, контекстного меню или панели **Standard**.

Примечание

Кроме того, существуют развитые средства форматирования текста, такие, как управление типом и размером шрифта, выравниванием и т. п. Большинство этих возможностей *(см. гл. 17)* реализуется при помощи панели инструментов Formatting (Форматирование).

2.3.3. Импорт текста

Mathcad позволяет импортировать фрагменты текста из других приложений (например, Notepad или Microsoft Word). Сделать это проще всего через буфер обмена:

1. Находясь в другом приложении, скопируйте нужный фрагмент в буфер обмена.
2. Перейдите в окно Mathcad и отметьте (щелчком мыши в желаемом месте) курсором ввода место вставки фрагмента.
3. Выберите один из двух путей:
   - создайте в документе Mathcad текстовый регион, нажав клавишу <">, и, находясь внутри него, вставьте содержимое буфера обмена сочетанием <Ctrl>+<V>. Фрагмент будет вставлен в документ в виде обычной текстовой области (рис. 2.25), которую затем можно редактировать стандартными для Mathcad средствами;
   - не создавая текстовой области, вставьте фрагмент из буфера нажатием <Ctrl>+<V>. Текст будет вставлен в виде объекта OLE (Object Linking and Embedding), т. е. для его редактирования каждый раз будет вызываться то приложение, в котором он был создан (рис. 2.26).

Рис. 2.25. Импорт фрагмента текста внутрь текстовой области

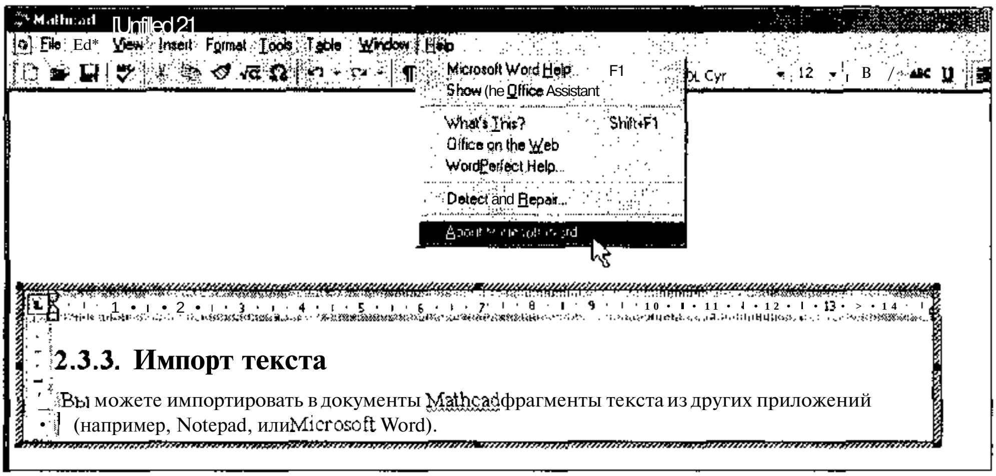


Рис. 2.26. Импорт текста из Microsoft Word в качестве объекта OLE

Применяйте второй способ в том случае, когда специальное форматирование текста невозможно осуществить средствами Mathcad. Чтобы вернуться к редактированию вставленного таким образом текста, следует дважды щелкнуть на нем мышью. При этом вместо меню и панелей инструментов в Mathcad появятся соответствующие средства другого приложения (чтобы подчеркнуть это, на рис. 2.26 открыто меню **Help,** свидетельствующее о работе в данный момент приложения Microsoft Word). С их помощью можно отредактировать объект и затем вернуться к обычному редактированию документа Mathcad, щелкнув в нем мышью за пределами объекта.

2.3.4. Математические символы внутри текста

Для качественного оформления документов, скорее всего, потребуются текстовые области, содержащие математические выражения. Для создания таких областей:

1. Щелкните в нужной части текстовой области.
2. Выберите команду **Insert/ Math Region** (Вставка /Математическая область) или нажмите клавиши <Ctrl> + <Shift>+<A>, чтобы создать пустой местозаполнитель внутри текста (рис. 2.27).
3. Введите математическое выражение в местозаполнитель так, как вводите обычные формулы *(см. разд. 2.2).*

Рис. 2.27. Вставка математических символов в текстовую область

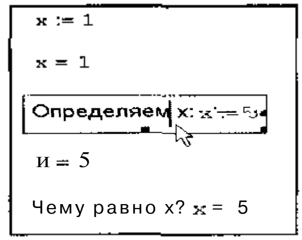


Рис. 2.28. Математические области в тексте влияют на вычисления

Помещая формулы в текст, помните о том, что они влияют на вычисления точно так же, как если бы были помещены в математический регион непосредственно в документе. На рис. 2.28 видны (сверху вниз): два математических региона, потом текстовый (который находится в процессе редактирования), в котором переменной x присвоено новое значение, и затем еще один математический и один текстовый регион, в котором выведено это значение x. Обратите внимание, что после переопределения внутри первого текста переменная x поменяла свое значение.

Примечание

Если необходимо, чтобы математическая область внутри текста не влияла на вычисления, отключите их. Для этого, находясь в режиме редактирования формулы, выполните команду Format / Properties (Формат / Свойства) и, перейдя в открывшемся диалоговом окне Properties (Свойства) на вкладку Calculations (Вычисления), установите флажок Disable Evaluations (Выключить вычисления) и нажмите кнопку OK.

2.3.5. Гиперссылки

Иногда необходимо сделать текстовую область одновременно и гиперссылкой, переводящей курсор на какое-либо иное место в активном документе, другой документ Mathcad либо на сайт в Интернете. Для вставки гиперссылки используется команда **Insert/ Hyperlink** (Вставка/ Гиперссылка). *(см. гл. 17)*.

2.4. Правка документа

В предыдущих разделах было разобрано, как осуществляется правка отдельных текстовых и математических областей. Наряду с этим, к частям документа, пустым или содержащим несколько регионов, применяются и стандартные методы редактирования. Перечислим кратко характерные приемы правки документов, учитывая, что смысл их стандартен для Windows-ттриложений.

Выделение части документа

- Чтобы выделить несколько регионов, расположенных последовательно друг за другом, нажмите вне крайнего из них левую кнопку мыши (определяя тем самым место курсора ввода) и протащите се указатель через все регионы, которые надо выделить. Выделенные регионы от курсора ввода до указателя мыши будут отмечены пунктиром (рис. 2.29).

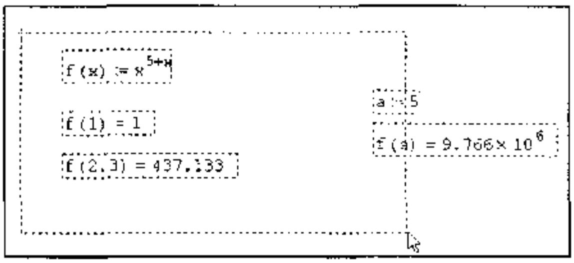


Рис. 2.29. Выделение области в документе

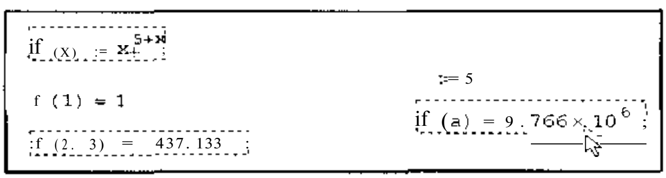


Рис. 2.30. Выделение нескольких разрозненных регионов

- Также можно выделить несколько соседних регионов, щелкая на крайнем из них, нажав клавишу <Shift> и, не отпуская ее, выполнить щелчок на другом крайнем регионе.
- Несколько разрозненных регионов можно выделить (рис. 2.30), щелкая на первом из них, нажав клавишу <Ctrl> и, не отпуская ее, последовательно щелкая на остальных регионах.
- Все содержание документа можно выделить при помощи команды **Edit / Select All** (Правка / Выделить все) или нажатием клавиш <Ctrl>+<A>.
- Для снятия выделения щелкните мышью в любой части документа.

Удаление части документа

- Выделенные регионы удаляются нажатием клавиши <Del> или <Ctrl>+<D>.
- Весь текущий регион удаляется нажатием клавиш <Ctrl>+<D> или командой **Edit / Delete** (Правка / Удалить).

3 Зак. 984

- Пустые строки в документе можно удалить, помещая щелчком мыши в их верхнюю часть курсор ввода и нажимая нужное число раз клавишу <Del>.
- Для вставки пустых строк ниже курсора ввода нажмите нужное число раз клавишу <Enter>.

Вырезка, копирование, вставка и перемещение части документа

- Для вырезки, копирования выделенных регионов в буфер обмена и для вставки их из буфера в документ используйте одно из стандартных средств:
  - верхнее меню **Edit** (Правка);
  - контекстное меню;
  - кнопки правки на панели инструментов **Standard** (Стандартная);
  - сочетание горячих клавиш <Ctrl>+<X>, <Ctrl>+<C>, <Ctrl>+<V>.
- Для перемещения и копирования выделенных регионов документа удобнее использовать технологию перетаскивания их мышью:
  - для перемещения поместите указатель мыши на один из выделенных регионов — он приобретет форму ладони. Затем нажмите левую кнопку мыши и перетащите при нажатой кнопке выделение в желаемое место (рис. 2.31). При отпускании кнопки мыши выделенные регионы переместятся на новое место;
  - для копирования выделенных регионов перетаскивайте их мышью при нажатой клавише <Ctrl>.

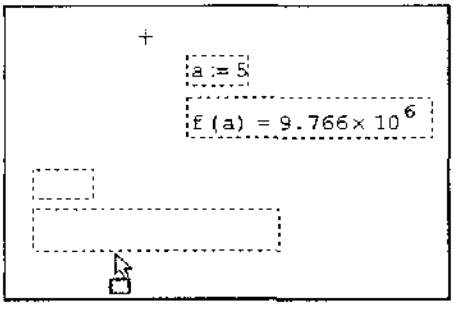


Рис. 2.31. Перетаскивание части документа в другое место

Выравнивание регионов

Чтобы документы воспринимались лучше, в Mathcad предусмотрены опции выравнивания регионов (и математических, и текстовых) по левому краю вдоль вертикальной линии и по верхнему краю вдоль горизонтальной. Для выравнивания выделите сначала несколько регионов и нажмите одну из двух кнопок выравнивания (рис. 2.32), или воспользуйтесь командой Format / Align Regions (Формат/ Выравнивание регионов) и выберите в открывающемся подменю (рис. 2.33) либо Across (Горизонтально), либо Down (Вертикально).

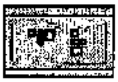


Рис. 2.32. Кнопки выравнивания регионов

Результат выравнивания показан в качестве примера на рис. 2.34. Для того чтобы расположить регионы в геометрически правильном порядке, возможно, потребуется применить различное выравнивание несколько раз.

Примечание

При попытке выровнять регионы может возникнуть ситуация, когда они станут перекрываться. В этом случае Mathcad задаст вопрос в диалоговом окне **"Selected regions may overlap. Align selected regions?"** (Выбранные регионы могут перекрываться. Выровнять их?). Если нажать в этом диалоге кнопку **Cancel** (Отмена), то операция выравнивания будет отменена.

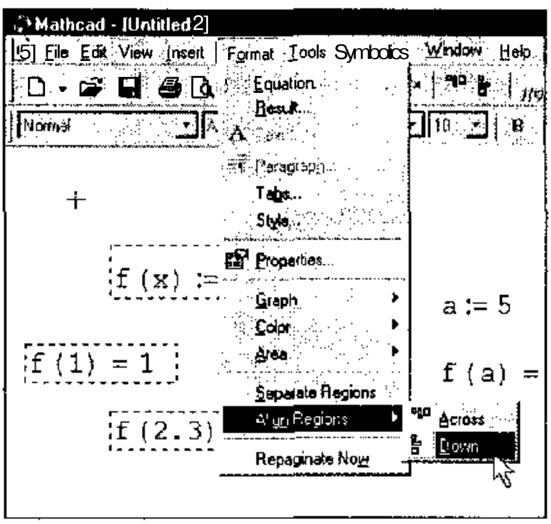


Рис. 2.33. Выравнивание регионов при помощи меню

$f(x) := x^{5+x}$	$a := 5$
$f(1) = 1$	$a = 5$
$f(2.3) = 437.133$	$f(a) = 9.766 \times 10^{6}$

Рис. 2.34. Результат вертикального и горизонтального выравнивания регионов

Обновление вида документа

Редактор Mathcad — довольно сложная программа, и в результате работы в нем на поверхности документа может время от времени появляться "мусор" — лишние символы, которых на самом деле в документе нет. Если вы подозреваете, что имеете дело именно с такой ситуацией, выполните команду **View/ Refresh** (Вид/ Обновить) или нажмите клавиши <Ctrl>+<R>. В результате все лишние символы должны исчезнуть.

Поиск и замена

Находясь в Mathcad, несложно организовать поиск символа, фрагмента или слова в документе (рис. 2.35):

1. Выполните команду **Edit/ Find** (Правка/ Найти) или нажмите клавиши <Ctrl>+<F> для вызова диалога **Find** (Поиск).
2. Введите в поле **Find what** (Найти) в верхней части диалога искомый текст.
3. Укажите, если это необходимо, опции поиска, устанавливая или снимая флажки:
   - **Match whole word only** (Искать совпадение только слов целиком);
   - **Match case** (Учитывать регистр);
   - **Find in Text Regions** (Искать в текстовых областях);
   - **Find in Math Regions** (Искать в математических областях).
4. При необходимости задайте направление поиска переключателем Up (Вверх) или **Down** (Вниз).
5. Нажмите кнопку **Find Next** (Найти) для поиска места, где указанный символ встречается в следующий раз.
6. Чтобы выйти из диалога, нажмите кнопку **Cancel. Вы** переместитесь в найденное место документа.

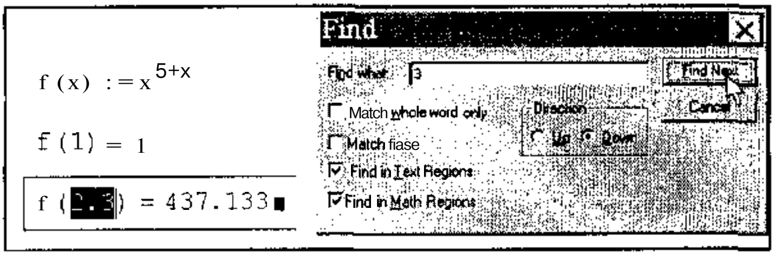


Рис. 2.35. Результат поиска символа в документе

Похожим образом можно автоматически заменить одни символы в документе другими (рис. 2.36):

1. Выберите меню **Edit/ Replace** (Правка/ Заменить) или нажмите клавиши <Ctrl>+<H> для вызова диалога **Find** (Поиск).
2. Введите текст, подлежащий замене, в поле **Find what** (Найти) в открывшемся диалоге **Replace** (Заменить).
3. Введите текст для замены в поле **Replace with** (Заменить).
4. Укажите, если это необходимо, рассмотренные выше опции поиска.
5. Нажмите одну из кнопок:
   - **Find Next** (Найти) — для поиска следующего вхождения указанного символа;
   - **Replace** (Заменить) — для замены следующего найденного символа другим;
   - **Replace All** (Заменить все) — для замены всех символов в документе, удовлетворяющих критериям поиска;
   - **Cancel** — для выхода из диалога **Replace.**

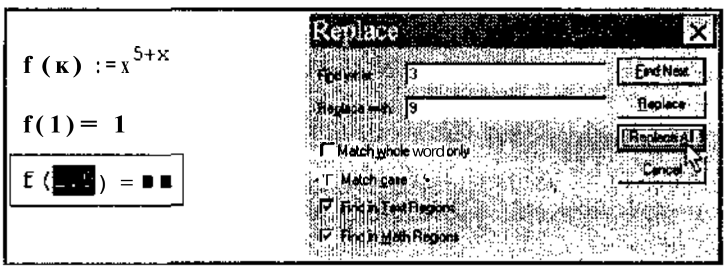


Рис. 2.36. Результат замены символов в документе

Проверка орфографии

Для проверки англоязычной орфографии выделите текстовые регионы, подлежащие проверке, и выполните команду **Edit / Check Spelling** (Правка / Проверка орфографии), либо нажмите кнопку с галочкой на стандартной панели инструментов. Если вы хотите проверить орфографию во всем документе, не выделяйте ни один текстовый регион, а поместите курсор ввода в точку, с которой требуется начать проверку.

Примечание

Орфография проверяется только внутри текстовых регионов.

Если в процессе проверки Mathcad обнаружит слово, отсутствующее в его словаре, оно будет выделено в документе, а пользователь увидит диалоговое окно **Check Spelling** (Проверка орфографии), показанное на рис. 2.37.

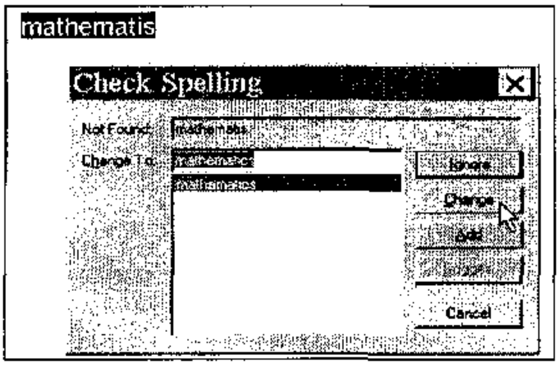


Рис. 2.37. Диалоговое окно Check **Spelling**

В диалоговом окне **Check Spelling** находятся следующие элементы:

- **Not Found** (Нет в словаре) — указание на то, что слово отсутствует в словаре. Проверить написание слова придется самостоятельно и затем ввести правильный вариант в поле ввода;
- **Change To** (Заменить на) — предложение наиболее близких слов из словаря для исправления. Выберите правильную замену из предложенного списка;
- **Change** (Заменить) — нажмите эту кнопку, чтобы заменить слово в документе на исправленное;
- **Ignore** (Пропустить) — оставить слово в документе неизменным;
- **Add** (Добавить) — оставить слово в документе и, кроме того, добавить его в словарь Mathcad, чтобы впоследствии он интерпретировал его как правильное;
- **Cancel** (Отмена) — оставить все как есть и выйти из диалогового окна, закончив проверку орфографии.

2.5. Печать документа

Чтобы распечатать экземпляр активного документа на принтере, нажмите клавиши <Ctrl>+<P> или кнопку с изображением принтера на стандартной панели инструментов.

Для более активного управления процессом печати служат следующие пункты меню **File** (Файл):

- **Page Setup** (Параметры страницы) — опции страницы вывода активного документа на печать (стандартный размер страницы, тип подачи бумаги, поля);
- **Print Preview** (Просмотр) — предварительный просмотр на экране вывода на печать активного документа;

- **Print** (Печать) — печать активного документа с возможностью выбора принтера (если установлено несколько принтеров), смены установок принтера (таких, как качество печати, разрешение, количество печатных копий документа и диапазон печатаемых страниц).

Отметим, что нажатие кнопки печати на панели инструментов приводит к мгновенной распечатке всего активного документа с текущими опциями печати и установками принтера.

2.6. Посылка документа по электронной почте

Послать документ по электронной почте можно с помощью почтового приложения (например, Microsoft Outlook), присоединяя файл с документом Mathcad к письму обычным образом. Но, кроме того, отправить активный документ по электронной почте легко, и не выходя из Mathcad. Для этого выберите команду **File/ Send** (Файл/ Отправить), в результате чего сразу появится окно **New Message** (Новое сообщение), изображенное на рис. 2.38 с автоматически присоединенным к нему файлом Mathcad. Пользователю остается лишь ввести в соответствующие поля окна электронный адрес получателя, тему и текст письма (последние два пункта необязательны) и отправить письмо.

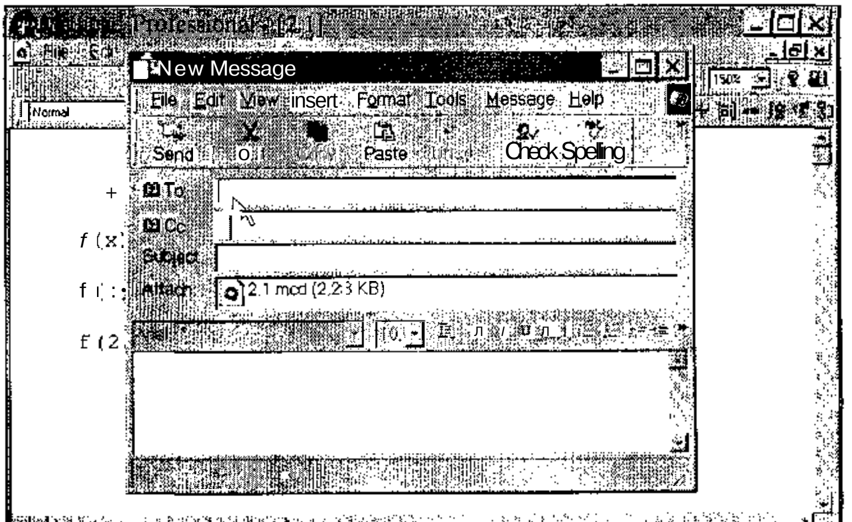


Рис. 2.38. Создание сообщения для отправки документа по электронной почте

Примечание

Для использования этой опции компьютер должен быть подключен к Интернету и на нем должно быть предварительно установлено соответствующее почтовое приложение.

ГЛАВА 3

Вычисления

Эта глава посвящена основам вычислений в Mathcad. Она содержит все необходимые сведения о применении переменных и функций, операторов присваивания, численного вывода и символьного вывода *(см. разд. 3.1)*, а также других операторов *(см. разд. 3.2)*. В заключение описываются основные средства управления процессом вычислений в Mathcad *(см. разд. 3.3)* и говорится несколько слов о том, каким образом происходит выдача сообщений об ошибках при вычислениях *(см. разд. 3.4)*.

3.1. Переменные и функции

Основные инструменты математика — это операции с переменными величинами и функциями. В Mathcad переменные, операторы и функции реализованы в интуитивной форме, т. е. выражения в редакторе вводятся и вычисляются так, как они были бы написаны на листе бумаги. Порядок вычислений в документе Mathcad также очевиден: математические выражения и действия воспринимаются процессором слева направо и сверху вниз.

Перечислим основные действия, которые пользователь может совершать для определения и вывода переменных и функций.

3.1.1. Определение переменных

Чтобы определить переменную, достаточно ввести ее имя и присвоить ей некоторое значение, для чего служит оператор присваивания *(см. следующий раздел)*.

3.1.2. Присваивание переменным значений

Чтобы присвоить переменной новое значение, например переменную x сделать равной 10:

1. Введите в желаемом месте документа имя переменной, например x.
2. Введите оператор присваивания с помощью клавиши <:> или нажатием соответствующей кнопки **Definition** (Присваивание) на панели инструментов **Calculator** или **Evaluation** (Выражения), как показано на рис. 3.1.

3. Введите в появившийся местозаполнитель новое значение переменной (ю).

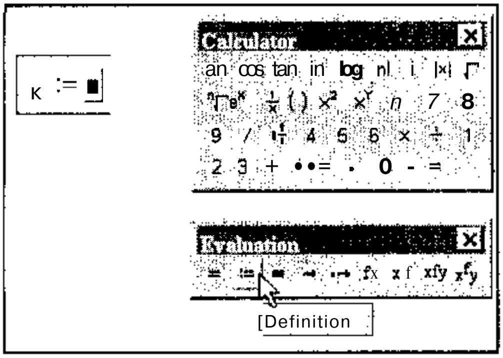


Рис. 3.1. Результат ввода оператора присваивания

Результат перечисленных действий показан в листинге 3.1.

Примечание

Кнопка оператора присваивания для удобства помещена сразу на две панели Calculator (Калькулятор) и **Evaluation** (Выражения).

Листинг 3.1. Присваивание переменной численного значения

x:= 10

Ввести новое значение переменной возможно как в виде числа, так и в виде математического выражения, содержащего другие переменные (листинг 3.2) и функции *(см. следующие разделы),* а также в виде строкового выражения (листинг 3.3.). В последнем случае будет создана переменная s не численного, а строкового типа.

Листинг 3.2. Присваивание переменной вычисленного значения выражения

$x := 10$

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

Листинг 3.3. Присваивание переменной строкового значения

s := "Hello,"

Если переменная с некоторым именем создается в данном документе впервые, то для ввода оператора присваивания, вместо двоеточия, допускается использовать символ равенства "=", который Mathcad автоматически заменит символом присваивания.

Примечание

В некоторых случаях это невозможно, в частности, когда значение присваивается переменной, имя которой зарезервировано Mathcad. Например, присвоить значение переменной с именем N можно, лишь вводя двоеточие, т. к. по умолчанию это имя обозначает в Mathcad размерность силы (Ньютон).

Чтобы переопределить значение переменной, определенной в документе, оператор присваивания следует вводить не знаком равенства, а двоеточием, либо пользоваться панелью инструментов.

Не вполне соответствующий общепринятому математическому стилю вид оператора присваивания (не =, а :=) является, на самом деле, компромиссом, связанным с назначением Mathcad как системы программирования. Этот оператор показывает, что он действует, в отличие от других, не слева направо, а справа налево, поскольку значение (справа) задается переменной (слева). И если непосвященного математика внешний вид этого оператора может ввести в некоторое заблуждение, то пользователю Mathcad он прямо говорит о действии, выполняемом в данном месте документа: значение переменной *не выводится* на экран (о чем говорит знак =), а некоторое значение *присваивается* (:=) данной переменной.

Для подготовки отчетов, тем не менее, может потребоваться изменить отображение оператора присваивания с принятых по умолчанию символов ":=" на символ равенства. Это делается для конкретного оператора присваивания с помощью пункта View Definition As контекстного меню (рис. 3.2) либо для всего документа с помощью команды Tools / Worksheet Options / Display (Сервис / Опции документа / Отображение) *(см. разд. "Управление отображением некоторых операторов" гл. 2).*

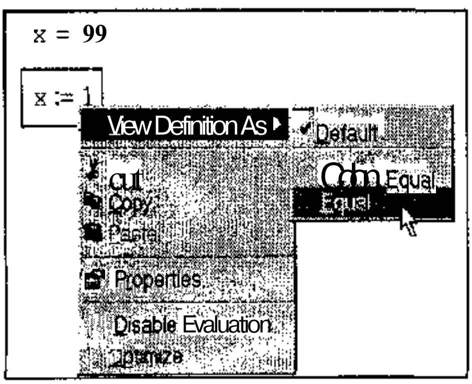


Рис. 3.2. Различное отображение оператора присваивания

Помимо разобранного оператора присваивания (а он применяется наиболее часто), существует также возможность *глобального присваивания.*

Глобальное присваивание будет рассмотрено в разд. 3.2.2.

3.1.3. Функции

Функции в Mathcad записываются в обычной для математика форме:

- f(x,...) — функция;
  - f — имя функции;
  - x, ... — список переменных.

Легче всего ввести написание функции в документ при помощи клавиатуры.

В Mathcad формально можно разделить функции на два типа:

- встроенные функции;
- функции, определенные пользователем.

Применение функций обоих типов в расчетах совершенно одинаково, с тем исключением, что любую встроенную функцию можно сразу использовать в любом месте документа *(о вставке встроенных функций в документ читайте в разд. "Знакомство с Mathcad" гл. 1),* а пользовательскую функцию необходимо предварительно определить в документе до момента вычисления ее значения.

3.1.4. Определение функции пользователя

Для того чтобы определить функцию пользователя, например $f(x,y)=x^2 \cdot \cos(x+y)$:

1. Введите в желаемом месте документа имя функции (f).
2. Введите левую скобку "(", имена переменных через запятую x, y и правую скобку ")". При вводе левой скобки и запятой автоматически будут появляться соответствующие местозаполнители.
3. Введите оператор присваивания с панели инструментов или нажатием клавиши <:>.
4. Введите в появившийся местозаполнитель выражение, определяющее функцию $x^2 \cdot \cos(x+y)$, пользуясь клавиатурой или панелями инструментов.

Результат ввода иллюстрируется листингом 3.4.

Листинг 3.4. Определение функции пользователя

$$f(x,y) := x^2 \cdot \cos(x+y)$$

Внимание!

Все переменные, присутствующие справа в выражении определения функции, либо должны входить в список аргументов функции (в скобках, слева после имени функции), либо должны быть определены ранее. В противном случае будет выведено сообщение об ошибке, причем имя неопределенной переменной будет выделено красным цветом (рис. 3.3).

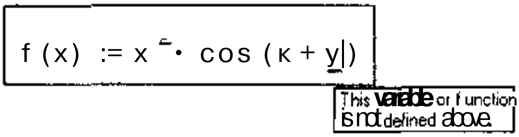


Рис. 3.3. Сообщение об ошибке ("Эта переменная или функция ранее не определена")

3.1.5. Вывод значений переменных и функций

Чтобы вычислить в документе некоторое математическое выражение, которое может состоять из переменных, операторов и функций (встроенных и определенных пользователем):

1. Введите это выражение, например x^y.
2. Нажмите клавишу <=>.

В результате справа от введенного знака равенства появится вычисленное значение выражения (листинг 3.5, предпоследняя строка). Нельзя изменять содержимое выражения справа от знака равенства, поскольку оно есть результат работы *вычислительного процессора* Mathcad, совершенно скрытой от глаз пользователя. Подчас (когда выражение содержит функции, реализующие разные численные методы, часто в сложных комбинациях) алгоритмы расчета бывают очень затейливыми и занимают существенное время. О том, что некоторое выражение документа находится в стадии вычисления, свидетельствует обрамляющая его зеленая рамка и невозможность предпринять какое-либо действие с программой Mathcad.

Листинг 3.5. Вычисление выражения

$x := 10$

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

$x^y = 1 \times 10^{50}$

$x = 10$

Заметьте, что, перед тем как вычислить значение математического выражения, Вы обязаны определить значение каждой входящей в него переменной (две первых строки листинга 3.5). Вычисляемое выражение может содержать любое количество переменных, операторов и функций. Вывод текущего значения той или иной переменной приведен в последней строке листинга 3.5, а значения функции — в листингах 3.6 и 3.7.

Листинг 3.6. Вывод значения функции

$f(x, y) := x^2 \cdot \cos(x + y)$

$f(2, 5.99) = -0.542$

$f(1.3, 7) = -0.729$

Листинг 3.7. Вывод **значения** функции *(продолжение листинга 3.6)*

```
x:=1.3
y:=7
f(x, y) = -0.729
```

Внимание!

При определении функций пользователя через различные переменные важную роль играет присутствие имен этих переменных в списке аргументов или определение их выше в тексте документа. Например, результаты вывода значения функции f(x,y) в листинге 3.6 остались бы точно такими же, если до или после определения функции присвоить переменным x и y некоторые значения. Так происходит потому, что значения аргумента заданы непосредственно в строке вычисления функции. Если же определить функцию f fx) так, как это сделано в листинге 3.8, то она будет зависеть от значения переменной y в момент определения f(x) (т. е. y=5), поскольку y не входит в список аргументов f!x). Фактически £(x)=x^2·cos(x+5). Даже если где-нибудь ниже в программе пользователь переопределит значение y, Mathcad все равно будет помнить функцию f(x) как выражение x^2·cos(x+5) (листинг 3.9).

Листинг 3.8. К определению функций пользователя

```
y := 5
f(x) := x^2 · cos(x + y)
f(1) = 0.96
```

Листинг 3.9. К определению функций пользователя *(продолжение листинга 3.8)*

```
y := 0
x := 1
f fx) = 0.96
x^2 · cos(x + y) = 0.54
x^2 · cos(x + 5) = 0.96
```

Примечание

Внимательнее относитесь к обязательному требованию совпадения количества аргументов при определении и выводе значения функций. Сравните, например, листинги 3.6 и 3.8, в которых, несмотря на одинаковое выражение в правой части определения функции f, создаются две существенно разные функции f(x,y) и f(x), соответственно.

Вводя знак равенства для вычисления математических выражений в Mathcad, Вы фактически применяете оператор *вычисления* или *численного вывода* (numerical evaluation). Его можно ввести также нажатием кнопки со знаком равенства на одной из панелей инструментов: **Calculator** (Калькулятор) или **Evaluation** (Выражения) (см. рис. 3.1). Оператор численного вывода означает, что все вычисления проводятся с числами, а различные встроенные алгоритмы реализуются соответствующими численными методами.

3.1.6. **Символьный вывод**

Наряду с численным выводом, в Mathcad имеется возможность символьного, или аналитического, вычисления значения выражения. Для символьных вычислений имеется ряд специальных средств, которые будут детально рассмотрены позднее *(см. гл. 5),* самое простое из них — это оператор *символьного вывода* (symbolic evaluation). Он обозначается символом → и в большинстве случаев применяется точно так же, как оператор численного вывода, однако внутреннее различие между действием этих двух операторов огромно. Если численный вывод — это в обычном смысле этого слова "запрограммированный" расчет по формулам и численным методам, скрытый от глаза пользователя, то символьный вывод — результат работы системы искусственного интеллекта, встроенной в Mathcad и называемой *символьным процессором.* Работа символьного процессора также невидима (и, чаще всего, даже трудно представима) пользователю и заключается в анализе самого текста математических выражений. Конечно, гораздо более узкий круг формул можно рассчитать символьно, хотя бы потому, что, вообще говоря, относительно не такая большая часть математических задач допускает аналитическое решение.

Чтобы попытаться вычислить символьно математическое выражение, например `B · sin(arcsin(C • x))`, где `B,C,x` — некоторые переменные:

1. Введите это выражение: `в • sin(asin(c - x))`.
2. Введите оператор символьного вывода сочетанием клавиш <Ctrl>+<.>, либо нажатием соответствующей кнопки (рис. 3.4) на панели **Symbolic** (Символика) или **Evaluation** (Выражения).

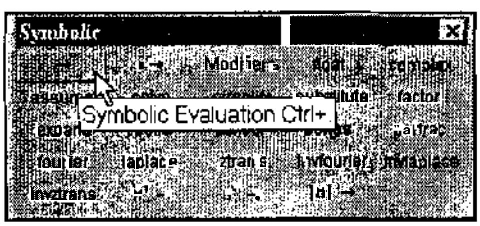


Рис. 3.4. Кнопка вставки оператора символьного вывода

После этого справа от символа оператора символьного вывода появится определенное аналитически значение выражения (листинг. 3.10) либо сообщение об ошибке **"No answer found"** (Ответ не найден). Если символьному процессору Mathcad не удается аналитически упростить выражение, то оно будет выдано справа от знака → в том же виде, что и слева.

Листинг 3.10. Символьный вывод выражения

$B \cdot \sin(\mathrm{asin}(C \cdot x)) \rightarrow B \cdot C \cdot x$

Листинг 3.11. Символьный вывод выражения, которое не удалось упростить

$x^2 \cdot \cos(x + y) \rightarrow x^2 \cdot \cos(x + y)$

Внимание!

Присмотритесь внимательнее к листингам 3.10 и 3.11: для символьного вывода не требуется предварительно определять переменные, входящие в левую часть выражения! Если же переменным были все-таки присвоены ранее некоторые значения, символьный процессор просто подставит их в упрощенную формулу и выдаст результат с учетом этих значений (см. в качестве примера два следующих листинга — 3.12 и 3.13).

Точно так же, как рассчитываются численно значения функций, можно вычислять их и с помощью символьного процессора. Сравните соответствующие результаты, которые представлены в листинге 3.12 (конечно, символьный и численный ответы равны: $9 \cdot \cos(8) = -1.31$). Аналогично можно символьно выводить значения переменных. Например, присвоить некоторой переменной значение функции или сложного выражения (листинг 3.13, вторая строка) и затем вывести значение переменной в символьном виде.

Листинг 3.12. Численный и символьный вывод значения функции

$f(x) := x^2 \cdot \cos(x + 5)$

$f(3) = -1.31$

$f(3) \rightarrow 9 \cdot \cos(8)$

Листинг 3.13. Численный и символьный вывод переменной

$f(x) := x^2 \cdot \cos(x + 5)$

$a := f(3)$

$a \rightarrow 9 \cdot \cos(8)$

$a = -1.31$

Как показывают приведенные примеры, преимущество символьных вычислений заключается в выдаче аналитического результата, который для математика часто является более ценным. Поэтому, исходя из специфики конкретных задач, решайте, стоит ли наряду с численными расчетами попытаться получить и символьное решение.

3.1.7. Допустимые имена переменныхифункций

В заключение перечислим, какие символы можно, а какие нельзя применять в именах, которые пользователь дает переменным и функциям, и перечислим ряд ограничений на присваивание имен. Допустимые символы:

- большие и маленькие буквы — Mathcad различает регистр: так, имена x и x определяют разные переменные. Кроме того, Mathcad различает и шрифт, например имена x и *x* воспринимаются как разные;
- числа от 0 до 9;
- символ бесконечности (клавиши <Ctrl>+<Shift>+<Z>);
- штрих (клавиши <Ctrl>+<F7>);
- греческие буквы — они вставляются с помощью панели Greek (Греческие символы);
- символ подчеркивания;
- символ процента;
- нижний индекс.

Внимание!

С осторожностью используйте нижний индекс в определении имен переменных и функций, не путая его с индексом векторной переменной. Чтобы ввести имя с нижним индексом, например, K_{max}: введите букву "K", затем точку ".", после чего линии ввода опустятся чуть ниже, и только затем сам нижний индекс max.

Теперь рассмотрим ограничения на имена переменных и функций:

- имя не может начинаться с цифры, символа подчеркивания, штриха или процента;
- символ бесконечности должен быть только первым в имени;
- все буквы в имени должны иметь один стиль и шрифт;
- имена не могут совпадать с именами встроенных функций, констант и размерностей, например sin или TOL. Тем не менее, допускается их переопределение, но тогда одноименная встроенная функция больше не будет использоваться по первоначальному назначению;

- Mathcad не различает имен переменных и функций: если сначала определить функцию `f(x)`, а потом переменную f, то в оставшейся части документа будет утерян доступ к функции f (x).

Имена, содержащие операторы и специальные символы

В некоторых случаях желательно использовать имена переменных и функций, содержащие символы операторов Mathcad или другие символы, которые нельзя вставлять в имена непосредственно. Для этого существуют две возможности.

Во-первых, имя, составленное из любых символов и заключенное в квадратные скобки, Mathcad будет воспринимать корректно (рис. 3.5, сверху). Например, чтобы ввести имя `[a+b]`:

1. Нажмите клавиши <Ctrl>+<Shift>+<J> — появится пара квадратных скобок с местозаполнителем внутри.
2. Введите в местозаполнитель последовательность любых символов, например `a+b`.

[a + b] := 7

$:= 5

a+b := 0

Рис. 3.5. Специальные символы в именах переменных

Во-вторых, если Вас не устраивает наличие квадратных скобок в имени, то вставить в него специальные символы можно чуть более сложным способом. Например, для ввода имени а+ь:

1. Введите первый символ (а), который должен быть допустимым для имен Mathcad.
2. Нажмите клавиши <Ctrl>+<Shift>+<K> для перехода в специальный "текстовый" режим редактирования,
3. Введите последовательность любых символов (+).
4. Еще раз нажмите клавиши <Ctrl>+<Shift>+<K>, чтобы вернуться в обычный режим редактирования. Теперь можно продолжать ввод допустимых символов в имя (b).

Результат этих действий показан в нижней строке рис. 3.5. Если требуется, чтобы имя начиналось со специального символа (средняя строка рис. 3.5), то необходимо выполнить все пункты 1—4, вводя в начале имени произвольный допустимый символ, а по завершении ввода просто стирая его.

3.2. Операторы

Каждый оператор в Mathcad обозначает некоторое математическое действие в виде символа. В полном согласии с терминологией, принятой в математике, ряд действий (например, сложение, деление, транспонирование матрицы и т. п.) реализован в Mathcad в виде встроенных операторов, а другие действия (например, sin, erf и т. п.) — в виде встроенных функций. Каждый оператор действует на одно или два числа (переменную или функцию), которые называют *операндами.* Если в момент вставки оператора одного или обоих операндов не хватает, то недостающие операнды будут отображены в виде местозаполнителей. Символ любого оператора в нужное место документа вводится одним из двух основных способов:

- нажатием соответствующей клавиши (или сочетания клавиш) на клавиатуре;
- нажатием указателем мыши соответствующей кнопки на одной из математических панелей инструментов.

Напомним, что большинство математических панелей содержат сгруппированные по смыслу математические операторы, а вызвать эти панели на экран можно нажатием соответствующей кнопки на панели **Math** (Математика).

Примечание

Везде в этом разделе будем рассматривать только второй способ вставки оператора. Те *же, кто предпочитает использовать клавиатуру, найдут перечень горячих клавиш в приложении 2.*

Выше мы рассмотрели особенности применения трех операторов: присваивания *(см. разд. 3.1.2),* численного *(см. разд. 3.1.5)* и символьного вывода *(см. разд. 3.1.6).* Разберем в данном разделе действие прочих операторов Mathcad и возможности определения операторов пользователя.

3.2.1. Арифметические операторы

Операторы, обозначающие основные арифметические действия, вводятся с панели **Calculator** (Калькулятор), показанной на рис. 3.6:

- сложение и вычитание: + — (листинг 3.14);
- умножение и деление: • / + (листинг 3.15);
- факториал: ! (листинг 3.16);
- модуль числа: |x| (листинг 3.16);
- квадратный корень: $\sqrt{\ }$ (листинг 3.17);
- корень n-й степени: $\sqrt[n]{\ }$ (листинг 3.17);

- возведение x в степень у: x^y (ЛИСТИНГ 3.17);
- изменение приоритета: скобки (листинг 3.18);
- численный вывод: = (все листинги).

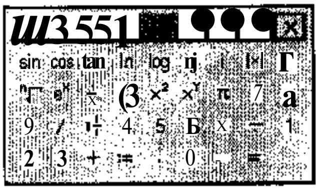


Рис. 3.6. Панель Calculator

Листинг 3.14. Операторы сложения, вычитания и отрицания

$1 + 3 - 7 = -3$

$-(-2) = 2$

Листинг 3.15. Операторы Деления и умножения

$\frac{5}{2} = 2.5$

$5 \div 2 = 2.5$

$2\frac{3}{4} = 2.75$

$1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$

Листинг 3.16. Операторы факториала и модуля

$5! = 120$

$|-10| = 10$

Листинг 3.17. Операторы извлечения корня и возведения в степень

$\sqrt{4} = 2$

$\sqrt[3]{125} = 5$

$e^{\ln(3)} = 3$

$3^2 = 9$

$10^{0.2} = 1.585$

Листинг 3.18. Оператор изменения приоритета ()

$(1 + 2) \cdot 3 = 9$

$1 + 2 \cdot 3 = 7$

Как видно, с помощью этой панели можно ввести не только перечисленные операторы, но и их часто используемые комбинации, например, возведение экспоненты в степень, смешанное произведение и деление, а также мнимую единицу и число π. Заметим, что допускается запись оператора деления как в одну, так и в две строки, что обеспечивается наличием двух соответствующих кнопок на панели **Calculator.**

Напомним, что в редакторе Mathcad можно выбирать отображение оператора умножения *(см. разд. "Управление отображением некоторых операторов" гл. 2)*. Для того чтобы поменять его:

1. Щелкните правой кнопкой мыши на выражении, содержащем оператор умножения.
2. Выберите первый пункт появившегося контекстного меню **View Multiplication** As (Представление умножения),
3. В подменю выберите пункт, соответствующий стилю представления умножения: в виде обычной точки **(Dot),** точки с уменьшенным расстоянием от него до сомножителей **(Narrow Dot),** толстой точки **(Large Dot),** крестика (X), без символа с небольшим расстоянием между сомножителями **(Thin Space),** вообще вместе **(No Space).** Чтобы просмотреть, как будет выглядеть выражение в двух последних представлениях, нужно снять с него выделение. Чтобы вернуть представление по умолчанию, выберите в подменю контекстного меню пункт **Default.**

Некоторых операторов, например таких, как оператор комплексного сопряжения, на панелях инструментов нет (листинг 3.19). Его приходится вводить исключительно с клавиатуры нажатием клавиши <"> в пределах математической области.

Листинг 3.19. Оператор комплексного сопряжения

$\overline{(3 + i)} = 3 - i$

3.2.2. Вычислительные операторы

Вычислительные операторы вставляются в документы при помощи панели инструментов **Calculus** (Вычисления). При нажатии любой из кнопок в документе появляется символ соответствующего математического действия, снабженный несколькими местозаполнителями. Количество и расположение местозаполнителей определяется типом оператора и в точности соответствует их общепринятой математической записи. Например, при вставке оператора суммы (рис. 3.7) необходимо задать четыре величины: переменную, по которой надо произвести суммирование, нижний и верхний пределы, а также само выражение, которое будет стоять под знаком суммы (пример заполненного оператора суммы см. ниже в листинге 3.22).

Для того чтобы вычислить неопределенный интеграл, следует заполнить два местозаполнителя: подынтегрального выражения и переменной интегрирования.

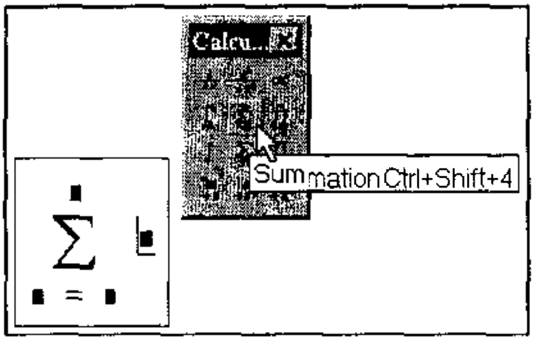


Рис. 3.7. Вставка оператора суммирования

После ввода какого-либо вычислительного оператора имеется возможность вычислить его значение либо численно, нажатием клавиши <=>, либо символьно, с помощью оператора символьного вывода.

Перечислим основные вычислительные операторы и приведем простейшие примеры их применения:

- дифференцирование и интегрирование;
  - производная (листинг 3.20);
  - N-Я производная (листинг 3.20);
  - определенный интеграл (листинг 3.21);
  - неопределенный интеграл (листинг 3.21).
- суммирование и вычисление произведения;
  - сумма (листинг 3.22);
  - произведение (листинг 3.22);
  - сумма ранжированной переменной (листинг 3.23);
  - произведение ранжированной переменной (листинг 3.23).
- пределы (листинг 3.24);
  - двусторонний;
  - левый;
  - правый.

Листинг 3.20. Операторы вычисления производных

$$\frac{d}{dx}\sin(x) \rightarrow \cos(x)$$

$$\frac{d^2}{dx^2}\sin(x) \rightarrow -\sin(x)$$

Листинг 3.21. Операторы интегрирования

$$\int_a^{\infty} \frac{1}{x^3}\,dx \rightarrow \frac{1}{2 \cdot a^2}$$

$$\int \ln(x)\,dx \rightarrow x \cdot \ln(x) - x$$

Листинг 3.22. Операторы суммирования и вычисления произведения

$$\sum_{i=1}^{10} \qquad \sum_{i=1}^{10}$$

$$\prod_{i=1}^{30} i = 2.653 \times 10^{32}$$

Листинг 3.23. Операторы суммирования и произведения ранжированной переменной

i := 1 - 5

$$\sum_i i^2 \cdot i! = 3.447 \times 10^3$$

$$\prod_i e^{i} = 3.269 \times 10^6$$

Примечание

О назначении и особенностях использования ранжированных переменных будет рассказано в следующей главе *(см. разд. "Ранжированные переменные" гл. 4).*

Листинг 3.24. Операторы символьного вычисления пределов

$$\lim_{x \to \infty} \frac{1 + 3 \cdot x}{x} \to 3$$

$$\lim_{x \to 0^+} \frac{1}{x} \to \infty$$

$$\lim_{x \to 0^-} \frac{1}{x} \to -\infty$$

Примечание

В отличие от других, операторы поиска предела могут быть вычислены только символьно *(см. гл. 5)*.

Операторы суммирования и вычисления произведения фактически являются более удобной записью операторов + и x с большим количеством операндов. А вот вычислительные операторы поиска производных и интегралов существенно отличаются от операторов умножения и сложения тем, что реализованы на основе определенных численных методов, которые в скрытой (невидимой для пользователя) форме запускаются вычислительным процессором Mathcad. При численном расчете интегралов и производных необходимо, хотя бы в общих чертах, представлять принцип работы соответствующих алгоритмов, чтобы избежать ошибок и неожиданностей при получении результатов *(численным методам интегрирования и дифференцирования посвящена гл. 7)*.

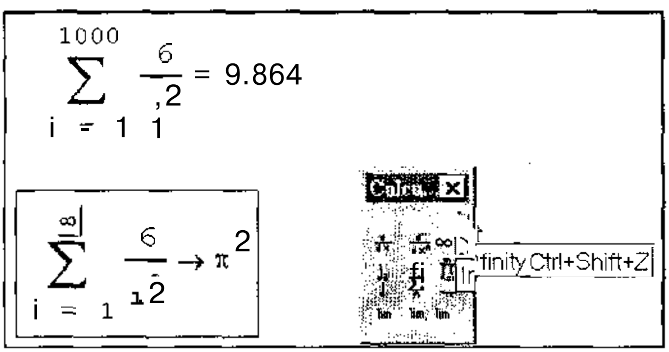


Рис. 3.8. Поиск бесконечного ряда

Важно отметить, что имеется возможность вычислять интегралы с одним или обоими бесконечными пределами, а также в символьной форме искать значения бесконечных пределов, сумм (рядов) и произведений. Для удобства ввода кнопка с символом бесконечности помешена на ту же панель инструментов **Calculus** (Вычисления). Пример вставки символа бесконечности в задаче поиска бесконечного ряда приведен на рис. 3.8.

3.2.3. Логические операторы

Результатом действия логических, или булевых, операторов являются только числа 0 (если логическое выражение, записанное с их помощью, истинно) или 1 (если логическое выражение ложно). Чтобы вычислить значение логического выражения, например 1=1 (рис. 3.9):

1. Вставьте с панели **Boolean** (Булевы операторы) соответствующий оператор =.
2. В появившиеся местозаполнители вставьте операнды (две единицы).
3. Нажмите клавишу <=>, чтобы получить ответ.

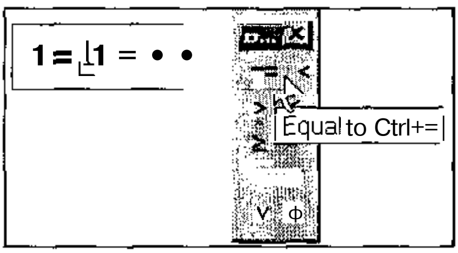


Рис. 3.9. Вставка логического оператора

Получается абсурдное на первый взгляд выражение 1=1=1. Однако на самом деле все правильно. Справа от оператора вывода записано логическое выражение 1=1 (обратите внимание, что логический знак равенства выглядит по-другому, нежели обычный), которое является истинным. Поэтому значение данного выражения равно 1, что и показано справа от знака равенства.

Перечислим логические операторы:

- больше **(Greater Than)** x>y;
- меньше **(Less Than)** x<y;
- больше или равно **(Greater Than or Equal)** x≥y;
- меньше или равно **(Less Than or Equal)** x≤y;
- равно **(Equal)** x=y;
- не равно **(Not Equal to)** x≠y;
- и **(And)** хлу;
- или **(Or)** xvy;
- исключающее или **(Exclusive or)** x©y;
- отрицание **(Not)** ¬x.

Примечание

Операнды в логических выражениях могут быть любыми числами. Однако если оператор по смыслу применим только к о и 1, то любое неравное нулю число по умолчанию принимается равным 1. Но в результате все равно может получиться либо 0, либо 1. Например, ¬(-0.33)=о.

Примеры действия логических операторов приведены в листингах 3.25 и 3.26.

Листинг 3.2S. Операторы сравнения

2 = 3 = 0	5 > 1 = 1	3 > 3 = 0
7 = 7 = 1	3 < ∞ = 1	3 ≥ 3 = 1
0 ≠ 0 = 0		

Листинг 3.26. Булевы операторы

1 v 0 = 1	1 л 0 = 0	1 ⊕ 0 = 1	¬1 = 0
0 v 0 = 0	0 л 0 = 0	0 ⊕ 0 = 0	¬0 = 1
1 v 1 = 1	1 Л 1 = 1	1 Φ 1 = 0	

Логические операторы чрезвычайно важны при записи подлежащих решению алгебраических уравнений и неравенств в приемлемой для Mathcad форме.

3.2.4. Матричные операторы

Матричные операторы предназначены для совершения различных действий над векторами и матрицами. Поскольку большинство из них реализует численные алгоритмы, о них будет подробно рассказано в части III *(см. гл. 9)*.

3.2.5. Операторы выражения

Почти все вычислительные операторы были рассмотрены выше *(см. разд. 3.1)*. Они сгруппированы на панели **Evaluation** (Выражения).

- Оценить численно **(Evaluate Numerically)** = *(см. разд. 3.1.5)*
- Вычислить символьно **(Evaluate Symbolically)** → *(см. разд. 3.1.6)*
- Присваивание **(Definition)** := *(см. разд. 3.1.2)*
- Глобальное присваивание **(Global Definition)** ≡

Рассмотрим различие между операторами обычного присваивания и *глобального присваивания* (процесс его вставки в документ показан на рис. 3.10). Для того чтобы вычислить выражение, содержащее некоторую переменную или функцию, необходимо, чтобы этой переменной ранее в документе было присвоено какое-либо значение. Иначе будет выдаваться сообщение об ошибке (рис. 3.11). Однако если в любой части документа (например в самом низу) вставить оператор глобального присваивания, то переменная будет определена в любой части документа (листинг 3.27).

Листинг 3.27. Действие операторов присваивания и глобального присваивания

x = 5

x:=10

x = 10

x ≡ 5

x = 5

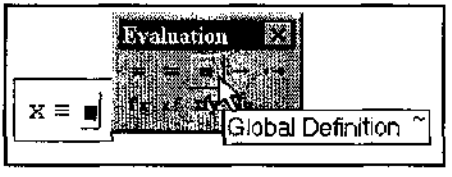


Рис. 3.10. Кнопка глобального присваивания на панели Evaluation

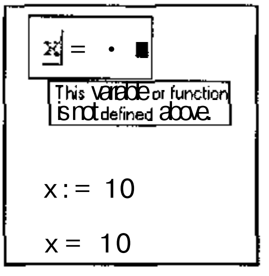


Рис. 3.11. Обычное присваивание сказывается только на нижеследующей части документа

Как видно из листинга 3.27, обычное, или *локальное,* присваивание переменной x действует от момента x:=ю до момента глобального присваивания x = 5. Вообще говоря, Mathcad анализирует документы на предмет присваивания переменных в два прохода: сначала распознаются все операторы глобального присваивания, и все выражения в документе сверху вниз и слева направо вычисляются в соответствии с ними, а при втором проходе в том же порядке анализируются операторы локального присваивания, и все выражения вычисляются с поправкой на них. Приведем важный пример взаимодействия глобального и локального присваивания (листинг 3.28).

Листинг 3.28. Взаимодействие глобального и локального присваивания

x ≡ 5

x = 5

x:=10

x= 10

y ≡ x^2

y = 25

Обратите внимание, что, несмотря на локальное присваивание переменной x:=ю в третьей строке листинга, значение переменной y вычисляется все-таки в соответствии с глобальным значением x ≡ 5, поскольку сама переменная y глобальным образом определена через переменную x.

Совет

Аккуратнее относитесь к определению глобальных переменных и, во избежание путаницы, старайтесь не переопределять их локально. Применяйте глобальное присваивание только для определения констант и, по возможности, избегайте случаев, когда оператор вывода предшествует оператору глобального присваивания для улучшения читаемости документов.

Точно так же как Вы глобально присваиваете значение переменной, допускается глобально определять функции (листинг 3.29).

Листинг 3.29. Глобальное определение функции пользователя

```
f(2) = 128

f(x) ≡ x^7
```

Примечание

Оператор глобального присваивания можно отображать не только в виде тождественного равенства, но и как обычный знак равенства. Для этого вызовите на операторе контекстное меню и в подменю пункта **View Definition As** выберите пункт **Equal** (Равенство).

3.2.6. Создание оператора пользователя

Запросы взыскательного пользователя могут отнюдь не исчерпываться набором встроенных операторов Mathcad. Для вставки в документы заранее созданных операторов пользователя применяется панель **Evaluation** (Выражения).

Выбор имени оператора

Оператор пользователя может иметь абсолютно любое имя *(см. ранее раздел "Имена, содержащие операторы и специальные символы" этой главы).* Однако, исходя из смысла операторов, логично давать им имена в виде символов. Это удобно делать с помощью коллекции символов, находящейся в справочной информации Mathcad. Выберите в верхнем меню **Help / QuickSheets** (Справка / Быстрые шпаргалки) и войдите затем в самый последний раздел **Extra Math Symbols** (Дополнительные символы) открывшегося содержания Шпаргалок. Там Вы увидите целую коллекцию символов, любой из которых можно просто перетащить указателем мыши в нужное место документа.

Присваивать оператору некоторое действие следует точно так же, как функции пользователя.

Создание бинарного оператора

Чтобы создать бинарный оператор, например реализующий действие $x \cdot y^2$:

1. Введите имя оператора, например, bin.
2. Наберите знак скобки <(>, затем список из двух операндов через запятую, <x>, <,>, <y>, затем закрывающую скобку <)>.
3. Введите оператор присваивания < :>.
4. Введите выражение, зависящее от операндов, действие которого необходимо присвоить оператору (xy^2).

Создание унарного оператора

Унарный оператор создается точно так же, только вместо двух операндов, отделенных запятой, Вам следует ввести лишь один операнд. Например, чтобы создать оператор с именем %, реализующий перевод доли числа в проценты и сводящийся к умножению его на 100 (листинг 3.30):

1. Введите имя оператора. Для этого нажмите клавиши <a>, <Ctrl>+ +<Shift>+<K>,<%>, затем снова <Ctrl>+<Shift>+<K>, потом сотрите в имени букву "a".
2. После знака % наберите скобку "(", далее "x", затем еще одну скобку ")".
3. Введите оператор присваивания, нажав клавишу <:>.
4. Введите выражение `x·100`.

Листинг 3.30. Создание унарного оператора пользователя

```
% (x) := x·100
```

Использование бинарного оператора

Возможны два вида вставки пользовательского бинарного оператора в документ, отличающиеся только отображением в документе. Чтобы вставить оператор в форме графа (или дерева):

1. Нажмите кнопку **Tree Operator** (Оператор дерево) на панели **Evaluation** (Выражения) (рис. 3.12, справа).
2. В появившиеся местозаполнители введите имя оператора (на вершине графа) и значения операндов (в ответвления дерева).
3. Введите оператор присваивания, нажав клавишу <=>.

Результат действия оператора показан на рис. 3.12, внизу слева.

Кроме древовидной формы оператора, допускается использование его в виде последовательности "операнд— имя оператора— другой операнд" (рис. 3.12, вторая строка слева). Чтобы ввести такую форму оператора, следует нажать соседнюю кнопку **Infix Operator** (Оператор внутри) с изображением **xfy**.

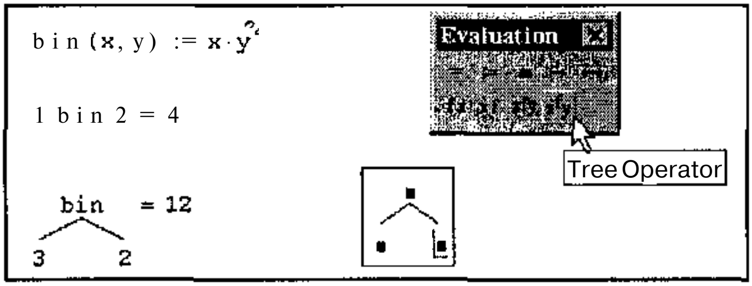


Рис. 3.12. Применение пользовательского бинарного оператора

Использование унарного оператора

Вставка унарного оператора совершенно аналогична, только вместо двух операндов требуется ввести один (рис. 3.13). Унарный оператор вставляется нажатием кнопки **Prefix Operator** (Оператор перед) на панели **Evaluation** (**Выражения**) либо кнопки **Postfix Operator** (Оператор после). Первый путь проиллюстрирован правой частью рис. 3.13 (в момент вставки) и результатом действия оператора (слева), а результат вставки оператора по второму пути — левой нижней строкой того же рисунка.

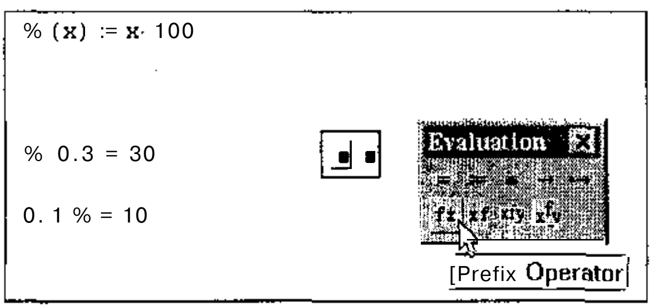


Рис. 3.13. Применение пользовательского унарного оператора

3.3. Управление вычислениями

Документ Mathcad — это в полном смысле этого слова компьютерная программа, а сама система Mathcad — настоящая система программирования, правда ориентированная на математика, а не на профессионального программиста. Большинство других сред программирования (знакомых читателю по реализации таких языков, как Си, Фортран, Бейсик и т. п.) разделяют редактирование кода программ и их выполнение, которое можно вызвать предназначенными для этого командами. В Mathcad и код программы, и результат их выполнения объединены в одном документе. Тем не менее, функции редактирования формул и их расчеты выполняются раздельно, и пользователь имеет возможность управлять всеми важнейшими опциями вычислений.

3.3.1. Режимы вычислений

Все примеры, которые мы рассматриваем в этой книге, неявно предполагают, что включен автоматический режим вычислений. Он включается по умолчанию при создании пустого документа, поэтому если вводятся выражения, содержащие операторы вывода, они вычисляются немедленно. Вообще говоря, имеется два режима вычислений:

- *автоматический режим* (automatic mode) — все вычисления проводятся автоматически по мере ввода формул;
- *ручной режим* (manual mode) — начало вычислений каждой формулы или всего документа производится пользователем.

Режим вычислений можно выбрать с помощью команды **Tools / Calculate / Automatic Calculation** (Сервис / Пересчитать / Считать автоматически), как показано на рис. 3.14. Если в этой строке меню установлен флажок проверки, значит, включен автоматический режим, если флажка нет, то редактируется документ в ручном режиме вычислений. Чтобы сменить режим, просто выберите этот пункт меню (например, нажав кнопку мыши в ситуации, показанной на рис. 3.14, включите ручной режим).

Примечание

Режим вычислений устанавливается независимо для каждого документа. Одновременно могут быть открыты несколько документов, вычисляемых в различных режимах.

Преимущества и недостатки каждого режима очевидны. С одной стороны, автоматические вычисления упрощают работу с документом, поскольку результаты расчетов появляются в реальном времени, и пользователь имеет возможность анализировать их сразу. С другой стороны, если вычисления сложные, то они могут отнимать много времени (что особенно заметно на компьютерах с не слишком мощным процессором и небольшим объемом оперативной памяти). Поэтому зачастую, чтобы продолжить редактирование документа, требуется довольно длительное ожидание завершения расчетов. В частности, если поменять какое-либо выражение в начале большого документа, которое влияет на последующие вычисления, то все они пересчитываются заново. В таких случаях часто удобнее редактировать текст в ручном режиме, а вычисления включать по мере необходимости.

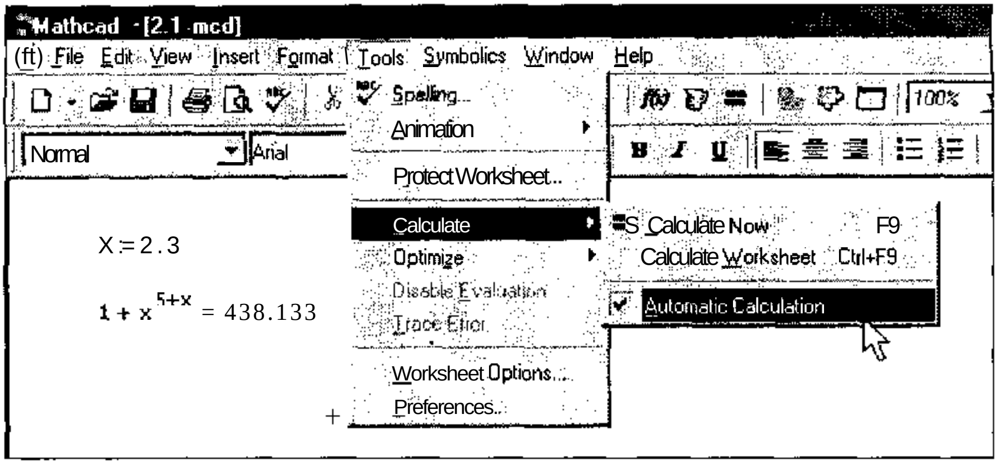


Рис. 3.14. Выбор режима вычислений

3.3.2. Прерывание вычислений

Mathcad осуществляет вычисления документа, как это принято в большинстве сред программирования: сверху вниз и слева направо. Пока очередное выражение находится в процессе расчета (вычислительным или символьным процессором), оно выделяется рамкой зеленого цвета (рис. 3.15), а любые действия пользователя по дальнейшему редактированию документа блокируются. Если у вас не слишком быстрый компьютер, а формулы достаточно сложные, то можно наблюдать, как зеленая рамка перескакивает с одного выражения на другое.

Чтобы прервать затянувшийся процесс вычислений, нажмите клавишу <Esc>. Появится диалоговое окно, показанное на рис. 3.16, в котором нужно подтвердить прерывание вычислений (**OK**). В этом случае выражения, которые Mathcad не успел вычислить, будут помечены в документы красным цветом. Прерванные вычисления возобновляются нажатием клавиши <F9> или командой **Tools/ Calculate/ Calculate Now** (Математика/ Пересчитать / Пересчитать).

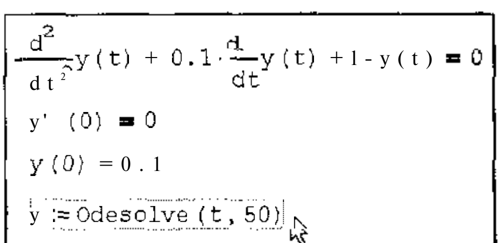


Рис. 3.15. Процесс вычисления выражения

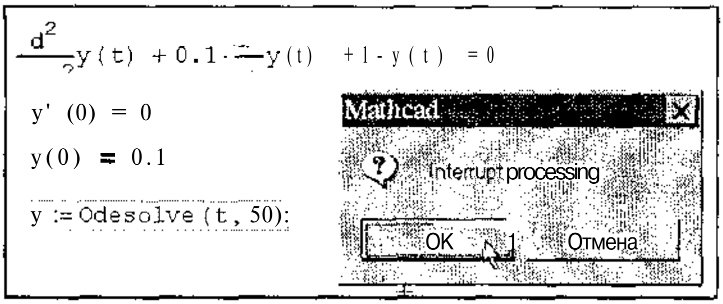


Рис. 3.16. Диалоговое окно прерывания вычислений

3.3.3. Вычисления в ручном режиме

Если флажок в строке команды **Tools/ Calculate/ Automatic Calculation** (Сервис / Пересчитать / Считать автоматически) снят, пользователь должен запускать вычисления самостоятельно.

- Для того чтобы вычислить все формулы во всем документе, выполните команду **Tools/ Calculate/ Calculate Worksheet** (Математика/ Пересчитать / Пересчитать все).
- Для вычисления всех формул в видимой части документа выберите пункт **Tools / Calculate / Calculate Now** (Сервис / Пересчитать / Пересчитать) либо нажмите клавишу <F9>, либо щелкните на кнопке с изображением знака равенства **(Calculate)** на стандартной панели инструментов.
- Прервать вычисления можно обычным образом, нажав клавишу <Esc>.

Совет

Управлять размером видимой части документа можно при помощи изменения масштаба отображения документа.

При редактировании текста в ручном режиме не выполняются ни вычисления, ни построение графиков, а соответствующие места в выражениях формально отмечаются местозаполнителями (рис. 3.17).

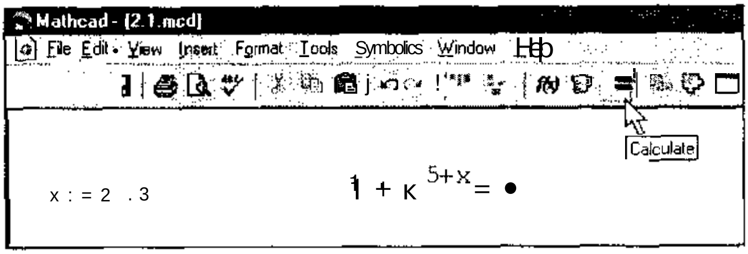


Рис. 3.17. Чтобы запустить вычисления в ручном режиме, нажмите кнопку Calculate

3.3.4. Отключение вычисления отдельныхформул

Mathcad позволяет отключить вычисление какой-либо формулы. При этом она не будет влиять на последующие вычисления. Чтобы не вычислять определенную формулу в документе:

1. Щелкните правой кнопкой мыши на формуле.
2. Выберите в контекстном меню пункт **Disable Evaluations** (Выключить вычисления), как показано на рис. 3.18.

Эквивалентный способ выключения вычисления отдельной формулы заключается в вызове диалогового окна Properties (Свойства) через одноименный пункт контекстного меню (см. рис. 3.18) или главного меню **Format** (Формат). В диалоге **Properties** следует перейти на вкладку **Calculation** (Вычисления) и установить там флажок **Disable Evaluations** (Выключить вычисления).

Результат выключения формулы из процесса вычислений проиллюстрирован листингом 3.31. На нем выключен второй из операторов присваивания, о чем можно судить по наличию черного квадрата сразу за формулой. Соответственно, в последней строке выведенное значение переменной x "не чувствует" выключенного присваивания и остается равным 3.

Листинг 3.31. Вычисление второго оператора присваивания выключено

$x := 3$

$x := 0$ ■

$x = 3$

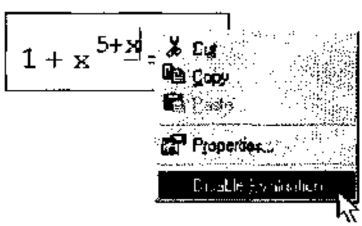


Рис. 3.18. Отключение вычисления формулы с помощью контекстного меню

3.3.5. Оптимизация вычислений

Отличительная черта новых версий Mathcad — улучшенные возможности ускорения численных вычислений за счет применения элементов символьной математики. Непосредственно перед численным расчетом Mathcad автоматически пытается упростить выражение, используя символьный процессор. Это называется *оптимизацией.* За счет того что от версии к версии качество работы символьного процессора улучшается, символьное преобразование зачастую существенно ускоряет расчеты. Режим оптимизации включается либо в документе целиком, либо для отдельных формул.

Чтобы включить или отключить режим оптимизации всех выражений в активном документе, выберите команду Tools / Optimize / Worksheet (Сервис / Оптимизация / Документ), как показано на рис. 3.19. Содержание документа, изображенного на этом же рисунке, помогает понять математический смысл режима оптимизации: для ускорения вычисления нижнего (определенного) интеграла выгодно использовать его аналитическое решение, определенное символьным процессором.

Чтобы изменить режим оптимизации для отдельной формулы, не меняя выбранного режима для остальных выражений документа, выделите эту формулу линиями ввода и выберите в верхнем меню Tools / Optimize / Equation (Сервис / Оптимизация / Уравнение).

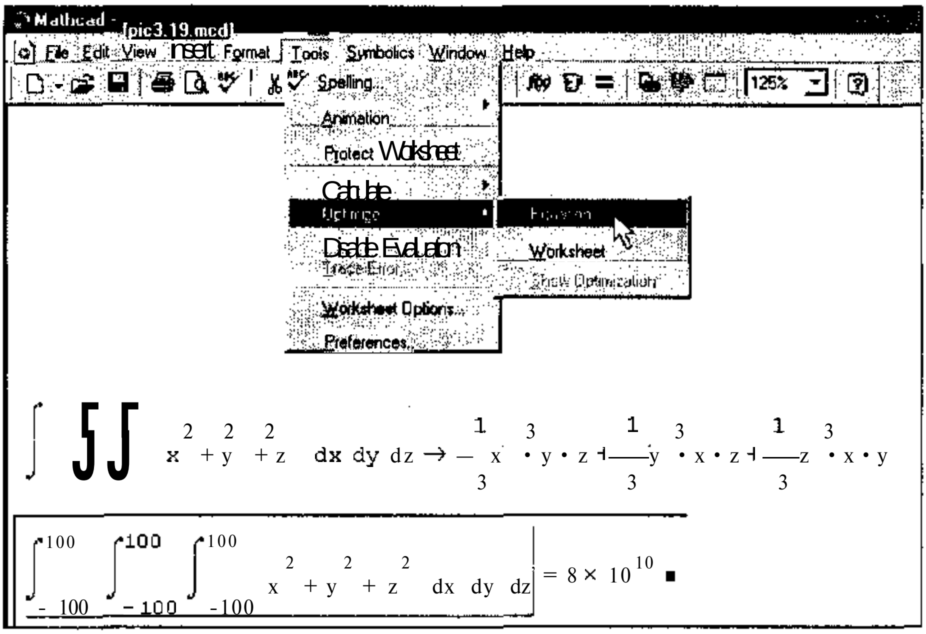


Рис. 3.19. Режим оптимизации вычислений

3.3.6. Диалоговое окно *Worksheet Options*

Наравне с изложенными способами установки режимов вычислений, их также удобно устанавливать для всего документа на вкладке Calculations

(Вычисления) диалогового окна **Worksheet Options** (Опции документа), вызываемого с помощью команды **Tools / Worksheet Options** (Сервис / Опции документа). Три флажка задают включение соответствующего режима вычислений (рис. 3.20).

- **Recalculate automatically** (Пересчитать автоматически) — включение режима автоматических вычислений.
- **Use strict singularity checking for matrices** (Использовать проверку матриц на сингулярность) — опция, появившаяся в версии Mathcad 2001, которая важна при некоторых операциях с матрицами. Она означает проведение дополнительной проверки на сингулярность матрицы перед использованием численных алгоритмов, что позволяет, во избежание неправильного применения численного метода, выдать заранее сообщение об ошибке, если матрица сингулярная.
- **Optimize expressions before calculating** (Оптимизировать выражения перед вычислением) — включение режима оптимизации.
- **Use exact equality for Boolean comparisons** (Использовать точное равенство для логического сравнения) — когда флажок выбран, применяется жесткий критерий точного равенства чисел (точнее, числа при сравнении считаются равными, если отличаются по модулю менее чем на 10^{-307}). Если флажок снят, используется более мягкий критерий (относительное различие чисел по модулю менее чем на 10^{-12}).

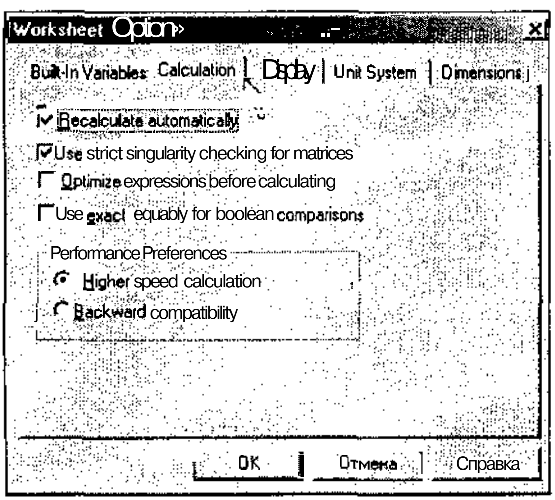


Рис. 3.20. Управление режимом вычислений в диалоговом окне Worksheet Options

Помимо флажков проверки, имеется также пара переключателей, которая позволяет реализовать новый режим *ускоренных вычислений* (higher speed calculation). Он включается выбором переключателя **Higher speed calculation** (Ускоренные вычисления), изображенного на рис. 3.20. Чтобы отключить режим ускоренных вычислений, выберите переключатель **Backward compatibility** (Обратная совместимость). В этом случае вычисления будут проводиться без дополнительной оптимизации по скорости, в точности так же, как в предыдущих версиях (Mathcad 2000 и ниже). Необходимость таких расчетов может возникнуть, если вдруг Вы столкнулись с сообщениями об ошибках в документах, созданных в предыдущих версиях Mathcad и корректно в них работающих.

3.4. Сообщения об ошибках

Когда процессор Mathcad по тем или иным причинам не может вычислить выражение, он вместо ответа выдает сообщение об ошибке (рис. 3.21). Если курсор находится вне формулы с ошибкой, то в ней имя функции или переменной, которая вызвала ошибку, отмечается красным цветом (сверху на рис. 3.21). При щелчке на такой формуле под ней появляется текстовое сообщение о типе ошибки, обрамленное черным прямоугольником (рис. 3.21, снизу).

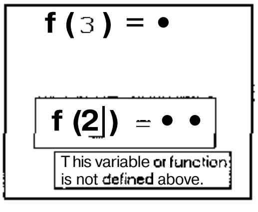


Рис. 3.21. Сообщение об ошибке

Если некоторые выражения вызывают ошибку, они просто игнорируются, а следующие выражения в документе по-прежнему вычисляются. Конечно, если формулы, вызвавшие ошибку, влияют на значения нижеследующих формул, то они будут также интерпретированы как ошибочные. Поэтому, встречая в документе сообщения об ошибках, найдите сначала самое первое из них. Часто ее устранение позволяет избавиться и от последующих ошибок.

Как бы хорошо вы ни овладели системой Mathcad, сообщения об ошибках все равно будут появляться в документах. Они могут быть связаны как с орфографическими ошибками, так и с более серьезными внутренними причинами, требующими знания численных алгоритмов расчетов. Искусство математика во многом состоит в умении анализировать ошибочные ситуации и находить правильный выход из них.

ГЛАВА 4

Типы данных

В этой главе рассматриваются типы данных, применяемых в документах Mathcad, и принципы их ввода-вывода в наиболее простой числовой форме. Обычные типы данных (действительные и комплексные числа, константы и строковые данные) перечисляются в начале главы *(см. разд. 4.1),* параллельно с объяснением принципов их ввода в документы. Наряду с обычными числами, в Mathcad имеется мощный аппарат работы с массивами. Массивы реализованы в виде векторов и матриц, что максимально приближает стиль вычислений к общепринятой математической форме *(см. разд. 4.3).* Описана возможность форматирования результатов численных расчетов *(см. разд. 4.4).*

Отличительной чертой среды Mathcad является возможность обращения с размерными переменными, снабженными физическими единицами измерений *(см. разд. 4.2).* Эти средства существенно облегчают инженерные и научные расчеты.

В некоторых случаях удобно осуществлять ввод данных при помощи элементов управления *(см. разд. 4.5).* Другие формы ввода-вывода (графики, анимация, ввод-вывод в файлы) разобраны в последней части книги *(см. гл. 16).*

4.1. Типы данных

Наиболее простой и распространенный ввод-вывод данных в Mathcad реализован присваиванием и выводом (либо численным, либо символьным) непосредственно в документе. Переменные и функции, посредством которых осуществляется ввод-вывод, могут иметь значения различных типов (числовые, строковые и т. д.). Перечислим основные типы данных, которые обрабатываются процессорами системы Mathcad:

- *числа* (в том числе, действительные, комплексные, а также встроенные константы) — Mathcad хранит все числа в формате двойной точности с плавающей точкой (не разделяя их на целые, булевы и т. д.);

- *строки* — любой текст, заключенный в кавычки;
- *массивы* (в том числе ранжированные переменные, векторы и матрицы) — упорядоченные последовательности чисел или строк.

Рассмотрим более подробно типы данных и то, как осуществляется их непосредственный ввод в документ с помощью присваивания значения переменным.

4.1.1. Действительные числа

Любое выражение, начинающееся с цифры, Mathcad интерпретирует как число. Поэтому для ввода числа просто начните его набирать на клавиатуре. Несмотря на то, что Mathcad хранит все числа в одинаковом формате, вводить их можно в наиболее подходящем *представлении* (notation), исходя из контекста документа:

- как *целое число;*
- как *десятичное число* (decimal notation) с любым количеством десятичных цифр после точки;
- в представлении *с порядком* (exponential notation) — в так называемом *научном формате* или *представлении* (scientific notation), для чего после ввода числа напечатайте символ умножения и введите ю в нужной степени;
- как число в другой системе счисления.

Три первых представления иллюстрируются содержанием соответствующей строки листинга 4.1.

Внимание!

При вводе целых чисел, больших или равных 1000, все цифры пишутся слитно (как показано в первой строке листинга 4.1) и ни в коем случае не разделяются на порядки запятыми. Например, ввод числа 1000 как 1,000 или 1.000 недопустим.

Листинг 4.1. Ввод действительных чисел

$a := 10000$

$b := 2.57285 \qquad c := 312.1$

$d := 4.17 \cdot 10^{-23} \qquad e := 345.1 \cdot 10^{3}$

Примечание

Если вы продолжите листинг 4.1 последовательным выводом всех переменных, то с удивлением обнаружите, что некоторые из чисел выглядят по-иному (например, число d=0). *Объяснение этому будет дано в разд. 4.2.*

Для ввода числа в других системах счисления: *двоичной* (binary), *восьмеричной* (octal) или *шестнадцатеричной* (hexadecimal) сделайте следующее:

1. Введите его представление в соответствующей системе, применяя лишь корректные символы (для двоичной системы допустимы только цифры 0 и 1; для восьмеричной — цифры от 0 до 7, для шестнадцатеричной — цифры от 0 до 9 и буквы от а до f). Например, число 34 в двоичной системе представлено такой последовательностью: 100010.
2. После ввода последнего символа числа введите b (для двоичного числа), о (для восьмеричного числа) или h (для шестнадцатеричного).

Использование чисел в других системах счисления иллюстрируется листингом 4.2. Обратите внимание, что вывод осуществляется все равно в десятичной системе.

Листинг 4.2. Ввод чисел в других системах исчисления

```
a := 100010b        a = 34
b := 37o            b = 31

c := 0af0h          c = 2.8 × 10^3
```

Примечание

В логических функциях используются битовые числа (ложь или истина). Они а Mathcad обозначаются обычными действительными числами о и 1.

4.1.2. Комплексные числа

Большинство операций в среде Mathcad по умолчанию осуществляются над *комплексными числами.* Комплексное число является суммой действительного и *мнимого числа,* получающегося путем умножения любого действительного числа на *мнимую единицу* (imaginary unit) i. По определению, $i = \sqrt{-1}$ или $i^2 = -1$.

Чтобы ввести мнимое число, например 3i:

1. Введите действительный сомножитель (з).
2. Введите символ "\" или "j" непосредственно после него.

Внимание!

Для ввода мнимой единицы надо нажать клавиши <1>, <i.>. Если просто ввести символ "i", то Mathcad интерпретирует его как переменную i. Кроме того, мнимая единица имеет вид 1i, только когда соответствующая формула выделена. В противном случае мнимая единица отображается просто как i (рис. 4.1).

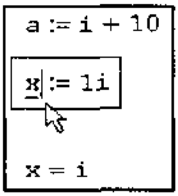


Рис. 4.1. Ввод мнимой единицы

Комплексное число можно ввести в виде обычной суммы действительной и мнимой частей или в виде любого выражения, содержащего мнимое число. Примеры ввода и вывода комплексных чисел иллюстрируются листингом 4.3.

Листинг 4.3. Комплексные числа

$x := 2i + 4$

$y := 19.785j + 0.1$

$z := 23 \cdot e^{0.1i}$

$x = 4 + 2i$

$y = 0.1 + 19.785i$

$z = 22.885 + 2.296i$

Для работы с комплексными числами имеются несколько простых функций и операторов *(см. разд. "Функции работы с комплексными числами" гл. 10),* действие которых показано в листинге 4.4.

Листинг 4.4. Функции работы с комплексными числами

$y := 19.785j + 0.1$

$\mathrm{Im}(y) = 19.785$ $\mathrm{Re}(y) = 0.1$

$z := 23 \cdot e^{0.1i}$

$|z| = 23$ $\arg(z) = 0.1$

Примечание

Можно выводить мнимую единицу в результатах вычислений не как i, а как *j*. Для смены представления выберите нужное в списке **Imaginary Value** (Мнимое значение) диалогового окна **Result Format** (Формат результата), доступного по команде **Format / Result / Display Options** (Формат / Результат / Опции отображения).

4.1.3. Встроенные константы

Некоторые имена в Mathcad зарезервированы под системные переменные, которые называются *встроенными константами* (built-in constants). Встроенные константы делятся на два типа: *математические,* хранящие значения некоторых общеупотребительных специальных математических символов, и *системные*, определяющие работу большинства численных алгоритмов, реализованных в Mathcad.

Математические константы *(math constants)*

- ∞ — символ бесконечности (вводится клавишами <Ctrl>+<Shift>+<z>);
- e — основание натурального логарифма (клавиша <e>);
- π — число "пи" (вводится клавишами <Ctrl>+<Shift>+<p>);
- i, j — мнимая единица (вводится клавишами <1>, <i> или <1>, <j>);
- % — символ процента, <%>, эквивалентный 0.01.

Математические константы по-разному интерпретируются при численных и символьных вычислениях. Вычислительный процессор просто воспринимает их как некоторые числа (листинг 4.5), а символьный распознает каждое из них, исходя из математического контекста, и способен выдавать математические константы в качестве результата.

Листинг 4.5. Значения математических констант

$\infty = 1 \times 10^{307}$

$e = 2.718$

$\pi = 3.142$

$i = i$

$j = i$

$\% = 0.01$ $\qquad$ $100 \cdot 25 \cdot \% = 25$

При желании можно изменить значение любой из перечисленных констант или использовать их в качестве переменных в расчетах (см. листинг 4.1, в котором переопределена константа e). Разумеется, если присвоить константе новое значение, прежнее станет недоступным.

Системные переменные *(system variables)*

- TOL — точность численных методов *(см. часть III)*;
- CTOL — точность выполнения выражений, используемая в некоторых численных методах *(см. часть III)*;
- ORIGIN — номер начального индекса в массивах *(см. разд. 4.3.1)*;
- PRNPRECISION — установка формата данных при выводе в файл;
- PRNCOLWIDTH — установка формата столбца при выводе в файл;
- CWD — строковое представление пути к текущей рабочей папке.

Предустановленные значения системных переменных перечислены в листинге 4.6. Их можно поменять в любой части документа, присвоив соответствующей переменной новое значение. Кроме того, переопределение значения переменной для всего документа производится при помощи команды **Tools / Worksheet Options / Built-in Variables** (Сервис / Опции документа / Встроенные переменные) в диалоговом окне **Worksheet Options** (Опции документа), приведенном на рис. 4.2. Чтобы в любой момент вернуть значения по умолчанию, нажмите кнопку **Restore Defaults** (Восстановить установки по умолчанию).

Листинг 4.6. Предустановленные значения системных переменных

$TOL = 1 \times 10^{-3}$

$CTOL = 1 \times 10^{-3}$

ORIGIN = 0

PRNPRECISION = 4

PRNCOLWIDTH = 8

CWD = "C:\Dima\MCAD\MathCad 2001\4 Data\"

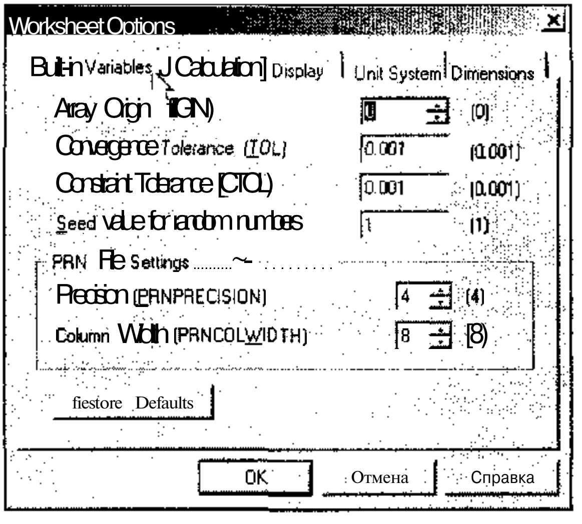


Рис. 4.2. Вкладка **Built-in Variables** диалога **Worksheet Options**

4.1.4. Строковые выражения

Значением переменной или функции может быть не только число, но и строка, состоящая из любой последовательности символов, заключенной в кавычки (листинг 4.7). Для работы со строками в Mathcad имеется несколько встроенных функций *(см. разд. "Строковые функции" гл. 10).*

Листинг 4.7. Ввод и вывод строк

```
s:="Hello,"

concat(s, " world! ") = "Hello, world! "
```

Примечание

Совершенно аналогичным образом можно определять пользовательские функции строкового типа.

4.2. Размерные переменные

В Mathcad числовые переменные и функции могут обладать *размерностью*. Сделано это для упрощения инженерных и физических расчетов. В Mathcad встроено большое количество *единиц измерения*, с помощью которых и создаются размерные переменные.

4.2.1. Создание размерной переменной

Чтобы создать размерную переменную, определяющую, например, силу тока в 10 А:

1. Введите выражение, присваивающее переменной `I` значение ю: `I:=10`.
2. Сразу после ввода `10` введите символ умножения "*".
3. Находясь в области местозаполнителя, выберите команду **Insert / Unit** (Вставка / Единицы) либо нажмите кнопку с изображением мерного стакана на стандартной панели инструментов, либо клавиши <Ctrl>+<U> (рис. 4.3).
4. В списке **Unit** (Единица измерения) диалогового окна **Insert Unit** (Вставка единицы измерений) выберите нужную единицу измерения **Ampere (A)**.
5. Нажмите кнопку ОК.

Если Вы затрудняетесь с выбором конкретной единицы измерения, но знаете, какова размерность переменной (в нашем случае это электрический ток), то попробуйте выбрать ее в списке **Dimension** (Размерность) диалогового окна **Insert Unit** (Вставка единицы измерений) (рис. 4.4). Тогда в списке **Unit** (Единица измерения) появятся допустимые для этой величины единицы измерений, из которых выбрать нужную будет легче (рис. 4.5).

Просмотреть вставку единиц измерения можно и без выхода из диалогового окна **Insert Unit**, нажимая вместо кнопки **OK** кнопку **Insert** (Вставить).

В этом случае Вы увидите, что единица измерений появилась в нужном месте документа, и можете поменять ее, оставаясь в диалоге **Insert Unit**.

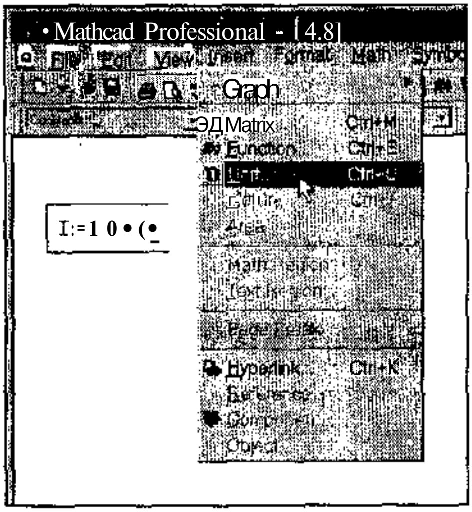


Рис. 4.3. Вставка единиц измерения размерной величины

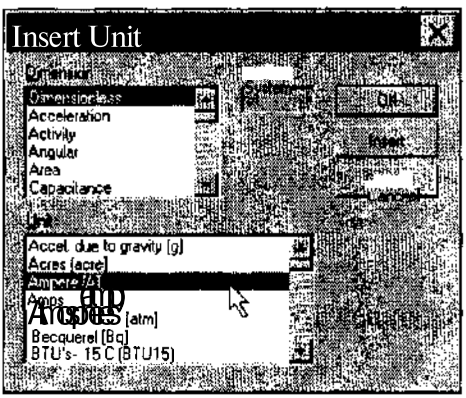


Рис. 4.4. Диалоговое окно Insert Unit

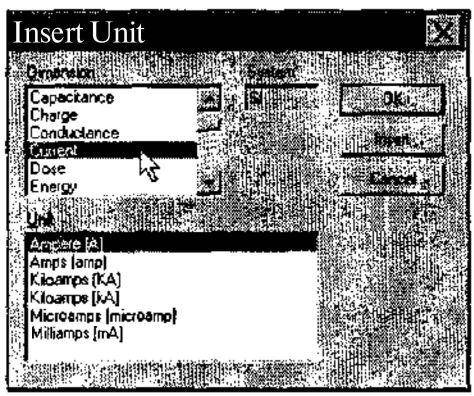


Рис. 4.5. Выбор размерности

4.2.2. Работа с размерными переменными

Работая с размерными переменными, приготовьтесь к тому, что Mathcad будет постоянно контролировать корректность расчетов. Например, нельзя складывать переменные разной размерности, в противном случае (рис. 4.6) будет получено сообщение об ошибке **"The units in this expression do not match"** (Размерности в этом выражении не совпадают). Тем не менее, позволяется складывать, например, амперы с килоамперами (см. рис. 4.9).

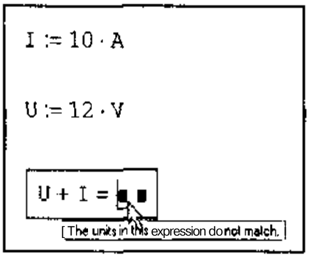


Рис. 4.6. Нельзя складывать переменные разной размерности

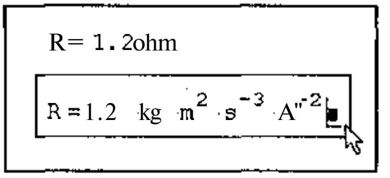


Рис. 4.7. Изменение единиц измерения в ответе

Над размерными переменными можно производить любые корректные с физической точки зрения расчеты. Пример расчета сопротивления через отношение напряжения к току приведен в листинге 4.8.

Листинг 4.8. Расчеты с размерными переменными

$$I := 10 \cdot A$$

$$U := 12 \cdot V$$

$$R := \frac{U}{I}$$

$$R = 1.2\ kg\,m^2\,s^{-3}\,A^{-2}$$

Обратите внимание, что результат в листинге 4.8 выдан не в омах. Тем не менее, легко перевести его в омы, как и в другие единицы. Для этого достаточно дважды щелкнуть на местозаполнителе, присутствующем после вычисленного значения формулы в момент, когда она выделена (рис. 4.7, внизу). В результате появляется то же самое диалоговое окно Insert Unit (Вставка единицы измерений), в котором можно поменять единицу измерений вычисленного ответа. В результате ответ будет пересчитан в соответствии с вновь введенной единицей измерения (как сделано для верхней формулы на рис. 4.7).

Примечание

Можно включить автоматический перевод единиц измерения в более простые единицы. Для этого перейдите в диалоговое окно **Result Format** (Формат результата) на вкладку, посвященную размерностям, с помощью команды **Format/Result/Unit Display** (Формат/ Результат/ Отображение размерности). Установите в ней флажок **Simplify units when possible** (Упрощать единицы, когда это возможно).

Процесс смены единиц в момент их выбора в диалоге Insert Unit (Вставка единицы измерений) показан на рис. 4.8. В результате приведенного выбора выведенное в амперах значение силы тока будет изменено на значение в килоамперах: 1.01 кА.

$$10 \cdot A + kA = 1.01 \times 10^{3} \cdot A$$

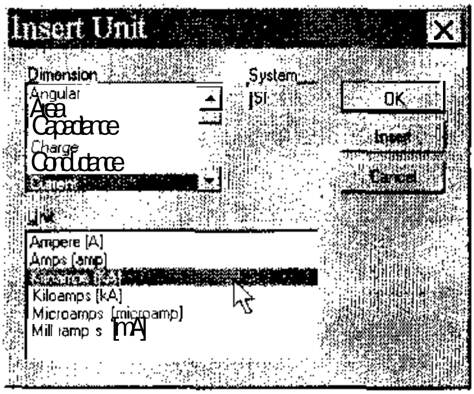


Рис. 4.8. Сложение переменных одной размерности, выраженных в разных единицах

4.2.3. Выбор системы единиц

Как легко заметить, во всех примерах этого раздела вставлялись единицы системы измерения SI. Об этом можно было судить как по перечню самих единиц, так и по недоступному списку System (Система) в диалоге Insert Unit (Вставка единицы измерений) с выбранным пунктом SI. Сменить систему единиц во всем документе можно, выполнив команду Tools / Document Options (Сервис / Опции документа) и переходя на вкладку Unit System (Система единиц). В ней следует выбрать один из переключателей, соответствующий желаемой системе.

4.2.4. Определение новой размерности

Чтобы определить новую (пользовательскую) единицу измерения, достаточно присвоить ее выражение через используемые размерности переменной с соответствующим именем. Пример создания новой единицы измерения "наноампер" приведен в листинге 4.9.

Листинг 4.9. Определение новой единицы измерения

$$nA := 10^{-9} \cdot A$$

$$3 \cdot A = 3 \times 10^{9}\, nA$$

Созданные пользователем единицы измерения недоступны в диалоговом окне Insert Unit (Вставка единицы измерений), поэтому их приходится вводить вручную с клавиатуры (как это сделано для новой единицы nA во второй строке листинга 4.9).

4.3. Массивы

Массивами (arrays) называют упорядоченные последовательности чисел или *элементов массива.* Доступ к любому элементу массива возможен по его *индексу,* т. е. номеру в последовательности чисел (в листинге 4.10 a — это массив, a_1 — его элемент). Применение массивов чрезвычайно эффективно в математических расчетах.

Листинг 4.10. Одномерный массив (вектор)

$$a := \begin{pmatrix} 14 \\ 1.4 \\ 4.7 \end{pmatrix}$$

$a_0 = 14$

$a_1 = 1.4$

$a_2 = 4.7$

В Mathcad условно выделяются два типа массивов:

- *векторы* (одноиндексные массивы, листинг 4.10), *матрицы* (двухиндексные, листинг 4.11) и *тензоры* (многоиндексные);
- *ранжированные переменные* (range variables) — векторы, элементы которых определенным образом зависят от их индекса.

Листинг 4.11. Двумерный массив (матрица)

$$a := \begin{pmatrix} 0.1 & 2.8 \\ 3.7 & 0 \end{pmatrix}$$

$a_{0,0} = 0.1$

$a_{1,0} = 3.7$

$a_{1,1} = 0$

4.3.1. Доступ к элементам массива

Доступ ко всему массиву осуществляется по имени векторной переменной. Например, последовательность символов "a", "=" в листингах 4.10 и 4.11 приведет к выводу соответствующего вектора или матрицы. В Mathcad имеются и операторы, и встроенные функции, которые действуют на векторы и матрицы целиком *(они рассматриваются в гл. 9),* например, транспонирование, матричное умножение и т. д.

Над элементами массива можно совершать действия как над обычными числами. Нужно только правильно задать соответствующий индекс или сочетание индексов массива. Например, чтобы получить доступ к нулевому элементу вектора a из листинга 4.10:

1. Введите имя переменной массива (a).
2. Нажмите кнопку Subscript (Нижний индекс) со значком $\mathbf{x_n}$ на панели **Matrix** (Матрица) либо введите [.
3. В появившийся справа снизу от имени массива местозаполнитель введите желаемый индекс (о).

Если после этого ввести знак численного вывода, то справа от него появится значение нулевого элемента вектора, как показано во второй строке листинга 4.10.

Чтобы получить доступ к элементу многоиндексного массива (например элементу $a_{1,0}$ матрицы a из листинга 4.11):

1. Введите имя переменной массива (a).
2. Перейдите к вводу нижнего индекса, введя [.
3. Введите в местозаполнитель индекса первый индекс (1), запятую "," и в появившийся после запятой местозаполнитель введите второй индекс (о).

В результате будет получен доступ к элементу, как показано в предпоследней строке листинга 4.11.

В рассмотренных листингах нумерация индексов массивов начинается с нуля, иными словами, первый элемент массива имеет индекс о. Стартовый индекс массива задается системной переменной ORIGIN, которая по умолчанию равна нулю. Если Вы привыкли нумеровать элементы векторов и матриц с единицы, присвойте этой переменной значение 1 (листинг 4.12). Обратите внимание, что в этом случае попытка выяснить значение нулевого элемента вектора приводит к ошибке, поскольку его значение не определено.

Помимо доступа к отдельным элементам массива, имеется возможность совершать действия над его подмассивами (например векторами-столбцами, образующими матрицу). Делается это с помощью оператора со значком $\mathbf{x}^{<>}$ на панели **Matrix** (Матрица) *(см. гл. 9)*.

Листинг 4.12. Изменение нумерации индексов массивов

$$ORIGIN := 1$$

$$a := \begin{pmatrix} 14 \\ 1.4 \\ 4.7 \end{pmatrix}$$

$$a_0 = \blacksquare$$

$a_1 = 14$

$a_2 = 1.4$

$a_3 = 4.7$

4.3.2. Ранжированные переменные

Ранжированные переменные в Mathcad являются разновидностью векторов и предназначены, главным образом, для создания *циклов* или итерационных вычислений. Простейший пример ранжированной переменной — это массив с числами, лежащими в некотором диапазоне с некоторым шагом.

Например, для создания ранжированной переменной s с элементами 0,1,2,3,4,5:

1. Поместите курсор ввода в нужное место документа.
2. Введите имя переменной (s) и оператор присваивания ":".
3. Нажмите кнопку **Range Variable** (Ранжированная переменная) на панели **Matrix** (Матрица), показанную на рис. 4.9, либо введите символ точки с запятой с клавиатуры.
4. В появившиеся местозаполнители (рис. 4.9) введите левую и правую границы диапазона изменения ранжированной переменной о и 5.

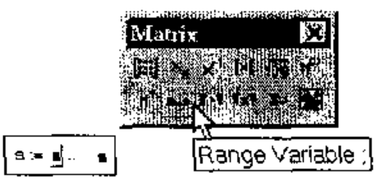


Рис. 4.9. Создание ранжированной переменной

Результат создания ранжированной переменной показан на рис. 4.10.

Чтобы создать ранжированную переменную с шагом, не равным 1, например, 0,2,4,6,8:

1. Создайте ранжированную переменную в диапазоне от о до 8 (см. рис.4.9).
2. Поместите линии ввода на значение начала диапазона (о).
3. Введите запятую.
4. В появившийся местозаполнитель (рис. 4.11) введите значение шага изменения ранжированной переменной (2).

Созданная ранжированная переменная будет иметь значения от о до 8 включительно, с шагом, равным 2.

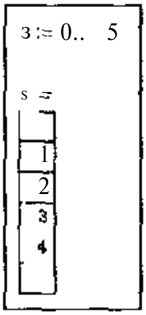


Рис. 4.10. Вывод ранжированной переменной

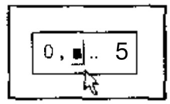


Рис. 4.11. Создание ранжированной переменной с шагом, не равным 1

Чаше всего ранжированные переменные используются:

- при параллельных вычислениях (листинги 4.13 и 4.14);
- для присвоения значений элементам других массивов (листинги 4.14 и 4.15).

Обратите внимание на типичный пример использования ранжированной переменной из листингов 4.13 и 4.14. Большинство математических действий, реализованных в Mathcad, совершаются над ранжированными переменными точно так же, как над обычными числами. В этом случае одно и то же действие осуществляется *параллельно* над всеми элементами ранжированной переменной.

Листинг 4.13. Ранжированная переменная при параллельных вычислениях

$i := 0, 2 .. 8$

$s(i) := i^2 + 1$

i =	s(i) =	sin(s(i))
0	1	0.841
2	5	-0.959
4	17	-0.961
6	37	-0.644
8	65	0.827

Примечание

Параллельные вычисления производятся точно так же и над произвольными векторами, не обязательно являющимися ранжированными переменными. Например, можно определить в листинге 4.14 вектор i, подобно вектору из листинга 4.10, и провести те же параллельные вычисления над его элементами.

Листинг 4.14. Ранжированная переменная при параллельных вычислениях

$i := 0 .. 5$

$s_i := i^2 + 1$

$$s = \begin{pmatrix} 1 \\ 2 \\ 5 \\ 10 \\ 17 \\ 26 \end{pmatrix} \qquad \sin(s) = \begin{pmatrix} 0.841 \\ 0.909 \\ -0.959 \\ -0.544 \\ -0.961 \\ 0.763 \end{pmatrix}$$

Листинг 4.15. Использование ранжированной переменной для определения матрицы

$i := 0 .. 3$

$j := 0 .. 5$

$C_{i,j} := i + j$

$$C = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}$$

Внимание!

Определяя массив с помощью ранжированных переменных (листинги 4.14 и 4,15), позаботьтесь о том, чтобы их значения пробегали все необходимые индексы массива. Например, если задать шаг изменения ранжированной переменной, равный 2, то половина элементов вектора будет не определена.

Помните о том, что ранжированные переменные — просто разновидности векторов с упрощенной формой задания элементов. Часто необходимо провести одни и те же вычисления циклически, большое количество раз, например, вычисление некоторой функции f (x) в некотором диапазоне x для построения подробного графика. Задание вручную всех значений аргумента {наподобие вектора из листинга 4.10) очень трудоемко, а с помощью задания ранжированной переменной x это делается в одну строку.

4.3.3. Создание массивов

Существует несколько способов создания массива:

- ввод всех элементов вручную с помощью команды **Insert Matrix**;
- определение отдельных элементов массива;
- создание таблицы данных и ввод в нее чисел;
- применение встроенных функций создания массива *(см. гл. 9)*;
- создание связи с другим приложением, например Excel или MATLAB;
- чтение из внешнего файла данных;
- импорт из внешнего файла данных.

Рассмотрим основные способы создания массива, учитывая, что две последних возможности будут разобраны в последней части книги Применяйте способ, который оптимален в смысле простоты и читаемости конкретного документа, либо ставший наиболее для Вас привычным.

Создание матрицы командой *Insert Matrix*

Самый простой и наглядный способ создания вектора или матрицы заключается в следующем:

1. Нажмите кнопку **Matrix or Vector** (Матрица или вектор) на панели **Matrix** (Матрица) (рис. 4.12) либо клавиши <Ctrl>+<M>, либо выберите пункт меню **Insert / Matrix** (Вставка / Матрица).
2. В диалоговом окне **Insert Matrix** (Вставка матрицы) задайте целое число столбцов и строк матрицы, которую хотите создать. Например, для создания вектора 3x1 введите показанные на рис. 4.12 значения.
3. Нажмите кнопку **OK** или **Insert** (Вставить) — в результате в документ будет вставлена заготовка матрицы с определенным числом строк и столбцов (рис. 4.13).
4. Введите значения в местозаполнители элементов матрицы. Переходить от одного элемента матрицы к другому можно с помощью указателя мыши либо клавиш со стрелками.

Добавление в уже созданную матрицу строк или столбцов производится точно так же:

1. Выделите линиями ввода элемент матрицы, правее и ниже которого будет осуществлена вставка столбцов и (или) строк.
2. Вставьте в него матрицу, как было описано выше. При этом допускается задание числа столбцов или строк равным нулю (рис. 4.14).
3. Заполните местозаполнители недостающих элементов матрицы.

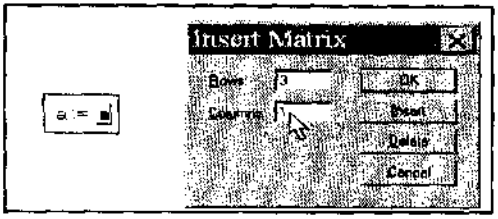


Рис. 4.12. Вставка матрицы

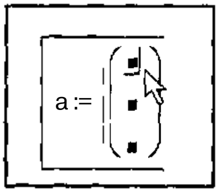


Рис. 4.13. Заполнение матрицы элементами

На рис. 4.14 и 4.15 показаны результаты последовательной вставки в матрицу столбца и строки после определения соответствующего числа столбцов и строк в диалоге Insert Matrix и нажатия в нем кнопки Insert (Вставить).

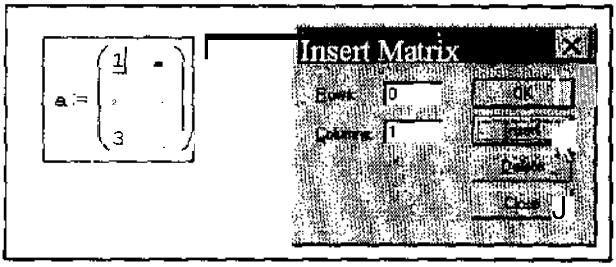


Рис. 4.14. Добавление одного столбца к матрице

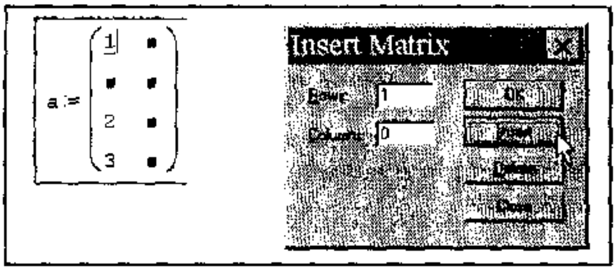


Рис. 4.15. Добавление одной строки к матрице

В местозаполнители элементов матрицы можно вставлять не только числа (действительные или комплексные), но и любые математические выражения, состоящие из переменных, операторов, встроенных и пользовательских функций (листинг 4.16, вторая строка).

Листинг 4.16. Использование переменных и функций при определении матрицы

$$x := 3$$

$$A := \begin{pmatrix} \sin(x) \\ x \end{pmatrix}$$

$$A = \begin{pmatrix} 0.141 \\ 3 \end{pmatrix}$$

Создание массива определением его отдельных элементов

Массив можно определить следующим образом:

- присваивая значения непосредственно отдельным элементам массива;
- применяя ранжированные переменные (см. листинг 4.15).

Любой из этих способов позволяет присвоить нужное значение как всем элементам массива (см. листинг 4.15), так и части из них, либо даже одному-единственному элементу. В последнем случае создается массив, размерность которого задается индексами введенного элемента (листинг 4.17), а неопределенным элементам по умолчанию присваиваются нулевые значения.

Листинг 4.17. Создание матрицы определением одного ее элемента

$s_{3,2} := 99$

$$s = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 99 \end{pmatrix}$$

В любом месте документа допускается как переопределение любого из элементов массива (листинг 4.18, первая строка), так и изменение его размерности. Чтобы поменять размерность всего массива, просто присвойте любое значение новому элементу, индексы которого выходят за границы прежней размерности (вторая строка листинга 4.18).

Примечание

В местозаполнители элементов матрицы допускается вставка любых функций, подобно применению обычного оператора присваивания.

Листинг 4.18. Изменение матрицы (*продолжение листинга 4.17*)

$s_{1,2} := 1$

$$s = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 99 \end{pmatrix}$$

$s_{4,4} := -7$

$$s = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 99 & 0 & 0 \\ 0 & 0 & 0 & 0 & -7 \end{pmatrix}$$

Создание тензора

Определение отдельных элементов — удобный способ создания тензоров (многоиндексных массивов). В Mathcad имеется непосредственная возможность работы только с векторами и матрицами. Тем не менее, можно создать тензор путем определения *вложенного массива* (nested array). Для этого необходимо присвоить каждому элементу матрицы значение в виде другого вектора или матрицы (листинг 4.19). Пользователь должен лишь позаботиться о корректности задания индексов тензора и не запутаться в индексировании вложенных матриц (последняя строка листинга).

Листинг 4.19. Создание тензора и доступ к его элементам

$s_{0,0} := \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ $\qquad s_{1,0} := \begin{pmatrix} 3 \\ 4 \end{pmatrix}$

$s_{0,1} := \begin{pmatrix} 5 \\ 6 \end{pmatrix}$ $\qquad s_{1,1} := \begin{pmatrix} 7 \\ 8 \end{pmatrix}$

$$s = \begin{pmatrix} \{2,1\} & \{2,1\} \\ \{2,1\} & \{2,1\} \end{pmatrix}$$

$\left(s_{1,0}\right)_0 = 1$ $\qquad \left(s_{1,0}\right)_1 = 4$

Совет

Процесс создания тензора автоматизирует применение ранжированных переменных.

Обратите внимание, что Mathcad по умолчанию не отображает трехмерную структуру тензора (предпоследняя строка листинга 4.19), а вместо этого показывает информацию о размерах каждого элемента матрицы s. Развернуть вложенные массивы можно с помощью команды **Format / Result / Display Options** (Формат / Результат / Опции отображения), устанавливая флажок **Expand Nested Arrays** (Разворачивать вложенные массивы) на вкладке **Display Options** (Опции отображения).

4.3.4. Отображение вывода векторов и матриц

Вы, вероятно, обратили внимание, что матрицы, векторы и ранжированные переменные отображались в различных примерах по-разному. Это связано с автоматическими установками отображения матриц, принятыми в Mathcad по умолчанию. Существуют два стиля отображения массива: в форме матрицы и в форме таблицы (рис. 4.16).

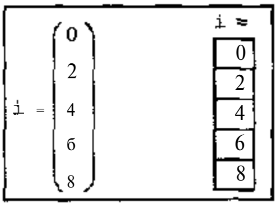


Рис. 4.16. Отображение массивов в форме матрицы (слева) и таблицы (справа)

Изменение стиля отображения какого-либо массива выполняется командой **Format / Result** (Формат / Результат), вызывающей диалог **Result Format** (Формат результата). В этом диалоге следует перейти на вкладку **Display Options** (Опции отображения) (рис. 4.17) и в списке **Matrix display style** (Стиль отображения матриц) выбрать один из стилей:

- **Automatic** (Авто) — стиль выбирается Mathcad;
- **Matrix** (Матрица);
- **Table** (Таблица).

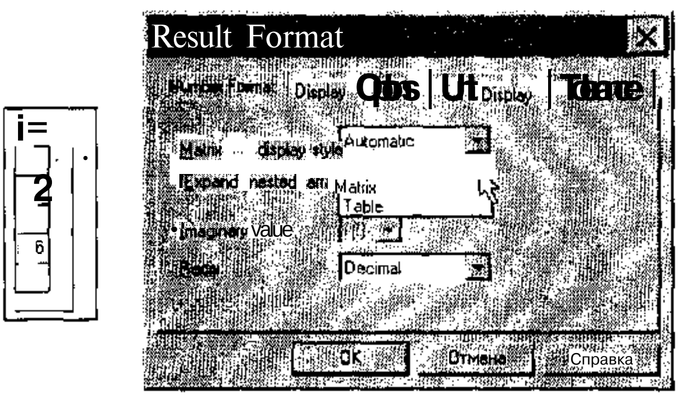


Рис. 4.17. Изменение стиля отображения массива

Стиль отображения в виде таблицы допускает различное выравнивание матрицы относительно выражения слева от оператора вывода (рис. 4,18). Для изменения выравнивания вызовите контекстное меню из области таблицы, наведите в нем указатель мыши на пункт **Alignment** (Выравнивание) и в подменю выберите тип выравнивания.

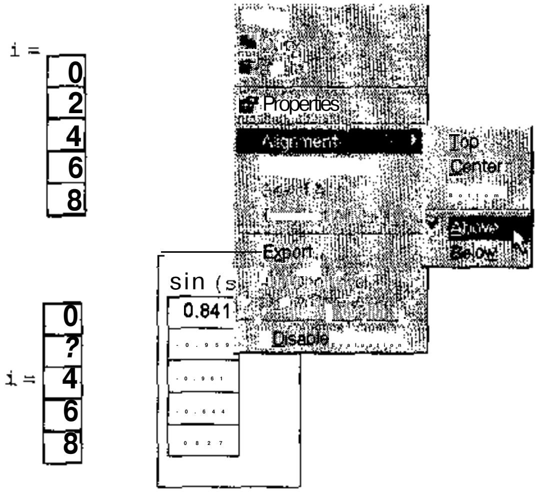


Рис. 4.18. Различные стили выравнивания матриц

В диалоговом окне **Result Format** (Формат результата), помимо стиля отображения матрицы, можно задать стиль отображения тензоров (вложенных массивов). Для того чтобы отображать тензоры в стиле, показанном на рис. 4.19, установите флажок **Expand nested arrays** (Разворачивать вложенные массивы). Чтобы отображать их в свернутой форме (см. листинг 4.19), снимите этот флажок.

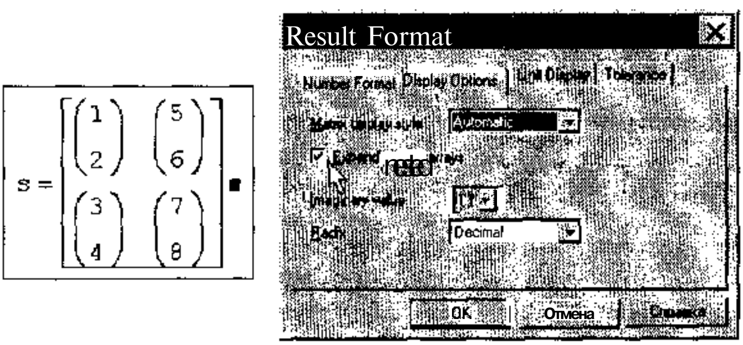


Рис. 4.19. Разворачивание вложенных массивов

Особенно наглядной формой отображения вектора является построение его в виде графика).

4.4. Формат вывода числовых данных

Несмотря на то, что невозможно влиять на результат, который отображается справа от оператора вывода значений переменных, функций и выражений, допускается изменять формат его отображения. Напомним, что как ввод, так и вывод данных может осуществляться в двух основных представлениях *(см. разд. 4.1.1)*:

- *десятичное* (decimal), например 13478.74559321;
- *с порядком* (exponential notation), например 1.348хю^4.

Выбор формата вывода числовых данных осуществляется при помощи диалогового окна **Result Format** (Формат результата). Оно вызывается командой **Format** / **Result** (Формат / Результат).

4.4.1. Формат результата

Управление представлением числа в десятичном представлении или представлении с порядком осуществляется при помощи следующих параметров:

- количество отображаемых десятичных знаков *(decimal places)* после точки. Например, число 122,5587 с четырьмя десятичными знаками при отображении с двумя знаками будет выглядеть как 122,56;
- отображение или скрытие незначащих нулей (trailing zeros) — опция, позволяющая показывать или скрывать незначащие нули в десятичном представлении числа, т. е. выводить, к примеру, "1,5" вместо "1,500" (даже если установлено количество десятичных знаков, равное 3);
- порядковый порог (exponential threshold), при превышении степени 10 которого число будет показываться с порядком. Например, при пороге 3 число 122,56 будет отображаться как десятичное, а при пороге 2 — уже как"1,23x10^2";

Примечание

Количество десятичных знаков левого сомножителя числа с порядком контролируется а некоторых форматах первым из трех перечисленных параметров.

- кроме того, число с порядком может представляться в эквивалентных видах: "1,23хЮ^2" или с порядком в инженерном формате (engineering format); "1.23E+002".

В Malhcad имеется несколько типов форматов, в каждом из которых разрешается изменение различных параметров представления числа. Формат выбирается на вкладке **Number Format** (Формат числа) диалогового окна **Result Format** (Формат результата) (рис. 4.20).

$$122.5889 = 1.226 \times 10^2$$

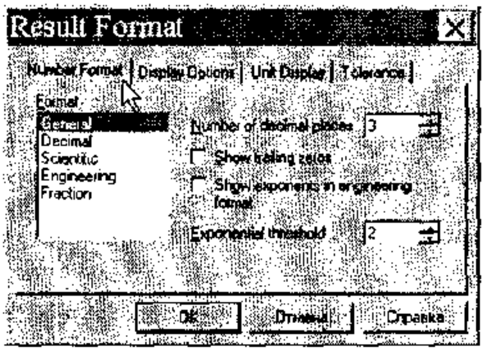


Рис. 4.20. Выбор формата вывода числа

Основной *(general)* формат

Этот формат принят при выводе чисел по умолчанию. Можно управлять и количеством отображаемых десятичных знаков (поле Number of decimal places), и порядковым порогом (поле Exponential threshold). При превышении порога число отображается с порядком (как показано на рис. 4.20). Несколько примеров вывода одного и того же числа в общем формате показано в листинге 4.21. В левой колонке приведены числа с порядковым порогом, равным 3, и количеством десятичных знаков (сверху вниз) 3, 4, 5, соответственно. Для нижнего числа установлен флажок отображения незначащих нулей. В правой колонке сгруппированы числа с порядковым порогом от 1 до 4 (сверху вниз).

Листинг 4.21. Основной формат результата

$152.5889 = 152.589$	$152.5889 = 1.526 \times 10^2$
$152.5889 = 152.5889$	$152.5889 = 1.5 \times 10^2$
$152.5889 = 152.58890$	$152.5889 = 152.58890$

Десятичный *(decimal)* формат

Числа отображаются только в десятичном представлении и никогда — в представлении с порядком.

Научный *(scientific)* формат

Числа отображаются только с порядком, причем количество десятичных знаков левого сомножителя, как и отображение незначащих нулей, определяется пользователем.

Инженерный *(engineering)* формат

Числа отображаются только с порядком, причем обязательно кратным 3; как и в научном формате, пользователю разрешается изменять количество десятичных знаков.

Дробный *(fraction)* формат

Этот формат сильно отличается от предыдущих, представляя число в виде дроби (рис. 4.21). Причем можно управлять как точностью представления числа с помощью поля **Level** of **accuracy** (Уровень точности), так и задать модификацию этого формата — отображение числа в виде целой и дробной части (как показано на рис. 4.21 внизу слева) посредством установки флажка Use mixed numbers (Смешанные числа).

Вид одного и того же числа в различных форматах приведен в листинге 4.22. В первой строке показан десятичный формат, во второй строке — научный с тремя десятичными знаками, в третьей — инженерный также с тремя десятичными знаками. В последних двух строках представлен дробный формат: в предпоследней с уровнем точности 5, в последней — 10. К тому же, для выражения последней строки установлен флажок Use mixed **numbers** (Смешанные числа).

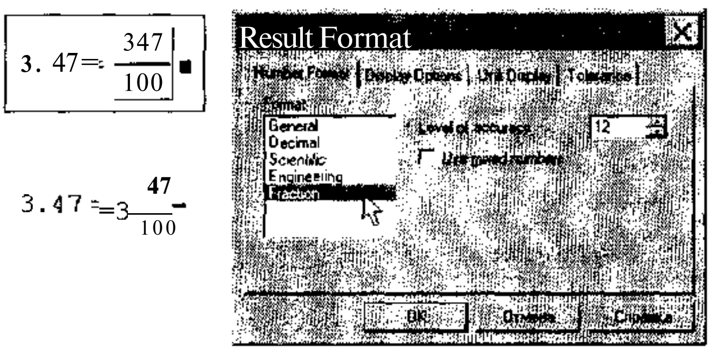


Рис. 4.21. Дробный формат

Листинг 4.22. Другие форматы результата вычислений

$12340.56789 = 12340.568$

$12340.56789 = 1.234 \times 10^{4}$

$12340.56789 = 12.341 \times 10^{3}$

$12340.56789 = \frac{999586}{81}$

$12340.56789 = 12340\,\frac{56789}{100000}$

4.4.2. Округление малых чисел до нуля

Mathcad автоматически округляет малые числа до нуля (листинг 4.23). Допускается установка порогового значения округления (в степенях 10), отдельно для действительной и мнимой части числа. При этом числа, по модулю меньшие порога, отображаются в виде нуля.

Внимание!

Помните, что это касается только отображения чисел. В памяти компьютера они хранятся корректно.

Листинг 4.23. Представление близких к нулю чисел

$2.15 \cdot 10^{-23} = 0$

$3.4 + i \cdot 10^{-11} = 3.4$

$-0.0000000000000001 = 0$

Чтобы изменить пороговые значения:

1. Щелкните на любом пустом месте документа.
2. Войдите в диалоговое окно **Result Format** (Формат результата): **Format / Result** (Формат / Результат).
3. Перейдите на вкладку **Tolerance** (Точность).
4. Установите пороговые значения для действительного нуля в поле **Zero threshold** (Порог нуля) и мнимого нуля в поле **Complex threshold** (Комплексный порог нуля).
5. Нажмите кнопку OK.

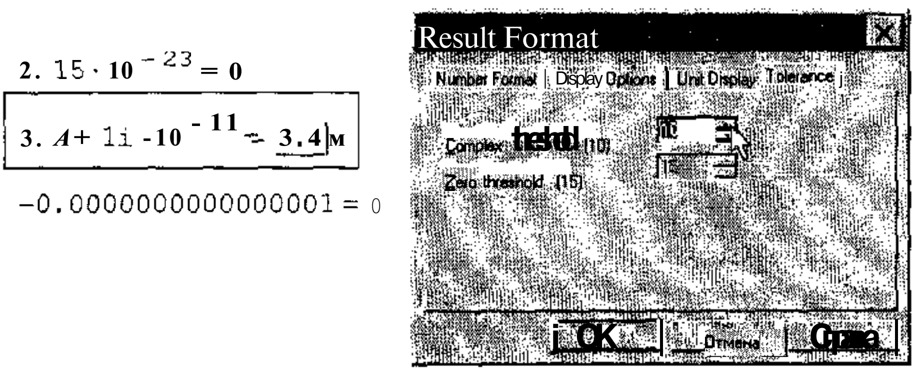


Рис. 4.22. Задание порога мнимого нуля

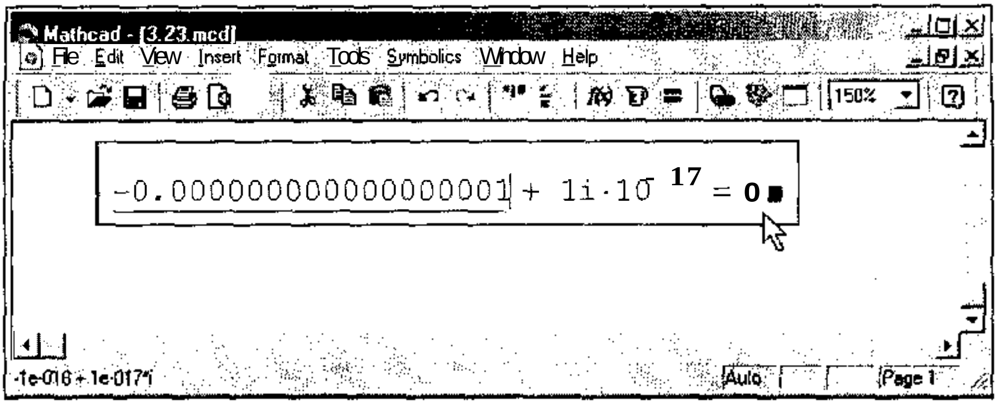


Рис. 4.23. Просмотр точного значения числа в строке состояния

Изменение порога мнимого нуля возможно и в режиме редактирования формулы (рис. 4.22), но изменение действительного порога нуля при этом недоступно.

Просмотреть число в точном представлении можно, нажав клавиши <Ctrl>+<Shift>+<N>. В этом случае на строке состояния (в самой нижней части окна Mathcad, слева) будет на короткое время выведен результат с максимальной точностью (рис. 4.23).

4.4.3. Вывод чисел в других системах счисления

Аналогично вводу чисел в других системах счисления *(см. разд. 4.1.1)*, вывести результат также возможно в виде десятичного, двоичного, восьмеричного или шестнадцатеричного числа (листинг 4.24, сверху вниз).

Листинг 4.24. Вывод чисел в других системах счисления

47 = 47

47 = 101111b

47 = 57o

47 = 2fh

Чтобы задать систему счисления, выберите команду **Format / Result / Display Options** (Формат/ Результат/ Опции отображения), а затем желаемый элемент списка **Radix** (Система счисления) (рис. 4.24). При отображении чисел в других системах счисления также доступно форматирование их представления на вкладке **Number Format** (Формат числа) того же диалога **Result Format** (Формат результата). В листинге 4.25 приведено несколько примеров форматирования чисел в двоичном представлении.

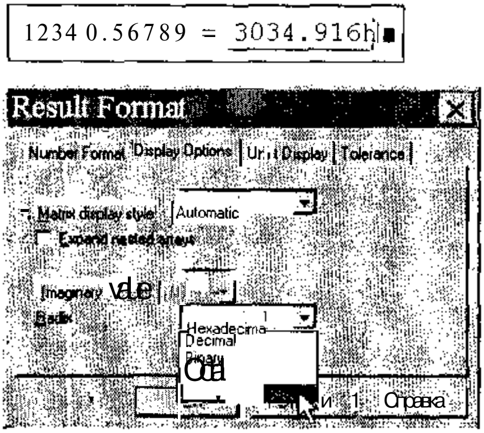


Рис. 4.24. Задание вывода результата в других системах счисления

Листинг 4,25. Форматирование вывода чисел в других системах счисления

$$47 = 101111b$$

$$47 = 1.1b \times 10b^{101b}$$

$$47 = 1.100b \times 10b^{101b}$$

Примечание

Мы рассмотрели в этой главе основные принципы наиболее простого численного ввода и вывода данных. О более впечатляющих формах ввода-вывода (графики, анимация, ввод-вывод в файлы) рассказывается в последней части книги.

4.5. Элементы управления (controls)

Одна из редко используемых возможностей Mathcad — ввод данных при помощи общеупотребительных элементов управления (такие, как поле ввода, ползунковый регулятор и т. п.). Очевидно, что такой способ ввода удобен, если Вы занимаетесь разработкой расчетов, которые предназначены для непрофессиональных пользователей Mathcad.

В Mathcad 11 имеются следующие элементы управления (рис. 4.25):

- **Check Box** (флажок проверки);
- **Radio Button** (переключатель);
- **Push Button** (кнопка);
- **Slider** (ползунковый регулятор);
- **Text Box** (поле текстового ввода);
- **List Box** (список).

5 Зак 984

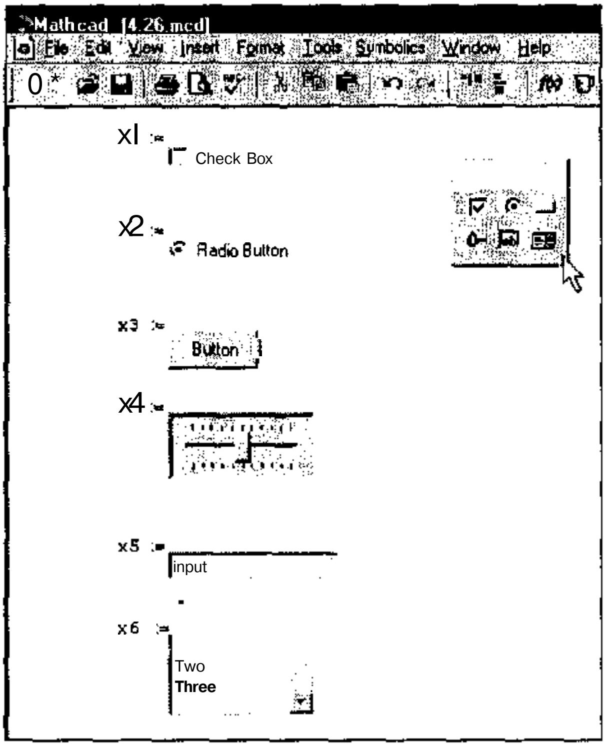


Рис. 4.25. Элементы управления MathSoft и панель **Controls**

Как видно из рис. 4.25, элементы управления в Mathcad используются для присваивания переменным значений, которые определяются действиями пользователя над элементами управления. К примеру, на рис. 4.26 два флажка проверки определяют переменные x0 и x1. Если флажок проверки выставлен, переменная принимает значение 1, а если снят — 0.

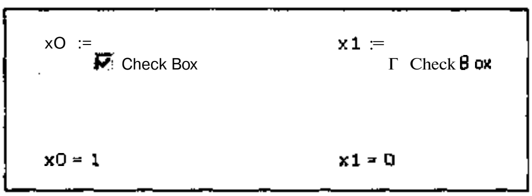
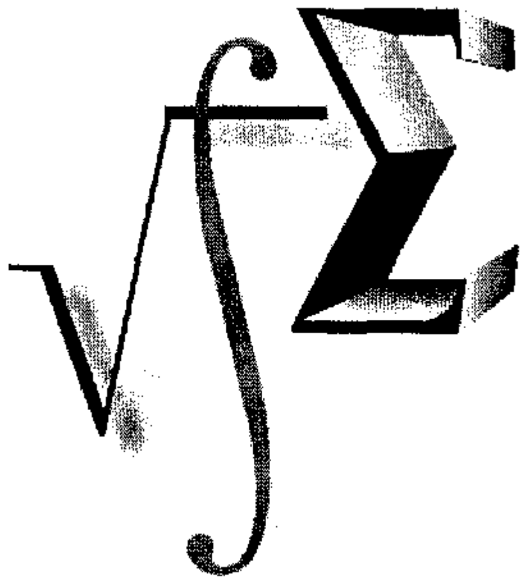


Рис. 4.26. Пример использования флажка проверки

Для того чтобы вставить элемент управления в документ, можно использовать либо команду меню Insert/ Controls (Вставка/ Элемент управления), либо панель инструментов Controls (Элементы управления), которую можно вызвать при помощи пункта Toolbars / Controls (Панели инструментов / Элементы управления) меню View (Вид). После нажатия кнопки с пиктограммой нужного элемента управления новый элемент управления появляется в документе вместе с местозаполнителем, который следует заменить именем переменной.

Чтобы отредактировать свойства самого элемента управления, следует вызвать на нем контекстное меню и выбрать в меню пункт Properties (Свойства). Большинство свойств имеет интуитивный смысл, и Вам будет несложно управлять характеристиками ввода данных посредством регулировки параметров самих элементов управления. Дополнительную информацию об использовании элементов управления Вы сможете найти в специальном справочном пособии, доступном по команде Help / Developer's Reference (Справка / Руководство разработчика).



ЧАСТЬ II

ТОЧНЫЕ ВЫЧИСЛЕНИЯ

ГЛАВА 5

Символьныевычисления

В данной главе рассматриваются возможности символьного процессора Mathcad. Он позволяет решить многие задачи математики аналитически, без применения численных методов и, соответственно, без погрешностей вычислений. В начале главы коротко говорится о путях проведения символьных вычислений в редакторе Mathcad *(см. разд. 5.1),* а основное содержание главы посвящено организации символьных вычислений для решения конкретных задач математики. Mathcad позволяет проводить широкий спектр аналитических преобразований, таких, как алгебраические и матричные операции *(см. разд. 5.2)*, основные действия математического анализа *(см. разд. 5.3)* и расчеты интегральных преобразований функций *(см. разд. 5.4).*

Необходимо отметить, что приемы многих символьных вычислений описываются также в третьей части данной книги, в рамках рассказа о решении конкретных вычислительных задач. В заключение главы приводится несколько практических приемов проведения эффективных символьных вычислений *(см. разд. 5.5).*

5.1. Способы символьных вычислений

Символьные вычисления в Mathcad можно осуществлять в двух различных вариантах:

- с помощью команд меню;
- с помощью оператора символьного вывода $\rightarrow$, ключевых слов символьного процессора и обычных формул (в справочной системе Mathcad этот способ называется *символьными вычислениями в реальном времени* — live symbolic evaluation).

Первый способ более удобен, когда требуется быстро получить какой-либо аналитический результат для однократного использования, не сохраняя сам ход вычислений. Второй способ более нагляден, т. к. позволяет записывать выражения в традиционной математической форме и сохранять символьные вычисления в документах Mathcad. Кроме того, аналитические преобразования, проводимые через меню, касаются только одного, выделенного в данный момент, выражения. Соответственно, на них не влияют формулы, находящиеся в документе Mathcad выше этого выделенного выражения (например, операторы присваивания значений каким-либо переменным). Оператор символьного вывода, напротив, учитывает все предыдущее содержимое документа и выдает результат с его учетом.

Примечание

В символьных вычислениях допускается использование большинства встроенных функций Mathcad.

Для символьных вычислений при помощи команд предназначено главное меню **Symbolics** (Символика), объединяющее математические операции, которые Mathcad умеет выполнять аналитически (рис. 5.1). Для реализации второго способа применяются все средства Mathcad, пригодные для численных вычислений (например, панели **Calculator, Evaluation** и т. д.), и специальная математическая панель инструментов, которую можно вызвать на экран нажатием кнопки **Symbolic Keyword Toolbar** (Панель символики) на панели **Math.** На панели **Symbolic** (Символические) находятся кнопки, соответствующие специфическим командам символьных преобразований (рис. 5.2). Например, таким, как разложение выражения на множители, расчет преобразования Лапласа и другим операциям, которые в Mathcad нельзя проводить численно и для которых, соответственно, не предусмотрены встроенные функции.

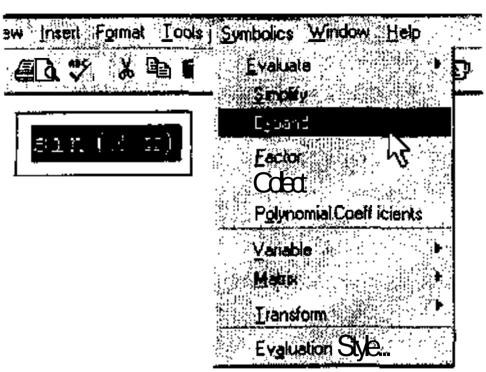


Рис. 5.1. Меню **Symbolics**

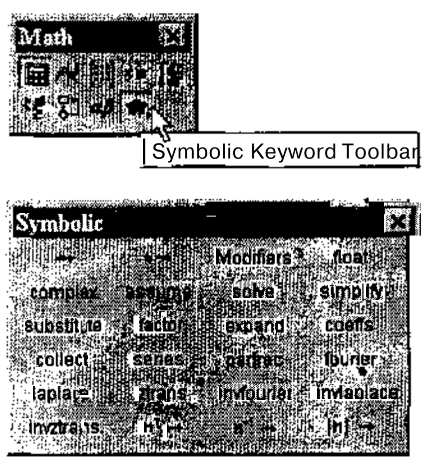


Рис. 5.2. Панель **Symbolic**

Рассмотрим оба типа символьных вычислений на простом примере разложения на сомножители выражения sin (2-х).

Первый способ (с помощью меню).

1. Введите выражение sin(2·x).
2. Выделите его целиком (см. рис. 5.1).
3. Выберите в главном меню пункты **Symbolics / Expand** (Символика / Разложить).

После этого результат разложения выражения появится чуть ниже в виде еще одной строки (рис. 5.3).

$$\sin(2 \cdot x)$$

$$2 \cdot \sin(x) \cdot \cos(x)$$

Рис. 5.3. Результат применения команды меню **Symbolics / Expand**

Внимание!

Символьные операции с помощью меню возможны лишь над каким-либо объектом (выражением, его частью или отдельной переменной). Для того чтобы правильно осуществить желаемое аналитическое преобразование, предварительно необходимо выделить тот объект, к которому оно будет относиться. В данном случае преобразование было применено ко всему выражению sin (2-х). Если же выделить часть формулы, как показано на рис. 5.4, то соответствующее преобразование будет отнесено к выделенной части (нижняя строка на этом рисунке).

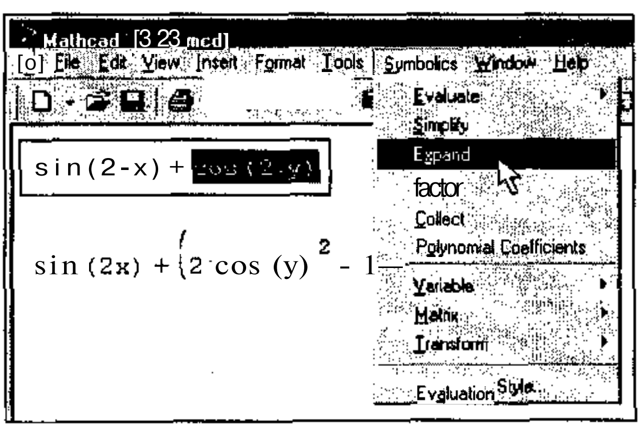


Рис. 5.4. Символьное разложение части выражения и его результат

Второй способ символьных преобразований (с помощью оператора →).

1. Введите выражение sin(2·x).
2. Нажмите кнопку **Expand** (Разложить) на панели **Symbolic.**

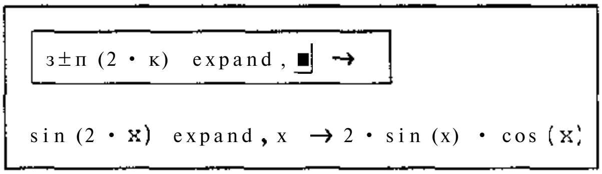


Рис. 5.5. Символьное разложение выражения

3. Введите в местозаполнитель после появившегося ключевого слова expand (рис. 5.5, сверху) имя переменной x либо нажмите клавишу <Del>, чтобы просто удалить местозаполнитель.
4. Введите оператор символьного вывода →.
5. Нажмите клавишу <Enter> либо просто щелкните мышью за пределами выражения.

Оператор символьного вывода, как Вы помните, можно ввести в редакторе Mathcad несколькими способами: нажатием кнопки → на любой из панелей **Evaluation** (Выражения) или **Symbolic** (Символика) либо сочетанием клавиш <Ctrl>+<.>. Результат символьного разложения выражения показан на рис 5.5, внизу.

Внимание!

Если символьные вычисления осуществляются вторым способом, символьный процессор учитывает все формулы, предварительно введенные в документе (рис. 5.6, внизу). Но если те же преобразования выполняются при помощи меню, символьный процессор "не видит" ничего, кроме одной формулы, и воспринимает все ее переменные аналитически, даже если им предварительно были присвоены какие-то значения (рис. 5.6, сверху). По этой причине, например, символьным преобразованиям через меню недоступны предварительные определения функций пользователя.

Совет

Если Вы можете выбрать способ символьных вычислений, рекомендуем второй путь — с помощью оператора →, поскольку при этом в документе сохраняются действия пользователя. Наличие специального меню символьных вычислений — своего рода дань прежним версиям Mathcad. В них аналитические преобразования были встроены не так гармонично и были доступны, главным образом, через меню.

Не всякое выражение поддается аналитическим преобразованиям. Если это так (либо в силу того, что задача вовсе не имеет аналитического решения, либо она оказывается слишком сложной для символьного процессора Mathcad), то в качестве результата выводится само выражение (листинг 5.1, внизу).

Листинг 5.1. Символьные преобразования

$\cos(2 \cdot x)$ expand , x $\rightarrow 2 \cdot \cos(x)^2 - 1$

$\cos(x)$ expand , x $\rightarrow \cos(x)$

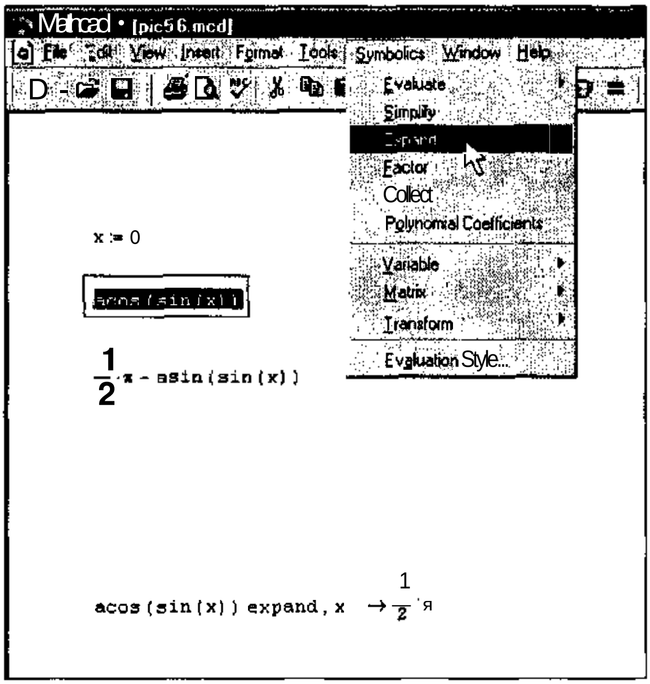


Рис. 5.6. Различив в символьных вычислениях при помощи меню (сверху) и оператора → (снизу)

Примечание

Далее в этой главе, рассматривая символьные вычисления с помощью меню, будем иллюстрировать результаты рисунками, а символьные вычисления с применением оператора → приводить в виде листингов.

5.2. Символьная алгебра

Символьный процессор Mathcad умеет выполнять основные алгебраические преобразования, такие, как упрощение выражений, разложение их на множители, символьное суммирование и перемножение.

5.2.1. Упрощение выражений (Simplify)

Упрощение выражений — наиболее часто применяемая операция. Символьный процессор Mathcad стремится так преобразовать выражение, чтобы оно приобрело более простую форму. При этом используются различные арифметические формулы, приведение подобных слагаемых, тригонометрические тождества, пересчет обратных функций и др. Чтобы упростить выражение с помощью меню (рис. 5.7):

1. Введите выражение.
2. Выделите выражение целиком или его часть, которую нужно упростить.
3. Выберите команду **Symbolics / Simplify** (Символика / Упростить).

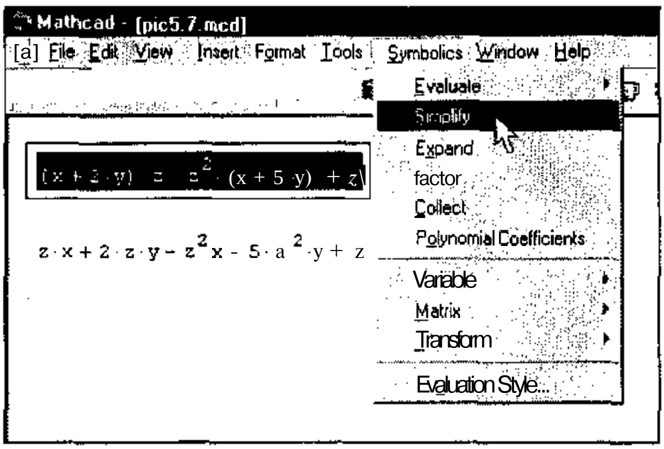


Рис. 5.7. Упрощение выражения

Для упрощения выражения при помощи оператора символьного вывода используйте ключевое слово simplify (листинг 5.2). Не забывайте, если некоторым переменным, входящим в выражение, ранее были присвоены некоторые значения, то они будут подставлены в него при выполнении символьного вывода (листинг 5.3).

Листинг 5.2. Упрощение выражения

$$(x+2\cdot y)\cdot z-z^2\cdot(x+5\cdot y)+z\ \mathrm{simplify}\ \rightarrow z\cdot x+2\cdot z\cdot y-z^2\cdot x-5\cdot z^2\cdot y+z$$

Листинг 5.3. Упрощение выражения с подстановкой значения переменных

$x:=10 \quad y:=1$

$$(x+2\cdot y)\cdot z-z^2\cdot(x+5\cdot y)+z\ \mathrm{simplify}\ \rightarrow 13\cdot z-15\cdot z^2$$

Упрощение выражений, содержащих числа, производится по-разному, в зависимости от наличия в числах десятичной точки. Если она есть, то выполняется непосредственное вычисление выражения (листинг 5.4).

Листинг 5.4. Упрощение выражения с числами

$\sqrt{3}$ simplify $\rightarrow \sqrt{3}$

$\sqrt{3.01}$ simplify $\rightarrow 1.7349351572897472412$

5.2.2. Разложение выражений (Expand)

Операция символьного *разложения,* или *расширения,* выражений противоположна по смыслу операции упрощения. В ходе разложения раскрываются все суммы и произведения, а сложные тригонометрические зависимости разлагаются с помощью тригонометрических тождеств. Разложение выражений производится путем выбора команды **Symbolics / Expand** (Символика / Разложить) либо использованием вместе с оператором символьного вывода ключевого слова expand. *Применение операции разложения было подробно рассмотрено в разд. 5.1* (см. рис. 5.3—5.6 и листинг 5.1).

5.2.3. Разложение на множители (Factor)

Разложение выражений на простые множители производится при помощи команды **Symbolics / Factor** (Символика / Разложить на множители) (рис. 5.8) либо использованием вместе с оператором символьного вывода ключевого слова factor (листинг 5.5). Эта операция позволяет разложить полиномы на произведение более простых полиномов, а целые числа — на простые сомножители. Применяя команду меню, не забывайте перед ее вызовом выделить все выражение или его часть, которую планируете разложить на множители.

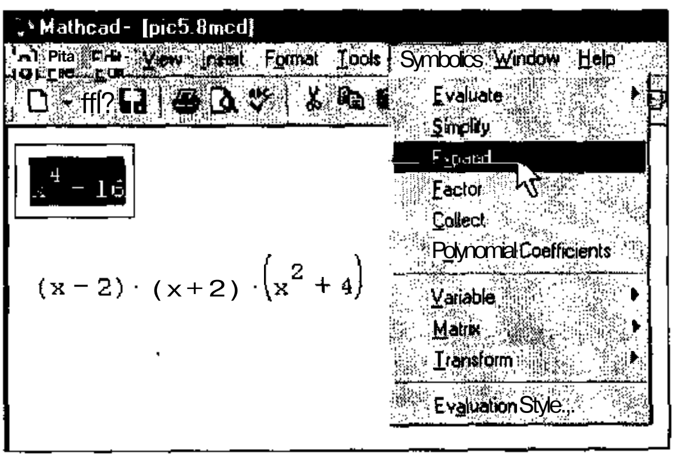


Рис. 5.8. Разложение выражения на множители

Листинг 5.5. Примеры разложения на множители

$$x^4 - 16 \text{ factor } \rightarrow (x-2)\cdot(x+2)\cdot\left(x^2+4\right)$$

$$28 \text{ factor } \rightarrow 2^2 \cdot 7$$

5.2.4. Приведение подобных слагаемых (Collect)

Чтобы привести подобные слагаемые полинома с помощью меню (рис. 5.9):

1. Введите выражение.
2. Выделите в выражении имя переменной, относительно которой надо привести подобные слагаемые (в примере на рис. 5.9 это переменная y).
3. Выберите команду **Symbolics / Collect** (Символика / Привести подобные).

В результате появится строка с результатом приведения подобных слагаемых (нижняя строка на рис. 5.9).

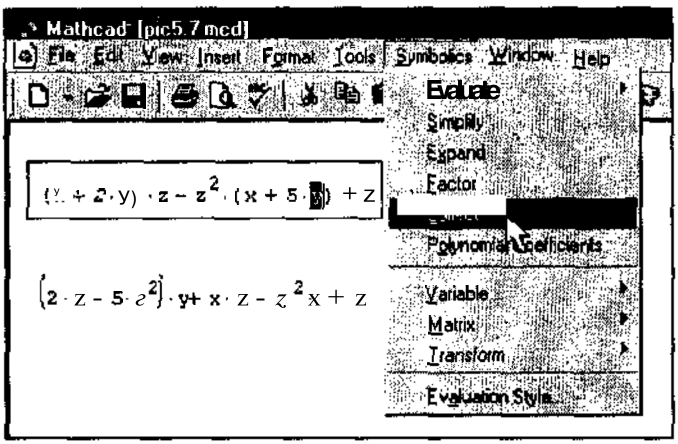


Рис. 5.9. Приведение подобных слагаемых

Чтобы привести подобные слагаемые с помощью оператора символьного вывода (листинг 5.6):

1. Введите выражение.
2. Нажмите кнопку **Collect** на панели **Symbolic** (Символика).
3. Введите в местозаполнитель после вставленного ключевого слова collect имя переменной, относительно которой требуется привести подобные слагаемые (в первой строке примера из листинга 5.6 это переменная x, во второй — y).
4. Введите оператор символьного вывода →.
5. Нажмите клавишу <Enter>.

Примечание

После ключевого слова `collect` допускается задание нескольких переменных через запятую. В этом случае, что иллюстрируется последней строкой листинга 5.6, приведение подобных слагаемых выполняется последовательно по всем переменным.

Листинг 5.6. Приведение подобных слагаемых по разным переменным

$$(x+2\cdot y)\cdot z - z^2\cdot y\cdot(x+5\cdot y)+z \text{ collect}, x \rightarrow \left(z-z^2\cdot y\right)\cdot x + 2\cdot z\cdot y - 5\cdot z^2\cdot y^2 + z$$

$$(x+2\cdot y)\cdot z - z^2\cdot y\cdot(x+5\cdot y)+z \text{ collect}, y \rightarrow -5\cdot z^2\cdot y^2 + \left(2\cdot z - z^2\cdot x\right)\cdot y + z\cdot x + z$$

$$(x+2\cdot y)\cdot z - z^2\cdot y\cdot(x+5\cdot y)+z \text{ collect}, x, y, z \rightarrow \left(z-z^2\cdot y\right)\cdot x + 2\cdot z\cdot y - 5\cdot z^2\cdot y^2 + z$$

5.2.5. Коэффициенты полинома (Polynomial Coefficients)

Если выражение является полиномом относительно некоторой переменной x, заданным не в обычном виде $a_0+a_1x+a_2x^2+\ldots$, а как произведение других, более простых полиномов, то коэффициенты $a_0, a_1, a_2\ldots$ легко определяются символьным процессором Mathcad. Коэффициенты сами могут быть функциями (подчас, довольно сложными) других переменных.

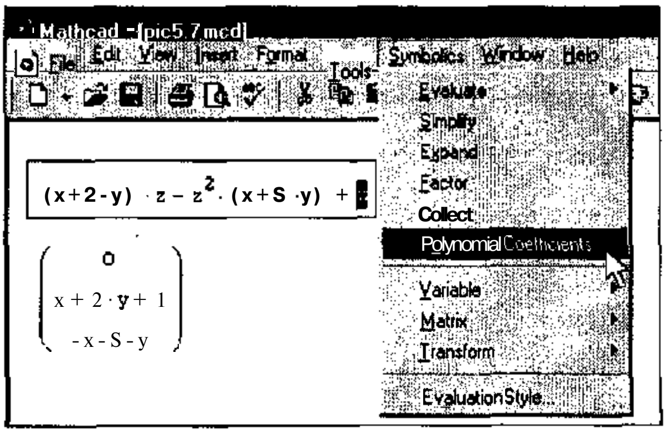


Рис. 5.10. Вычисление коэффициентов полинома

Чтобы вычислить полиномиальные коэффициенты в выражении при помощи меню (рис. 5.10):

1. Введите выражение.
2. Выделите в нем имя переменной или выражение, для которого требуется рассчитать полиномиальные коэффициенты (в примере на рис. 5.10 это переменная z).

3. Выполните команду **Symbolic / Polynomial Coefficients** (Символика / Коэффициенты полинома).

В результате под выражением появится вектор, состоящий из полиномиальных коэффициентов. Первым элементом вектора является свободный член a_0, вторым — a_1, и т. д.

Примечание

Конкретная задача, требующая вычисления полиномиальных коэффициентов, приведена в разделе, посвященном численному отделению корней полинома *(см. разд. "Корни полинома" гл. 8).*

Чтобы вычислить полиномиальные коэффициенты с помощью оператора символьного вывода:

1. Введите выражение.
2. Нажмите кнопку **Coeffs** на панели **Symbolic** (Символика).
3. Введите в местозаполнитель после вставленного ключевого слова coeffs аргумент полинома.
4. Введите оператор символьного вывода →.
5. Нажмите клавишу <Enter>.

Примеры вычисления коэффициентов полинома приведены в листингах 5.7 и 5.8. Листинг 5.7 показывает расчет коэффициентов для разных аргументов. Последний листинг демонстрирует возможность определения коэффициентов не только для отдельных переменных, но для более сложных выражений, входящих в рассматриваемую формулу в качестве составной части.

Листинг 5.7. Вычисление коэффициентов полинома

$$(x+2\cdot y)\cdot z - z^2\cdot y\cdot(x+5\cdot y)+z \text{ coeffs}, z \rightarrow \begin{pmatrix} 0 \\ x+2\cdot y+1 \\ -y\cdot x-5\cdot y^2 \end{pmatrix}$$

$$(x+2\cdot y)\cdot z - z^2\cdot y\cdot(x+5\cdot y)+z \text{ coeffs}, x \rightarrow \begin{pmatrix} 2\cdot z\cdot y-5\cdot z^2\cdot y^2+z \\ z-z^2\cdot y \end{pmatrix}$$

Листинг 5.8. Вычисление полиномиальных коэффициентов для простой переменной и выражения

$$(x-4)\cdot(x-7)\cdot x+99 \text{ coeffs}, x \rightarrow \begin{pmatrix} 99 \\ 28 \\ -11 \\ 1 \end{pmatrix}$$

$$(x-4)^3 + (x-4)\cdot(x-7)\cdot x + 99 \text{ coeffs}, x-4 \rightarrow \begin{pmatrix} 99 \\ x^2 - 7\cdot x \\ 0 \\ 1 \end{pmatrix}$$

5.2.6. Ряды и произведения

Чтобы вычислить символьно конечную или бесконечную сумму или произведение:

1. Введите выражение, используя панель **Calculus** (Вычисления) для вставки соответствующих символов суммирования или произведения *(см. разд. "Вычислительные операторы" гл. 3)*. При необходимости введите в качестве предела ряда символ бесконечности (клавиши <Ctrl>+<Shift>+<Z>).
2. В зависимости от желаемого стиля символьных вычислений выберите команду **Symbolics / Simplify** (Символика / Упростить) или введите оператор символьного вывода →.

Примеры численного и символьного вычисления рядов и произведений приведены в листингах 5.9 и 5.10.

Листинг 5.9. Символьные и численные расчеты рядов

$$\overset{10}{\underset{i=0}{}} \; . \; 047 \times 10^3 \qquad \sum_{i=0}^{10} 2^i \rightarrow 2047$$

$$\sum_{i=0}^{\infty} a^i \rightarrow \frac{-1}{(a-1)}$$

$$\sum_{n=0}^{\infty} \frac{x^n}{2^n\cdot n!} \rightarrow \exp\left(\frac{1}{2}\cdot x\right) \qquad \sum_{n=0}^{\infty} \frac{1^n}{2^n\cdot n!} \rightarrow \exp\left(\frac{1}{2}\right) = 1.649$$

$$\overset{100}{\underset{n=0}{}} \frac{n}{2\cdot n!} = 1.649$$

Листинг 5.10. Символьный расчет произведения

$$\prod_{n=1}^{\infty} \frac{1}{n^3+1} \rightarrow 0 \qquad \prod_{n=1}^{\infty} \sqrt{n} \rightarrow \infty$$

5.2.7. Разложение на элементарные дроби (Convert to Partial Fractions)

Чтобы разложить сложную дробь на более простые дроби, следует либо выполнить команду **Symbolics / Variable / Convert to Partial Fractions** (Символика / Переменная / Разложить на элементарные дроби) (рис. 5.11), либо указать ключевое слово `parfrac` (листинг 5.11). Применяя первый способ (меню), не забывайте перед выбором его команды выделить переменную, по которой будет производиться разложение, а если используется второй способ (с оператором символьного вывода), то имя переменной следует указать после ключевого слова `parfrac`. В общем, последовательность действий при разложении на дроби та же самая, что и обычно *(см., например, разд. 5.2.4).*

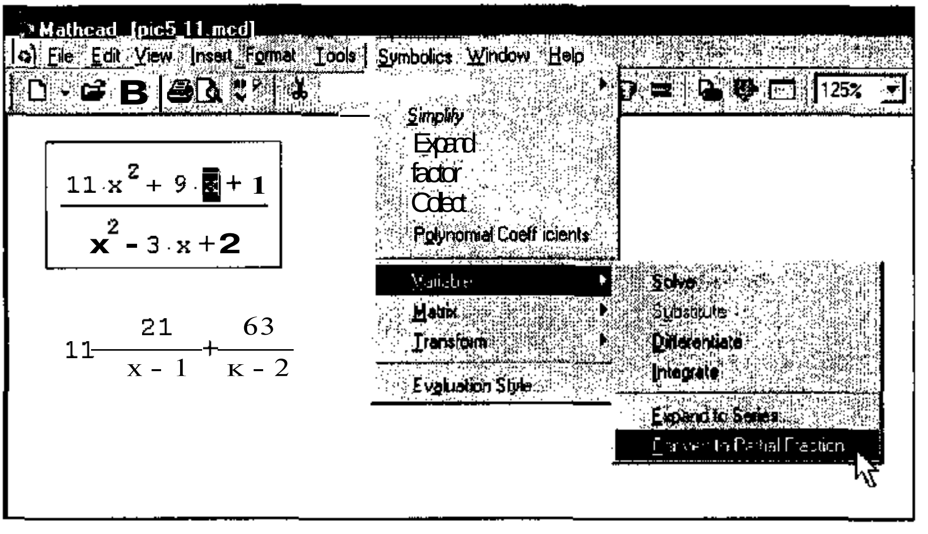


Рис. 5.11. Разложение сложной дроби на элементарные дроби

Листинг 5.11. Разложение на элементарные дроби

$$\frac{11\cdot x^2+9\cdot x+1}{x^2-3\cdot x+2}\ \text{convert}, \texttt{parfrac}, x \ \rightarrow 11-\frac{21}{(x-1)}+\frac{63}{(x-2)}$$

5.2.8. Подстановка переменной (Substitute)

Очень удобная возможность символьных вычислений — это операция подстановки значения переменной в выражение. При помощи меню подстановка производится следующим образом (рис. 5.12):

1. Выделите значение переменной, которое необходимо подставить в некоторое выражение. Значение переменной может быть любым выражением

относительно любых переменных (на рис. 5.12 в качестве подстановки взята самая первая строка документа).

2. Скопируйте значение переменной в буфер обмена, например, нажатием клавиш <Ctrl>+<C> или кнопки **Copy** (Копировать) на панели инструментов **Standard** (Стандартная).
3. Выделите в выражении, в которое требуется подставить значение из буфера обмена, переменную, которая будет заменяться (во второй строке на рис. 5.12 выделена переменная x).
4. Выполните команду **Symbolics / Variable / Substitute** (Символика / Переменная / Подставить).

Результат этих действий иллюстрируется нижней строкой в документе на рис. 5.12.

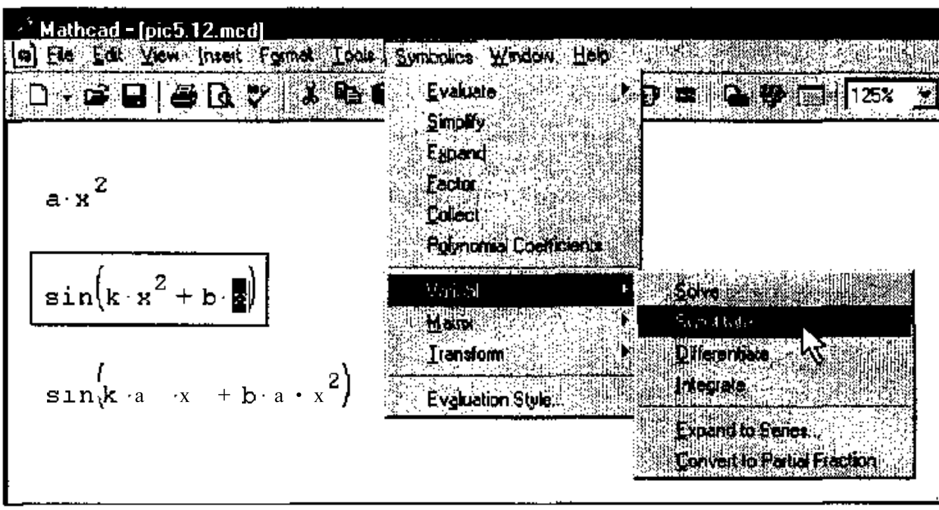


Рис. 5.12. Подстановка значения переменной

Для осуществления той же операции в совокупности с оператором символьного вывода используйте ключевое слово substitute, которое вставляется в документ одноименной кнопкой на панели **Symbolic** (Символика). После ключевого слова substitute необходимо ввести в местозаполнители логическое выражение, показывающее, какую именно переменную какой формулой следует заменить (листинг 5.12).

Листинг 5.12. Подстановка значения переменной

$$\sin\left(k\cdot x^2+b\cdot x\right) \text{ substitute}, k=a\cdot x^2 \rightarrow \sin\left(a\cdot x^4+b\cdot x\right)$$

5.2.9. Матричная алгебра

Символьный процессор Mathcad позволяет аналитически выполнять самые разные матричные вычисления. Помня о том, что большинство операций и встроенных функций осуществляются над матрицами точно так же, как над обычными числами, к матричным вычислениям можно применять рассмотренную выше команду упрощения **(Simplify)** из меню символьных вычислений.

Кроме того, имеется ряд специфичных матричных операций, которые можно организовать либо с помощью пункта меню **Symbolics / Matrix** (Символика / Матрица), либо с помощью нескольких кнопок на панели **Symbolic** (Символика), относящихся к матрицам (см. рис. 5.2). Это следующие матричные операции:

- Transpose (Транспонирование);
- **Invert** (Обратная матрица);
- **Determinant** (Определитель).

Выполняются действия с матрицами в той же последовательности, что и рассмотренные символьные операции со скалярными переменными. Перед их применением не забывайте выделить в выражении матрицу, к которой будет относиться операция.

5.3. Математический анализ

Наиболее ярким проявлением возможностей символьного процессора в Mathcad являются аналитические вычисления пределов, производных, интегралов и разложений в ряд, а также решение алгебраических уравнений. Все эти операции при выполнении их посредством меню **Symbolics** (Символика) находятся в его подменю **Variable** (Переменная). Соответственно, требуется предварительное выделение в выражении переменной, относительно которой будет совершаться операция. Для выделения переменной достаточно поместить ее между линиями ввода, но для большей наглядности лучше выделить ее черным цветом путем протаскивания указателя мыши через нужную часть выражения.

Все перечисленные операции можно осуществлять и при помощи оператора символьного вывода. Применение этого способа описывается в соответствующих главах *части III (за исключением разложения в ряд, освещенного в разд. 5.3.3).* Ниже в этом разделе приводятся сведения о проведении операций математического анализа посредством меню.

Примечание

Символьный поиск предела функции описан в *разд. "Вычислительные операторы"гл.3).*

5.3.1. Дифференцирование (Differentiate)

Чтобы аналитически продифференцировать выражение по некоторой переменной, выделите в нем эту переменную и выберите команду Symbolics / **Variable** /Differentiate (Символика/Переменная/Дифференцировать) (рис. 5.13).

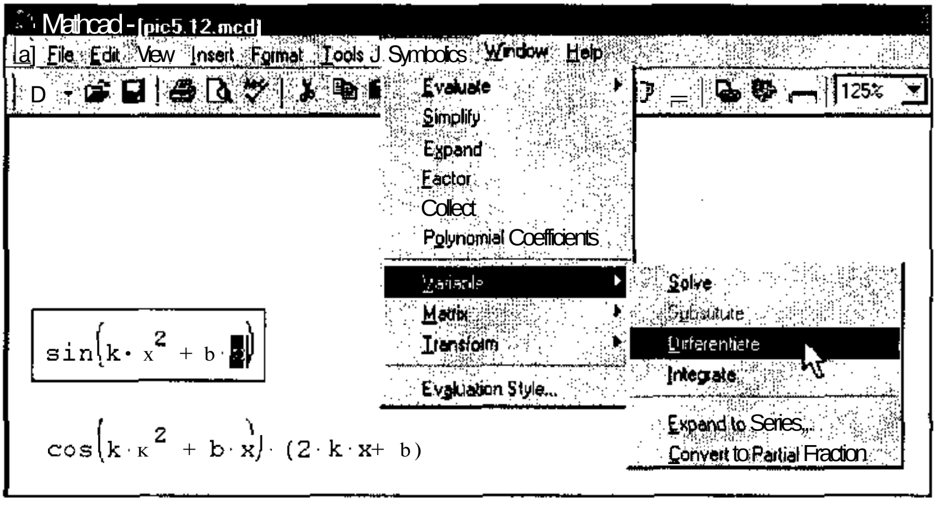


Рис. 5.13. Дифференцирование по переменной

В результате в следующей строке за выражением появится значение ее производной. Для того чтобы найти вторую производную, повторно примените эту последовательность действий, но уже к полученному результату дифференцирования. Так же находятся и производные высших порядков.

5.3.2. Интегрирование (Integrate)

Для вычисления неопределенного интеграла от некоторого выражения по определенной переменной выделите в выражении переменную и выполните команду Symbolics / Variable / Integrate (Символика / Переменная / Интегрировать) (рис. 5.14). Вычисленное аналитическое представление неопределенного интеграла появится ниже. При этом результат может содержать как встроенные в Mathcad функции *(см. гл. 10 и приложение 3),* так и другие спецфункции, которые нельзя непосредственно рассчитать в Mathcad, но символьный процессор "умеет" выдавать их в качестве результата некоторых символьных операций.

Примечание

Более подробную информацию о символьном решении алгебраических уравнений, дифференцировании и интегрировании (с применением оператора символьного вывода), включая вычисление производных высших порядков, определенных кратных интегралов, можно найти в *части III* этой книги *(см. гл. 7).*

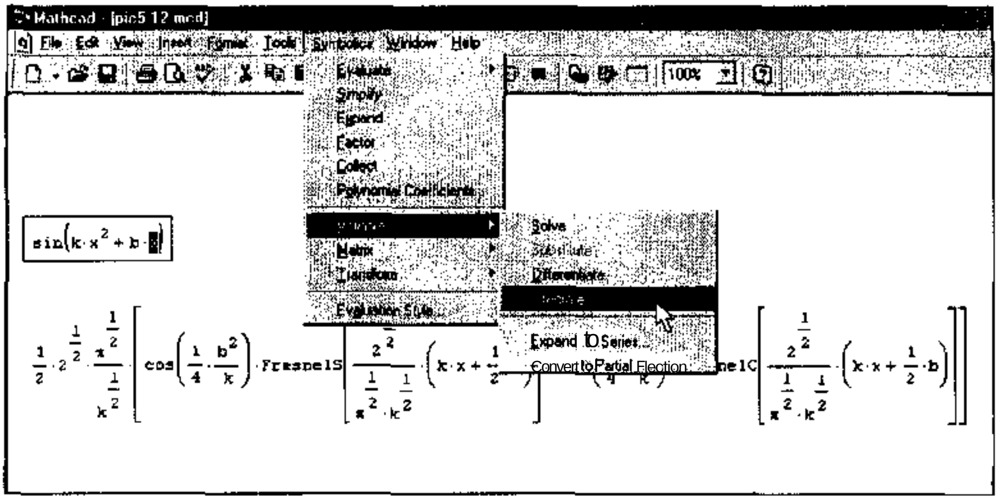


Рис. 5.14. Интегрирование по переменной

5.3.3. Разложение в ряд (Expand to Series)

С помощью символьного процессора Mathcad возможно получить разложение выражения в *ряд Тейлора* по любой переменной x в точке x=o, т. е. представить выражение в окрестности точки x суммой вида $a_0+a_1x+a_2x^2+a_3x^3+\ldots$ Здесь a_i — некоторые коэффициенты, не зависящие от x, но, возможно, являющиеся функциями других переменных, входящих в исходное выражение. Если выражение имеет в точке x=0 особенность, то соответствующее разложение называют *рядом Лорана*.

Чтобы разложить выражение в ряд:

1. Введите выражение.
2. Выделите значение переменной, по которой требуется получить разложение в ряд.
3. Выполните команду **Symbolics / Variable / Expand to Series** (Символика / Переменная / Разложить в ряд) (рис. 5.15).
4. В появившемся диалоговом окне (рис. 5.16) введите желаемый порядок аппроксимации **(Order of Approximation)** и нажмите кнопку OK.

Результат разложения появится под выражением (рис. 5.17).

Внимание!

Не забывайте, что разложение строится только в точке x=o. Чтобы получить разложение в другой точке x=a, можно, к примеру, подставить вместо переменной x значение x-a *(см. разд. 5.2.8).*

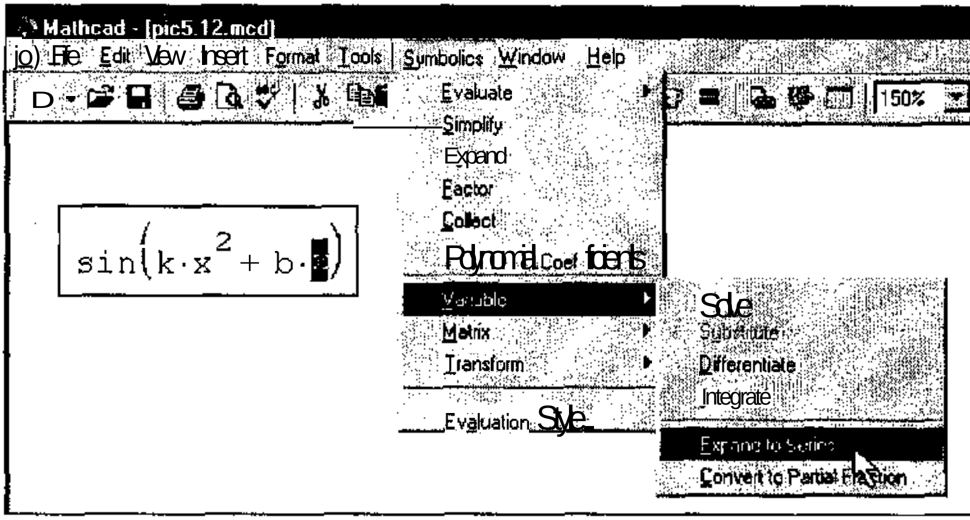


Рис. 5.15. Подготовка выражения для разложения в ряд по переменной x

Рис. 5.16. Разложение в ряд Тейлора

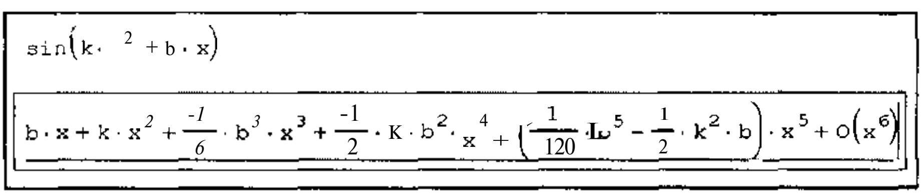


Рис. 5.17. Результат разложения в ряд Тейлора

Для разложения в ряд альтернативным способом, с помощью оператора символьного вывода, используйте ключевое слово series, вставляя его одноименной кнопкой панели Symbolic (Символика). После ключевого слова series, через запятую, указывается имя переменной, по которой производится разложение, и порядок аппроксимации (листинги 5.13 и 5.14). Сравнение функции и ее разложений в ряды с разными порядками аппроксимации (для k=b=1) иллюстрируется рис. 5.18. Видно, что разложение в ряд хорошо работает в окрестности точки x=0, а по мере удаления от нее все сильнее и сильнее отличается от функции.

Листинг 5.13. Разложение выражения в ряд с разным порядком аппроксимации

$$\sin\left(k\cdot x^2+b\cdot x\right)\ \text{series}, x, 2 \rightarrow b\cdot x$$

$$\sin\left(k\cdot x^2+b\cdot x\right)\ \text{series}, x, 3 \rightarrow k\cdot x^2+b\cdot x$$

$$\sin\left(k\cdot x^2+b\cdot x\right)\ \text{series}, x, 4 \rightarrow b\cdot x+k\cdot x^2-\frac{1}{6}\cdot b^3\cdot x^3$$

$$\sin\left(k\cdot x^2+b\cdot x\right)\ \text{series}, x, 5 \rightarrow b\cdot x+k\cdot x^2-\frac{1}{6}\cdot b^3\cdot x^3-\frac{1}{2}\cdot k\cdot b^2\cdot x^4$$

Листинг 5.14. Разложение выражения в ряд по разным переменным

$$\sin\left(k\cdot x^2+b\cdot x\right)\ \text{series}, k, 3 \rightarrow \sin(b\cdot x)+\cos(b\cdot x)\cdot x^2\cdot k-\frac{1}{2}\cdot\sin(b\cdot x)\cdot x^4\cdot k^2$$

$$\sin\left(k\cdot x^2+b\cdot x\right)\ \text{series}, b, 3 \rightarrow \sin\left(k\cdot x^2\right)+\cos\left(k\cdot x^2\right)\cdot x\cdot b-\frac{1}{2}\cdot\sin\left(k\cdot x^2\right)\cdot x^2\cdot b^2$$

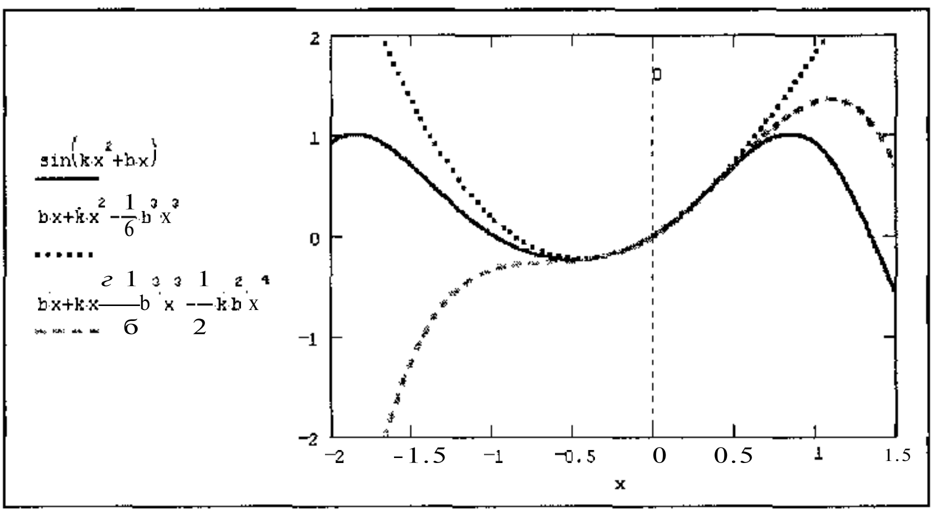


Рис. 5.18. функция и ее разложения в ряды Тейлора

5.3.4. Решение уравнений (Solve)

С помощью символьного процессора можно вычислить аналитически значение переменной, при котором выражение обращается в ноль. Для этого:

1. Введите выражение.
2. Выделите переменную, относительно которой будет решаться уравнение, приравнивающее выражение к нулю.

3. Выберите в меню Symbolics (Символика) пункт Variable/ Solve (Переменная / Решить) (рис. 5.19).

Примечание

Подробная информация о символьном решении алгебраических уравнений изложена в *части IU (см. гл. 8).* В частности, там рассказано о возможности решения систем уравнений и задании уравнений в привычной для нас форме логического равенства.

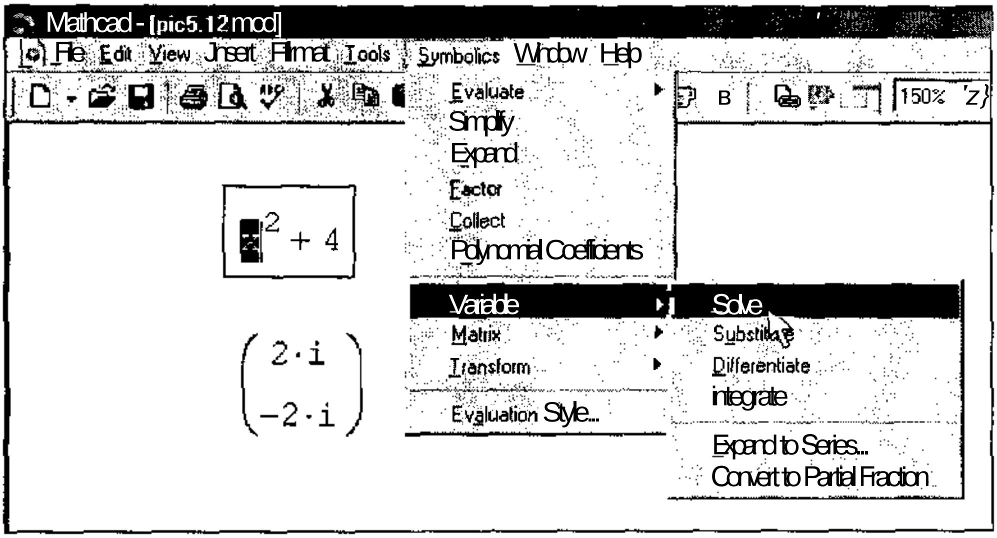


Рис. 5.19. Символьное решение уравнения

5.4. Интегральные преобразования

Интегральные преобразования, по определению, ставят в соответствие некоторой функции f(x) другую функцию от другого аргумента F(ω). Причем это соответствие f(x)→F(ω) задается интегральной зависимостью. Символьный процессор Mathcad позволяет осуществлять три вида интегральных преобразований функций — преобразование Фурье, Лапласа и Z-преобразование. Наряду с прямыми преобразованиями, имеется возможность совершать любое из этих трех обратных преобразований, т. е. F(ω)→f(x).

Выполняются все символьные интегральные преобразования аналогично уже рассмотренным операциям. Для вычисления преобразования выражения выделяется переменная, по которой будет осуществляться преобразование, и затем выбирается соответствующий пункт меню. Преобразования с применением оператора символьного вывода используются с одним из соответствующих ключевых слов, вслед за которым требуется указать имя нужной переменной.

Приведем примеры символьного расчета каждого из трех интегральных преобразований.

5.4.1. Преобразование Фурье (Fourier)

Преобразование Фурье представляет функцию `f(x)` в виде интеграла по гармоническим функциям, называемого *интегралом Фурье:*

$$F(\omega) = \int_{-\infty}^{\infty} f(x) \cdot \exp(-i\omega x)\,dx.$$

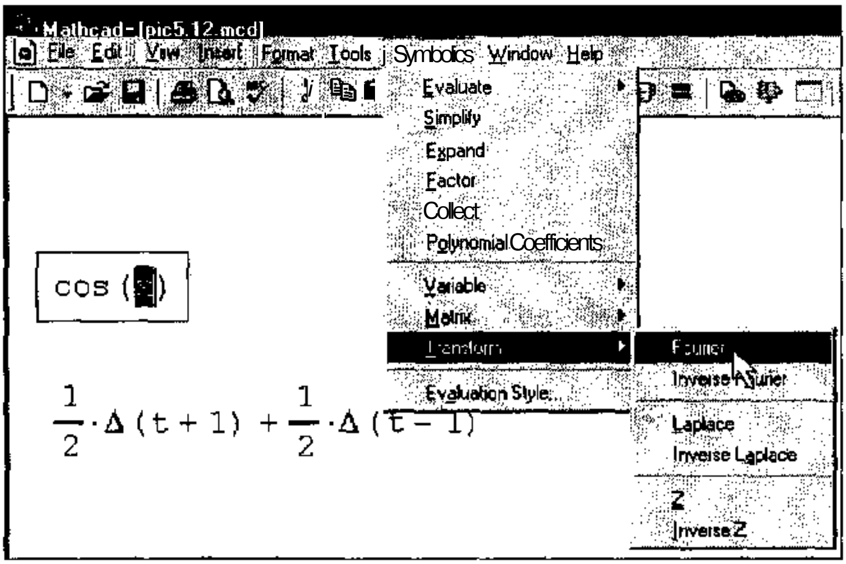


Рис. 5.20. Расчет Фурье-преобразования при помощи меню

Аналитический расчет преобразования Фурье при помощи меню показан на рис. 5.20. В листинге 5.15 приведены два примера вычисления прямого преобразования Фурье с применением ключевого слова `fourier` и оператора символьного вывода →. Листингом 5.16 иллюстрируется обратное преобразование Фурье одной из функций предыдущего листинга.

Примечание

В Mathcad преобразование Фурье можно вычислить и с помощью численного процессора, использующего популярный алгоритм БПФ *(см. разд. "Преобразование Фурье" гл. 14).*

Листинг 5.15. Прямое преобразование Фурье

$\cos(x)$ fourier, $x \to \pi\cdot\Delta(\omega-1)+\pi\cdot\Delta(\omega+1)$

x^2+4 fourier, $x \to -2\,\pi\cdot\Delta(2,\omega)+8\cdot\pi\cdot\Delta(\omega)$

Листинг 5.16. Обратное преобразование

$-2 \cdot я \cdot д(2, \omega) + 8 \cdot \pi \cdot \Delta(\omega)$

$+ 8 \cdot \pi \cdot \mathrm{Dirac}(\omega)$ invfourier

5.4.2. Преобразование Лапласа (Laplace)

Преобразованием Лапласа называют интеграл от f (x) следующего вида:

$$F(s) = \int_0^{\infty} f(x) \cdot \exp(-sx)\,dx.$$

Рассчитывается преобразование Лапласа совершенно аналогично Фурье-преобразованию *(см. предыдущий раздел).* Примеры преобразования Лапласа приведены в листинге 5.17.

Листинг 5.17. Прямое и обратное преобразование Лапласа

$x^2 + 4$ laplace, $x \rightarrow \frac{2}{s^3} + \frac{4}{s}$

$\frac{2}{s^3} + \frac{4}{s}$ invlaplace, $s \rightarrow t^2 + 4$

5.4.3. Z-преобразование (Z)

Z-преобразование функции f(x) определяется через бесконечную сумму следующего вида:

$$F(Z) = \sum_{n=0}^{\infty} f(n) \cdot z^{-n}.$$

Пример Z-преобразования приведен в листинге 5.18.

Листинг 5.18. Прямое и обратное Z-преобразование

$x^2 + 4$ ztrans, $x \rightarrow z \cdot \frac{(-7 \cdot z + 5 + 4 \cdot z^2)}{(z-1)^3}$

$z \cdot \frac{(-7 \cdot z + 5 + 4 \cdot z^2)}{(z-1)^3}$ invztrans, $z \rightarrow 4 + n^2$

5.5. Дополнительные возможности символьного процессора

Выше в этой главе были разобраны основные приемы символьных вычислений в Mathcad. Они, как правило, были показаны на простых примерах, которые иллюстрировали ту или иную символьную операцию. Тем не менее, при проведении разнообразных (и численных тоже) расчетов в Mathcad возможности символьного процессора можно использовать более эффективно. Отметим некоторые из них.

5.5.1. Применение функций пользователя

При проведении символьных вычислений с оператором символьного вывода функции пользователя и переменные, определенные ранее в документе Mathcad, воспринимаются символьным процессором корректно. Таким образом, имеется мощный аппарат включения символьных расчетов в программы пользователя. Примеры применения функции пользователя приведены в листингах 5.19 и 5.20. Сравните последние строчки этих листингов. Несмотря на их идентичность слева от знака символьного вывода, полученные результаты потличаются. Это связано с тем, что в листинге 5.20 предварительно переменной x присвоено значение 4. Поскольку значения переменных влияют на символьные вычисления, то результат учитывает подстановку вместо x числа 4.

Листинг 5.19. Функция пользователя в символьных вычислениях

$f(k,x) := \cos(k \cdot x) + 4 \cdot x^{2-k}$

$f(k,x)$ substitute, $k = \sqrt{x} \rightarrow \cos\left(x^{\frac{3}{2}}\right) + 4 \cdot x^{\left(2 - x^{\frac{1}{2}}\right)}$

$f(k,x)$ series, k, $2 \rightarrow 1 + 4 \cdot x^2 - 4 \cdot x^2 \cdot \ln(x) \cdot k$

Листинг 5.20. Значения переменных влияют на результат символьных вычислений

$f(k,x) := \cos(k \cdot x) + 4 \cdot x^{2-k}$

$x := 4$

$f(k,x)$ series, k, $2 \rightarrow 65 - 64 \cdot \ln(4) \cdot k$

Напротив, при осуществлении символьных операций через меню Symbolics (Символика), символьный процессор "не видит" ничего, кроме выражения, в пределах которого находятся линии ввода. Поэтому ни функции пользователя, ни предварительно определенные значения каких-либо переменных никак не влияют на вычисления.

Совет

Используйте меню Symbolics (Символика), если требуется "сиюминутно" провести некоторые аналитические действия с выражением и получить ответ в общем виде, не учитывающем текущие значения переменных, входящих в выражение.

5.5.2. Получение численного значения выражения

С помощью символьного процессора можно рассчитать численное значение выражения (действительное или комплексное). Иногда такой путь представляется более удобным, чем применение численного процессора {т. е. знака обычного равенства). Чтобы рассчитать значение некоторого выражения (рис. 5.21), выберите команду **Symbolics/ Evaluate/ Symbolically** (Символика / Вычислить / Символьно), либо пункт **Symbolics** / **Evaluate** / **Floating Point** (Символика/ Вычислить/ С плавающей точкой). В последнем случае Вам будет предложено с помощью диалога **Floating Point Evaluation** (Вычисления с плавающей точкой) задать точность вывода. В итоге применения данных команд Mathcad заменяет символьные результаты, где это возможно, значениями в виде чисел с плавающей точкой.

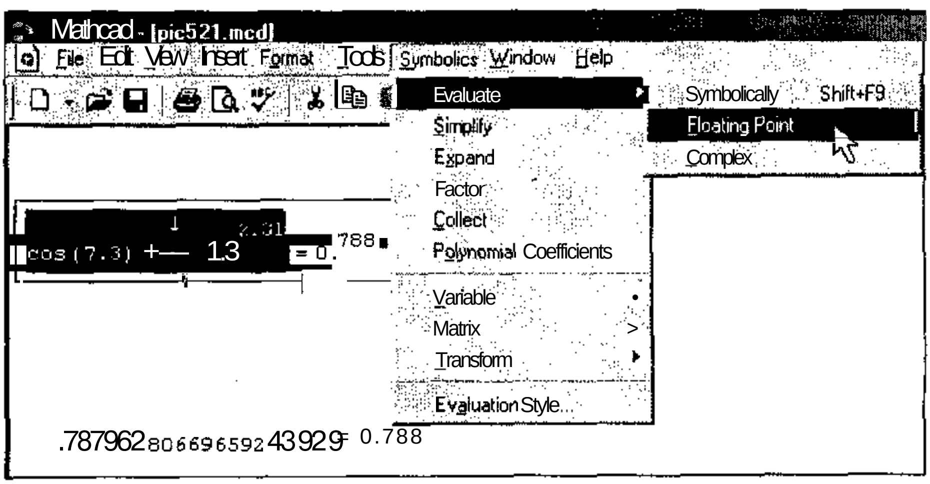


Рис. 5.21. Вычисление выражения с плавающей точкой

Еще один пункт меню **Symbolics** / **Evaluate** / **Complex** (Символика / Вычислить/ Комплексно) позволяет представить выражение в виде a+ь· i.

Аналогичные по действию ключевые слова `float` и complex можно использовать в документах, вводя их с панели **Symbolic** (Символика). Ключевое слово `float` применяется вместе со значением точности вывода результата с плавающей точкой (листинг 5.21). С помощью слова complex можно преобразовывать выражения как в символьном виде, так и с учетом численных значений, если они были ранее присвоены переменным (несколько примеров приведено в листинге 5.22).

Листинг 5.21. Вычисление выражения с плавающей точкой

$x := 3 \qquad k := 2.4$

$\cos(k\cdot x) + 4\cdot x^{2-k}$ float, 3 $\rightarrow 3.19$

$\cos(k\cdot x) + 4\cdot x^{2-k}$ float, 10 $\rightarrow 3.185927374$

$\cos(k\cdot x) + 4\cdot x^{2-k}$ float, 20 $\rightarrow 3.1859273744412716730$

Листинг 5.22. Комплексные преобразования выражений

e^{z+2i} complex $\rightarrow \exp(z)\cdot\cos(2) + i\cdot\exp(z)\cdot\sin(2)$

$4.2\cdot 2i^{1.8-3i}$ complex $\rightarrow 1193.452397093084618 + 1107.3477730509390980\cdot i$

$x := i$

$4\cdot x^{3}$ complex $\rightarrow -4\cdot i$

$4\cdot x^{3.1}$ complex $\rightarrow .62573786016092347604 - 3.9507533623805509048\cdot i$

5.5.3. Последовательности символьных команд

Символьные вычисления допускается проводить с применением цепочек из ключевых слов. Для этого ключевые слова, соответствующие последовательным символьным операциям, должны быть введены по очереди с панели **Symbolic** (Символика). Принцип организации цепочек символьных вычислений очень похож на применение встроенного языка программирования Mathcad *(см. следующую главу)*. Несколько примеров использования последовательности символьных операторов приводится в листингах 5.23 и 5.24.

Примечание

Последовательности символьных команд допускают введение дополнительных условий в расчеты, например таких, как ограничение на действительную или комплексную форму результата. Это делается с помощью ключевого слова assume. Более подробную информацию читатель найдет в справочной системе Mathcad.

Листинг 5.23. Фурье-преобразование, разложение в ряд с заданной точностью

e^{-x} fourier , x $\rightarrow 2\cdot\pi\cdot\Delta(\omega - i)$

e^{-x^2} | fourier , x | series , ω , 5 $\rightarrow \pi^{\frac{1}{2}} - \frac{1}{4}\cdot\pi^{\frac{1}{2}}\cdot\omega^2 + \frac{1}{32}\cdot\pi^{\frac{1}{2}}\cdot\omega^4$

e^{-x^2} | fourier , x | series , ω , 5 | float , 3 $\rightarrow 1.77 - .443\cdot\omega^2 + 5.54\cdot 10^{-2}\cdot\omega^4$

Листинг 5.24. Z-преобразование и разложение на простые дроби

$x^2 + 4$ ztrans , x $\rightarrow z\cdot\frac{(-7\cdot z + 5 + 4\cdot z^2)}{(z-1)^3}$

$x^2 + 4$ | ztrans , x | convert , parfrac , z $\rightarrow 4 + \frac{2}{(z-1)^3} + \frac{3}{(z-1)^2} + \frac{5}{(z-1)}$

ГЛАВА 6

Программирование

Создатели Mathcad изначально поставили перед собой такую задачу, чтобы дать возможность профессионалам-математикам, физикам и инженерам самостоятельно проводить сложные расчеты, не обращаясь за помощью к программистам. Несмотря на блестящее воплощение этих замыслов, выяснилось, что вовсе без программирования Mathcad серьезно теряет в своей силе, в основном, из-за недовольства пользователей, знакомых с техникой создания программ и желающих осуществить свои расчеты в привычном для себя программистском стиле. Вместо знакомых принципов программирования, пользователям старых версий Mathcad предлагалось комбинировать несколько специфичных встроенных функций и ранжированные переменные *(см. разд. 6.1).*

Последние версии Mathcad имеют не очень мошный, но весьма элегантный собственный язык *(см. разд. 6.2).* С одной стороны, он дает возможность программисту эффективно применять программный код в документах Mathcad. С другой, простота и интуитивность языка программирования позволяет быстро ему обучиться. Наконец, программные модули внутри документа Mathcad сочетают в себе и обособленность (поэтому их легко отличить от остальных формул), и простоту смыслового восприятия.

Несмотря на небольшое число операторов, язык программирования Mathcad позволяет решать самые различные, в том числе и довольно сложные, задачи и является серьезным подспорьем для расчетов *(см. разд. 6.3).*

6.1. Программирование без программирования

В ранних версиях Mathcad встроенного языка программирования не было. Чтобы применять привычные операции проверки условий и организовывать циклы, приходилось изобретать причудливую смесь из встроенных функций

6 Зак 984

условия if (листинг 6.1) и until и комбинаций ранжированных переменных (листинг 6.2).

Примечание

В связи с устоявшимися традициями применения языка программирования функцию until настоятельно не рекомендуется использовать в дальнейшей работе (тем не менее, она действует в Mathcad 11, но отнесена к устаревшим функциям).

Листинг 6.1. Функция условия

```
f(x) := if (x< 0 , "negative" , "positive" )
f(1) = "positive"
f(-1) = "negative"
```

Листинг 6.2. Организация цикла при помощи ранжированной переменной

```
i:=0..  10
        .2
x_i := i
```

Фактически, использование ранжированных переменных — мощный аппарат Mathcad, похожий на применение циклов в программировании. В подавляющем большинстве случаев намного удобнее организовать циклы (в том числе вложенные) с помощью ранжированных переменных, чем заниматься для этого программированием. Полезнее освоить технику, связанную с ранжированными переменными, векторами и матрицами, поскольку на ней основаны главные принципы расчетов в Mathcad, в частности подготовка графиков. *(Более подробную информацию о ранжированных переменных и связанными с ними возможностями можно получить в гл. 4.)*

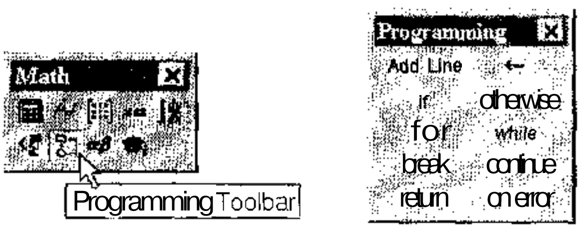


Рис. 6.1. Панель инструментов Programming

6.2. Язык программирования Mathcad

Для вставки программного кода в документы в Mathcad имеется специальная панель инструментов **Programming** (Программирование), которую можно вызвать на экран нажатием кнопки **Programming Toolbar** на панели **Math** (Математика), как показано на рис. 6.1. Большинство кнопок этой панели выполнено в виде текстового представления операторов программирования, поэтому их смысл легко понятен.

Изложим последовательно основные составные элементы языка программирования Mathcad и рассмотрим примеры его использования.

6.2.1. Что такое программа?

Основными инструментами работы в Mathcad являются математические выражения, переменные и функции. Нередко записать формулу, использующую ту или иную внутреннюю логику (например возвращение различных значений в зависимости от условий), в одну строку не удается. Назначение программных модулей как раз и заключается в определении выражений, переменных и функций в несколько строк, часто с применением специфических программных операторов.

Сравните определение функции f(x) из листинга 6.1 с определением f(x) с помощью программного модуля (листинг 6.3).

Листинг 6.3, Функция условия, определенная с помощью программы

```
f(x) := | "negative"  if x < 0
        | "positive"  if x > 0
        | "zero"  otherwise
f(1) → "positive"
f(-1) = "negative"
f(0) = "zero"
```

Несмотря на принципиальную эквивалентность определения функций и переменных через встроенные функции Mathcad или программные модули, программирование имеет ряд существенных преимуществ, которые в ряде случаев делают документ более простым и читаемым:

- возможность применения циклов и условных операторов;
- простота создания функций и переменных, требующих нескольких простых шагов (как в примере листинга 6.3);
- возможность создания функций, содержащих закрытый для остального документа код, включая преимущества использования локальных переменных и обработку исключительных ситуаций (ошибок).

Как видно из листинга 6.3, программный модуль обозначается в Mathcad вертикальной чертой, справа от которой последовательно записываются операторы языка программирования.

6.2.2. Создание программы (Add Line)

Чтобы создать программный модуль, например, представленный в предыдущем разделе (см. листинг 6.3):

1. Введите часть выражения, которая будет находиться слева от знака присваивания и сам знак присваивания. В нашем примере это имя функции f(x).
2. При необходимости вызовите на экран панель инструментов Programming (Программирование) (см. рис. 6.1).
3. Нажмите на этой панели кнопку Add Line (Добавить линию).
4. Если приблизительно известно, сколько строк кода будет содержать программа, можно создать нужное количество линий повторным нажатием кнопки Add Line (Добавить линию) соответствующее число раз (на рис. 6.2 показан результат трехкратного нажатия).
5. В появившиеся местозаполнители введите желаемый программный код, используя программные операторы. В рассматриваемом примере в каждый местозаполнитель вводится строка, например, "positive" (рис. 6.3), затем нажимается кнопка If (Если) на панели Programming (Программирование) и в возникший местозаполнитель вводится выражение x>0 (рис. 6.4).

После того как программный модуль полностью определен и ни один местозаполнитель не остался пустым, функция может использоваться обычным образом, как в численных, так и в символьных расчетах.

Внимание!

Не вводите с клавиатуры имена программных операторов. Для их вставки можно применять лишь сочетания клавиш, которые приведены в тексте всплывающей подсказки (рис. 6.2 и 6.3).

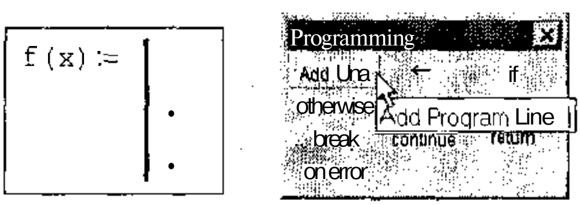


Рис. 6.2. Начало создания программного модуля

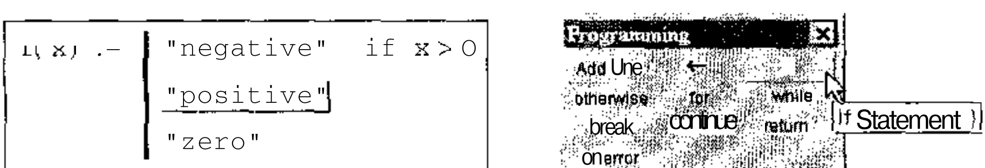


Рис. 6.3. Вставка программного оператора

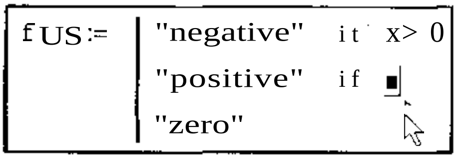


Рис. 6.4. Вставка условия в программу

6.2.3. Разработка программы

Вставить строку программного кода в уже созданную программу можно в любой момент с помощью той же самой кнопки Add Line (Добавить линию). Для этого следует предварительно поместить на нужное место внутри программного модуля линии ввода. Например, расположение линии ввода на строке, показанной на рис. 6.5, приведет к появлению новой линии с местозаполнителем перед этой строкой. Если передвинуть вертикальную линию ввода из начала строки (как на рис. 6.5) в ее конец, то новая линия появится после строки. Если выделить строку не целиком, а лишь некоторую ее часть (рис. 6.6), то это повлияет на положение в программе новой строки кода (результат нажатия кнопки Add Line показан на рис. 6.7).

Совет

Не забывайте, что для желаемого размещения линий ввода внутри формулы можно использовать не только мышь и клавиши со стрелками, но и пробел. С помощью последовательных нажатий пробела линии ввода "захватывают" разные части формулы.

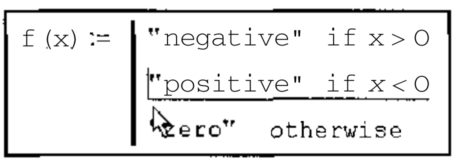


Рис. 6.5. Вставка новой строки в существующую программу

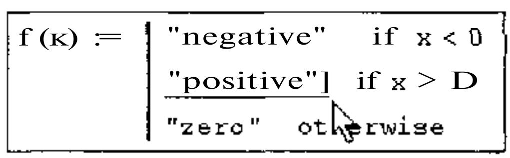


Рис. 6.6. Положение линий ввода влияет на положение новой линии

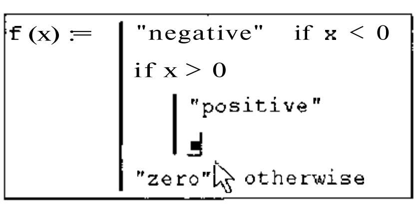


Рис. 6.7. Результат вставки новой линии в программу (из положения рис. 6.6)

Зачем может потребоваться вставка новой линии в положение, показанное на рис. 6.7? Новая вертикальная черта с двумя линиями выделяет фрагмент программы, который относится к условию x>0, находящемуся в его заголовке. Пример возможного дальнейшего программирования показан в листинге 6.4.

Листинг 6.4. Пример усовершенствования программы

```
f(x) := | "negative"  if x < 0
        | if x > 0
        |    | "positive"
        |    | "big positive"  if x > 1000
        | "zero"  otherwise
f(1) = "positive"
f(10^5) = "big positive"
```

В режиме выполнения программы, а это происходит при любой попытке вычислить f(x), выполняется последовательно каждая строка кода. Например, в предпоследней строке листинга 6.4 вычисляется f(1). Рассмотрим работу каждой строки кода этого листинга.

1. Поскольку x=1, то условие x<o не выполнено, и в первой строке ничего не происходит.
2. Условие второй строки x>o выполнено, поэтому выполняются обе следующие строки, объединенные короткой вертикальной чертой в общий фрагмент.
3. Функции f(x) присваивается значение f(x)="positive".
4. Условие x>юоо не выполнено, поэтому значение "big positive" не присваивается f(x), она так и остается равной строке "positive".
5. Последняя строка не выполняется, т. к. одно из условий (x>o) оказалось истинным, и оператор otherwise (т. е. "иначе") не понадобился.

Таким образом, основной принцип создания программных модулей заключается в правильном расположении строк кода. Ориентироваться в их действии довольно легко, т. к. фрагменты кода одного уровня сгруппированы в программе с помощью вертикальных черт.

6.2.4. Локальное присваивание (←)

Язык программирования Mathcad не был бы эффективным, если бы не позволял создавать внутри программных модулей локальные переменные, которые "не видны" извне, из других частей документа. Присваивание в пределах программ, в отличие от документов Mathcad, производится с помощью оператора **Local Definition** (Локальное присваивание), который вставляется нажатием кнопки с изображением стрелки ← на панели **Programming** (Программирование).

Внимание!

Ни оператор присваивания :=, ни оператор вывода = в пределах программ не применяются.

Локальное присваивание иллюстрируется листингом 6.5. Переменная z существует только внутри программы, выделенной вертикальной чертой. Из других мест документа получить ее значение невозможно.

Листинг 6.5. Локальное присваивание в программе

$$f(x) := \left| \begin{array}{l} z \leftarrow 4 \\ z + x \end{array} \right.$$

$f(1) = 5$

6.2.5. Условные операторы *(if, otherwise)*

Действие условного оператора if состоит из двух частей. Сначала проверяется логическое выражение (условие) справа от него. Если оно истинно, выполняется выражение слева от оператора if. Если ложно — ничего не происходит, а выполнение программы продолжается переходом к ее следующей строке. Вставить условный оператор в программу можно следующим образом (см. рис. 6.8):

1. Если необходимо, введите левую часть выражения и оператор присваивания.
2. Создайте новую строку программного кода, нажав на панели **Programming** (Программирование) кнопку **Add Line** (Добавить строку).
3. Нажмите кнопку условного оператора if.
4. Справа от оператора if введите условие. Пользуйтесь логическими операторами, вводя их с панели **Boolean** (Булевы операторы).
5. Выражение, которое должно выполняться, если условие истинно, введите слева от оператора if.
6. Если в программе предусматриваются дополнительные условия, добавьте в программу еще одну строку нажатием кнопки **Add Line** и введите их таким же образом, используя оператор if или otherwise.

Оператор otherwise используется совместно с одним или несколькими условными операторами if и указывает на выражение, которое будет выполняться, если ни одно из условий не оказалось истинным. Примеры использования операторов if и otherwise приведены в предыдущих разделах (см. листинги 6.3 и 6.4).

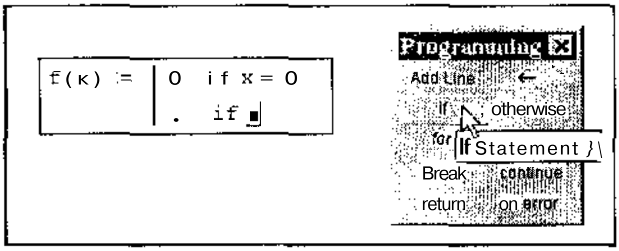


Рис. 6.8. Вставка условного оператора

6.2.6. Операторы цикла (for, while, break, continue)

В языке программирования Mathcad имеются два оператора цикла: for и while. Первый из них дает возможность организовать цикл по некоторой переменной, заставляя ее пробегать некоторый диапазон значений. Второй создает цикл с выходом из него по некоторому логическому условию. Чтобы вставить в программный модуль оператор цикла:

1. Создайте в программном модуле новую линию.
2. Вставьте один из операторов цикла for или while нажатием одноименной кнопки на панели **Programming** (Программирование).
3. Если выбран оператор for (рис. 6.9), то вставьте в соответствующие местозаполнители имя переменной и диапазон ее значений (листинги 6.6 и 6.7), а если while — то логическое выражение, при нарушении которого должен осуществляться выход из цикла (листинг 6.8).

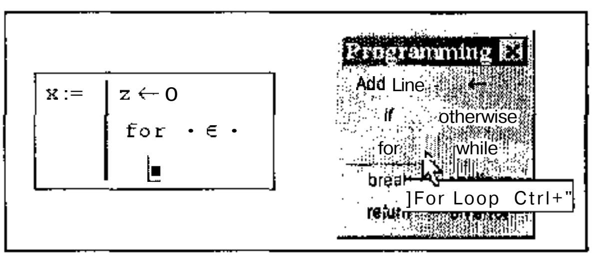


Рис. 6.9. Вставка оператора цикла

4. В нижний местозаполнитель введите тело цикла, т. е. выражения, которые должны выполняться циклически.

При необходимости дополните программу другими строками и введите в них нужный код.

Примечание

Диапазон значений переменной в условии цикла `for` можно задать как с помощью диапазона ранжированной переменной (листинг 6.6), так и с помощью вектора (листинг 6.7).

Листинг 6.6. Оператор цикла `for` с ранжированной переменной

```
x := | z ← 0
     | for  i ∈ 0.. 5
     |   z ← z + i
x = 15
```

Листинг 6.7. Оператор цикла `for` с вектором

```
x := | z ← 0
     | for  i ∈ ( 1  2  3 )
     |   z ← z + i
x = 6
```

Листинг 6.8. Оператор цикла `while`

```
x := | z ← 0
     | while  z < 10
     |   z ← z + 1
x = 10
```

Иногда необходимо досрочно завершить цикл, т. е. не по условию в его заголовке, а в некоторой строке в теле цикла. Для этого предназначен оператор break. Модификации листингов 6.6 и 6.8 с прерыванием цикла оператором break приведены в листингах 6.9 и 6.10, соответственно. Например в листинге 6.9, как только значение переменной цикла i достигает 2, цикл, благодаря оператору break в последней строке программного модуля, прерывается. Соответственно, значение переменной x остается равным 0+1+2=3.

Листинг 6.9. Оператор `break` внутри цикла `for`

```
x := | z ← 0
     | for  i ∈ 0 .. 5
     |    | z ← z + i
     |    | break  if  i = 2
x = 3
```

Листинг 6.10. Оператор break внутри цикла while

```
x := | z ← 0
     | while  z < 10
     |     | z ← z + 1
     |     | break   if  z > 5
x = 6
```

Примечание

Чтобы четче обозначить границы завершения тела цикла, в его конце может использоваться дополнительная строка с оператором `continue`, который вводится одноименной кнопкой панели **Programming.** Примеры, модернизирующие листинги 6.7 и 6.8, иллюстрируются листингами 6.11 и 6.12, соответственно. Как видно, на результат программы наличие оператора `continue` не влияет.

Листинг 6.11. Оператор continue в конце цикла while

```
x := | z ← 0
     | while  z < 10
     |     | z ← z + 1
     |     | continue
x = 10
```

Листинг 6.12. Оператор continue в конце цикла for

```
x := | z ← 0
     | for  i ∈ { 1  2  3 }
     |     | z ← z + i
     |     | continue
x = 6
```

6.2.7. Возврат значения *(return)*

Если для определения переменной или функции применяется программный модуль, то его строки исполняются последовательно при вычислении в документе этой переменной или функции. Соответственно, по мере выполнения программы рассчитываемый результат претерпевает изменения. В качестве окончательного результата выдается последнее присвоенное значение (примеры можно найти в листингах 6.3—6.12). Чтобы подчеркнуть возврат программным модулем определенного значения, можно взять за правило делать это в последней строке программного модуля (листинг 6.13).

Листинг 6.13. Возврат значения обозначен явно в последней строке программы

```
f(x) := | y ← x²
        | z ← y + 1
        | 2
f(2) = 5
```

Вместе с тем, можно прервать выполнение программы в любой ее точке (например с помощью условного оператора) и выдать некоторое значение, применив оператор return. В этом случае при выполнении указанного условия (листинг 6.14) значение, введенное в местозаполнитель после return, возвращается в качестве результата, а никакой другой код больше не выполняется. Вставляется в программу оператор return с помощью одноименной кнопки панели **Programming** (Программирование).

Листинг 6.14. Применение оператора return

```
f(x) := | z ← x²
        | return  "zero"  if x = 0
        | return  "i"  if x = i
        | z
f(-1) = 1
f(2) = 4
f(0) = "zero"
f(i) = "i"
```

6.2.8. Перехват ошибок *(on error)*

Программирование в Mathcad позволяет осуществлять дополнительную обработку ошибок. Если пользователь предполагает, что выполнение кода в каком-либо месте программного модуля способно вызвать ошибку (например деление на ноль), то эту ошибку можно перехватить с помощью оператора on error. Чтобы вставить его в программу, надо поместить линии ввода в ней в нужное положение и нажать кнопку с именем оператора on error на панели **Programming** (Программирование). В результате появится строка с двумя местозаполнителями и оператором on error посередине (рис. 6.10).

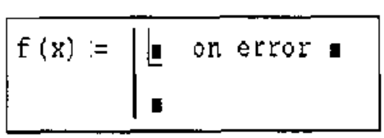


Рис. 6.10. Вставка оператора перехода по ошибке

В правом местозаполнителе следует ввести выражение, которое должно выполняться в данной строке программы. В левом — выражение, которое будет выполнено вместо правого выражения, если при выполнении последнего возникнет ошибка. Приведем пример применения оператора on error (листинг 6.15) в программном модуле, который рассчитывает функцию обратного числа значению n. Если n≠0, то и присвоенное значение z≠0, поэтому в последней строке программы выполняется правое выражение расчета 1/z. Так происходит при расчете f(-2). Если попытаться вычислить f(0) как в конце листинга, то выполнение программы, заложенной в f(n), вызовет ошибку деления на ноль в последней строке программы. Соответственно, вместо выражения справа от оператора on error будет выполнено левое выражение, присваивающее функции f(n) строковое значение "user error: cannot divide by zero" (пользовательская ошибка: деление на ноль невозможно). Конечно, этой строке можно присвоить и текст на русском языке.

Листинг 6.15. Перехват ошибки деления на ноль

```
f(n) := | z ← n
        | "user error: can't divide by zero"   on error   1/z

f(-2) → -1/2

f(0) = "user error: can't divide by zero"
```

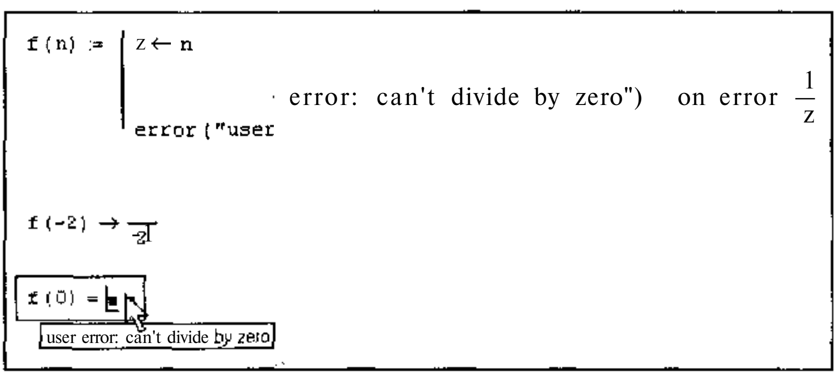


Рис. 6.11. Перехват ошибки деления на ноль

Оператор перехвата ошибок удобно применять в комбинации со встроенной функцией error(S). Она приводит к генерации ошибки в обычной для Mathcad форме с сообщением s. Пример усовершенствования листинга 6.15 для такого стиля обработки ошибки деления на ноль показан на рис. 6.11.

Обратите внимание, что сделанные изменения свелись к помещению текста сообщения об ошибке в аргумент функции `error`.

6.3. Примеры программирования

Рассмотрим два простых примера использования программных модулей в Mathcad для численных (листинг 6.16) и символьных (листинг 6.17) расчетов. В двух приведенных листингах используется большинство операторов, рассмотренных в данной главе. Когда вы станете сами разрабатывать свои программные модули в Mathcad, не забывайте, что операторы программирования вставляются в текст программы с помощью кнопок панели инструментов **Programming** (Программирование). Их имена нельзя ни в коем случае просто набивать на клавиатуре, поскольку они не будут восприняты Mathcad корректно.

Примечание

С помощью средств программирования можно создавать намного более сложные программы. Несколько примеров достаточно эффективного применения программирования вы найдете в *разд. "Фазовый портрет динамической системы" гл. 11 v\ разд. "Разностные схемы для ОДУ" гл. 12.*

Листинг 6.16. Программирование в численных расчетах

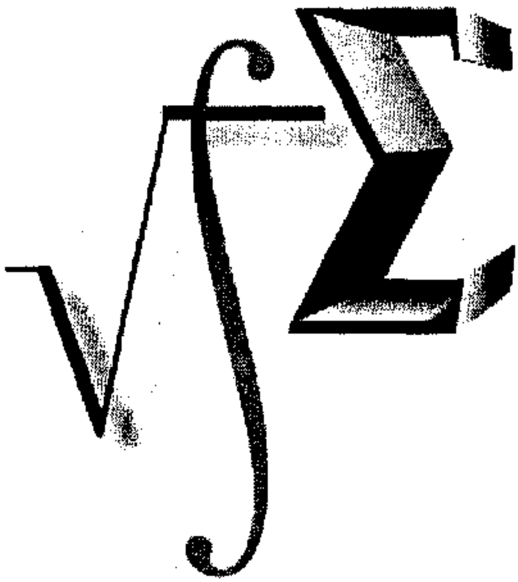
```
f(n) := | return  -99  if n < 0
        | z ← 1
        | for i ∈ 1 .. n
        |   z ← z · i
        | z
f(-2) → -99
f(0) = 0
f(3.9) = 6
f(3) = 6
f(10) = 3.629 × 10^6
```

Листинг 6.17. Программирование в символьных расчетах

$$f(n) := \left|\begin{array}{ll} -1 & \text{if } n < 0 \\ x \text{ on error } \dfrac{d^n}{dx^n}x & \text{otherwise} \end{array}\right.$$

$$f(1) \to 10 \cdot x^9$$

```
f(10)  → 3628800
f(-3)  → -1
f(2.1) → x
```



ЧАСТЬ III

ЧИСЛЕННЫЕ МЕТОДЫ

ГЛАВА 7

Интегрирование и дифференцирование

В этой главе рассматриваются основные математические операции, к которым мы отнесли численное дифференцирование и интегрирование функций.

Интегрирование *(см. разд. 7.1)* и дифференцирование *(см. разд. 7.2)* — самые простые, с вычислительной точки зрения, операции, реализованные в Mathcad в виде операторов. Тем не менее, если расчеты выполняются с помощью вычислительного процессора, необходимо хорошо представлять себе особенности численных алгоритмов, действие которых остается для пользователя "за кадром". В тех же разделах *(см. 7.1 и 7.2)* упоминается и об особенностях символьных операций интегрирования и дифференцирования.

7.1. Интегрирование

Интегрирование в Mathcad реализовано в виде вычислительного оператора. Допускается вычислять интегралы от скалярных функций в пределах интегрирования, которые также должны быть скалярами. Несмотря на то что пределы интегрирования обязаны быть действительными, подынтегральная функция может иметь и комплексные значения, поэтому и значение интеграла может быть комплексным. Если пределы интегрирования имеют размерность *(см. разд. "Размерные переменные" гл. 4),* то она должна быть одной и той же для обоих пределов.

7.1.1. Операторы интегрирования

Интегрирование, дифференцирование, как и множество других математических действий, устроено в Mathcad по принципу "как пишется, так и вводится". Чтобы вычислить определенный интеграл, следует напечатать его обычную математическую форму в документе. Делается это с помощью панели Calculus (Вычисления) нажатием кнопки со значком интеграла или вводом с клавиатуры сочетания клавиш <Shift>+<7> (или символа "&"). Появится символ интеграла с несколькими местозаполнителями (рис. 7.1), в которые нужно ввести нижний и верхний интервалы интегрирования, подынтегральную функцию и переменную интегрирования.

Внимание!

Можно вычислять интегралы с одним или обоими бесконечными пределами. Для этого на месте соответствующего предела введите символ бесконечности, воспользовавшись, например, той же самой панелью **Calculus** (Вычисления). Чтобы ввести -∞ (минус бесконечность), добавьте знак минус к символу бесконечности, как к обычному числу.

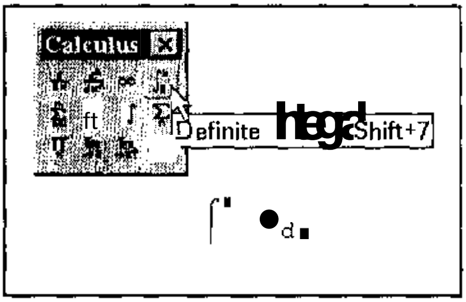


Рис. 7.1. Оператор интегрирования

Чтобы получить результат интегрирования, следует ввести знак равенства или символьного равенства. В первом случае интегрирование будет проведено численным методом, во втором — в случае успеха, будет найдено точное значение интеграла с помощью символьного процессора Mathcad. Эти два способа иллюстрирует листинг 7.1. Конечно, символьное интегрирование возможно только для небольшого круга несложных подынтегральных функций.

Листинг 7.1. Численное и символьное вычисление определенного интеграла

$$\int_0^{\pi} \sin(x)\, dx = 2$$

$$\int_0^{\pi} \sin(x)\, dx \rightarrow 2$$

Примечание

Подынтегральная функция может зависеть от любого количества переменных. Именно для того чтобы указать, по какой переменной Mathcad следует вычислять интеграл, и нужно вводить ее имя в соответствующий местозаполнитель.

Помните, что для численного интегрирования по одной из переменных предварительно следует задать значение остальных переменных, от которых зависит подынтегральная функция и для которых вы намерены вычислить интеграл (листинг 7.2).

Листинг 7.2. Интегрирование функции двух переменных по разным переменным

$$\alpha := 2$$

$$\int_0^{\pi} \alpha \cdot \sin(x)\ dx = 4$$

$$x := 1$$

$$\int_1^{10} \alpha \cdot \sin(x)\ d\alpha = 42.074$$

Примечание

Оператор интегрирования может использоваться точно так же, как и другие операторы: для определения функций, в циклах и при вычислении ранжированных переменных. Пример присваивания пользовательской функции g(x) значения определенного интеграла и вычисления нескольких ее значений приведен в листинге 7.3.

Листинг 7.3. Использование оператора интегрирования в функции пользователя

$$g(\alpha) := \int \alpha \cdot \sin(x)\ dx$$

$$i := 1 .. 5$$

$g(i) =$

2
4
6
8
10

7.1.2. Об алгоритмах интегрирования

Результат численного интегрирования — это не точное, а приближенное значение интеграла, определенное с погрешностью, которая зависит от встроенной константы `TOL`. Чем она меньше, тем с лучшей точностью будет найден интеграл, но и тем больше времени будет затрачено на расчеты. По умолчанию TOL=0.001. Для того чтобы ускорить вычисления, можно установить меньшее значение TOL.

Совет

Если скорость расчетов имеет для Вас принципиальное значение, например при многократном вычислении интеграла внутри цикла, проявите осторожность, выбирая значение точности. Обязательно поэкспериментируйте на тестовом примере с характерной для Ваших расчетов подынтегральной функцией. Посмотрите, как уменьшение константы TOL сказывается на погрешности интегрирования, вычислив интеграл для разных ее значений и выбрав оптимальное, исходя из соотношения точность / скорость вычислений.

Отдавайте себе отчет в том, что при вводе в редакторе Mathcad оператора численного интегрирования, Вы, фактически, создаете самую настоящую программу. Например, программой является первая строка листинга 7.1, просто большая часть ее скрыта от Вашего взора разработчиками компании MathSoft. В большинстве случаев об этом не приходится специально задумываться, можно полностью положиться на Mathcad. Но иногда может потребоваться умение управлять параметрами этой программы, как мы уже рассмотрели на примере выбора константы TOL. Кроме нее, пользователь имеет возможность выбирать сам алгоритм численного интегрирования. Для этого:

1. Щелкните правой кнопкой мыши в любом месте на левой части вычисляемого интеграла.
2. В появившемся контекстном меню выберите один из четырех численных алгоритмов (рис. 7.2).

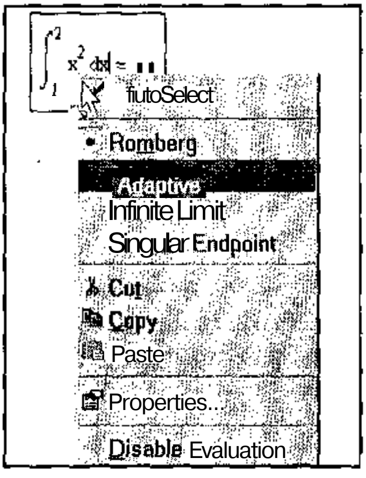


Рис. 7.2. Выбор алгоритма численного интегрирования

Обратите внимание, что, перед тем как один из алгоритмов выбран впервые, как показано на рис. 7.2, флажок проверки в контекстном меню установлен возле пункта **AutoSelect** (Автоматический выбор). Это означает, что алгоритм определяется Mathcad, исходя из анализа пределов интегрирования и особенностей подынтегральной функции. Как только один из алгоритмов выбран, этот флажок сбрасывается, а избранный алгоритм отмечается точкой.

Разработчиками Mathcad И запрограммированы четыре численных метода интегрирования:

- **Romberg** (Ромберга) — для большинства функций, не содержащих особенностей;
- **Adaptive** (Адаптивный) — для функций, быстро меняющихся на интервале интегрирования;
- **Infinite Limit** (Бесконечный предел) — для интегралов с бесконечными пределами ();
- **Singular Endpoint** (Сингулярная граница) — для интегралов с сингулярностью на конце. Модифицированный алгоритм Ромберга для функций, не определенных на одном или обоих концах интервала интегрирования.

Старайтесь все-таки оставить выбор численного метода за Mathcad, установив флажок **AutoSelect** (Автоматический выбор) в контекстном меню. Попробовать другой метод можно, например, чтобы сравнить результаты расчетов в специфических случаях, когда у Вас закрадываются сомнения в их правильности.

Если подынтегральная функция "хорошая", т. е. не меняется на интервале интегрирования слишком быстро и не обращается на нем в бесконечность, то численное решение интеграла не принесет никаких неприятных сюрпризов. Приведем основные идеи итерационного алгоритма Ромберга, который применяется для большинства таких функций.

- Сначала строится несколько интерполирующих полиномов, которые заменяют на интервале интегрирования подынтегральную функцию `f(x)`. В качестве первой итерации полиномы вычисляются по 1,2 и 4 интервалам. Например, первый полином, построенный по 1 интервалу, — это просто прямая линия, проведенная через две граничные точки интервала интегрирования, второй — квадратичная парабола и т. д.
- Интеграл от каждого полинома с известными коэффициентами легко вычисляется аналитически. Таким образом, определяется последовательность интегралов от интерполирующих полиномов: I_1, I_2, I_4, ... Например, по правилу трапеций $I_1=(b-a)\cdot(f(a)+f(b))/2$ и т. д.

- Из-за интерполяции по разному числу точек вычисленные интегралы I_1, I_2, ... несколько отличаются друг от друга. Причем, чем больше точек используется для интерполяции, тем интеграл от интерполяционного полинома ближе к искомому интегралу, стремясь к нему в пределе бесконечного числа точек. Поэтому определенным образом осуществляется экстраполяция последовательности I_1, I_2, I_4, ... до нулевой ширины элементарного интервала. Результат этой экстраполяции J принимается за приближение к вычисляемому интегралу.
- Осуществляется переход к новой итерации с помощью еще более частого разбиения интервала интегрирования, добавления нового члена последовательности интерполирующих полиномов и вычисления нового (N-го) приближения Ромберга J^N.
- Чем больше количество точек интерполяции, тем ближе очередное приближение Ромберга к вычисляемому интегралу и, соответственно, тем меньше оно отличается от приближения предыдущей итерации. Как только разница между двумя последними итерациями $|J^N-J^{N-1}|$ становится меньше погрешности TOL или меньше $TOL \cdot |J^N|$, итерации прерываются, и J^N появляется на экране в качестве результата интегрирования.

Примечание

Об алгоритме полиномиальной сплайн-интерполяции см. гл. 15.

7.1.3. О расходящихся интегралах

Если интеграл расходится (равен бесконечности), то вычислительный процессор Mathcad может выдать сообщение об ошибке, выделив при этом оператор интегрирования, как обычно, красным цветом. Чаще всего ошибка будет иметь тип **"Found a number with a magnitude greater than 10^307"** (Найдено число, превышающее значение 10^{307}) или **"Can't converge to a solution"** (Не сходится к решению), как, например, при попытке вычислить интеграл $\int_0^\infty \frac{1}{\sqrt{x}}\,dx$. Тем не менее, символьный процессор справляется с этим интегралом, совершенно правильно находя его бесконечное значение (листинг 7.4).

Листинг 7.4. Символьное вычисление расходящегося интеграла

$$\int_0^\infty \frac{1}{\sqrt{x}}\,dx \rightarrow \infty$$

Примечание

Символьный процессор предоставляет замечательные возможности аналитического вычисления интегралов, в том числе зависящих от параметров и неопределенных интегралов, как показано в листингах 7.5 и 7.6. Об *этом и вычислении интегралов с помощью меню Symbolics (Символика), упоминалось в гл. 5.*

Листинг 7.5. Символьное вычисление интеграла с переменным пределом

$$\int_0^a \frac{1}{\sqrt{x}}\, dx \rightarrow 2 \cdot a^{\left(\frac{1}{2}\right)}$$

Листинг 7.6. Символьное вычисление неопределенного интеграла

$$\int \frac{1}{\sqrt{x}}\, dx \rightarrow 2 \cdot x^{\left(\frac{1}{2}\right)}$$

При попытке численного решения задачи из листинга 7.4 методом, отличным от алгоритма вычисления интегралов с бесконечными пределами (Infinite Limit), будет получено неверное решение (листинг 7.7) — вместо бесконечности выдано большое, но конечное число, немного не дотягивающее до численной бесконечности, являющейся для вычислительного процессора просто большим числом 10^{307} *(см. разд. "Встроенные константы" гл. 4)*. Отметим, что Mathcad в режиме автоматического выбора алгоритма (AutoSelect) предлагает именно алгоритм Infinite Limit.

Листинг 7.7. Плохо выбранный численный алгоритм неверно находит расходящийся интеграл

$$\int_0^{\infty} \frac{1}{\sqrt{x}}\, dx = 6.325 \times 10^{153}$$

7.1.4. Кратные интегралы

Для того чтобы вычислить кратный интеграл:

1. Введите, как обычно, оператор интегрирования.
2. В соответствующих местозаполнителях введите имя первой переменной интегрирования и пределы интегрирования по этой переменной.

3. На месте ввода подынтегральной функции введите еще один оператор интегрирования.
4. Точно так же введите вторую переменную, пределы интегрирования и подынтегральную функцию (если интеграл двукратный) или следующий оператор интегрирования (если более чем двукратный) и т. д., пока выражение с многократным интегралом не будет введено окончательно.

Пример символьного и численного расчета двукратного интеграла в бесконечных пределах приведен в листинге 7.8. Обратите внимание, что символьный процессор "угадывает" точное значение интеграла π, а вычислительный определяет его приближенно и выдает в виде числа 3.142.

Листинг 7.8. Символьное и численное вычисление кратного интеграла

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\left(x^2+y^2\right)}\,dx\,dy \to \pi$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\left(x^2+y^2\right)}\,dx\,dy = 3.142$$

Внимание!

Аккуратнее вводите в редакторе Mathcad кратные интегралы, если они имеют различные пределы интегрирования по разным переменным. Не перепутайте пределы, относящиеся к разным переменным. Если Вы имеете дело с такого рода задачами, обязательно разберитесь с листингом 7.9, в котором символьный процессор вычисляет двукратный интеграл. В первой строке пределы интегрирования [a, b] относятся к переменной y, а во второй строке — к переменной x.

Листинг 7.9. Символьное вычисление кратных интегралов

$$\int_{a}^{b}\int_{-1}^{1} x+y^3\,dx\,dy \to \frac{1}{4}\cdot b^4 - \frac{1}{4}\cdot a^4$$

$$\int_{a}^{b}\int_{-1}^{1} x+y^3\,dy\,dx \to b^2 - a^2$$

7.2. Дифференцирование

С помощью Mathcad можно вычислять производные скалярных функций любого количества аргументов, от 0-го до 5-го порядка включительно. И функции, и аргументы могут быть как действительными, так и комплексными. Невозможно дифференцирование функций только вблизи точек их сингулярности.

Вычислительный процессор Mathcad обеспечивает превосходную точность численного дифференцирования. Но больше всего пользователь оценит возможности символьного процессора, который позволяет с легкостью осуществить рутинную работу вычисления производных громоздких функций, поскольку, в отличие от всех других операций, символьное дифференцирование выполняется успешно для подавляющего большинства аналитически заданных функций.

В Mathcad 11 для ускорения и повышения точности численного дифференцирования функций, заданных аналитически, автоматически задействуется символьный процессор *(см. разд. "Оптимизация вычислений" гл. 3).*

7.2.1. Первая производная

Для того чтобы продифференцировать функцию f(x) в некоторой точке:

1. Определите точку x, в которой будет вычислена производная, например **x:=1.**
2. Введите оператор дифференцирования нажатием кнопки **Derivative** (Производная) на панели **Calculus** (Вычисления) или введите с клавиатуры вопросительный знак <?>.
3. В появившихся местозаполнителях (рис. 7.3) введите функцию, зависящую от аргумента x, т. е. f(x), и имя самого аргумента x.
4. Введите оператор <=> численного или <→> символьного вывода для получения ответа.

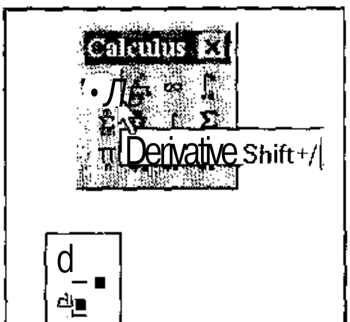


Рис. 7.3. Оператор дифференцирования

Пример дифференцирования функции `f(x)=cos(x)·ln(x)` приведен в листинге 7.10.

Листинг 7.10. Численное и символьное дифференцирование

$$x := 0.01$$

$$\frac{d}{dx}\cos(x)\cdot\ln(x) = 100.041$$

$$\frac{d}{dx}\cos(x)\cdot\ln(x) \rightarrow -\sin\left(1.\cdot10^{-2}\right)\cdot\ln\left(1.\cdot10^{-2}\right)+1.\cdot10^{2}\cdot\cos\left(1.\cdot10^{-2}\right)$$

Внимание!

Не забывайте предварительно определять точку, в которой производится численное дифференцирование, как это сделано в первой строке листинга 7.10. Иначе будет выдано сообщение об ошибке, показанное на рис. 7.4, гласящее, что переменная или функция, входящая в выражение, ранее не определена. Между тем, символьное дифференцирование не требует обязательного явного задания точки дифференцирования. В этом случае вместо значения производной (числа или числового выражения) будет выдана аналитическая зависимость (см. верхнюю часть рис. 7.4).

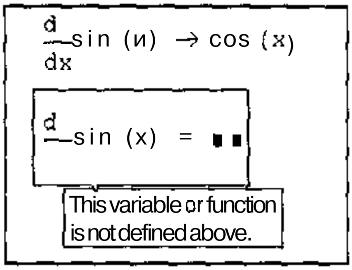


Рис. 7.4. Ошибка в применении оператора дифференцирования

Конечно, можно, как и при использовании других операторов, предварительно определить функцию в отдельном выражении, а затем посчитать ее производную (см. листинг 7.11); или применить оператор дифференцирования для определения собственных функций пользователя (см. листинг 7.12).

Листинг 7.11. Символьное и численное дифференцирование функции пользователя

$$f(x) := \frac{1}{x}$$

$$\frac{d}{dx}f(x) \rightarrow \frac{-1}{x^{2}}$$

$$x := 0.1$$

$$\frac{d}{dx}f(x) = -100$$

$\frac{d}{dx}f(x) \rightarrow -(1.\cdot 10^2)$

Листинг 7.12. Определение функции через оператор дифференцирования

$f(x) := \frac{1}{x}$

$g(x) := \frac{d}{dx}f(x)$

$g(0.1) = -100$

$g(0.1) \rightarrow -(1..10^2)$

В обоих листингах первой строкой определяется функция `f(x)=1/x`. Во второй строке листинга 7.11 с помощью символьного процессора находится аналитическое выражение ее производной, а в оставшейся части, подобно листингу 7.10, сначала численно, а затем аналитически определяются значения этой производной в точке x=o.1. В листинге 7.12 через производную от f(x) определяется еще одна пользовательская функция д<x) и затем находится ее конкретное значение в той же точке `x=0.1`.

Как Вы заметили, оператор дифференцирования, в основном, соответствует его общепринятому математическому обозначению. Однако в некоторых случаях при его вводе следует проявить осторожность. Рассмотрим один показательный пример, приведенный в листинге 7.13. Его первые две строки вычисляют производную sin(x) в точке x=o.5. Последняя строка демонстрирует неправильное применение оператора дифференцирования. Вместо вычисления производной sin(x) в той же точке, как этого можно было ожидать, получено нулевое значение. Это случилось потому, что аргумент функции sin(x) введен не в виде переменной x, а в виде числа. Поэтому Mathcad воспринимает последнюю строку как вычисление сначала значения синуса в точке x=o.5, а затем дифференцирование этого значения (т. е. константы) также в точке x=o.5, в соответствии с требованием первой строки листинга. Поэтому ответ, на самом деле, неудивителен — в какой точке ни дифференцируй константу, результатом будет ноль.

Листинг 7.13. Пример правильного и неправильного применения дифференцирования

x := 0.5

$\frac{d}{dx}\sin(x) = 0.878$

$$\frac{d}{dx}\sin(0.5) = 0$$

Для численного дифференцирования Mathcad применяет довольно сложный алгоритм, вычисляющий производную с колоссальной точностью до 7—8-го знака после запятой. Этот алгоритм (метод Риддера) описан во встроенной справочной системе Mathcad, доступной через меню **Help** (Справка). Погрешность дифференцирования не зависит от констант TOL ИЛИ CTOL, В противоположность большинству остальных численных методов, а определяется непосредственно алгоритмом.

Исключение составляют функции, которые дифференцируются в окрестности сингулярной точки; например для рассмотренной нами функции f(x)=1/x это будут точки вблизи x=o. При попытке найти ее производную при x=0 будет выдано сообщение об одной из ошибок деления на ноль **"Can't divide by zero"** (Деление на ноль невозможно) или **"Found a singularity while evaluating this expression. You may be dividing by zero"** (Найдена сингулярность при вычислении этого выражения. Возможно, Вы делите на ноль). Если попробовать численно определить производную очень близко к нулю, например, при x=ю$^{-100}$, то может появиться сообщение об ошибке **"Can't converge to a solution"** (Невозможно найти решение). Встретившись с одной из упомянутых ошибок, присмотритесь повнимательнее к дифференцируемой функции и убедитесь, что Вы не имеете дело с точкой сингулярности.

7.2.2. Производные высших порядков

Mathcad позволяет численно определять производные высших порядков, от 0-го до 5-го включительно. Чтобы вычислить производную функции f(x) N-ГО порядка в точке x, нужно проделать те же самые действия, что и при взятии первой производной *(см. разд. 7.2.1),* за тем исключением, что вместо оператора производной необходимо применить оператор N-Й производной (Nth **Derivative).** Этот оператор вводится с той же панели **Calculus** (Вычисления) либо с клавиатуры нажатием клавиш <Ctrl>+<?> и содержит еще два местозаполнителя, в которые следует поместить число N. В полном соответствии с математическим смыслом оператора, определение порядка производной в одном из местозаполнителей приводит к автоматическому появлению того же числа в другом из них.

"Производная" при N=O ПО определению равна самой функции, при N=1 получается обычная первая производная. Листинг 7.14 демонстрирует численное и символьное вычисление второй производной. Обратите внимание, что, как и при вычислении обычной производной, необходимо перед оператором дифференцирования присвоить аргументу функции значение, для которого будет вычисляться производная.

Листинг 7.14. Численное и символьное вычисление второй производной

x := 0.1

$$\frac{d^2}{dx^2}\cos(x)\cdot x^2 = 1.94$$

$$\frac{d^2}{dx^2}\cos(x)\cdot x^2 \rightarrow 1.99\cdot\cos(.1) - .4\cdot\sin(.1)$$

Примечание

Убедиться в том, что символьный процессор Mathcad в последней строке листинга 7.14 дает тот же результат, что и вычислительный процессор в предыдущей строке, можно, упростив его. Для этого следует выделить полученное последнее выражение и выбрать в меню Symbolics (Символика) пункт **Symplify** (Упростить). После этого ниже появится еще одна строка с численным результатом выделенного выражения.

Чтобы вычислить производную порядка выше 5-го, следует последовательно применить несколько раз оператор N-й производной, подобно тому как вводились операторы кратного интегрирования *(см. разд. 7.1.4)*. Однако для символьных вычислений этого не потребуется — символьный процессор умеет считать производные порядка выше 5-го. Сказанное иллюстрирует листинг 7.15, в котором сначала численно, а затем символьно вычисляется седьмая производная синуса в точке x=0.1.

Листинг 7.15. Численное и символьное вычисление седьмой производной

x := 0.1

$$\frac{d^5}{dx^5}\frac{d^2}{dx^2}\sin(x) = -0.995$$

$$\frac{d^7}{dx^7}\sin(x) \rightarrow -\cos(.1)$$

Расчет производных высших порядков производится тем же вычислительным методом Риддера, что и расчет первых производных. Причем для первой производной этот метод обеспечивает точность до 7—8 значащих разрядов числа, а при повышении порядка производной на каждую единицу точность падает примерно на один разряд.

Из сказанного ясно, что падение точности при численном расчете высших производных может быть очень существенно. В частности, если попытаться определить девятую производную синуса, подобно идее листинга 7.15, то в качестве результата будет выдан нуль, в то время как истинное значение девятой производной равно cos(0.1).

7.2.3. Частные производные

С помощью обоих процессоров Mathcad можно вычислять производные функций любого количества аргументов. В этом случае, как известно, производные по разным аргументам называются *частными*. Чтобы вычислить частную производную, необходимо, как обычно, ввести оператор производной с панели **Calculus** (Вычисления) и в соответствующем местозаполнителе напечатать имя переменной, по которой должно быть осуществлено дифференцирование. Пример приведен в листинге 7.16, в первой строке которого определена функция двух переменных, а в двух следующих строках символьным образом вычислены ее частные производные по обеим переменным — x и y — соответственно. Чтобы определить частную производную численным методом, необходимо предварительно задать значения всех аргументов, что и сделано в следующих двух строках листинга. Последнее выражение в листинге снова (как и в третьей строке) определяет символьно частную производную по y. Но поскольку переменным x и y уже присвоено конкретное значение, то в результате получается число, а не аналитическое выражение.

Листинг 7.16. Символьное и численное вычисление частных производных

$f(x, y) := x^{2 \cdot y} + \cos(x) \cdot y$

$\frac{\partial}{\partial x} f(x, y) \rightarrow 2 \cdot x^{(2 \cdot y)} \cdot \frac{y}{x} - \sin(x) \cdot y$

$\frac{\partial}{\partial y} f(x, y) \rightarrow 2 \cdot x^{(2 \cdot y)} \cdot \ln(x) + \cos(x)$

$x := 1 \qquad y := 0.1$

$\frac{\partial}{\partial y} f(x, y) = 0.54$

$\frac{\partial}{\partial y} f(x, y) \rightarrow \cos(1)$

Частные производные высших порядков рассчитываются точно так же, как и обычные производные высших порядков *(см. разд. 7.2.2)*. Листинг 7.17

иллюстрирует расчет вторых производных функции из предыдущего примера по переменным x, y и смешанной производной.

Листинг 7.17. Вычисление второй частной производной

$$\frac{\partial^2}{\partial x^2}\left(x^{2\cdot y}+\cos(x)\cdot y\right)\rightarrow 4\cdot x^{(2\cdot y)}\cdot\frac{y^2}{x^2}-2\cdot x^{(2\cdot y)}\cdot\frac{y}{x^2}-\cos(x)\cdot y$$

$$\frac{\partial^2}{\partial y^2}\left(x^{2\cdot y}+\cos(x)\cdot y\right)\rightarrow 4\cdot x^{(2\cdot y)}\cdot\ln(x)^2$$

$$\frac{\partial}{\partial x}\frac{\partial}{\partial y}\left(x^{2\cdot y}+\cos(x)\cdot y\right)\rightarrow 4\cdot x^{(2\cdot y)}\cdot\frac{y}{x}\cdot\ln(x)+2\cdot\frac{x^{(2\cdot y)}}{x}-\sin(x)$$

Возможно, Вы обратили внимание, что в обоих листингах 7.16 и 7.17 оператор дифференцирования записан в форме частной производной. Подобно тому как существует возможность выбирать вид, например оператора присваивания, можно записывать операторы дифференцирования в виде обычной или частной производной. Запись оператора не влияет на вычисления, а служит лишь более привычной формой представления расчетов. Для того чтобы изменить вид оператора дифференцирования на представление частной производной, следует:

1. Вызвать контекстное меню из области оператора дифференцирования нажатием правой кнопки мыши.
2. Выбрать в контекстном меню верхний пункт View Derivative As (Показывать производную как).
3. В появившемся подменю (рис. 7.5) выбрать пункт Partial Derivative (Частная производная).

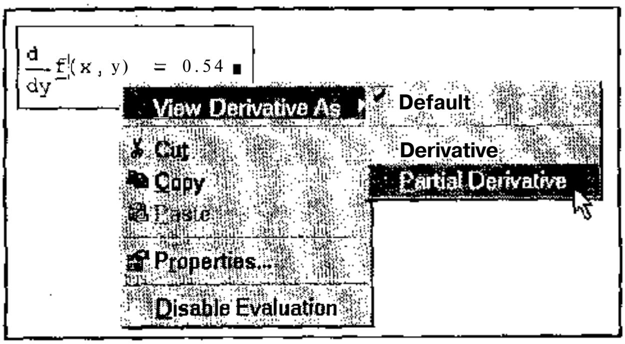


Рис. 7.5. Изменение вида оператора дифференцирования

Чтобы вернуть вид производной, принятый по умолчанию, выберите в подменю пункт Default (По умолчанию) либо, для представления ее в обычном виде — Derivative (Производная).

Завершим разговор о частных производных двумя примерами, которые нередко встречаются в вычислительной практике. Программная реализация первого из них, посвященная вычислению *градиента* функции двух переменных, приведена в листинге 7.18. В качестве примера взята функция f(x,y), определяемая в первой строке листинга, график которой показан в виде линий уровня на рис. 7.6. Как известно, градиент функции f(x,y) является векторной функцией тех же аргументов, что и f(x,y), определенной через ее частные производные, согласно второй строке листинга 7.18. В оставшейся части этого листинга задаются ранжированные переменные и матрицы, необходимые для подготовки графиков функции и ее градиента.

Листинг 7.18. Вычисление градиента

$$f(x,y) := x^2 + 0.1\cdot y^3$$

$$\mathrm{grad}(x,y) := \begin{pmatrix} \dfrac{\partial}{\text{Эх}} f(x,y) \\[2ex] \dfrac{\partial}{\partial y} f(x,y) \end{pmatrix}$$

$$N := 5$$

$$i := 0 .. 2\cdot N \qquad j := 0 .. 2 - N$$

$$F_{i,j} := f(i - N, j - N)$$

$$V_{i,j} := \mathrm{grad}(i - N, j - N)$$

$$X_{i,j} := (V_{i,j})_0 \qquad Y_{i,j} := (V_{i,j})_1$$

Векторное поле рассчитанного градиента функции f(x,y} показано на рис. 7.7. Как можно убедиться, сравнив рис. 7.6 и 7.7, математический смысл градиента состоит в задании в каждой точке (x,y) направления на плоскости, в котором функция f(x,y) растет наиболее быстро.

До сих пор в данной главе мы рассматривали скалярные функции, к которым, собственно, и можно применять операторы дифференцирования. Часто приходится иметь дело с вычислением производных векторных функций. Например, в различных областях математики *(см. разд. "Жесткие системы ОДУ" гл. 11)* мы сталкиваемся с проблемой вычисления *якобиана* (или *матрицы Якобы)* — матрицы, составленной из частных производных векторной функции по всем ее аргументам. Пример вычисления якобиана векторной функции f(x) векторного аргумента x приведен в листинге 7.19. В нем для определения частных производных якобиана каждый i-й скалярный компонент $f(x)_i$ дифференцируется символьным процессором Mathcad.

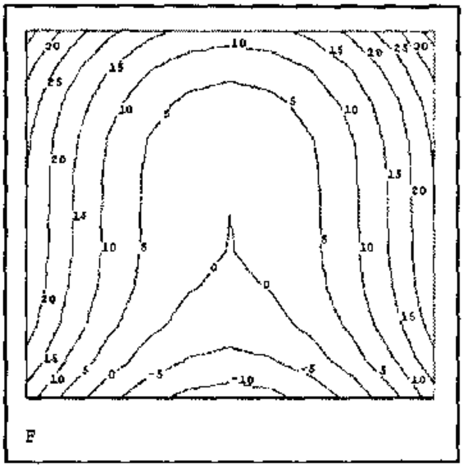


Рис. 7.6. График линий уровня функции f (x,y) (листинг 7.18)

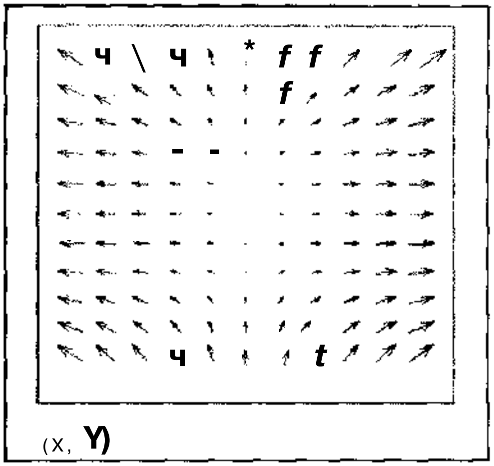


Рис. 7.7. Векторное поле градиента функции f (x,y) (листинг 7.18)

Тот же самый якобиан можно вычислить и несколько по-другому, если определить функцию не одного векторного, а трех скалярных аргументов f(x,y,z) (листинг 7.20).

Листинг 7.19. Вычисление якобиана векторной функции векторного аргумента

$$f(x) := \begin{bmatrix} x_0 \cdot \sin(x_1) \\ -(x_1)^{x_0} + x_2 \\ 3 \cdot x_1 + x_2 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial}{\partial x}\left[f\left(\begin{pmatrix} x \\ y \\ z \end{pmatrix}\right)_0\right] & \frac{\partial}{\partial y}\left[f\left(\begin{pmatrix} x \\ y \\ z \end{pmatrix}\right)_0\right] & \frac{\partial}{\partial z}\left[f\left(\begin{pmatrix} x \\ y \\ z \end{pmatrix}\right)_0\right] \\ \frac{\partial}{\partial x}\left[f\left(\begin{pmatrix} x \\ y \\ z \end{pmatrix}\right)_1\right] & \frac{\partial}{\partial y}\left[f\left(\begin{pmatrix} x \\ y \\ z \end{pmatrix}\right)_1\right] & \frac{\partial}{\partial z}\left[f\left(\begin{pmatrix} x \\ y \\ z \end{pmatrix}\right)_1\right] \\ \frac{\partial}{\partial x}\left[f\left(\begin{pmatrix} x \\ y \\ z \end{pmatrix}\right)_2\right] & \frac{\partial}{\partial y}\left[f\left(\begin{pmatrix} x \\ y \\ z \end{pmatrix}\right)_2\right] & \frac{\partial}{\partial z}\left[f\left(\begin{pmatrix} x \\ y \\ z \end{pmatrix}\right)_2\right] \end{bmatrix} \rightarrow \begin{pmatrix} \sin(y) & x \cdot \cos(y) & 0 \\ -y^{x} \cdot \ln(y) & -y^{x} \cdot \frac{x}{y} & 1 \\ 0 & 3 & 1 \end{pmatrix}$$

Листинг 7.20. Вычисление якобиана векторной функции трех скалярных аргументов

$$f(x,y,z) := \begin{pmatrix} x\cdot\sin(y) \\ -y^{x}+z \\ 3\cdot y+z \end{pmatrix}$$

$$\begin{bmatrix} \frac{\partial}{\partial x}\left(f(x,y,z)_0\right) & \frac{\partial}{\partial y}\left(f(x,y,z)_0\right) & \frac{\partial}{\partial z}\left(f(x,y,z)_0\right) \\ \frac{\partial}{\partial x}\left(f(x,y,z)_1\right) & \frac{\partial}{\partial y}\left(f(x,y,z)_1\right) & \frac{\partial}{\partial z}\left(f(x,y,z)_1\right) \\ \frac{\partial}{\partial x}\left(f(x,y,z)_2\right) & \frac{\partial}{\partial y}\left(f(x,y,z)_2\right) & \frac{\partial}{\partial z}\left(f(x,y,z)_2\right) \end{bmatrix} \to \begin{pmatrix} \sin(y) & x\cdot\cos(y) & 0 \\ -y^{x}\cdot\ln(y) & -y^{x}\cdot\frac{x}{y} & 1 \\ 0 & 3 & 1 \end{pmatrix}$$

Не забывайте, что для численного определения якобиана необходимо сначала определить точку, в которой он будет рассчитываться, т. е. вектор `x` в терминах листинга 7.19, или все три переменных `x`, `y`, `z` в обозначениях листинга 7.20.

ГЛАВА 8

Алгебраические уравнения и оптимизация

В этой главе рассматривается решение алгебраических нелинейных уравнений и систем таких уравнений.

Задача ставится следующим образом. Пусть имеется одно алгебраическое уравнение с неизвестным x

$$f(x)=0,$$

или система N алгебраических уравнений

$$\begin{cases} f_1(x_1, \dots, x_M) = 0, \\ \dots \\ f_N(x_1, \dots, x_M) = 0, \end{cases}$$

где $f(x)$ и $f_i(x_1, \ldots, x_M)$ — некоторые функции. Требуется найти корни уравнения, т. е. все значения x, которые переводят уравнение (или, соответственно, систему уравнений) в верное равенство (равенства).

Примечание

Решение систем линейных уравнений, у которых все функции имеют вид $f_i(x) = a_{i1} \cdot x_1 + a_{i2} \cdot x_2 + \ldots + a_{iN} \cdot x_N$, представляет собой отдельную задачу вычислительной линейной алгебры. *Она рассматривается в гл. 9.*

Как правило, отыскание корней численными методами связано с несколькими задачами:

- ❒ исследование существования корней в принципе, определение их количества и примерного расположения;
- ❒ отыскание корней с заданной погрешностью TOL.

Последнее означает, что надо найти значения x_0, при которых $f(x_0)$ отличается от нуля не более чем на TOL. почти все встроенные функции системы

Mathcad, предназначенные для решения нелинейных алгебраических уравнений, нацелены на решение второй задачи, т. е. предполагают, что корни уже приблизительно локализованы. Чтобы решить первую задачу (предварительной локализации корней), можно использовать, например, графическое представление f(x) *(см. разд. 8.1)* или последовательный поиск корня, начиная из множества пробных точек, покрывающих расчетную область *(сканирование)*. Mathcad предлагает несколько встроенных функций, которые следует применять в зависимости от специфики уравнения, т. е. свойств f(x). Для решения одного уравнения с одним неизвестным служит функция root, реализующая метод секущих *(см. разд. 8.1);* для решения системы — вычислительный блок Given/Find, сочетающий различные градиентные методы *(см. разд. 8.3, 8.4)*. Если f(x) — это полином, то вычислить все его корни можно также с помощью функции polyroots *(см. разд. 8.2)*. Кроме того, в некоторых случаях приходится сводить решение уравнений к задаче поиска экстремума *(см. разд. 8.5)*. Различные приемы нахождения экстремумов функций реализуются при помощи встроенных функций Minerr, Maximize и Minimize *(см. разд. 8.5, 8.6)*. В конце данной главы рассказывается о символьном решении уравнений *(см. разд. 8.7)* и о возможной программной реализации эффективного метода решения серии алгебраических уравнений или задач оптимизации, зависящих от параметра *(см. разд. 8.8)*.

8.1. Одно уравнение с одним неизвестным

Рассмотрим одно алгебраическое уравнение с одним неизвестным x.

$$f(x)=0, \tag{1}$$

например,

$$\sin(x)=0.$$

Для решения таких уравнений Mathcad имеет встроенную функцию root, которая, в зависимости от типа задачи, может включать либо два, либо четыре аргумента и, соответственно, работает несколько по-разному.

- `root(f(x),x);`
- `root(f(x),x,a,b);`
  - f(x) — скалярная функция, определяющая уравнение (1);
  - x — скалярная переменная, относительно которой решается уравнение;
  - a,b — границы интервала, внутри которого происходит поиск корня.

Первый тип функции root требует дополнительного задания *начального значения* (guess value) переменной x. Для этого нужно просто предварительно присвоить x некоторое число. Поиск корня будет производиться вблизи этого числа. Таким образом, присвоение начального значения требует априорной информации о примерной локализации корня.

Приведем пример решения очень простого уравнения sin(x)=0, корни которого известны заранее.

Листинг 8.1. Поиск корня нелинейного алгебраического уравнения

```
x := 0.Ь
£ (x) := sin (x)
solution := root ( f (x) , x)
solution = -6.2 × 10
```

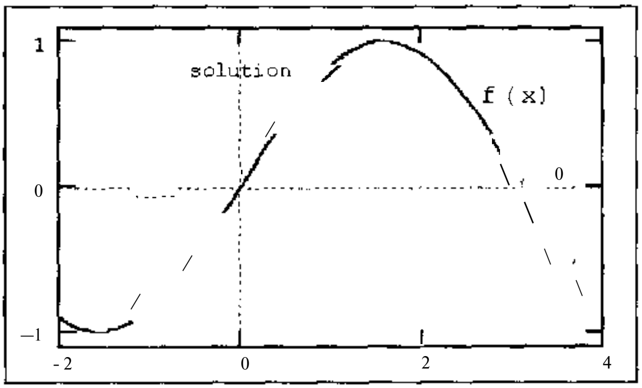


Рис. 8.1. Графическое решение уравнения sin(x)=0

График функции f(x)=sin(x) и положение найденного корня показаны на рис. 8.1. Обратите внимание, что, хотя уравнение имеет бесконечное количество корней x_N=N·π (N=0,±1,±2,...), Mathcad находит (с заданной точностью) только один из них, x_0, лежащий наиболее близко к x=0.5. Если задать другое начальное значение, например x=3, то решением будет другой корень уравнения x_1=π и т. д. Таким образом, для поиска корня средствами Mathcad требуется его предварительная локализация. Это связано с особенностями выбранного численного метода, который называется *методом секущих* и состоит в следующем (рис. 8.2):

1. Начальное приближение принимается за 0-е приближение к корню: x0=x.
2. Выбирается шаг h=TOL·x и определяется первое приближение к корню x1=x0+h. Если x=0, то принимается h=TOL.
3. Через эти две точки проводится секущая — прямая линия, которая пересекает ось x в некоторой точке x2. Эта точка принимается за второе приближение.

4. Новая секущая проводится через первую и вторую точки, тем самым определяя третье приближение, и т. д.
5. Если на каком-либо шаге оказывается, что уравнение выполнено, т. е. |f(x)|<TOL, то итерационный процесс прерывается, и x выдается в качестве решения.

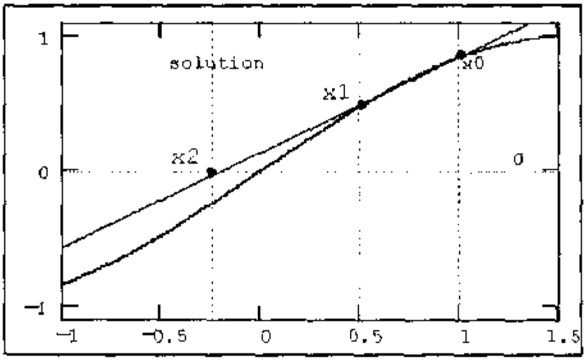


Рис. 8.2. Иллюстрация метода секущих

Результат, показанный на рис. 8.2, получен для погрешности вычислений, которой в целях иллюстративности предварительно присвоено значение TOL=0.5. Поэтому для поиска корня с такой невысокой точностью оказалось достаточно одной итерации. В вычислениях, приведенных в листинге 8.1, погрешность TOL=0.001 была установлена по умолчанию, и решение, выданное численным методом, лежало намного ближе к истинному положению корня x=о. Иными словами, чем меньше константа TOL, тем ближе к нулю будет значение f(x) в найденном корне, но тем больше времени будет затрачено вычислительным процессором Mathcad на его поиск.

Примечание

Соответствующий пример можно найти в Быстрых шпаргалках, на странице Ресурсов Mathcad. Он расположен в разделе "Solving Equations" (Решение уравнений) и называется "Effects of TOL on Solving Equations" (Влияние константы TOL на решение уравнений).

Если уравнение неразрешимо, то при попытке найти его корень будет выдано сообщение об ошибке. Кроме того, к ошибке или выдаче неправильного корня может привести и попытка применить метод секущих в области локального максимума или минимума f(x). В этом случае секущая может иметь направление, близкое к горизонтальному, выводя точку следующего приближения далеко от предполагаемого положения корня. Для решения таких уравнений лучше применять другую встроенную функцию Minerr *(см. разд. 8.5)*. Аналогичные проблемы могут возникнуть, если начальное приближение выбрано слишком далеко от настоящего решения или f(x) имеет особенности типа бесконечности.

Примечание

Для решения уравнения с одним неизвестным применимы и градиентные методы, относящиеся в Mathcad к системам уравнений. *Информация об этом приведена в разд. 8.3.*

Иногда удобнее задавать не начальное приближение к корню, а интервал |a,b|, внутри которого корень заведомо находится. В этом случае следует использовать функцию root с четырьмя аргументами, а присваивать начальное значение x не нужно, как показано в листинге 8.2. Поиск корня будет осуществлен в промежутке между a и b альтернативным численным методом (Риддера или Брента).

Листинг 8.2. Поиск корня алгебраического уравнения в заданном интервале

```
solution := root(sin(x), x, -1, 1)
solution = 0
```

Обратите внимание, что явный вид функции f(x) может быть определен непосредственно в теле функции root.

Когда root имеет четыре аргумента, следует помнить о двух ее особенностях:

- внутри интервала [a,b] не должно находиться более одного корня, иначе будет найден один из них, заранее неизвестно, какой именно;
- значения f(a) и f(b) должны иметь разный знак, иначе будет выдано сообщение об ошибке.

Если уравнение не имеет действительных корней, но имеет мнимые, то их также можно найти. В листинге 8.3 приведен пример, в котором уравнение $x^2+1=0$, имеющее два чисто мнимых корня, решается два раза с разными начальными значениями. При задании начального значения 0.5 (первая строка листинга) численный метод отыскивает первый корень (отрицательную мнимую единицу -i), а при начальном значении -0.5 (третья строка листинга) находится и второй корень (i).

Листинг 8.3. Поиск мнимого корня

$x := 0.5$

$\mathrm{root}(x^2+1, x) = -i$

$x := -0.5$

$\mathrm{root}(x^2+1, x) = i$

Для решения этого уравнения второй вид функции root (с четырьмя, а не с двумя аргументами) неприменим, поскольку f(x) является положительноопределенной, и указать интервал, на границах которого она имела бы разный знак, невозможно.

Остается добавить, что f(x) может быть функцией не только x, а любого количества аргументов. Именно поэтому в самой функции root необходимо определить, относительно какого из аргументов следует решить уравнение. Эта возможность проиллюстрирована листингом 8.4 на примере функции двух переменных $f(x,y)=x^2-y^2+3$. В нем сначала решается уравнение f(x,0)=0 относительно переменной x, а потом— другое уравнение f(1,y)=0 относительно переменной y.

Листинг 8,4. Поиск корня уравнения, заданного функцией двух переменных

$f(x,y) := x^2 - y^2 + 3$

x := 1

y := 0

root (f(x,y) , x) = -1.732i

root (f(x, y) , y) = 2

В первой строке листинга определяется функция f(x,y), во второй и третьей — значения, для которых будет производиться решение уравнения по у и x, соответственно. В четвертой строке решено уравнение f(x,o)=o, а в последней —уравнение f(1,y)=0. Не забывайте при численном решении уравнений относительно одной из переменных предварительно определить значения остальных переменных. Иначе попытка вычислить уравнения приведет к появлению ошибки **"This variable or function is not defined above"**, в данном случае говорящей о том, что другая переменная ранее не определена. Конечно, можно указать значение других переменных непосредственно внутри функции root, беспрепятственно удалив, например, вторую и третью строки листинга 8.4 и введя его последние строки в виде root (f(x, 0) ,x) = и root (f(1,y),y)=, соответственно.

Примечание

Для того чтобы отыскать зависимость корней уравнения, вычисленных по одной переменной, от других переменных, разработаны специальные эффективные алгоритмы. *Об одной из возможностей читайте в разд. 8,8.*

8.2. Корни полинома

Если функция £(x) является *полиномом*, то все его корни можно определить, используя встроенную функцию

polyroots(v),

где v — вектор, составленный из коэффициентов полинома.

Поскольку полином N-Й степени имеет ровно N корней (некоторые из них могут быть кратными), вектор v должен состоять из N+1 элемента. Результатом действия функции poiyroots является вектор, составленный из N корней рассматриваемого полинома. При этом нет надобности вводить какое-либо начальное приближение, как для функции root. Пример поиска корней полинома четвертой степени иллюстрируется листингом 8.5.

Листинг 8.5. Поиск корня полинома

$v := (3 \quad -10 \quad 12 \quad -6 \quad 1)^T$

$$\text{poiyroots}(v) = \begin{pmatrix} 1 \\ 1 - 5.113i \times 10^{-6} \\ 1 + 5.113i \times 10^{-6} \\ 3 \end{pmatrix}$$

Коэффициенты рассматриваемого в примере полинома

$$f(x) = (x-3) \cdot (x-1)^3 = x^4 - 6x^3 + 12x^2 - 10x + 3 \qquad (1)$$

записаны в виде вектора в первой строке листинга. Первым в векторе должен идти свободный член полинома, вторым — коэффициент при x^1 и т. д. Соответственно, последним N+1 элементом вектора должен быть коэффициент при старшей степени x^N.

Совет

Иногда исходный полином имеется не в развернутом виде, а, например, как произведение нескольких полиномов. В этом случае определить все его коэффициенты можно, выделив его и выбрав в меню **Symbolics** (Символика) пункт **Expand** (Разложить). В результате символьный процессор Mathcad сам преобразует полином в нужную форму, пользователю надо будет только корректно ввести ее в аргументы функции poiyroots.

Во второй строке листинга 8.5 показано действие функции poiyroots. Обратите внимание, что численный метод вместо двух из трех действительных единичных корней (иными словами, кратного корня 1) выдает два мнимых числа. Однако малая мнимая часть этих корней находится в пределах погрешности, определяемой константой TOL, И не должна вводить пользователей в заблуждение. Просто нужно помнить, что корни полинома могут быть комплексными, и ошибка вычислений может сказываться как на действительной, так и на комплексной части искомого корня.

Для функции poiyroots можно выбрать один из двух численных методов — метод полиномов Лаггера (он установлен по умолчанию) или метод парной матрицы.

Для смены метода:

1. Вызовите контекстное меню, щелкнув правой кнопкой мыши на слове polyroots.
2. В верхней части контекстного меню выберите либо пункт **LaGuerre** (Лаггера), либо **Companion Matrix** (Парная матрица).
3. Щелкните вне действия функции polyroots — если включен режим автоматических вычислений, будет произведен пересчет корней полинома в соответствии с вновь выбранным методом.

Для того чтобы оставить за Mathcad выбор метода решения, установите флажок **AutoSelect** (Автоматический выбор), выбрав одноименный пункт в том же самом контекстном меню.

8.3. Системы уравнений

Рассмотрим решение системы N нелинейных уравнений с M неизвестными

$$\begin{cases} f_1(x_1, \ldots, x_M) = 0, \\ \ldots \\ f_N(x_1, \ldots, x_M) = 0, \end{cases} \qquad (1)$$

Здесь $f_1(x_1, \ldots, x_M), \ldots, f_N(x_1, \ldots, x_M)$ — некоторые скалярные функции от скалярных переменных $x_1, x_2, \ldots, x_M$ и, возможно, от еще каких-либо переменных. Уравнений может быть как больше, так и меньше числа переменных. Заметим, что систему (1) можно формально переписать в виде

$$\mathbf{f}(\mathbf{x}) = 0, \qquad (2)$$

где $\mathbf{x}$ — вектор, составленный из переменных $x_1, x_2, \ldots, x_M$, а $\mathbf{f}$(x) — соответствующая векторная функция.

Для решения систем имеется специальный *вычислительный блок,* состоящий из трех частей, идущих последовательно друг за другом:

- Given — ключевое слово;
- система, записанная логическими операторами в виде равенств и, возможно, неравенств;
- $\mathrm{Find}(x_1, \ldots, x_M)$ — встроенная функция для решения системы относительно переменных $x_1, \ldots, x_M$.

Вставлять логические операторы следует, пользуясь панелью инструментов **Boolean** {Булевы операторы). Если Вы предпочитаете ввод с клавиатуры, помните, что логический знак равенства вводится сочетанием клавиш <Ctrl>+<=>. Блок Given/Find использует для поиска решения итерационные методы, поэтому, как и для функции root, требуется задать начальные значения для всех $x_1, \ldots, x_M$. Сделать это необходимо до ключевого слова

Given. Значение функции Find есть вектор, составленный из решения по каждой переменной. Таким образом, число элементов вектора равно числу аргументов Find.

В листинге 8.6 приведен пример решения системы двух уравнений.

Листинг 8.6. Решение системы уравнений

$f(x,y) := x^4 + y^2 - 3$

$g(x,y) := x + 2 \cdot y$

$x := 1 \quad y := 1$

Given

$f(x,y) = 0$

$g(x,y) = 0$

$v := \mathrm{Find}(x,y)$

$v = \begin{pmatrix} 1.269 \\ -0.635 \end{pmatrix}$

$f(v_0, v_1) = -1.954 \times 10^{-7}$

$g(v_0, v_1) = 0$

В первых двух строках листинга вводятся функции, которые определяют систему уравнений. Затем переменным x и y, относительно которых она будет решаться, присваиваются начальные значения. После этого следует ключевое слово Given и два логических оператора, выражающих рассматриваемую систему уравнений. Завершает вычислительный блок функция Find, значение которой присваивается вектору v. Следующая строка показывает содержание вектора v, т. е. решение системы. Первый элемент вектора есть первый аргумент функции Find, второй элемент— ее второй аргумент. В последних двух строках осуществлена проверка правильности решения уравнений.

Совет

Часто бывает очень полезно проверить точность решения уравнений, вычислив значения образующих их функций в найденных вычислительным процессором корнях, как это сделано в конце листинга 8.6.

Отметим, что уравнения можно определить непосредственно внутри вычислительного блока. Таким образом, можно не определять заранее функции f(x,y) и g(x,y), как это сделано в первых двух строках листинга 8.6, а сразу написать:

Given

$x^4 + y^2 = 3$

$x + 2 \cdot y = 0$

Такая форма представляет уравнения в более привычной и наглядной форме, особенно подходящей для документирования работы.

Графическая интерпретация рассмотренной системы представлена на рис. 8.3. Каждое из уравнений показано на плоскости XY графиком. Первое — сплошной кривой, второе — пунктиром. Поскольку второе уравнение линейное, то оно определяет на плоскости XY прямую. Две точки пересечения кривых соответствуют одновременному выполнению обоих уравнений, т. е. искомым действительным корням системы. Как нетрудно убедиться, в листинге найдено только одно из двух решений — находящееся в правой нижней части графика. Чтобы отыскать и второе решение, следует повторить вычисления, изменив начальные значения так, чтобы они лежали ближе к другой точке пересечения графиков, например `x=-1, y=-1`.

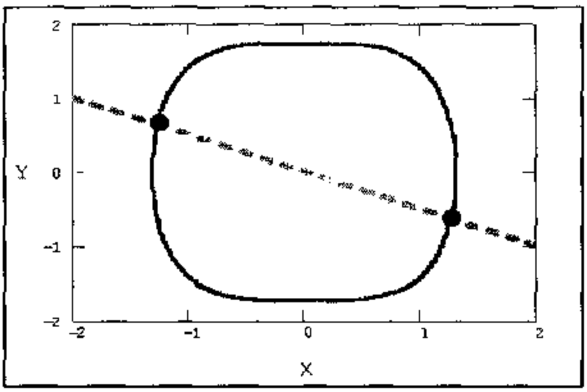


Рис. 8.3. Графическое решение системы двух уравнений

Пока мы рассмотрели пример системы из двух уравнений и таким же числом неизвестных, что встречается наиболее часто. Но число уравнений и неизвестных может и не совпадать. Более того, в вычислительный блок можно добавить дополнительные условия в виде неравенств. Например, введение ограничения на поиск только отрицательных значений x в рассмотренный выше листинг 8.6 приведет к нахождению другого решения, как это показано в листинге 8.7.

Листинг 8.7. Решение системы уравнений и неравенств

$x := 1 \quad y := 1$

Given

$x^4 + y^2 = 3$

$x + 2 \cdot y = 0$

$x < 0$

$$\text{Find}(x, y) = \begin{pmatrix} -1.269 \\ 0.635 \end{pmatrix}$$

Обратите внимание, что, несмотря на те же начальные значения, что и в листинге 8.6, мы получили в листинге 8.7 другой корень. Это произошло именно благодаря введению дополнительного неравенства, которое определено в блоке Given в предпоследней строке листинга 8.7.

Если предпринять попытку решить несовместную систему, Mathcad выдаст сообщение об ошибке, гласящее, что ни одного решения не найдено, и предложение попробовать поменять начальные значения или значение погрешности.

Примечание

Вычислительный блок использует константу CTOL В качестве погрешности выполнения уравнений, введенных после ключевого слова Given. Например, если CTOL=0.001, то уравнение x=10 будет считаться выполненным и при x=10.001, и при x=9.999. Другая константа TOL определяет условие прекращения итераций численным алгоритмом *(см. разд. 8.4)*, Значение CTOL может быть задано пользователем так же как и TOL, например, CTOL:=0.01. По умолчанию принято, что CTOL=TOL=0.001, но Вы по желанию можете переопределить их.

Особенную осторожность следует соблюдать при решении систем с числом неизвестных большим, чем число уравнений. Например, можно удалить одно из двух уравнений из рассмотренного нами листинга 8.6, попытавшись решить единственное уравнение д(x,y)=o с двумя неизвестными x и y. В такой постановке задача имеет бесконечное множество корней: для любого x и, соответственно, y=-x/2 условие, определяющее единственное уравнение, выполнено. Однако, даже если корней бесконечно много, численный метод будет производить расчеты только до тех пор, пока логические выражения в вычислительном блоке не будут выполнены (в пределах погрешности). После этого итерации будут остановлены и выдано решение. В результате будет найдена всего одна пара значений (x,y), обнаруженная первой.

Примечание

О том, как найти все решения рассматриваемой задачи, рассказывается в разд. 8.7.

Вычислительным блоком с функцией Find можно найти и корень уравнения с одним неизвестным. Действие Find в этом случае совершенно аналогично уже рассмотренным в данном разделе примерам. Задача поиска корня рассматривается как решение системы, состоящей из одного уравнения. Единственным отличием будет скалярный, а не векторный тип числа, возвращаемого функцией Find. Пример решения уравнения из предыдущего раздела приведен в листинге 8.8.

Листинг 8.8. Поиск корня уравнения с одним неизвестным с помощью функции Find

```
x := 0.5
Given
sin(x) = 0
Find(x) = -3.814 × 10^-7
```

В чем же отличие приведенного решения от листинга 8.1 с функцией root? Оно состоит в том, что одна и та же задача решена различными численными методами. В данном случае выбор метода не влияет на окончательный результат, но бывают ситуации, когда применение того или иного метода имеет решающее значение.

8.4. О численных методах решения систем уравнений

Если Вы решаете "хорошие" уравнения, как все те, которые были приведены в предыдущих разделах, то можете никогда не задумываться, как именно Mathcad ищет их корни. Однако даже в этом случае полезно представлять, что происходит "за кадром", т. е. какие действия совершаются в промежутке между введением необходимых условий после ключевого слова Given и получением результата после применения функции Find. Это важно хотя бы с позиций выбора начальных значений переменных перед вычислительным блоком. Рассмотрим в данном разделе некоторые особенности численных методов и возможности установки их различных параметров, которые предоставляет Mathcad.

Функция Find реализует градиентные численные методы. Покажем их основную идею на примере уравнения с одним неизвестным f(x)=0 для функции $f(x)=x^2+5x+2$, график которой показан на рис. 8.4. Основная идея градиентных методов состоит в последовательных приближениях к истинному решению уравнения, которые вычисляются с помощью производной от f(x). Приведем наиболее простую форму алгоритма, называемого методом Ньютона:

1. За нулевую итерацию принимается введенное пользователем начальное значение x0=x.
2. В точке x0 методом конечных разностей вычисляется производная f'(x0).
3. Пользуясь разложением Тейлора, можно заменить f(x) в окрестности x0 касательной — прямой линией $f(x) \approx f(x0)+f'(x0)\cdot(x-x0)$.

4. Определяется точка x1, в которой прямая пересекает ось x (см. рис. 8.4).
5. Если f(x1)<TOL, то итерации прерываются, и значение x1 выдается в качестве решения. В противном случае x1 принимается за новую итерацию, и цикл повторяется: строится касательная к f(x) в точке x1, определяется x2 — точка ее пересечения с осью x и т. д.

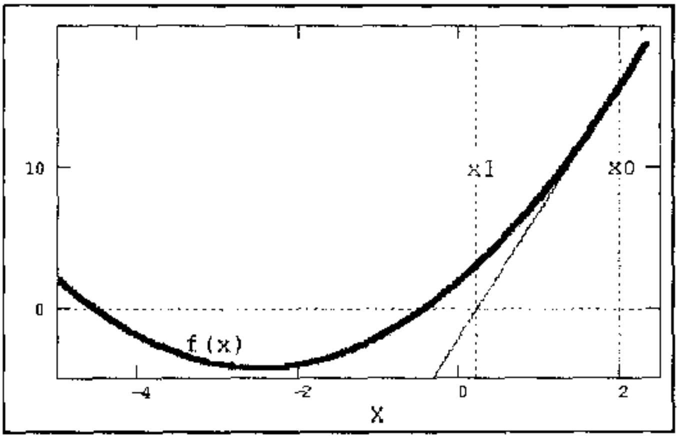


Рис. 8.4. Иллюстрация метода Ньютона

Модификация алгоритма Ньютона для решения системы нескольких уравнений заключается в линеаризации соответствующих функций многих переменных, т. е. аппроксимации их линейной зависимостью с помощью частных производных. Например, для нулевой итерации в случае системы двух уравнений используются выражения типа:

$$f_1(x,y) \approx f_1(x0,y0) + \frac{\partial f_1(x0,y0)}{\partial x}(x - x0) + \frac{\partial f_1(x0,y0)}{\partial y}(y - y0),$$

$$f_2(x,y) \approx f_2(x0,y0) + \frac{\partial f_2(x0,y0)}{\partial x}(x - x0) + \frac{\partial f_2(x0,y0)}{\partial y}(y - y0).$$

Чтобы отыскать точку, соответствующую каждой новой итерации, требуется приравнять оба равенства нулю, т. е. решить на каждом шаге полученную систему линейных уравнений.

Mathcad предлагает три различных вида градиентных методов. Чтобы поменять численный метод:

1. Щелкните правой кнопкой мыши на названии функции Find.
2. Наведите указатель мыши на пункт **Nonlinear** (Нелинейный) в контекстном меню.
3. В появившемся подменю (рис. 8.5) выберите один из трех методов: **Conjugate Gradient** (Сопряженных фадиентов), **Quasi-Newton** (Квази-Ньютоновский) или **Levenberg-Marquardt** (Левенберга).

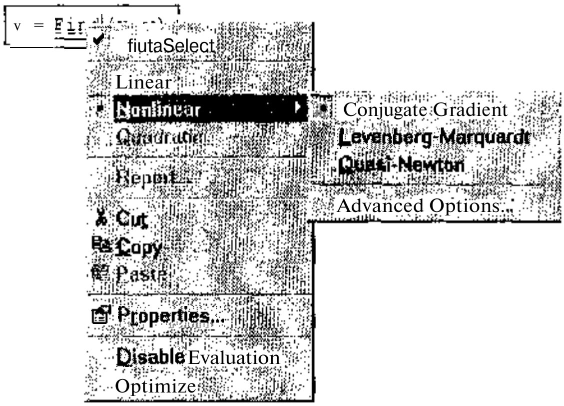


Рис. 8.5. Смена численного метода

Чтобы вернуть автоматический выбор типа численного метода, в контекстном меню надо выбрать пункт **AutoSelect** (Автоматический выбор). Если установлена опция автоматического выбора (о чем говорит флажок, установленный в пункте **AutoSelect**), то текущий тип численного метода можно узнать, вызвав то же самое подменю и посмотрев, который из них отмечен точкой. Два последних метода являются *квази-Ньютоновскими*, основная идея которых была рассмотрена выше. Первый из них, метод сопряженных градиентов, является *двухшаговым* — для поиска очередной итерации он использует как текущую, так и предыдущую итерации. Алгоритм Левенберга подробно описан в справочной системе Mathcad, а детальную информацию о методах Ньютона и сопряженных градиентов можно найти в большинстве книг по численным методам.

Помимо выбора самого метода, имеется возможность устанавливать их некоторые параметры. Для этого нужно вызвать с помощью того же контекстного меню диалоговое окно **Advanced Options** (Дополнительные параметры), выбрав в контекстном меню пункты **Nonlinear / Advanced options** (Нелинейный / Дополнительные параметры). В этом диалоговом окне (рис. 8.6) имеется пять групп переключателей, по два в каждой.

В первой строке **Derivative estimation** (Аппроксимация производной) определяется метод вычисления производной **Forward** (Вперед) или **Central** (Центральная). Они соответствуют аппроксимации производной либо правой (двухточечная схема "вперед"), либо центральной (трехточечная симметричная схема) конечной разностью.

Примечание

Обратите внимание, что вычисление производной в градиентных численных методах решения уравнений производится более экономичным способом, нежели при численном дифференцировании (*см. гл. 7*).

Во второй строке **Variable estimation** (Аппроксимация переменных) можно определить тип аппроксимации рядом Тейлора. Для рассмотренного нами в этом разделе случая аппроксимации касательной прямой линией выберите переключатель **Tangent** (Касательная), для более точной квадратичной аппроксимации (параболой) выберите **Quadratic** (Квадратичная). Следующая группа переключателей **Linear variable check** (Проверка линейности) позволяет в специфических задачах сэкономить время вычислений. Если Вы уверены, что нелинейности всех функций, входящих в уравнение, мало сказываются на значениях всех их частных производных, то установите переключатель Yes (Да). В этом случае производные будут приняты равными константам и не будут вычисляться на каждом шаге.

Совет

С осторожностью изменяйте параметры численных методов. Пользуйтесь ими, когда решение не находится при выставленных по умолчанию параметрах или когда расчеты занимают очень продолжительное время.

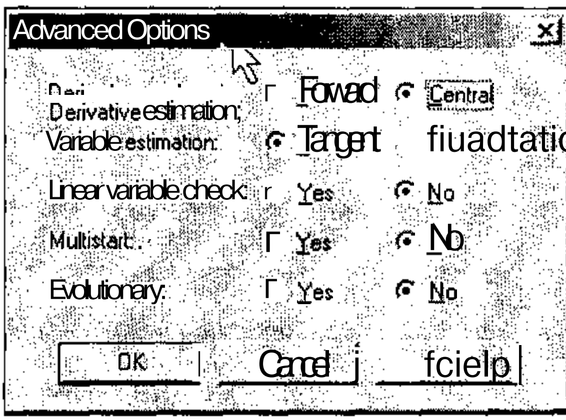


Рис. 8.6. Диалоговое окно Advanced Options

Пара переключателей **Multistart** (Сканирование) задает опцию поиска глобального или локального минимума или максимума. Если выставлен переключатель Yes (Да), Mathcad будет пытаться найти наиболее глубокий экстремум из области, близкой к начальному приближению. Эта опция предназначена, в основном, для настройки (тех же самых, градиентных) алгоритмов поиска экстремума, а не для решения алгебраических уравнений.

Наконец, последний переключатель **Evolutionary** (Эволюционный алгоритм), если установить его в положение Yes (Да), позволяет использовать модификацию численного метода для решения уравнений, определяемых не обязательно гладкими функциями. Как мы убедились в этом разделе, все градиентные методы, реализованные в функции Find, требуют многократного вычисления производных. Если Вы работаете с достаточно гладкими функциями, то градиентные методы обеспечивают быстрый и надежный поиск корня. Для поиска корня недостаточно гладких функций одной переменной, следует либо выставить данную опцию функции Find, либо использовать метод секущих (функцию root). Помните, что правильный выбор численного метода и его параметров может помочь при решении нестандартной задачи, которая при стандартных установках может и не поддаваться решению.

8.5. Приближенное решение уравнений

Иногда приходится заменять задачу определения корней системы уравнений задачей поиска экстремума функции многих переменных. Например, когда невозможно найти решение с помощью функции Find, можно попытаться потребовать вместо точного выполнения уравнений условий минимизировать их невязку. Для этого следует в вычислительном блоке вместо функции Find использовать функцию Minerr, имеющую тот же самый набор параметров. Она также должна находиться в пределах вычислительного блока:

1. $x_1:=C_1$ ... $x_M:=C_M$ — начальные значения для неизвестных.
2. Given — ключевое слово.
3. Система алгебраических уравнений и неравенств, записанная логическими операторами.
4. Minerr(x_1 ... x_M) — приближенное решение системы относительно переменных x_1, ... ,x_M, минимизирующее невязку системы уравнений.

Примечание

В функции Minerr реализованы те же самые алгоритмы, что и в функции Find, иным является только условие завершения работы численного метода. Поэтому пользователь может тем же самым образом, с помощью контекстного меню *(см. разд. 8.4)*, выбирать численный алгоритм приближенного решения для функции Minerr.

Пример использования функции Minerr показан в листинге 8.9. Как видно, достаточно заменить в вычислительном блоке имя функции на Minerr, чтобы вместо точного (с точностью до TOL) получить приближенное решение уравнения, заданного после ключевого слова Given.

Листинг 8.9. Приближенное решение уравнения, имеющего корень (x=0,y=0)

$x := 1 \qquad y := 1$

$k := 10^6$

Given

$k \cdot x^2 + y^2 = 0$

v := Minerr (x, y)

$$v = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Листинг 8.9 демонстрирует приближенное решение уравнения $k \cdot x^2 + y^2 = 0$, которое при любом значении коэффициента k имеет единственный точный корень $(x=0, y=0)$. Тем не менее, при попытке решить его функцией Find для больших к, порядка принятых в листинге, происходит генерация ошибки "No solution was found" (Решение не найдено). Это связано с иным поведением функции $f(x,y)=k \cdot x^2+y^2$ вблизи ее корня, по сравнению с функциями, приводимыми в качестве примеров выше в этой главе (см. рис. 8.1, 8.2). В отличие от них, f (x,y) не пересекает плоскость f (x,y)=o, а лишь касается ее (рис. 8.7) в точке $(x=0, y=0)$. Поэтому и найти корень изложенными в предыдущем разделе градиентными методами сложнее, поскольку вблизи корня производные f (x,y) близки к нулю, и итерации могут уводить предполагаемое решение далеко от корня.

Ситуация еще более ухудшается, если наряду с корнем типа касания (см. рис. 8.7) имеются (возможно, весьма удаленные) корни типа пересечения. Тогда попытка решить уравнение или систему уравнений с помощью функции Find может приводить к нахождению корня второго типа, даже если начальное приближение было взято очень близко к первому. Поэтому если Вы предполагаете, что система уравнений имеет корень типа касания, намного предпочтительнее использовать функцию Minerr, тем более, что всегда есть возможность проверить правильность решения уравнений простой подстановкой в них полученного решения (см. листинг 8.6).

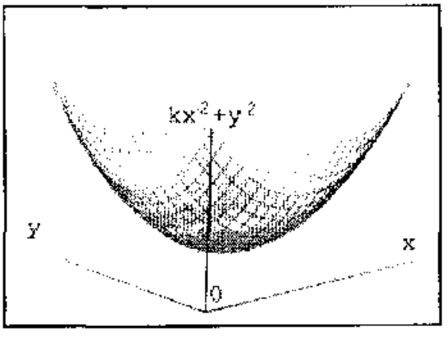


Рис. 8.7. График функции $k \cdot x^2 + y^2$

В листинге 8.9 мы рассмотрели пример нахождения существующего решения уравнения. Приведем в заключение пример нахождения функцией Minerr приближенного решения несовместной системы уравнений и неравенств (листинг 8.10). Решение, выдаваемое функцией Minerr, минимизирует невязку данной системы.

Примечание

Согласно своему математическому смыслу, функция Minerr может применяться для построения регрессии серии данных по закону, заданному пользователем *(см. разд. 15.2).*

Листинг 8.10. Приближенное решение несовместной системы уравнений и неравенств

$x := 1 \qquad y := 1$

Given

$x^2 + y^2 = -1$

$x > 0.1$

$y \le -0.2$

$\mathrm{Minerr}(x, y) = \begin{pmatrix} -0.042 \\ -0.085 \end{pmatrix}$

Как видно из листинга, в качестве результата выдаются значения переменных, наилучшим образом удовлетворяющие уравнению и неравенствам внутри вычислительного блока. Внимательный читатель может обнаружить, что решение, выщаваемое функцией Minerr в рассматриваемом примере, не является единственным, поскольку множество пар значений (x,y) в равной степени минимизирует невязку данной системы уравнений и неравенств. Поэтому для различных начальных значений будут получаться разные решения, подобно тому как разные решения выдаются функцией Find в случае бесконечного множества корней *(см. обсуждение листинга 8.6 в разд. 8.3).* Еще более опасен случай, когда имеются всего несколько локальных минимумов функции невязки. Тогда неудачно выбранное начальное приближение приведет к выдаче именно этого локального минимума, несмотря на то, что другой (глобальный) минимум невязки может удовлетворять системе гораздо лучше.

8.6. Поиск экстремума функции

Задачи поиска *экстремума* функции означают нахождение ее *максимума* (наибольшего значения) или *минимума* (наименьшего значения) в некоторой области определения ее аргументов. Ограничения значений аргументов, задающих эту область, как и прочие дополнительные условия, должны быть определены в виде системы неравенств и (или) уравнений. В таком случае говорят о задаче на *условный экстремум.*

Для решения задач поиска максимума и минимума в Mathcad имеются Встроенные функции Minerr, Minimize И Maximize. Все ОНИ ИСПОЛЬЗУЮТ те же градиентные численные методы, что и функция Find для решения уравнений. Поэтому Вы можете выбирать численный алгоритм минимизации из уже рассмотренных нами численных методов *(см. разд. 8.4).*

8.6.1. Экстремум функции одной переменной

Поиск экстремума функции включает в себя задачи нахождения *локального* и *глобального* экстремума. Последние называют еще *задачами оптимизации.* Рассмотрим конкретный пример функции `f(x)`, показанной графиком на рис. 8.8 на интервале (-2,5). Она имеет глобальный максимум на левой границе интервала, глобальный минимум, локальный максимум, локальный минимум и локальный максимум на правой границе интервала (в порядке слева направо).

В Mathcad с помощью встроенных функций решается только задача поиска локального экстремума. Чтобы найти глобальный максимум (или минимум), требуется либо сначала вычислить все их локальные значения и потом выбрать из них наибольший (наименьший), либо предварительно *просканировать* с некоторым шагом рассматриваемую область, чтобы выделить из нее подобласть наибольших (наименьших) значений функции и осуществить поиск глобального экстремума, уже находясь в его окрестности. Последний путь таит в себе некоторую опасность уйти в зону другого локального экстремума, но часто может быть предпочтительнее из соображений экономии времени.

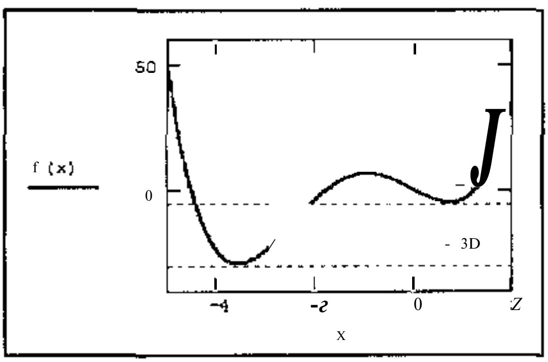


Рис. 8.8. График функции $f(x)=x^4+5 \cdot x^3-10 \cdot x$

Для поиска локальных экстремумов имеются две встроенные функции, которые могут применяться как в пределах вычислительного блока, так и автономно.

- `Minimize(f,x1 ... xM)` — вектор значений аргументов, при которых функция f достигает минимума;
- `Maximize(f,x1, ...,xM)` — вектор значений аргументов, при которых функция f достигает максимума;
  - `f(x1, .. .,xM, ...)` — функция;
  - `x1,...,xM` — аргументы, по которым производится минимизация (максимизация).

Всем аргументам функции f предварительно следует присвоить некоторые значения, причем для тех переменных, по которым производится минимизация, они будут восприниматься как начальные приближения. Примеры вычисления экстремума функции одной переменной (рис. 8.8) без дополнительных условий показаны в листингах 8.11—8.12. Поскольку никаких дополнительных условий в них не вводится, поиск экстремумов выполняется для любых значений x от $-\infty$ до ∞.

Листинг 8.11. Минимум функции одной переменной

```
f(x) := x^4 + 5·x^3 − 10·x
x := −1
Minimize ff,x) = −3.552
x := 1
Minimize(f,x) = 0.746
```

Листинг 8.12. Максимум функции одной переменной

```
f(x) := x^4 + 5·x^3 − 10·x
x := 1
Maximize(f,x) = −0.944
x := -10
Maximize(f,x) = ■
```

Как видно из листингов, существенное влияние на результат оказывает выбор начального приближения, в зависимости от чего в качестве ответа выдаются различные локальные экстремумы. В последнем случае численный метод вообще не справляется с задачей, поскольку начальное приближение x=-10 выбрано далеко от области локального максимума, и поиск решения уходит в сторону увеличения f(x), т. е. расходится к $x \to \infty$.

8.6.2. Условный экстремум

В задачах на условный экстремум функции минимизации и максимизации должны быть включены в вычислительный блок, т. е. им должно предшествовать ключевое слово Given. В промежутке между Given и функцией поиска экстремума с помощью булевых операторов записываются логические выражения (неравенства, уравнения), задающие ограничения на значения аргументов минимизируемой функции. В листинге 8.13 показаны примеры поиска условного экстремума на различных интервалах, определенных неравенствами. Сравните результаты работы этого листинга с двумя предыдущими.

Листинг 8.13. Три примера поиска условного экстремума функции

```
f (x) := x^4 + 5·x^3 - 10·x
x := 1
Given
-5 < x < -2
Minimize (f,x) =-3.552
x := 1
Given
x > O
Minimize (f, x) = 0,746
x := -1C
Given
-3 < x < 0
Maximize (f. x) = -0.944
```

Не забывайте о важности выбора правильного начального приближения и в случае задач на условный экстремум. Например, если вместо условия -3<x<o в последнем примере листинга задать -5<x<0, то при том же самом начальном x=-10 будет найден максимум Maximize(f,x)=-0.944, что неверно, поскольку максимальное значение достигается функцией f(x} на левой границе интервала при x=-5. Выбор начального приближения x=-4 решает задачу правильно, выдавая в качестве результата Maximize(f,x) =-5.

8.6.3. Экстремум функции многих переменных

Вычисление экстремума функции многих переменных не несет принципиальных особенностей по сравнению с функциями одной переменной. Поэтому ограничимся примером (листинг 8.14) нахождения максимума и минимума функции, показанной в виде графиков трехмерной поверхности и линий уровня на рис. 8.9. Привлечем внимание читателя только к тому, как с помощью неравенств, введенных логическими операторами, задается область на плоскости (X,Y).

Листинг 8.14. Экстремум функции двух переменных

```
f (x, y) := 2·(x - 5.07)^2 + (y - 10.03)^2 - 0.2·(x - 5.07)^3
x := 3          y := 3
Given
0 < x < 15
0 < y < 20
```

$$\text{Minimize}\ (f, x, y) = \begin{pmatrix} 5.07 \\ 10.03 \end{pmatrix}$$

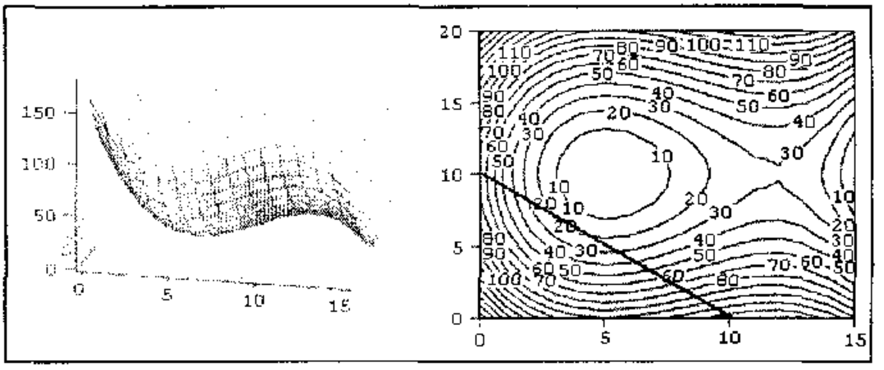


Рис. 8.9. График функции f (x,y) и отрезок прямой x+y=10

Дополнительные условия могут быть заданы и равенствами. Например, определение после ключевого слона Given уравнения x+y=ю приводит к такому решению задачи на условный экстремум:

$$\text{Minimize}\ (f, x, y) = \begin{pmatrix} 3.589 \\ 6.411 \end{pmatrix}$$

Как нетрудно сообразить, еще одно дополнительное условие привело к тому, что численный метод ищет минимальное значение функции f(x,y) вдоль отрезка прямой, показанного на рис. 8.9.

Примечание

Поиск минимума можно организовать и с помощью функции Minerr. Для этого в листинге 8.14 надо поменять имя функции Minimize на Minerr, а после ключевого слова Given добавить выражение, приравнивающее функции f(x,y) значение, заведомо меньшее минимального, например f (x,y) =0.

8.6.4. Линейное программирование

Задачи поиска условного экстремума функции многих переменных часто встречаются в экономических расчетах для минимизации издержек, финансовых рисков, максимизации прибыли и т. п. Целый класс экономических задач оптимизации описывается системами линейных уравнений и неравенств. Они называются задачами *линейного программирования.* Приведем характерный пример т. н. *транспортной задачи,* которая решает одну из проблем оптимальной организации доставки товара потребителям с точки зрения экономии транспортных средстн (листинг 8.15).

Листинг 8.15. Решение задачи линейного программирования

$$a := \begin{pmatrix} 145 \\ 210 \\ 160 \end{pmatrix} \qquad b := \begin{pmatrix} 237 \\ 278 \end{pmatrix}$$

$$\sum a = 515 \qquad \sum b = 515$$

$M := \mathrm{rows}(a)$ $\qquad N := \mathrm{rows}(b)$

$M = 3$ $\qquad N = 2$

$$c := \begin{pmatrix} 11.5 & 7 & 12 \\ 6.2 & 10 & 9.0 \end{pmatrix}$$

$$f(x) := \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} c_{i,j} \cdot x_{i,j}$$

$CTOL := 0.5$

$x_{N-1,M-1} := 0$

Given

$x_{0,0} + x_{1,0} = a_0$ $\qquad x_{0,0} \ge 0 \qquad x_{1,0} \ge 0$

$x_{0,1} + x_{1,1} = a_1$ $\qquad x_{0,1} \ge 0 \qquad x_{1,1} \ge 0$

$x_{0,2} + x_{1,2} = a_2$ $\qquad x_{0,2} \ge 0 \qquad x_{1,2} \ge 0$

$x_{0,0} + x_{0,1} + x_{0,2} = b_0$

$x_{1,0} + x_{1,1} + x_{1,2} = b_1$

$sol := \mathrm{Minimize}(f, x)$

$$sol = \begin{pmatrix} 0 & 210 & 27 \\ 145 & 0 & 133 \end{pmatrix} \qquad f(sol) = 3.89 \times 10^3$$

Модель типичной транспортной задачи следующая. Пусть имеется N предприятий-производителей, выпустивших продукцию в количестве $b_0, \ldots, b_{N-1}$ тонн. Эту продукцию требуется доставить м потребителям в количестве $a_0, \ldots, a_{M-1}$ тонн каждому. Численное определение векторов а и ь находится в первой строке листинга. Сумма всех заказов потребителей a_i равна сумме произведенной продукции, т. е. всех b_i (проверка равенства во второй строке). Если известна стоимость перевозки тонны продукции от i-го производителя к j-му потребителю c_{ij}, то решение задачи задает оптимальное распределение соответствующего товаропотока x_{ij} с точки зрения минимизации суммы транспортных расходов. Матрица с и минимизируемая функция f(x) матричного аргумента x находятся в середине листинга 8.15.

Условия, выражающие неотрицательность товаропотока, и равенства, задающие сумму произведенной каждым предприятием продукции и сумму заказов каждого потребителя, находятся после ключевого слова Given. Решение, присвоенное матричной переменной sol, выведено в последней строке листинга вместе с соответствующей суммой затрат. Обратите внимание, что для получения результата пришлось увеличить погрешность CTOL, задающую максимальную допустимую невязку дополнительных условий. В строке, предшествующей ключевому слову Given, определяются нулевые начальные значения для x простым созданием нулевого элемента матрицы $x_{N-1,M-1}$.

Внимание!

Если взять другие начальные значения для x, решение, скорее всего, будет другим! Возможно, Вы сумеете отыскать другой локальный минимум, который еще больше минимизирует транспортные затраты. Это еще раз доказывает, что задачи на глобальный минимум, к классу которых относится линейное программирование, требуют аккуратного отношения в смысле выбора начальных значений. Часто ничего другого не остается, кроме сканирования всей области начальных значений, чтобы из множества локальных минимумов выбрать наиболее глубокий.

8.7. Символьное решение уравнений

Некоторые уравнения можно решить точно с помощью символьного процессора Mathcad. Делается это очень похоже на численное решение уравнений с применением вычислительного блока. Присваивать неизвестным начальные значения нет необходимости. Листинги 8.16 и 8.17 демонстрируют символьное решение уравнения с одним неизвестным и системы двух уравнений с двумя неизвестными соответственно.

Листинг 8.16. Символьное решение алгебраического уравнения с одним неизвестным

Given

$x^2 + 2 \cdot x - 4 = 0$

$\mathrm{Find}(x) \rightarrow (\sqrt{5} - 1 \quad -1 - \sqrt{5})$

Листинг 8.17. Символьное решение системы алгебраических уравнений

Given

$x^4 + y^2 - 3 = 0$

$x + 2 \cdot y = 0$

$$\text{Find}(x, y) \rightarrow \begin{bmatrix} \frac{-1}{4}\left(-2+Э\cdot\sqrt{193}\right)^{\frac{1}{2}} & \frac{1}{A}\cdot\left(-2+2\cdot\sqrt{193}\right)^{\frac{1}{2}} & \frac{-1}{4}\left(-2-2\cdot\sqrt{193}\right)^{\frac{1}{2}} & \frac{1}{4}\cdot\left(-2-2\cdot\sqrt{193}\right)^{\frac{1}{2}} \\ \frac{1}{8}\cdot\left(-2+2\cdot\sqrt{193}\right)^{\frac{1}{2}} & \frac{-1}{6}\cdot\left(-2+2\cdot\sqrt{193}\right)^{\frac{1}{2}} & \frac{1}{S}\cdot\left(-2-22\cdot\sqrt{193}\right)^{\frac{1}{2}} & \frac{-1}{8}\left(-2-2\cdot\sqrt{193}\right)^{\frac{1}{2}} \end{bmatrix}$$

Как видно, вместо знака равенства после функции Find в листингах следует знак символьных вычислений, который можно ввести с панели **Symbolic** (Символика) или, нажав клавиши <Ctrl>+<.>. Не забывайте, что сами уравнения должны иметь вид логических выражений, т. е. знаки равенства нужно вводить с помощью панели **Booleans** (Булевы операторы). Обратите внимание, что в листинге 8.17 вычислены как два первых действительных корня, которые мы уже находили численным методом *(см. разд. 8.3),* так и два других мнимых корня. Эти два последних корня чисто мнимые, т. к. множитель, входящий в них, равен $(-2-2\cdot\sqrt{193})^2 \approx 5.46\cdot i$.

С помощью символьного процессора решить уравнение с одним неизвестным можно и по-другому:

1. Введите уравнение, пользуясь панелью **Booleans** (Булевы операторы) или нажав клавиши <Ctrl>+<.> для получения логического знака равенства, например $x^2+2\cdot x-4=0$.
2. Щелчком мыши выберите переменную, относительно которой Вы собираетесь решить уравнение.
3. Выберите в меню **Symbolics** (Символика) пункт **Variable / Solve** (Переменная / Решить).

После строки с уравнением появится строка с решением или сообщение о невозможности символьного решения этого уравнения.

В данном примере после осуществления описанных действий появляется вектор, состоящий из двух корней уравнения

$$\begin{pmatrix} \sqrt{5}-1 \\ -1-\sqrt{5} \end{pmatrix}.$$

Символьные вычисления могут производиться и над уравнениями, в которые, помимо неизвестных, входят различные параметры. В листинге 8.18 приведен пример решения уравнения четвертой степени с параметром а. Как видите, результат получен в аналитической форме.

Листинг 8.18. Символьное решение уравнения, зависящего от параметра

```
Given

x - a = 0

Find (x) → ( a  -a  i·a  -i·a )
```

В следующем разделе мы рассмотрим более подробно, как с помощью Mathcad можно численными методами решать подобные задачи.

8.8. Метод продолжения по параметру

Решение "хороших" нелинейных уравнений и систем типа тех, которые были рассмотрены в предыдущих разделах этой главы, представляет собой несложную, с вычислительной точки зрения, задачу. В реальных инженерных и научных расчетах очень распространена более сложная проблема: решение не одного уравнения (или системы), а целой серии уравнений, зависящих от некоторого параметра (или нескольких параметров). Для таких задач существуют очень эффективные методы, которые называются *методами продолжения.* Эти методы непосредственно не встроены в Mathcad, но могут быть легко запрограммированы с помощью уже рассмотренных нами средств. Будем далее говорить об одном уравнении, имея в виду, что всегда возможно обобщение результатов на случай системы уравнений.

Пусть имеется уравнение f(a,x>=o, зависящее не только от неизвестного x, но и от параметра a. Требуется определить зависимость его корня x от параметра a, т. е. x(a}. Простой пример такой задачи был приведен в предыдущем разделе в листинге 8.18. Тогда нам повезло, и решение в общем виде было найдено с помощью символьных вычислений. Рассмотрим еще один, чуть более сложный пример, аналитическое решение которого также заранее известно, но с которым символьный процессор, тем не менее, не справляется. Обобщим задачу, добавив зависимость от параметра a следующим образом:

$\sin(a\cdot x)=0.$ (1)

Аналитическое решение этого уравнения находится из соотношения $a\cdot x = N\cdot\pi$, где $N=0,\pm1,\pm2,\ldots$, т.е. имеется бесконечное количество (для каждого N) семейств решений $x_N=N\cdot\pi/a$. Несколько из семейств для N=1,2,3 показаны на рис. 8.10 тремя сплошными кривыми. Кроме того, следует иметь в виду, что для N=0, т. е. x=0, решением является прямая, совпадающая с осью x. Заметим (листинг 8.19), что символьный процессор Mathcad выдает в качестве решения только эту серию x=0.

Листинг 8.19. Символьное решение уравнения sin(a·x)=0

```
root (sin (a-x) , x) → O
```

Забудем на время, что аналитическое решение известно, и подойдем к уравнению (1) как к любой другой новой задаче. Решим его методом секущих, применяя для этого встроенную функцию root. Самый простой, но далеко не лучший способ иллюстрируется листингом 8.20. Корень уравнения (1) требуется определить численно для каждого значения параметра а. Для этого первые две строки листинга создают ранжированную переменную i, с помощью которой определяется вектор значений параметра a_i, для которых будут производиться расчеты. Его элементы пробегают значения от о.ою до 0.025 с шагом 0.0005 (эти числа взяты ради примера, Вы можете поэкспериментировать с другими значениями). Последняя строка листинга присваивает элементам еще одного вектора у вычисленные с помощью функции root значения корней уравнения (1) для каждого a_i. Но, для того чтобы функция root заработала, необходимо предварительно задать начальное приближение к решению, что сделано в третьей строке листинга. Ключевой момент метода, примененного в листинге 8.20, заключается в том, что одно и то же начальное значение x=300 использовано для всех a_i.

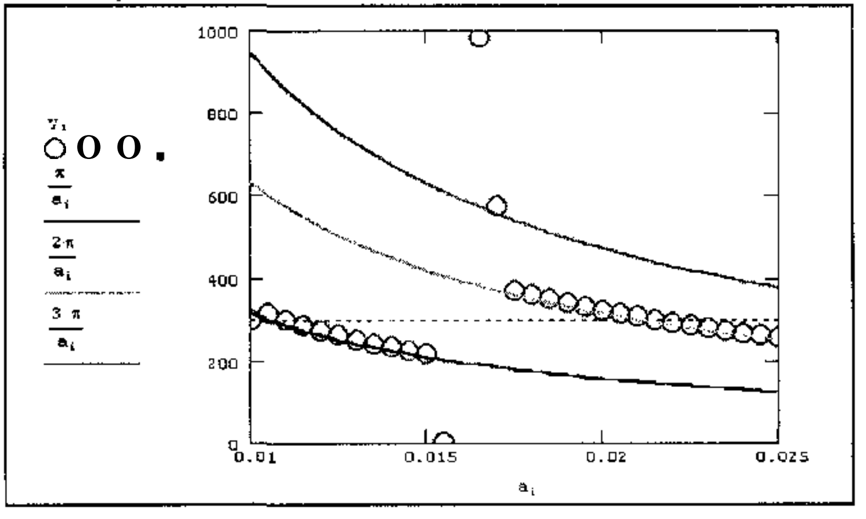


Рис. 8.10. Решение уравнения sin(a·x)=0 (см. листинг 8.21)

Листинг 8.20. Один из способов численного решения уравнения sin(a·x)=0

$i := 0 .. 30$

$a_i := 0.01 + i \cdot 0.0005$

$x := 300$

$y_{i+1} := \mathrm{root}(\sin(a_i \cdot x), x)$

Результат расчетов y_i показан на том же рис. 8.10 точками, причем самая левая точка отвечает начальному значению y_0=300. Обратите внимание, что, по мере увеличения а, кривая корней уравнения сначала идет по семейству решений $y=\pi/a$, а потом (в районе $a \approx 0.17$) "перепрыгивает" на другое семейство $y=2\cdot\pi/a$. С вычислительной точки зрения такая ситуация чаще всего крайне неблагоприятна, поскольку хотелось бы отыскать непрерывное семейство решений. Скачки зависимости y(a) могут вводить пользователя в заблуждение, вовсе скрывая от него существование семейства решений при a>o . 17.

Почему же происходят эти скачки с одного семейства решений на другое? Конечно, причина кроется в выборе начального значения для вычисления каждого из корней. Линия начальных значений $y=300$ обозначена на рис. 8.Ю в виде пунктирной горизонтальной прямой. Для $a_0=0.01$, и вообще для нескольких первых a_i начальное значение $y=300$ находится ближе всего к нижнему семейству решений $y=\pi/a$. Поэтому неудивительно, что численный метод находит именно эти корни. В правой части графика на рис. 8.10 к линии начальных значений ближе второе семейство решений $y=2\cdot\pi/a$, к ним-то и приводит численный метод.

Приведенные соображения диктуют очень простой рецепт избавления от скачков и нахождения одного из семейств непрерывных решений. Для этого требуется при поиске каждого (i+1)-го корня взять начальное значение, по возможности близкое к отыскиваемому семейству. Неплохим вариантом будет выбор приближения в виде предыдущего 1-го корня, который был найден для прошлого значения параметра a_i. Возможный вариант воплощения этого метода, называемого *продолжением по параметру,* приведен в листинге 8.21. В нем функция root применена внутри функции пользователя f(x0,a), определенной в самом начале листинга с помощью средств программирования. Назначение функции f{xo,a) заключается в том, что она выдает значение корня для заданного значения параметра а и начального приближения к решению хо. В остальном, смысл листинга 8.21 повторяет предыдущий, за исключением того, что задается явно (в его предпоследней строке) только начальное значение y_0= зоо только для поиска y_1. Для всех последующих точек, как следует из последней строки листинга, взято начальное значение, равное предыдущему корню y_i.

Листинг 8.21. Решение уравнения sin(a-x)=o методом продолжения по параметру

```
f (xO , a) := | x ← x0
              | root ( sin ( a • x) ,x)
i := 0 .. 30
a_i := 0.01 + i·0.0005
y_0 := 300
y_i+1 := f(y_i, a_i)
```

Результат вычислений, приведенный на рис. 8.11, разительно отличается от предыдущего. Как видно, столь малое изменение идеологии применения численного метода привело к определению непрерывного семейства корней. Отметим, что получить результат рис. 8.10 (без продолжения по параметру) в терминах введенной нами функции f(x0,a) можно, изменив ее первый аргумент в последней строке листинга 8.21 на константу: f(300,a_i).

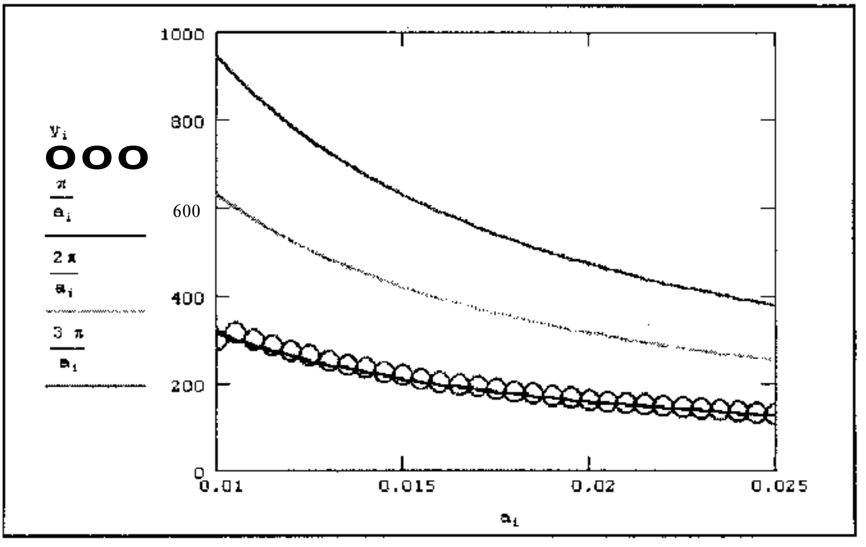


Рис. 8.11. Решение уравнения sin(a·x)=0 методом продолжения для y_0=300 (листинг 8.21)

Чтобы найти другое семейство решений, нужно взять соответствующее первое начальное значение y_0, например y_0=600. Результат действия листинга 8.21 для этого случая показан на рис. 8.12. Если взять y_0 ближе к третьему семейству решений (например y_0=900), то оно и будет найдено вычислительным процессором Mathcad, и т. д.

С помощью метода продолжения можно решать и соответствующие задачи оптимизации, зависящие от параметра. Идеология в этом случае остается точно такой же, но вместо функций решения нелинейных уравнений root или Find вам следует применить одну из функций поиска экстремума Minerr, Maximize или Minimize.

Мы привели основную идею и один из возможных способов реализации метода продолжения по параметру. Безусловно, Вы можете предложить иные как математические, так и программистские решения этой проблемы. В частности, для выбора очередного начального приближения к корню можно использовать результат экстраполяции уже найденной зависимости x{a), придумать более сложные алгоритмы для ветвящихся семейств решений и т. д.

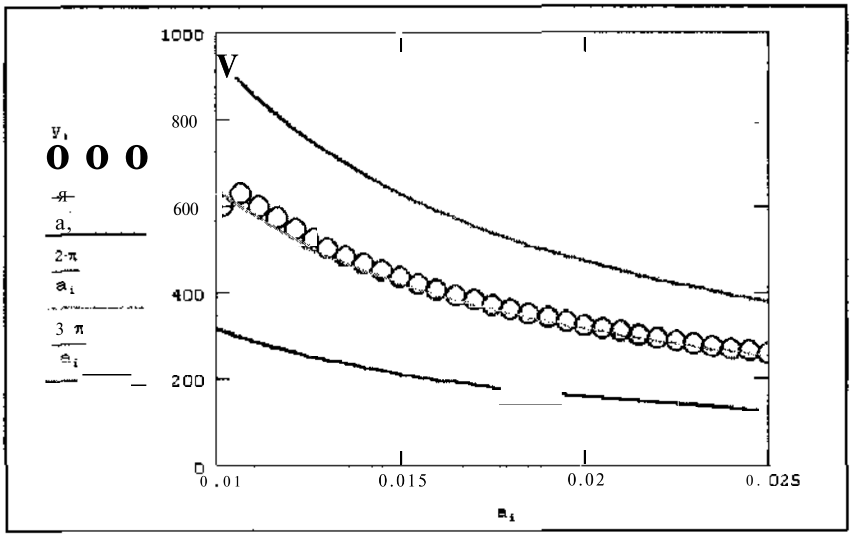


Рис. 8.12. Решение уравнения $\sin(a \cdot x)=0$ методом продолжения по параметру для $y_0=600$

ГЛАВА 9

Матричные вычисления

Матричные вычисления можно условно разделить на несколько типов. Первый тип — это простейшие действия, которые реализованы операторами *(см. разд. 9.1)* и несколькими функциями, предназначенными для создания, объединения, сортировки, получения основных свойств матриц и т. п. *(см. разд. 9.2).* Второй тип — это более сложные функции, которые реализуют алгоритмы вычислительной линейной алгебры, такие, как решение систем линейных уравнений *(см. разд. 9.3),* вычисление собственных векторов и собственных значений *(см. разд. 9.4)*, различные матричные разложения *(см.разд. 9.5).*

9.1. Простейшие операции с матрицами

Простейшие операции матричной алгебры реализованы в Mathcad в виде операторов. Написание операторов по смыслу максимально приближено к их математическому действию. Каждый оператор выражается соответствующим символом. Рассмотрим матричные и векторные операции Mathcad 11. Векторы являются частным случаем матриц размерности `N×1`, поэтому для них справедливы все те операции, что и для матриц, если ограничения особо не оговорены (например некоторые операции применимы только к квадратным матрицам NXN). Какие-то действия допустимы только для векторов (например скалярное произведение), а какие-то, несмотря на одинаковое написание, по-разному действуют на векторы и матрицы.

Внимание!

Непосредственное проведение векторных операций над строками, т. е. матрицами `1×N`, невозможно; для того чтобы превратить строку в вектор, ее нужно предварительно транспонировать.

8 Зак. 984

9.1.1. Транспонирование

Транспортированием называют операцию, переводящую матрицу размерности $M \times N$ в матрицу размерности $N \times M$, делая столбцы исходной матрицы строками, а строки — столбцами. Пример приведен в листинге 9.1. Ввод символа транспонирования (transpose) осуществляется с помощью панели инструментов Matrix (Матрица) или нажатием клавиш <Ctrl>+<1>. Не забывайте, что для вставки символа транспонирования матрица должна находиться между линиями ввода.

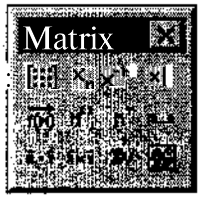


Рис. 9.1. Панель инструментов Matrix

Листинг 9.1. Транспонирование векторов и матриц

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}^T = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$$

9.1.2. Сложение

В Mathcad можно как складывать матрицы, так и вычитать их друг из друга. Для этих операторов применяются символы <+> или <—>, соответственно. Матрицы должны иметь одинаковую размерность, иначе будет выдано сообщение об ошибке. Каждый элемент суммы двух матриц равен сумме соответствующих элементов матриц-слагаемых (листинг 9.2).

Листинг 9.2. Сложение и вычитание матриц

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \qquad B := \begin{pmatrix} 1 & 0 & 0 \\ -1 & -3 & -4 \end{pmatrix}$$

$$A + B = \begin{pmatrix} 2 & 2 & 3 \\ 3 & 2 & 2 \end{pmatrix}$$

$$A-B = \begin{pmatrix} 0 & 2 & 3 \\ 5 & 8 & 10 \end{pmatrix}$$

Кроме сложения матриц, Mathcad поддерживает операцию сложения матрицы со скаляром (листинг 9.3). Каждый элемент результирующей матрицы равен сумме соответствующего элемента исходной матрицы и скалярной величины.

Листинг 9.3. Сложение матрицы со скаляром

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

$$x := 1$$

$$A + x = \begin{pmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \end{pmatrix}$$

$$A - x = \begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{pmatrix}$$

Результат смены знака матрицы эквивалентен смене знака всех ее элементов. Для того чтобы изменить знак матрицы, достаточно ввести перед ней знак минуса, как перед обычным числом (листинг 9.4).

Листинг 9.4. Смена знака матрицы

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \qquad -A = \begin{pmatrix} -1 & -2 & -3 \\ -4 & -5 & -6 \end{pmatrix}$$

9.1.3. Умножение

При умножении следует помнить, что матрицу размерности `M×N` допустимо умножать только на матрицу размерности NXP (P может быть любым). В результате получается матрица размерности `M×P`.

Чтобы ввести символ умножения, нужно нажать клавишу со звездочкой <*> или воспользоваться панелью инструментов **Matrix** (Матрица), нажав на ней кнопку **Dot Product** (Умножение) (рис. 9.1). Умножение матриц обозначается по умолчанию точкой, как показано в листинге 9.5. Символ умножения матриц можно выбирать точно так же, как и в скалярных выражениях *(см. разд. "Управление отображением некоторых операторов" гл. 2).*

Листинг 9.5. Умножение матриц

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \qquad B := \begin{pmatrix} 1 & 0 & 0 \\ -1 & -3 & -4 \end{pmatrix}$$

$$C := B^T$$

$$C = \begin{pmatrix} 1 & -1 \\ 0 & -3 \\ 0 & -4 \end{pmatrix}$$

$$A \cdot C = \begin{pmatrix} 1 & -19 \\ 4 & -43 \end{pmatrix}$$

Еще один пример, относящийся к умножению вектора на матрицу-строку и, наоборот, строки на вектор, приведен в листинге 9.6. Во второй строке этого листинга показано, как выглядит формула при выборе отображения оператора умножения **No Space** (Вместе).

Листинг 9.6. Умножение вектора и строки

$$\begin{pmatrix} 3 & 4 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 11 \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} \begin{pmatrix} 3 & 4 \end{pmatrix} = \begin{pmatrix} 3 & 4 \\ 6 & 8 \end{pmatrix}$$

Внимание!

Тот же самый оператор умножения действует на два вектора по-другому (см. *разд. 9.1.6)*.

Аналогично сложению матриц со скаляром определяется умножение и деление матрицы на скалярную величину (листинг 9.7). Символ умножения вводится так же, как и в случае умножения двух матриц. На скаляр можно умножать любую матрицу M×N.

Листинг 9.7. Умножение матрицы на скаляр

$$A \cdot 2 = \begin{pmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{pmatrix}$$

$$\frac{A}{2} = \begin{pmatrix} 0.5 & 1 & 1.5 \\ 2 & 2.5 & 3 \end{pmatrix}$$

9.1.4. Определитель квадратной матрицы

Определитель (Determinant) матрицы обозначается стандартным математическим символом. Чтобы ввести оператор нахождения определителя матрицы, можно нажать кнопку **Determinant** (Определитель) на панели инструментов **Matrix** (Матрица) (рис. 9.2) или набрать на клавиатуре <|> {нажав клавиши

<Shift>+<\>). В результате любого из этих действий появляется местозаполнитель, в который следует поместить матрицу. Чтобы вычислить определитель уже введенной матрицы (именно этот случай показан на рис. 9.2), нужно:

1. Переместить курсор в документе таким образом, чтобы поместить матрицу между линиями ввода (напоминаем, что линии ввода — это вертикальный и горизонтальный отрезки синего цвета, образующие уголок, указывающий на текущую область редактирования).
2. Ввести оператор нахождения определителя матрицы.
3. Ввести знак равенства, чтобы вычислить определитель.

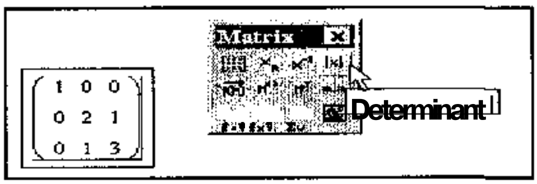


Рис. 9.2 Ввод символа определителя матрицы

Результат вычисления определителя приведен в листинге 9.8.

Листинг 9.8. Поиск определителя квадратной матрицы

$$\left|\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 3 \end{pmatrix}\right| = 5$$

9.1.5. Модуль вектора

Модуль вектора (vector magnitude) обозначается тем же символом, что и определитель матрицы. По определению, модуль вектора равен квадратному корню из суммы квадратов его элементов (листинг 9.9).

Листинг 9.9. Поиск модуля вектора

$$\left|\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}\right| = 3.742$$

9.1.6. Скалярное произведение векторов

Скалярное произведение векторов (vector inner product) определяется как скаляр, равный сумме попарных произведений соответствующих элементов.

Векторы должны иметь одинаковую размерность, скалярное произведение имеет ту же размерность. Скалярное произведение двух векторов u и v равно $u \cdot v = |u| \cdot |v| \cdot \cos\theta$, где в — угол между векторами. Если векторы ортогональны, их скалярное произведение равно нулю. Обозначается скалярное произведение тем же символом умножения (листинг 9.10). Для обозначения скалярного произведения пользователь также может выбирать представление оператора умножения.

Совет

Никогда не применяйте для обозначения скалярного произведения символ x, который является общеупотребительным символом векторного произведения *(см. разд. 9.1.7).*

Листинг 9.10. Скалярное произведение векторов

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} = 32$$

Внимание!

С осторожностью перемножайте несколько (более двух) векторов. По-разному расставленные скобки полностью изменяют результат умножения. Примеры такого умножения см. в листинге 9.11.

Листинг 9.11. Скалярное произведение векторов, умноженное на третий вектор

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} \cdot \begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix} = \begin{pmatrix} 224 \\ 256 \\ 288 \end{pmatrix}$$

$$\left[\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}\right] \cdot \begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix} = \begin{pmatrix} 224 \\ 256 \\ 288 \end{pmatrix} \qquad \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \cdot \left[\begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} \cdot \begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix}\right] = \begin{pmatrix} 122 \\ 244 \\ 366 \end{pmatrix}$$

9.1.7. Векторное произведение

Векторное произведение (cross product) двух векторов u и v с углом 8 между ними равно вектору с модулем $|u| \cdot |v| \cdot \sin\theta$, направленным перпендикулярно плоскости векторов u и v. Обозначают векторное произведение символом x, который можно ввести нажатием кнопки **Cross Product** (Векторное произведение) в панели **Matrix** (Матрица) или сочетанием клавиш <Ctrl>+<8>. Пример приведен в листинге 9.12.

Листинг 9.12. Векторное произведение

$$\begin{pmatrix}1\\2\\3\end{pmatrix}\times\begin{pmatrix}4\\5\\6\end{pmatrix}=\begin{pmatrix}-3\\6\\-3\end{pmatrix}$$

9.1.8. Сумма элементов вектора и след матрицы

Иногда бывает нужно вычислить сумму всех элементов вектора. Для этого существует вспомогательный оператор (листинг 9.13, первая строка), задаваемый кнопкой **Vector Sum** (Сумма вектора) на панели **Matrix** (Матрица) или сочетанием клавиш <Ctrl>+<4>. Этот оператор чаще оказывается полезным не в векторной алгебре, а при организации циклов с индексированными переменными.

На том же листинге 9.13 (снизу) показано применение операции суммирования диагональных элементов квадратной матрицы. Эту сумму называют *следом* (trace) матрицы. Данная операция организована в виде встроенной функции `tr`:

- tr(A) — след квадратной матрицы A.

Листинг 9.13. Суммирование элементов вектора и диагонали матрицы

$$\sum\begin{pmatrix}1\\2\\3\end{pmatrix}=6$$

$$A:=\begin{pmatrix}3&4\\7&2\end{pmatrix}\qquad tr(A)=5$$

9.1.9. Обратная матрица

Поиск *обратной матрицы* возможен, если матрица квадратная и ее определитель не равен нулю (листинг 9.14). Произведение исходной матрицы на обратную по определению является единичной матрицей. Для ввода оператора поиска обратной матрицы нажмите кнопку **Inverse** (Обратная матрица) на панели инструментов **Matrix** (Матрица).

Листинг 9.14. Поиск обратной матрицы

$$\begin{pmatrix}1&0&0\\0&2&0\\0&0&3\end{pmatrix}^{-1}=\begin{pmatrix}1&0&0\\0&0.5&0\\0&0&0.333\end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.333 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.999 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.333 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.999 \end{pmatrix}$$

9.1.10. Возведение матрицы в степень

К квадратным матрицам можно формально применять операцию возведения в степень n. Для этого n должно быть целым числом. Результат данной операции приведен в табл. 9.1. Ввести оператор возведения матрицы M в степень n можно точно так же, как и для скалярной величины: нажав кнопку **Raise to Power** (Возвести в степень) на панели **Calculator** (Калькулятор) или нажав клавишу <^>. После появления местозаполнителя в него следует ввести значение степени n.

Таблица 9.1. *Результаты возведения матрицы в степень*

n	M^n
0	единичная матрица размерности матрицы M
1	сама матрица M
-1	M^{-1} — матрица, обратная M
2, 3, ...	$M \cdot M$, $(M \cdot M) \cdot M$, ...
-2, -3, ...	$M^{-1} \cdot M^{-1}$, $(M^{-1} \cdot M^{-1}) \cdot M^{-1}$, ...

Некоторые примеры возведения матриц в степень приведены в листинге 9.15.

Листинг 9.15. Примеры возведения квадратной матрицы в целую степень

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.333 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3.003 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}^{-2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.111 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}^2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 9 \end{pmatrix}$$

9.1.11. Векторизация массивов

Векторная алгебра Mathcad включает несколько необычный оператор, который называется *оператором векторизации* (vectorize operator). Этот оператор предназначен, как правило, для работы с массивами. Он позволяет провести однотипную операцию над всеми элементами массива (т. е. матрицы или вектора), упрощая тем самым программирование циклов. Например, иногда требуется умножить каждый элемент одного вектора на соответствующий элемент другого вектора. Непосредственно такой операции в Mathcad нет, но ее легко осуществить с помощью векторизации (листинг 9.16). Для этого:

1. Введите векторное выражение, как показано во второй строчке листинга (обратите внимание, что в таком виде символ умножения обозначает оператор скалярного произведения векторов).
2. Переместите курсор таким образом, чтобы линии ввода выделяли все выражение, которое требуется подвергнуть векторизации (рис. 9.3).
3. Введите оператор векторизации, нажав кнопку **Vectorize** (Векторизация) на панели **Matrix** (Матрица) (рис. 9.3), или сочетанием клавиш <Ctrl>+<->.
4. Введите <=>, чтобы получить результат.

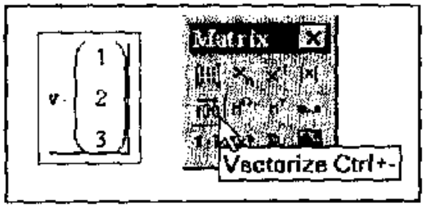


Рис. 9.3. Оператор векторизации

Листинг 9.16. Использование векторизации для перемножения элементов вектора

$$v := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

$$v \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = 14$$

$$\overrightarrow{\left[v\cdot\begin{pmatrix}1\\2\\3\end{pmatrix}\right]}=\begin{pmatrix}1\\4\\9\end{pmatrix}$$

Оператор векторизации можно использовать только с векторами и матрицами одинакового размера.

Большинство неспецифических функций Mathcad не требуют векторизации для проведения одной и той же операции над всеми элементами вектора. Например, аргументом тригонометрических функций по определению является скаляр. Если попытаться вычислить синус векторной величины, Mathcad осуществит векторизацию по умолчанию, вычислив синус каждого элемента и выдав в качестве результата соответствующий вектор. Пример показан в листинге 9.17.

Листинг 9.17. Векторизация необязательна для большинства функций Mathcad

$$\sin(v)=\begin{pmatrix}0.841\\0.909\\0.141\end{pmatrix}\qquad \sin\left(\vec{v}\right)=\begin{pmatrix}0.841\\0.909\\0.141\end{pmatrix}$$

9.1.12. Символьные операции с матрицами

Все матричные и векторные операторы, о которых шла речь выше, допустимо использовать в символьных вычислениях. Мощь символьных операций заключается в возможности проводить их не только над конкретными числами, но и над переменными. Несколько примеров приведены в листинге 9.18.

Листинг 9.1 в. Примеры символьных операций над векторами и матрицами

$$\begin{pmatrix}a&b\\c&d\\f&g\end{pmatrix}\cdot\begin{pmatrix}o&p&q\\r&s&t\end{pmatrix}\to\begin{pmatrix}a\cdot o+b\cdot r&a\cdot p+b\cdot s&a\cdot q+b-t\\c\cdot o+d\cdot r&c\cdot p+d\cdot s&c\cdot q+d\cdot t\\f\cdot o+g\cdot r&f\cdot p+g\cdot s&f\cdot q+g\cdot t\end{pmatrix}$$

$$\left|\begin{pmatrix}a&0&0\\0&b&1\\0&1&c\end{pmatrix}\right|\to a\cdot(b\cdot c-1)$$

$$\begin{pmatrix}a&b\\c&d\end{pmatrix}^{-1}\to\begin{bmatrix}\dfrac{d}{(a\cdot d-b\cdot c)}&\dfrac{-b}{(a-d-b\cdot c)}\\\dfrac{-c}{(a\cdot d-b\cdot c)}&\dfrac{a}{(a\cdot d-b\cdot c)}\end{bmatrix}$$

$$\begin{pmatrix}a\\b\\c\end{pmatrix}\times\begin{pmatrix}r\\s\\t\end{pmatrix}\to\begin{pmatrix}b\cdot t-c\cdot s\\c\cdot r-a\cdot t\\a\cdot s-b\cdot r\end{pmatrix}$$

$$\sum \begin{pmatrix} a \\ b \\ c \end{pmatrix} \rightarrow a + b + c$$

Совет

Смело используйте символьный процессор в качестве мощного математического справочника. Например, когда Вы хотите вспомнить какое-либо определение из области линейной алгебры (так, правила перемножения и обращения матриц показаны в первых строках листинга 9.18).

9.2. Матричные функции

Перечислим основные встроенные функции, предназначенные для облегчения работы с векторами и матрицами. Они нужны для создания матриц, слияния и выделения части матриц, получения основных свойств матриц и т. п.

9.2.1. Функции создания матриц

Самым наглядным способом создания матрицы или вектора является применение первой кнопки панели инструментов **Matrix** (Матрицы) *(см. разд. "Массивы" гл. 4)*. Однако в большинстве случаев, в частности при программировании сложных проектов, удобнее бывает создавать массивы с помощью встроенных функций.

Определение элементов матрицы через функцию

- `matrix(M,N,f)` — создание матрицы размера M×N, каждый i,j элемент которой есть `f(i,j)` (листинг 9.19);
  - `M` — количество строк;
  - N — количество столбцов;
  - f (i, j) — функция.

Листинг 9.19. Создание матрицы

$f(i,j) := i + 0.5 \cdot j$

$A := \mathrm{matrix}(2, 3, f)$

$$A = \begin{pmatrix} 0 & 0.5 & 1 \\ 1 & 1.5 & 2 \end{pmatrix}$$

Для создания матриц имеются еще две специфические функции, применяемые, в основном, для быстрого и эффектного представления каких-либо зависимостей в виде трехмерных графиков (типа поверхности или пространственной кривой). Все их аргументы, кроме первого (функции), необязательны. Рассмотрим первую из функций.

- CreateSpace(F(или f1,f2 ,f3) ,t0 ,t1,tgrid,fmap) — Создание вложенного массива, представляющего x-, y- и z-координаты параметрической пространственной кривой, заданной функцией F;
  - F(t) — векторная функция из трех элементов, заданная параметрически относительно единственного аргумента t;
  - f1 (t),f2 (t),f3(t) — скалярные функции;
  - to — нижний предел t (по умолчанию -5);
  - t1 — верхний предел t (по умолчанию 5);
  - tgrid — число точек сетки по переменной t (по умолчанию 20);
  - fmap — векторная функция от трех аргументов, задающая преобразование координат.

Примечание

О вложенных массивах читайте в разд. "Создание тензора" гл. 4,

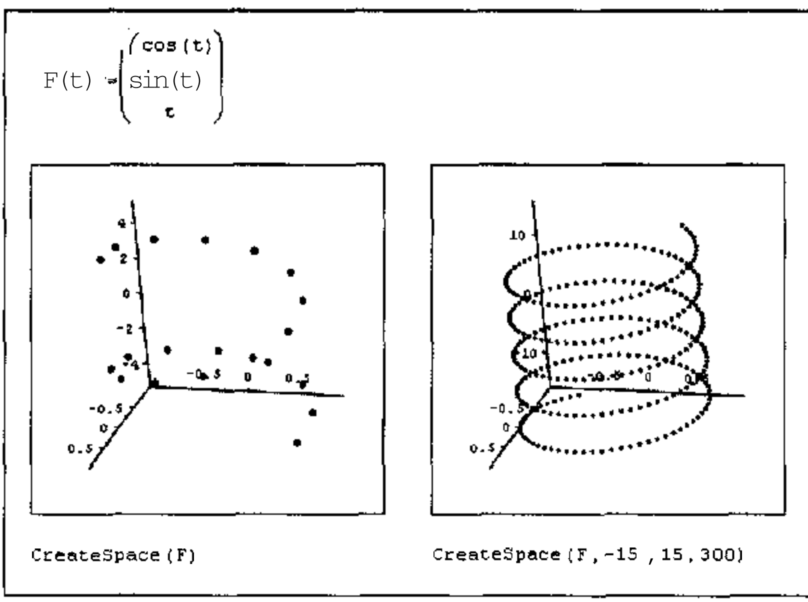
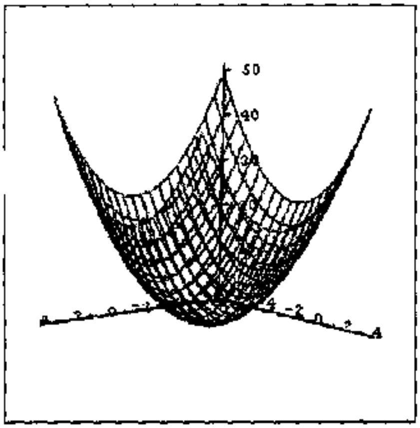


Рис. Э.4. Использование функции CreateSpace с разным набором параметров

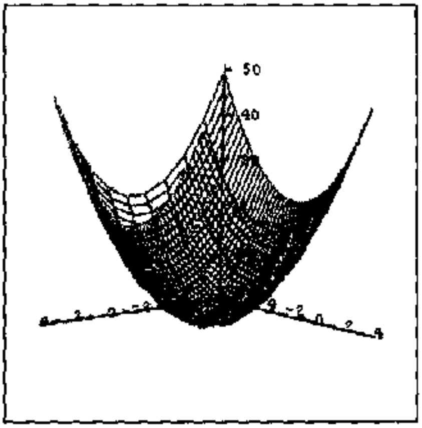
Пример использования функции createspace показан на рис. 9.4. Заметьте, для построения графика спирали не потребовалось никакого дополнительного кода, кроме определения параметрической зависимости в вектор-функции F!

Функция создания матрицы для графика трехмерной поверхности устроена совершенно аналогично, за тем исключением, что для определения поверхности требуется не одна, а две переменных. Пример ее использования иллюстрирует рис. 9.5.

$g(s,t) := s^2 + t^2$



CreateMesh (g)



CreateMesh(g, 10, 50)

Рис. 9.5. Использование функции CreateMesh с разным набором параметров

- CreateMesh(F (или g, или f1,f2,f3),s0,s1,t0,t1,sgrid,tgrid,fmap) — создание вложенного массива, представляющего x-, y- и z-координаты параметрической поверхности, заданной функцией F;
  - F(s,t) — векторная функция из трех элементов, заданная параметрически относительно двух аргументов s и t;
  - g(s,t) — скалярная функция;
  - f1(s,t),f2(s,t),f3(s,t) — скалярные функции;
  - s0, t0 — нижние пределы аргументов s, t (по умолчанию -5);
  - s1,t1 — верхние пределы аргументов s, t (по умолчанию 5);
  - sgrid, tgrid — число точек сетки по переменным s и t (по умолчанию 20);

* fmap — векторная функция из трех элементов от трех аргументов, задающая преобразование координат.

Примеры вложенных массивов, которые создаются функциями CreateMesh и createspace, приведены в листинге 9.20. Каждая матрица из числа трех вложенных матриц, образующих массив, определяет x-, y- и z-координаты точек поверхности или кривой, соответственно.

Листинг 9.20. Результат действия функций CreateMesh и CreateSpace (рис. 9.4—9.5)

$$\mathrm{CreateMesh}(g,2,3)=\left[\left[\begin{pmatrix}-5 & -5 & -5\\ 5 & 5 & 5\end{pmatrix}\right.\right.$$

$$\begin{pmatrix}-5 & 0 & 5\\ -5 & 0 & 5\end{pmatrix}$$

$$\left.\left.\begin{pmatrix}50 & 25 & 50\\ 50 & 25 & 50\end{pmatrix}\right]\right]$$

$$\mathrm{CreateSpace}(F,3)=\left[\left[\begin{pmatrix}0.284\\ 1\\ 0.284\end{pmatrix}\right.\right.$$

$$\begin{pmatrix}0.959\\ 0\\ -0.959\end{pmatrix}$$

$$\left.\left.\begin{pmatrix}-5\\ 0\\ 5\end{pmatrix}\right]\right]$$

Создание матриц специального вида

В Mathcad легко создать матрицы определенного вида с помощью одной из встроенных функций. Примеры использования этих функций приведены в листинге 9.21.

- identity(N) — единичная матрица размера N×N;
- diag(v) — диагональная матрица, на диагонали которой находятся элементы вектора v;
- geninv(A) — создание матрицы, обратной (слева) матрице A;

- rref (A) — преобразование матрицы или вектора A в ступенчатый вид;
  - N — целое число;
  - V — вектор;
  - A — матрица из действительных чисел.

Примечание

Размер N×M матрицы A ДЛЯ функции geninv должен быть таким, чтобы N≥M.

Листинг 0.21. Создание матриц специального вида

$$\mathrm{identity}(2) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\mathrm{diag}\left(\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}\right) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

$$A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

$$\mathrm{geninv}(A) = \begin{pmatrix} -1.333 & -0.333 & 0.667 \\ 1.083 & 0.333 & -0.417 \end{pmatrix} \qquad \mathrm{geninv}(A) \cdot A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\mathrm{rref}(A) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$\mathrm{rref}\left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}\right) = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \end{pmatrix}$$

9.2.2. Слияние и разбиение матриц

Из матрицы или вектора можно выделить либо подматрицу, либо вектор-столбец, либо отдельный элемент. И обратно, можно "склеить" несколько матриц в одну.

Выделение части матрицы

Часть матрицы выщеляется одним из следующих способов:

- для выделения одного элемента предназначен оператор нижнего индекса *(подробнее об этом операторе рассказано е разд. "Доступ к элементам массива" гл. 4).* Оператор вводится нажатием кнопки **Subscript** (Нижний индекс) со значком x_n на панели **Matrix** (Матрица), либо нажатием клавиши <[> (листинг 9.22, вторая строка сверху);

- для выделения из матрицы столбца примените оператор выделения столбца нажатием кнопки **Matrix Column** с изображением угловых скобок О на панели **Matrix,** либо сочетанием клавиш <Ctrl>+<6> (листинг 9.22). Этот оператор называют еще, по аналогии с предыдущим, оператором *верхнего индекса;*
- чтобы выделить из матрицы строку, примените тот же оператор О к транспонированной матрице (листинг 9.22, снизу);
- для выделения подматрицы используйте встроенную функцию `submatrix(A,ir,jr,ic,jc)`, возвращающую часть матрицы A, находящуюся между строками `ir`, `jr` и столбцами `ic`, `jc` включительно (листинг 9.23).

Примечание

Выделить из матрицы один столбец или строку можно и с помощью функции `submatrix`.

Листинг 9.22. Доступ к отдельным элементам, столбцам и строкам матрицы

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

$$A_{0,1} = 2 \qquad A_{1,1} = 5$$

$$A^{\langle 2 \rangle} = \begin{pmatrix} 3 \\ 6 \end{pmatrix} \qquad A^{\langle 0 \rangle} = \begin{pmatrix} 1 \\ 4 \end{pmatrix} \qquad A^{\langle 1 \rangle} = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$$

$$\left(A^T\right)^{\langle 0 \rangle} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \qquad \left(A^1\right) = (1 \quad 2 \quad 3$$

Листинг 9.23. Выделение подматрицы

$$\mathrm{submatrix}\left[\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, 0, 1, 0, 1\right] = \begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix}$$

$$\mathrm{submatrix}\left[\begin{pmatrix} 1 & 0 & 0 \\ -1 & -3 & -4 \end{pmatrix}, 0, 0, 0, 1\right] = (1 \quad 0)$$

Те же операции применимы к матрицам-векторам и матрицам-строкам. Следует помнить только, что размер их составляет `N×1` И `1×N`, соответственно (листинг 9.24).

Листинг 9.24. Выделение частей из векторов и строк

$$(1\ \ 2\ \ 3)^{\langle 0\rangle} = (1) \qquad \left(\begin{pmatrix}1\\2\\3\end{pmatrix}^T\right)^{\langle 1\rangle} = (2)$$

$$\mathrm{submatrix}\left[\begin{pmatrix}1\\2\\3\end{pmatrix}, 0, 1, 0, 0\right] = \begin{pmatrix}1\\2\end{pmatrix}$$

Слияние матриц

Для того чтобы составить из двух или более матриц одну, в Mathcad предусмотрены две матричные функции (листинг 9.25):

- `augment(A,B,C,...)` — матрица, сформированная слиянием матриц-аргументов слева направо;
- `stack(A,B,C,...)` — матрица, сформированная слиянием матриц-аргументов сверху вниз;
  - `A,B,C, ...` — векторы или матрицы соответствующего размера.

Листинг 9.25. Примеры слияния матриц

$$A := \begin{pmatrix}1 & 2 & 3\\4 & 5 & 6\end{pmatrix} \qquad B := \begin{pmatrix}0 & 0 & 0\\0 & 0 & 0\end{pmatrix}$$

$$\mathrm{stack}(A, B) = \begin{pmatrix}1 & 2 & 3\\4 & 5 & 6\\0 & 0 & 0\\0 & 0 & 0\end{pmatrix}$$

$$\mathrm{augment}(A, B) = \begin{pmatrix}1 & 2 & 3 & 0 & 0 & 0\\4 & 5 & 6 & 0 & 0 & 0\end{pmatrix}$$

$$\mathrm{augment}\left[\begin{pmatrix}1\\1\end{pmatrix}, A, B, \begin{pmatrix}2\\3\end{pmatrix}\right] = \begin{pmatrix}1 & 1 & 2 & 3 & 0 & 0 & 0 & 3\\1 & 4 & 5 & 6 & 0 & 0 & 0 & 3\end{pmatrix}$$

9.2.3. Вывод размера матриц

Для получения сведений о характеристиках матриц или векторов предусмотрены следующие встроенные функции (листинг 9.26):

- rows (A) — число строк;

- cols (A) — число столбцов;
- length (v) — число элементов вектора;
- last(v) — индекс последнего элемента вектора;
  - A — матрица или вектор;
  - v — вектор.

Примечание

Число элементов вектора и индекс его последнего элемента совпадают, если индексы нумеруются с 1, т. е. системная константа ORIGIN равна 1 *(см. гл. 4).*

Листинг 9.26. Размер матриц и векторов

$w := (1\ \ 2\ \ 3)$

$A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$ $\qquad v := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$

```
rows (A) =3          rows (v) =3          rows (w) = 1
cols (A) = 2         cols (v) = 1         cols (w) = 3
length (v) = 3
last (v) = 2
```

9.2.4. Сортировка матриц

Часто бывает нужно переставить элементы матрицы или вектора, расположив их в определенной строке или столбце в порядке возрастания или убывания. Для этого имеются несколько встроенных функций, которые позволяют гибко управлять сортировкой матриц:

- sort(v) — сортировка элементов вектора в порядке возрастания (листинг 9.27);
- csort(A,i) — сортировка строк матрицы выстраиванием элементов 1-го столбца в порядке возрастания (листинг 9.28);
- rsort(A,i) — сортировка столбцов матрицы выстраиванием элементов i-й строки в порядке возрастания (листинг 9.29);
- reverse(v) — перестановка элементов вектора в обратном порядке (листинг 9.27);
  - v — вектор;
  - A — матрица;
  - i — индекс строки или столбца.

Примечание

Если элементы матриц или векторов комплексные, то сортировка ведется по действительной части, а мнимая часть игнорируется.

Листинг 9.27. Сортировка векторов

$$v := \begin{pmatrix} 3 \\ 1 \\ 4 \\ 2 \end{pmatrix} \qquad \mathrm{sort}(v) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \qquad \mathrm{reverse}(v) = \begin{pmatrix} 2 \\ 4 \\ 1 \\ 3 \end{pmatrix}$$

Листинг 9.28. Сортировка матриц по столбцу

$$A := \begin{pmatrix} 1 & 9 \\ 3 & 0 \\ 2 & 8 \end{pmatrix}$$

$$\mathrm{csort}(A, 1) = \begin{pmatrix} 3 & 0 \\ 2 & 8 \\ 1 & 9 \end{pmatrix} \qquad \mathrm{csort}(A, 0) = \begin{pmatrix} 1 & 9 \\ 2 & 8 \\ 3 & 0 \end{pmatrix}$$

Листинг 9.29. Сортировка матриц по строке (матрица A из листинга 9.28)

$$\mathrm{rsort}(A, 1) = \begin{pmatrix} 9 & 1 \\ 0 & 3 \\ 8 & 2 \end{pmatrix} \qquad \mathrm{rsort}(A, 2) = \begin{pmatrix} 1 & 9 \\ 3 & 0 \\ 2 & 8 \end{pmatrix}$$

9.2.5. Норма квадратной матрицы

В линейной алгебре используются различные матричные *нормы* (norm), которые ставят в соответствие матрице некоторую скалярную числовую характеристику. Норма матрицы отражает порядок величины матричных элементов. В разных специфических задачах линейной алгебры применяются различные виды норм. Mathcad имеет четыре встроенные функции для расчета разных норм квадратных матриц:

- `norm1(A)` — норма в пространстве L1;
- `norm2(A)` — норма в пространстве L2;
- `norme(A)` — евклидова норма (euclidean norm);
- `normi(A)` — max-норма, или ∞-норма (infinity norm);
  - A — квадратная матрица.

Примеры расчета различных норм двух матриц А и в с различающимися на два порядка элементами приведены в листинге 9.30. В последней строке этого листинга пояснено определение евклидовой нормы, которое похоже на определение длины вектора.

Совет

В большинстве задач неважно, какую норму использовать. Как видно, в обычных случаях разные нормы дают примерно одинаковые значения, хорошо отражая порядок величины матричных элементов. Определение остальных норм заинтересованный читатель отыщет в справочниках по линейной алгебре или в справочной системе Mathcad (раздел **Mathcad Resources**).

Листинг 9.30. Нормы матриц

$A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ $\qquad B := \begin{pmatrix} 100 & 200 \\ 300 & 400 \end{pmatrix}$

$norm1(A) = 6$ $\qquad norm1(B) = 600$

$norm2(A) = 5.465$ $\qquad norm2(B) = 546.499$

$normi(A) = 7$ $\qquad normi(B) = 700$

$norme(A) = 5.477$ $\qquad norme(B) = 547.723$

$\sqrt{1^2 + 2^2 + 3^2 + 4^2} = 5.477$

9.2.6. Число обусловленности квадратной матрицы

Еще одной важной характеристикой матрицы является ее *число обусловленности* (condition number). Число обусловленности является мерой чувствительности системы линейных уравнений А • x=ь, определяемой матрицей А, К погрешностям задания вектора ь правых частей уравнений. Чем больше число обусловленности, тем сильнее это воздействие и тем более неустойчив процесс нахождения решения. Число обусловленности связано с нормой матрицы и вычисляется по-разному для каждой из норм;

- cond1(A) — число обусловленности в норме L1;
- cond2 (A) — число обусловленности в норме L2;
- conde(A) — число обусловленности в евклидовой норме;
- condi(A) — число обусловленности в ∞-норме;
   - А — квадратная матрица.

Расчет чисел обусловленности для двух матриц A и B показан в листинге 9.31. Обратите внимание, что первая из матриц является хорошо обусловленной, а вторая — плохо обусловленной (две ее строки определяют очень близкие системы уравнений, с точностью до множителя з). Вторая строка листинга дает формальное определение числа обусловленности как произведения норм исходной и обратной матриц. В других нормах определение точно такое же.

Примечание

Как нетрудно понять, матрицы A и в из предыдущего листинга 9.30 обладают одинаковыми числами обусловленности, т. к. B=100·A, И, следовательно, обе матрицы определяют одну и ту же систему уравнений.

Листинг 9.31. Числа обусловленности матриц

$A := \begin{pmatrix} 1 & \\ 3 & \end{pmatrix}$:) $B := \begin{pmatrix} 1 & 2 \\ 3 & 6.01 \end{pmatrix}$

$\mathrm{norme}(A) \cdot \mathrm{norme}(A^{-1}) = 15$

$\mathrm{conde}(A) = 15$ $\mathrm{conde}(B) = 5.012 \times 10^3$

$\mathrm{cond1}(A) = 21$ $\mathrm{cond1}(B) = 7.217 \times 10^3$

$\mathrm{cond2}(A) = 14.933$ $\mathrm{cond2}(B) = 5.012 \times 10^3$

$\mathrm{condi}(A) = 21$ $\mathrm{condi}(B) = 7.217 \times 10^3$

9.2.7. Ранг матрицы

Рангом (rank) матрицы называют наибольшее натуральное число к, для которого существует не равный нулю определитель k-ГО порядка подматрицы, составленной из любого пересечения k столбцов и к строк матрицы.

Для вычисления ранга в Mathcad предназначена функция rank.

- rank (A) — ранг матрицы;
  - A — матрица.

Листинг 9.32. Ранг матрицы

$\mathrm{rank}\begin{pmatrix} 1 & 2 \\ 3 & 6 \end{pmatrix} = 1$ $\mathrm{rank}\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = 2$

$$\mathrm{rank}\begin{pmatrix}1 & 2\\ 3 & 6\\ 1 & 2\end{pmatrix} = 1 \qquad \mathrm{rank}\begin{pmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{pmatrix} = 2$$

9.3. Системы линейных алгебраических уравнений

Центральным вопросом вычислительной линейной алгебры является решение *систем линейных алгебраических уравнений* (СЛАУ), т. е. систем уравнений вида

$$a_{i1} \cdot x_1 + a_{i2} \cdot x_2 + \ldots + a_{iN} \cdot x_N = b_i. \qquad (1)$$

В матричной форме СЛАУ записывается в эквивалентном виде:

$$A \cdot x = b, \qquad (2)$$

где A — матрица коэффициентов СЛАУ размерности N×N, X — вектор неизвестных, b — вектор правых частей уравнений.

Примечание

К системам линейных уравнений сводится множество, если не сказать большинство, задач вычислительной математики. *Один из таких примеров приведен в разд. "Разностные схемы для ОДУ" гл. 12.*

СЛАУ имеет единственное решение, если матрица A является *невырожденной*, или, по-другому, *несингулярной,* т. е. ее определитель не равен нулю. С вычислительной точки зрения, решение СЛАУ не представляет трудностей, если матрица A не очень велика. С большой матрицей проблем также не возникнет, если она не очень плохо обусловлена. В Mathcad СЛАУ можно решить как в более наглядной форме (1), так и в более удобной для записи форме (2). Для первого способа следует использовать вычислительный блок Given/Find *(см. гл. 8),* а для второго — встроенную функцию lsolve.

- lsolve(A,b) — решение системы линейных уравнений;
  - A — матрица коэффициентов системы;
  - b — вектор правых частей.

Применение функции lsolve показано в листинге 9.33. При этом матрица A может быть определена любым из способов *(см. разд. "Массивы" гл. 4),* необязательно явно, как во всех примерах этого раздела. Встроенную функцию lsolve допускается применять и при символьном решении СЛАУ (листинг 9.34).

Примечание

Соответствующая матрице A и вектору ь система уравнений выписана явно в листинге 9.35.

Листинг 9.33. Решение СЛАУ

$$A := \begin{pmatrix} 1 & 5 & 2 \\ 0.7 & 12 & 5 \\ 3 & 0 & 4 \end{pmatrix} \qquad Ь := \begin{pmatrix} 1 \\ 2.9 \\ 3.1 \end{pmatrix}$$

$$\mathtt{lsolve}(A, Ь) = \begin{pmatrix} -0.186 \\ -0.129 \\ 0.915 \end{pmatrix}$$

Листинг 9.34. Символьное решение СЛАУ (продолжение листинга 9.33)

$$\mathtt{lsolve}(A, Ь) \rightarrow \begin{pmatrix} -.18648648648648648648649 \\ -.12864864864864864864865 \\ .91486486486486486486486 \end{pmatrix}$$

В некоторых случаях, для большей наглядности представления СЛАУ, его можно решить точно так же, как систему нелинейных уравнений *(см. гл. 8)*. Пример численного решения СЛАУ из предыдущих листингов показан в листинге 9.35. Не забывайте, что при численном решении всем неизвестным требуется присвоить начальные значения (это сделано в первой строке листинга 9.35). Они могут быть произвольными, т. к. решение СЛАУ с невырожденной матрицей единственно.

Примечание

При решении СЛАУ с помощью функции Find Mathcad автоматически выбирает линейный численный алгоритм, в чем можно убедиться, вызывая на имени Find контекстное меню.

Листинг 9.35. Решение СЛАУ с помощью вычислительного блока

$x := 0 \qquad y := 0 \qquad z := 0$

Given

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

$0.7x + 12y + 5z = 2.9$

$3x + 0y + 4z = 3.1$

$$\mathtt{find}(x, y, z) = \begin{pmatrix} -0.186 \\ -0.129 \\ 0.915 \end{pmatrix}$$

9.4. Собственные векторы и собственные значения матриц

Вторая по частоте применения задача вычислительной линейной алгебры — это задача поиска собственных векторов x и собственных значений Я матрицы A, т. е. решения матричного уравнения $A \cdot x = \lambda \cdot x$. Такое уравнение имеет решения в виде собственных значений $\lambda1, \lambda2, \ldots$ и соответствующих им собственных векторов x1, x2, . .. Для решения таких задач на собственные векторы и собственные значения в Mathcad встроено несколько функций, реализующих довольно сложные вычислительные алгоритмы:

- eigenvals(A) — вычисляет вектор, элементами которого являются собственные значения матрицы A;
- eigenvecs (A) — вычисляет матрицу, содержащую нормированные собственные векторы, соответствующие собственным значениям матрицы A;

  п-й столбец вычисляемой матрицы соответствует собственному вектору п-го собственного значения, вычисляемого eigenvals;
- eigenvec(A,λ) — вычисляет собственный вектор для матрицы A и заданного собственного значения *X;*

  • A — квадратная матрица.

Применение этих функций иллюстрирует листинг 9.36. Проверка правильности нахождения собственных векторов и собственных значений приведена в листинге 9.37. Причем проверка правильности выражения $A \cdot x = \lambda \cdot x$ проведена дважды — сначала на числовых значениях x и *X,* а потом путем перемножения соответствующих матричных компонентов.

Листинг 9.36. Поиск собственных векторов и собственных значений

$$A := \begin{pmatrix} 1 & 5 & 2 \\ 0.7 & 12 & 5 \\ 3 & 0 & 4 \end{pmatrix}$$

$$\mathrm{eigenvals}(A) = \begin{pmatrix} 0.938 \\ 3.024 \\ 13.037 \end{pmatrix}$$

$$\mathrm{eigenvecs}(A) = \begin{pmatrix} 0.68936 & -0.27605 & 0.3989 \\ 0.26171 & -0.45121 & 0.90738 \\ -0.6755 & 0.84865 & 0.13242 \end{pmatrix}$$

$$\mathrm{eigenvec}(A, 0.938) = \begin{pmatrix} -0.68936 \\ -0.26171 \\ 0.6755 \end{pmatrix}$$

$$\mathrm{eigenvec}(A, 3.024) = \begin{pmatrix} -0.27605 \\ -0.45121 \\ 0.84865 \end{pmatrix}$$

Листинг 9.37. Проверка правильности нахождения собственных векторов собственных значений *(продолжение листинга 9.36)*

$$\begin{pmatrix} 1 & 5 & 2 \\ 0.7 & 12 & 5 \\ 3 & 0 & 4 \end{pmatrix} \cdot \begin{pmatrix} 0.68936 \\ 0.26171 \\ -0.6755 \end{pmatrix} = \begin{pmatrix} 0.647 \\ 0.246 \\ -0.634 \end{pmatrix}$$

$$0.938 \cdot \begin{pmatrix} 0.68936 \\ 0.26171 \\ -0.6755 \end{pmatrix} = \begin{pmatrix} 0.647 \\ 0.245 \\ -0.634 \end{pmatrix}$$

$$A \cdot \mathrm{eigenvecs}(A)^{\langle 0 \rangle} = \begin{pmatrix} 0.647 \\ 0.246 \\ -0.634 \end{pmatrix}$$

$$\mathrm{eigenvals}(A)_0 \cdot \mathrm{eigenvecs}(A)^{\langle 0 \rangle} = \begin{pmatrix} 0.647 \\ 0.246 \\ -0.634 \end{pmatrix}$$

Помимо рассмотренной проблемы поиска собственных векторов и значений, иногда рассматривают более общую задачу, называемую задачей на *обобщенные собственные значения*: $A \cdot x = \lambda \cdot B \cdot x$. В ее формулировке помимо матрицы A присутствует еще одна квадратная матрица в. Для задачи на обобщенные собственные значения имеются еще две встроенные функции, действие которых аналогично рассмотренным (листинги 9.38 и 9.39):

- genvals(A,B) — вычисляет вектор v собственных значений, каждый из которых удовлетворяет задаче на обобщенные собственные значения;
- genvecs(A,B} — вычисляет матрицу, содержащую нормированные собственные векторы, соответствующие собственным значениям в векторе v, который вычисляется с помощью `genvals`. В этой матрице i-й столбец является собственным вектором x, удовлетворяющим задаче на обобщенные собственные значения;
  - `A, B` — квадратные матрицы.

Листинг 9.38. Поиск обобщенных собственных векторов и собственных значений

$$A := \begin{pmatrix} 1 & 5 & 2 \\ 0.7 & 12 & 5 \\ 3 & 0 & 4 \end{pmatrix} \qquad B := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

$$\mathrm{genvals}(A, B) = \begin{pmatrix} 0.5 \\ -0.674 \end{pmatrix}$$

$$\mathrm{genvecs}(A, B) = \begin{pmatrix} -0.3067 & -0.70544 \\ 0.15571 & -0.23715 \\ -0.93898 & 0.66792 \end{pmatrix}$$

Листинг 9.39. Проверка правильности нахождения собственных векторов и собственных значений (*продолжение листинга 9.38*)

$$A \cdot \mathrm{genvecs}(A, B)^{\langle 0 \rangle} = \begin{pmatrix} -1.406 \\ -3.041 \\ -4.676 \end{pmatrix}$$

$$\mathrm{genvals}(A, B)_0 \cdot B \cdot \mathrm{genvecs}(A, B)^{\langle 0 \rangle} = \begin{pmatrix} -1.406 \\ -3.041 \\ -4.676 \end{pmatrix}$$

$$A \cdot \mathrm{genvecs}(A, B)^{\langle 1 \rangle} = \begin{pmatrix} -0.555 \\ 0 \\ 0.555 \end{pmatrix}$$

$$\mathrm{genvals}(A, B)_1 \cdot B \cdot \mathrm{genvecs}(A, B)^{\langle 1 \rangle} = \begin{pmatrix} -0.555 \\ 0 \\ 0.555 \end{pmatrix}$$

9.5. Матричные разложения

Современная вычислительная линейная алгебра — бурно развивающаяся наука. Главная проблема, рассматриваемая ею, — это проблема решения систем линейных уравнений. В настоящее время разработано множество методов, упрощающих эту задачу, которые, в частности, зависят от структуры матрицы СЛАУ. Большинство методов основано на представлении матрицы в виде произведения других матриц специального вида или *матричных разложениях*. Как правило, после определенного разложения матрицы задача линейной алгебры существенно упрощается. В Mathcad имеется несколько встроенных функций, реализующих алгоритмы наиболее популярных матричных разложений.

9,5.1. Разложение **Холецкого**

Разложением Холецкого симметричной матрицы A является представление вида $A=L \cdot L^T$, где L — *треугольная матрица* (т. е. матрица, по одну из сторон от диагонали которой находятся одни нули). Алгоритм Холецкого реализован ВО Встроенной **функции** cholesky.

- `cholesky(A)` — разложение Холецкого;
  - A — квадратная, положительно-определенная матрица.

Пример разложения Холецкого приведен в листинге 9.40. Обратите внимание, что в результате получается верхняя треугольная матрица (нули сверху от диагонали), а транспонированная матрица является нижней треугольной. В последней строке листинга приведена проверка правильности найденного разложения.

Листинг 9.40. Разложение Холецкого

$$A:=\begin{pmatrix} 13 & 7 & 4 \\ 7 & 9 & -3 \\ 4 & -3 & 9 \end{pmatrix}$$

L := cholesky (A)

$$L=\begin{pmatrix} 3.606 & 0 & 0 \\ 1.941 & 2.287 & 0 \\ 1.109 & -2.253 & 1.64 \end{pmatrix}$$

$$L \cdot L^T=\begin{pmatrix} 13 & 7 & 4 \\ 7 & 9 & -3 \\ 4 & -3 & 9 \end{pmatrix}$$

9.5.2. QR-разложение

QR-разложением матрицы A называется разложение вида $A=Q \cdot R$, где Q — ортогональная матрица, а R — верхняя треугольная матрица.

- qr(A) —QR-разложение;
  - A — вектор или матрица любого размера.

Результатом действия функции qr(A) является матрица L, составленная из матриц Q и R, соответственно. Чтобы выделить сами матрицы QR-разложения, необходимо применить функцию выделения подматрицы submatrix (листинг 9.41).

Листинг 9.41. QR-разложение

$$A:=\begin{pmatrix} 13 & 7 & 4 \\ 7 & 9 & -3 \\ 4 & -3 & 9 \end{pmatrix}$$

$$L:=\mathrm{qr}(A)$$

$$L=\begin{pmatrix} 0.85 & 0.122 & 0.513 & 15.297 & 9.283 & 4.38 \\ 0.458 & -0.654 & -0.603 & 0 & -7.268 & 9.171 \\ 0.261 & 0.747 & -0.612 & 0 & 0 & -1.646 \end{pmatrix}$$

$$Q:=\mathrm{submatrix}\left(L, 0, \mathrm{rows}(L)-1, 0, \frac{\mathrm{cols}(L)-1}{2}\right)$$

$$Q=\begin{pmatrix} 0.85 & 0.122 & 0.513 \\ 0.458 & -0.654 & -0.603 \\ 0.261 & 0.747 & -0.612 \end{pmatrix}$$

$$R:=\mathrm{submatrix}\left(L, 0, \mathrm{rows}(L)-1, \frac{\mathrm{cols}(L)+1}{2}, \mathrm{cols}(L)-1\right)$$

$$R=\begin{pmatrix} 15.297 & 9.283 & 4.38 \\ 0 & -7.268 & 9.171 \\ 0 & 0 & -1.646 \end{pmatrix}$$

$$Q\cdot R=\begin{pmatrix} 13 & 7 & 4 \\ 7 & 9 & -3 \\ 4 & -3 & 9 \end{pmatrix}$$

9.5.3. LU-разложение

LU-разложением матрицы A, или *треугольным* разложением, называется матричное разложение вида $P \cdot A = L \cdot U$, где L и U — нижняя и верхняя треугольные матрицы, соответственно. P, A, L, U — квадратные матрицы одного порядка.

- `lu(A)` — LU-разложение матрицы;
  - A — квадратная матрица.

Примечание

Фактически, треугольное разложение матрицы системы линейных уравнений производится при ее решении численным методом Гаусса.

Функция LU-разложения, подобно предыдущей функции QR-разложения, выдает составную матрицу в (листинг 9.42). Выделить матрицы P, L, U несложно при помощи встроенной функции submatrix.

Листинг 9.42. LU-разложение

$$A := \begin{pmatrix} 13 & 7 & 4 \\ 7 & 9 & -3 \\ 4 & -3 & 9 \end{pmatrix}$$

$B := lu(A)$

$$B = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 13 & 7 & 4 \\ 0 & 1 & 0 & 0.538 & 1 & 0 & 0 & 5.231 & -5.154 \\ 0 & 0 & 1 & 0.308 & -0.985 & 1 & 0 & 0 & 2.691 \end{pmatrix}$$

$$P := \mathrm{submatrix}\left(B, 0, \mathrm{rows}(B) - 1, 0, \frac{\mathrm{cols}(B) - 1}{3}\right)$$

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$L := \mathrm{submatrix}\left(B, 0, \mathrm{rows}(B) - 1, \frac{\mathrm{cols}(B) + 1}{3}, \frac{\mathrm{cols}(B) - 1}{3} \cdot 2\right)$$

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0.538 & 1 & 0 \\ 0.308 & -0.985 & 1 \end{pmatrix}$$

$$U := \mathrm{submatrix}\left(B, 0, \mathrm{rows}(B) - 1, \frac{\mathrm{cols}(B) + 1}{3} \cdot 2, \mathrm{cols}(B) - 1\right)$$

$$U = \begin{pmatrix} 13 & 7 & 4 \\ 0 & 5.231 & -5.154 \\ 0 & 0 & 2.691 \end{pmatrix}$$

$$P \cdot A - L \cdot U = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

9.5.4. Сингулярное разложение

Сингулярным разложением (singular value decomposition) матрицы A размера N×M (причем N≥M) является разложение вида $A=U \cdot s \cdot V^T$, где U и V — ортогональные матрицы размером N×N и M×M, соответственно, а s — диагональная матрица с сингулярными числами матрицы A на диагонали.

- svds(A) — вектор, состоящий из сингулярных чисел;
- `svd(A)` — сингулярное разложение;
  - A — действительная матрица.

Примеры поиска сингулярных чисел невырожденной и сингулярной матрицы приведены в листингах 9.43 и 9.44, соответственно. Проверка правильности сингулярного разложения приведена в листинге 9.45. Вычисленные сингулярные числа находятся на главной диагонали средней матрицы (ее остальные элементы, по определению, равны нулю). Сравнивая матрицы из листингов 9.44 и 9.45, Вы без труда разберетесь, каким образом следует выделять искомые матрицы сингулярного разложения из результата, поставляемого функцией `svd`.

Листинг 9.43. Сингулярные числа и собственные значения невырожденной матрицы

$$A:=\begin{pmatrix} 13 & 7 & 4 \\ 7 & 9 & -3 \\ 4 & -3 & 9 \end{pmatrix}$$

$$\mathrm{eigenvals}(A)=\begin{pmatrix} 18.522 \\ 11.629 \\ 0.85 \end{pmatrix} \qquad \mathrm{svds}(A)=\begin{pmatrix} 18.522 \\ 11.629 \\ 0.85 \end{pmatrix}$$

Листинг 9.44. Сингулярное разложение сингулярной матрицы

$$A:=\begin{pmatrix} 1 & 2 & 3 \\ 3 & 6 & 9 \\ 0 & 0 & 0 \end{pmatrix} \qquad \mathrm{svds}(A)=\begin{pmatrix} 11.832 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathrm{svd}(A)=\begin{pmatrix} -0.316 & -0.949 & 0 \\ -0.949 & 0.316 & 0 \\ 0 & 0 & 1 \\ -0.267 & 0.964 & 0 \\ -0.535 & -0.148 & 0.832 \\ -0.802 & -0.222 & -0.555 \end{pmatrix}$$

Листинг 9.45. Проверка сингулярного разложения (*продолжение листинга 0.44*)

$$\begin{pmatrix} -0.316 & -0.949 & 0 \\ -0.949 & 0.316 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 11.832 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} -0.267 & 0.964 & 0 \\ -0.535 & -0.148 & 0.832 \\ -0.802 & -0.222 & -0.555 \end{pmatrix}^{T} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 6 & 9 \\ 0 & 0 & 0 \end{pmatrix}$$

ГЛАВА 10

Специальные функции

Данная глава посвящена вычислению различных математических функций, встроенных в Mathcad. Причем описываются как специальные функции (Бесселя, Эйри и т. п.), так и элементарные функции (синус, экспонента, гиперболические функции), а также функции, специфичные для тех или иных областей (финансовые функции). Кроме того, упоминаются очень простые, с точки зрения программной реализации, но часто очень полезные функции типа ступеньки, дельта-функции и т. д.

Несмотря на то, что большинство функций, о которых пойдет речь в этой главе, рассчитываются без привлечения специальных численных методов, мы совместили рассказ о них в одной главе, чтобы читателю было удобнее найти описание нужной функции. Перечень специальных функций разбит на разделы по их математическому смыслу и (или) области применения.

Примечание

Вставлять в документ не очень знакомую спецфункцию легче всего, пользуясь диалоговым окном Insert Function (Вставить функцию), которое вызывается нажатием кнопки с надписью *f(x)* на стандартной панели инструментов *(см. разд. "Знакомство с Mathcad" гл. 1).* В этом диалоге функции разбиты на несколько групп, поэтому несложно выбрать из них нужную. При выделении какой-либо группы в левом списке упомянутого диалога справа обнаруживается список функций, принадлежащих этой группе. Названия групп функций, появляющихся в левом списке диалогового окна Insert Function, приведены в скобках после названия каждого раздела этой главы.

10.1. Функции Бесселя (Bessel)

Функции Бесселя, по определению, являются решениями различных краевых задач для некоторых обыкновенных дифференциальных уравнений (ОДУ).

10.1.1. Обычные функции Бесселя

Функции Бесселя первого и второго рода обычно возникают как решения волнового уравнения с цилиндрическими граничными условиями.

Примечание

Конкретный вид соответствующих дифференциальных уравнений можно без труда отыскать в справочниках по спецфункциям или в справочной системе Mathcad.

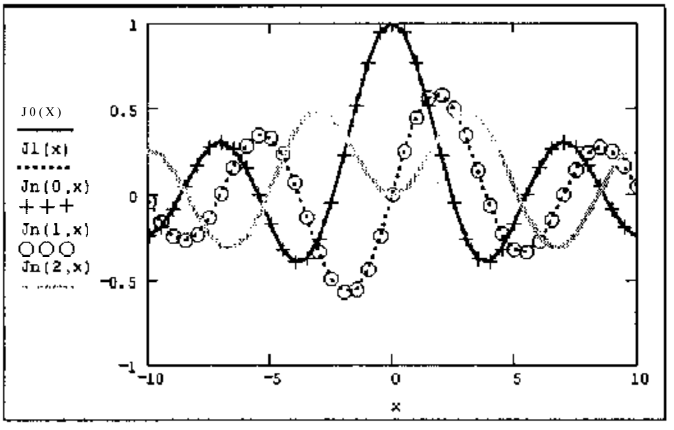


Рис. 10.1. Функции Бесселя первого рода

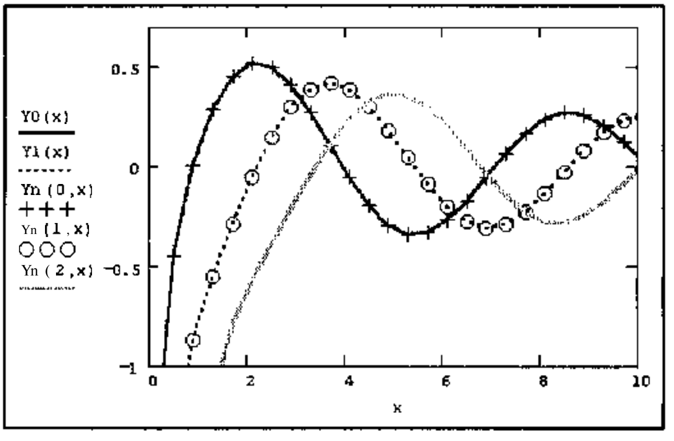


Рис. 10.2. Функции Бесселя второго рода

- JO(z) — функция Бесселя первого рода нулевого порядка;
- J1(z) — функция Бесселя первого рода первого порядка;
- Jn(m,z) — функция Бесселя m-го порядка;

- `Y0(z)` — функция Бесселя второго рода нулевого порядка, x>o;
- `Y1(z)` — функция Бесселя второго рода первого порядка, x>o;
- `Yn(m, z)` — функция Бесселя второго рода `m`-го порядка, x>o;
  - `z` — действительный или комплексный безразмерный скаляр;
  - m — порядок, целое число о<т<юо.

Внешний вид нескольких первых функций Бесселя первого и второго рода показан на рис. 10.1 и 10.2, соответственно.

10.1.2. Модифицированные функции Бесселя

Перечислим их:

- `I0(z)` — модифицированная функция Бесселя первого рода нулевого порядка;
- `I1(z)` — модифицированная функция Бесселя первого рода первого порядка;
- `In(m,z)` — модифицированная функция Бесселя первого рода m-го порядка;
- `K0(z)` — модифицированная функция Бесселя второго рода нулевого порядка, x>o;
- `K1(z)` — модифицированная функция Бесселя второго рода первого порядка, x>o;
- `Kn(m,z)` — модифицированная функция Бесселя второго рода m-го порядка, x>o;
  - `z` — действительный или комплексный безразмерный скаляр;
  - `m` — порядок, целое число `0<m<100`.

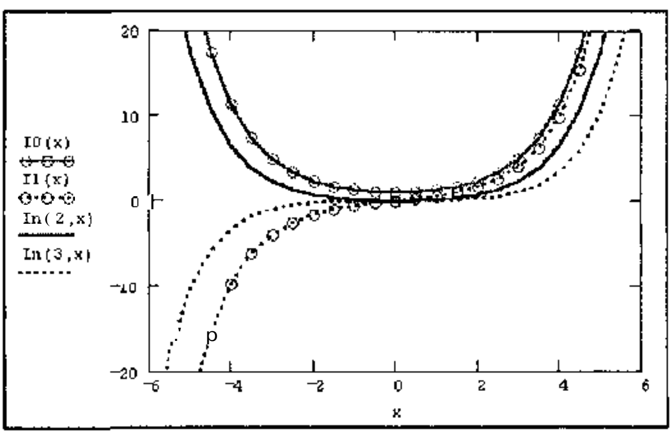


Рис. 10.3. Модифицированные функции Бесселя первого рода

Примеры нескольких первых модифицированных функций Бесселя показаны на рис. 10.3 и 10.4.

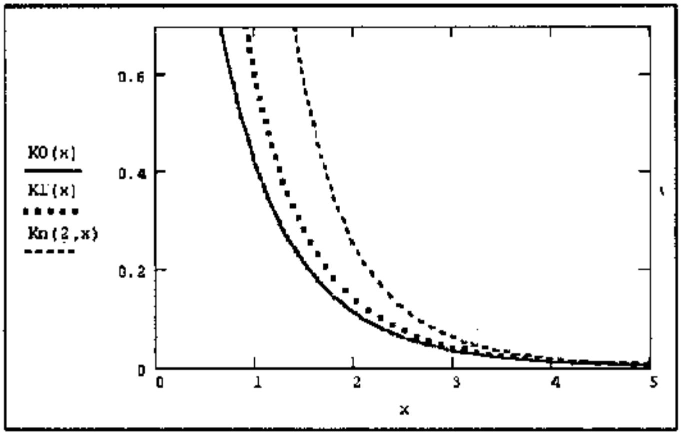


Рис. 10.4. Модифицированные функции Бесселя второго рода

10.1.3. Функции Эйри

Функции Эйри являются независимыми решениями ОДУ `y''=zy`. Их вид показан на рис. 10.5. Итак:

- Ai(z) — функция Эйри первого рода;
- Bi(z) — функция Эйри второго рода;
  - z — действительный или комплексный безразмерный скаляр, `x<103.892`.

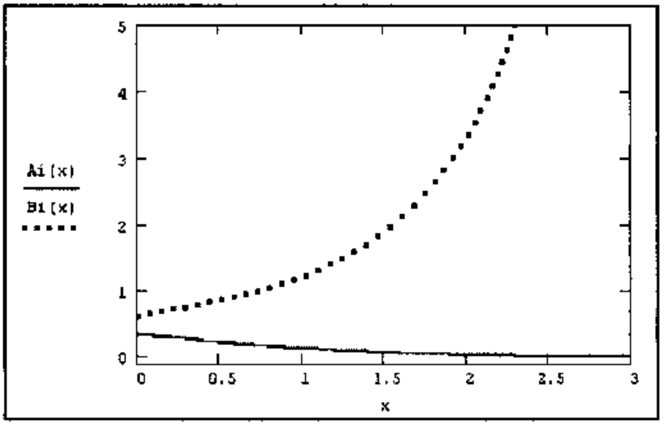


Рис. 10.5. Функции Эйри

10.1.4. Функции Бесселя-Кельвина

Комплексная комбинация функций Бесселя-Кельвина вида ber(n,x)+ +i·bei(n,x) является решением соответствующего ОДУ, зависящего от параметра n. Вид графиков функции bei для n=1 и 2 показан на рис. 10.6.

- bei (n, x) — мнимая часть функции Бесселя-Кельвина порядка n;
- ber(n,x) — действительная часть функции Бесселя-Кельвина порядка n;
  - n — порядок (безразмерное неотрицательное целое число);
  - x — действительный безразмерный скаляр.

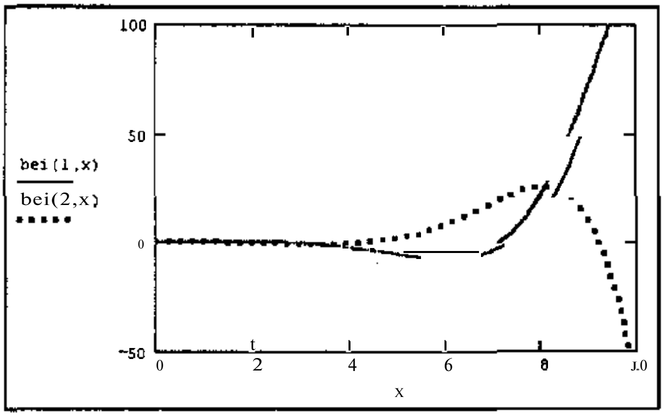


Рис. 10.6. Функции Бесселя-Кельвина

10.1.5. Сферические функции Бесселя

График сферических функций Бесселя первого порядка показан на рис. 10.7.

- js(n,z) — сферическая функция Бесселя первого рода порядка n, x>0;
- ys (n, z) — сферическая функция Бесселя второго рода порядка n, x>0;
  - n — порядок (целое число), n≥200;
  - z — действительный или комплексный безразмерный скаляр, x>0.

10.2. Функции работы с комплексными числами (Complex Numbers)

В Mathcad имеется несколько функций, облегчающих работу с комплексными числами.

- Re(z) — действительная часть комплексного числа z;
- Im(z) — мнимая часть комплексного числа z;

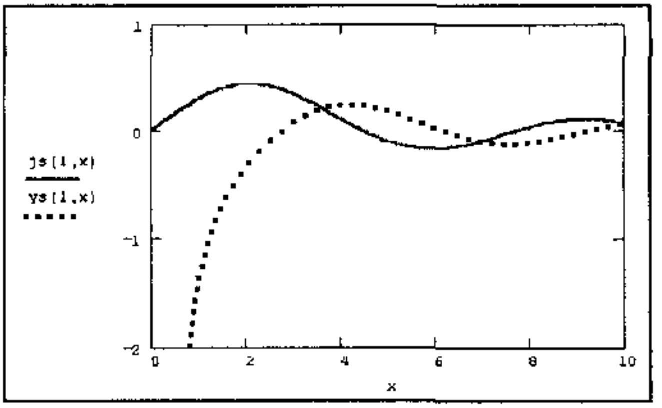


Рис. 10.7. Сферические функции Бесселя первого порядка

- arg(z) — аргумент комплексного числа z, $-\pi < \arg(z) \le \pi$;
- csgn(z) — функция комплексного знака числа (возврашает либо о, если z=0; либо 1, если Re(z)>o, или если Re(z)=0 и Im(z)>0; либо -1 — в остальных случаях);
- signum(z) — возвращает 1, если z=o, и z/|z| — в остальных случаях;
  - z — действительное, мнимое или комплексное число.

Комплексное число можно ввести как обычно, в виде суммы действительной и мнимой частей, либо как результат любого комплексного выражения. Несколько примеров действия функций работы с комплексными числами приведены в листингах 10.1—10.3.

Листинг 10.1. Базовые функции работы с комплексными числами

$\mathrm{Re}(3.9 + 2.4i) = 3.9$ $\mathrm{Im}(3.9 + 2.4i) = 2.4$

$\left|1.7 \cdot e^{0.1i}\right| = 1.7$ $\arg\left(1.7 \cdot e^{0.1i}\right) = 0.1$

Листинг 10.2. Пример действия функции csgn

csgn(0) = 0 csgn(0 - i) = -X

csgn(i) = 1 csgn(0 + i) = 1

csgn(0.1) = 1 csgn(0.1 + 2i) = 1

csgn(-0.1) = -1 csgn(-0.1 - 3i) = -1

Листинг 10.3. Пример действия функции signum

signum(0) = 1 signum(0 - i) = -i

signum(i) = i signum(0 + i) = i

signum(0.1) =1 signum(0.1+2i) = 0.05 + 0.999i

signum(-0.1) = -1 signum(-0.1-3i) =-0.033 -0.999i

10.3. Логарифмы и экспонента (Log and Exponential)

Перечислим без комментариев хорошо известные логарифмические функции (рис. 10.8) и экспоненциальную функцию:

- exp(z) — значение e (основание натурального логарифма) в степени z;
- ln(z) — натуральный логарифм;
- log(z) — десятичный логарифм;
- log(z,b) — логарифм z по основанию b.
- lnGamma(z) — логарифм гамма-функции Эйлера *(см. разд. 10.6)*;

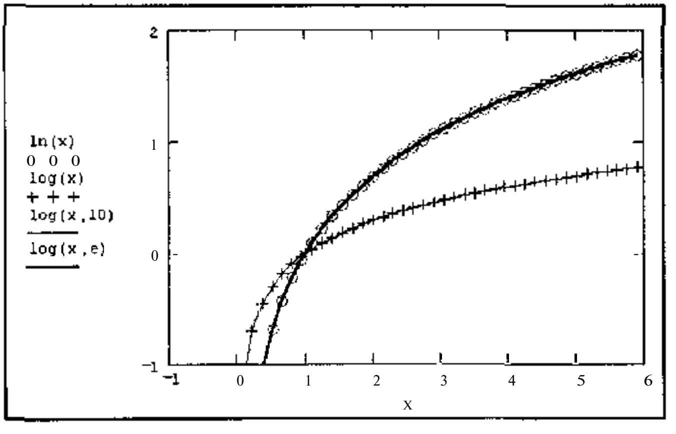


Рис. 10.8. Логарифмические функции

10.4. Тригонометрические функции (Trigonometric)

- acos(z) — арккосинус;
- acoc(z) — котангенс;
- acsc(z) — арккосеканс (листинг 10.4);
- angle(x,y) — угол между точкой (x,y) и осью ox;
- asec(z) — арксеканс;
- asin(z) — арксинус (листинг 10.4);
- atan(z) — арктангенс;

- atan2(x,y) — угол, отсчитываемый от оси ox до точки (x,y) (листинг 10.5);

- cos(z) — косинус;
- cot(z) — котангенс;
- csc(z) — косеканс (листинг 10.4);
- sec(z) — секанс;
- sin(z) — синус (листинг 10.4);
- tan(z) — тангенс;
  - z — безразмерный скаляр.

Примечание

Аргумент тригонометрических функций и результат обратных тригонометрических функций выражаются в радианах. Чтобы использовать значение угла в градусах, его необходимо перевести в радианы (листинг 10.6).

Аргумент тригонометрических функций может быть комплексным.

Листинг 10.4. Примеры тригонометрических функций

```
sin(0.5) =0.479            1/csc(0.5) = 0.479
asin(0.479) =0.5           acsc(1/0.479) = 0.5
```

Листинг 10.5. Примеры расчета угла между прямой и осью ox

```
atan2 (1,1) =0.785       atan2 (-1,-1) =-2.356
angle(1,1) = 0.785       angle (-1,-1) =3.927
```

Листинг 10.6. Расчет тригонометрических функций в градусах

$z := 47$

$\cos\left(\frac{\pi \cdot z}{180}\right) = 0.682$

$\mathrm{acos}(0.682) \cdot \frac{180}{\pi} = 47$

10.5. Гиперболические функции (Hyperbolic)

Гиперболические функции, согласно определению, выражаются через различные комбинации e^z и e^{-z} (пример приведен в листинге 10.7). Аргумент гиперболических функций также может быть комплексным. Графики трех основных гиперболических функций показаны на рис. 10.9.

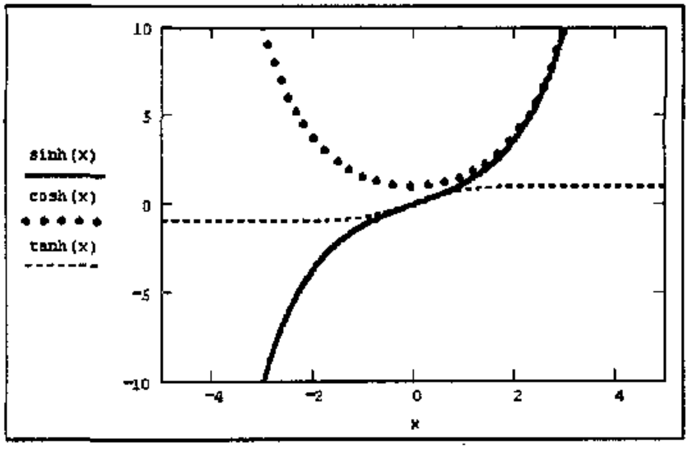


Рис. 10.9. Основные гиперболические функции

- acosh(z) — гиперболический арккосинус;
- acoth(z) — гиперболический котангенс;
- asinh(z) — гиперболический арксинус;
- acsch(z) — гиперболический арккосеканс;
- atanh(z) — обратный гиперболический тангенс;
- asech(z) — обратный гиперболический секанс;
- cosh(z} — гиперболический косинус;
- coth(z) — гиперболический котангенс;
- sinh(z) — гиперболический синус;
- csch(z) — гиперболический косеканс;
- tanh(z) — гиперболический тангенс;
- sech(z) — гиперболический секанс;
  - z — безразмерный скаляр.

Листинг 10.7. Пример гиперболических функций

z :=1.27

$$\frac{e^{z} + e^{-z}}{2} = 1.921$$

cosh(z) = 1.921

acosh(1.921) = 1.27

10.6. Другие спецфункции (Special)

Приведем перечень остальных спецфункций, которые рассчитываются Mathcad встроенным образом. Действие некоторых функций иллюстрируется листингом 10.8, а некоторые полиномы — графиками на рис. 10.11—10.13.

- erf (z) — функция ошибок *(см. разд. "Нормальное (Гауссово) распределение" гл. 14)*;
- erfc(z)≡1-erf(z);
  - z — скаляр.
- fhyper(a,b,c,x) — Гауссова гипергеометрическая функция;
- mhyper(a,b,x) — конфлюэнтная гипергеометрическая функция;
  - a,ь,c — параметры;
  - x — действительный скаляр, -1<x<1.
- Gamma(z) — гамма-функция Эйлера;
  - z — скаляр, |z|<1.
- Gamma(a,x) — неполная гамма-функция порядка a;
  - x — действительный положительный скаляр.

Примечание

Гамма-функция а документе Mathcad отображается греческой буквой Г (листинг 10.8).

- Her(n,x) — полином Эрмита порядка п с аргументом x (рис. 10.10);
  - п — порядок (неотрицательное целое число);
  - x — скаляр.

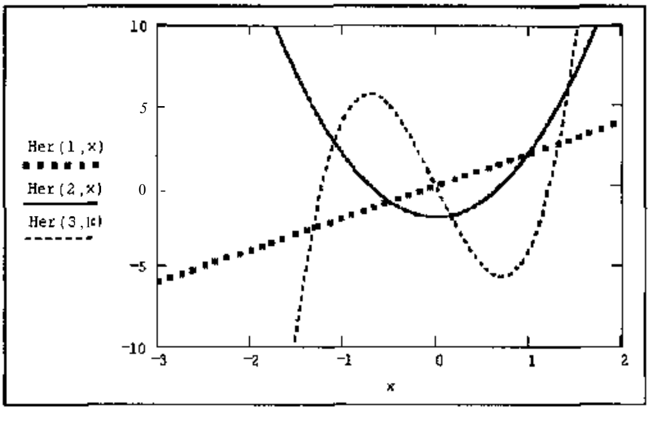


Рис. 10.10. Полиномы Эрмита

- ibeta(a,x,y) — неполная бета-функция для x и y с параметром a;
  - a — действительный скаляр, $0 \le a \le 1$;
  - x,y — действительные скаляры, x>0, y>0.
- Jac(n,a,b,x) — полином Якоби степени n в точке x с параметрами a и b;
- Lag(n, x) — полином Лагерра степени n в точке x (рис. 10.11);

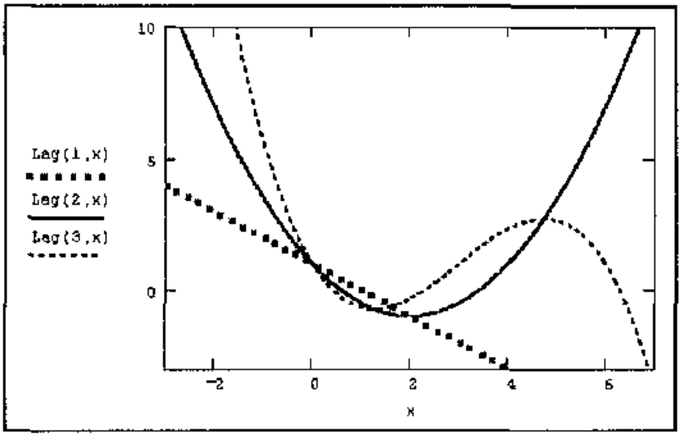


Рис. 10.11. Полиномы Лаггера

- Leg(n,x) — полином Лежандра степени n в точке x (рис. 10.12);
  - n — порядок (неотрицательное целое число);
  - x - действительный скаляр;
  - a,b — действительные скаляры, a>-1, b>-1.
- Tcheb(n,x) — полином Чебышева первого рода степени n в точке x (рис. 10.13);
- ucheb{n,x) — полином Чебышева второго рода степени n в точке x (рис. 10.13);
  - n — порядок (неотрицательное целое число);
  - x — действительный скаляр.

Листинг 10.8. Примеры вычисления некоторых спецфункций

```
fhyper (1,2,3, 0.34) =1.306
Γ(0.7i) =-0.29 - 0.961i
Γ(1.3 , 7.7) = 8.655 x 10^-4
Jac(1, 2 , 1,-0.13) =0.175
```

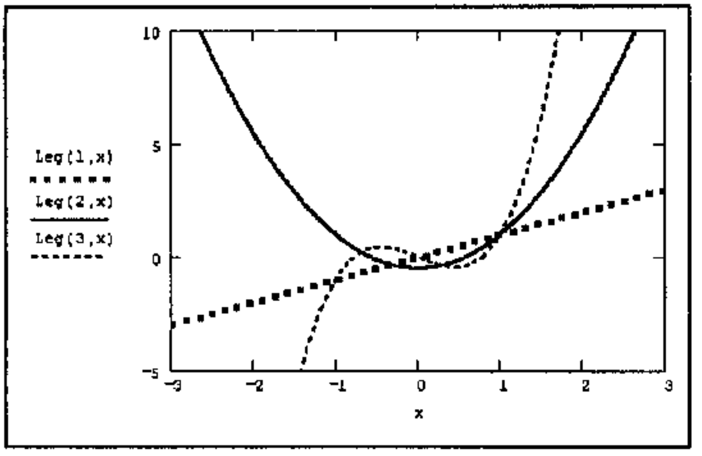


Рис. 10.12. Полиномы Лежандра

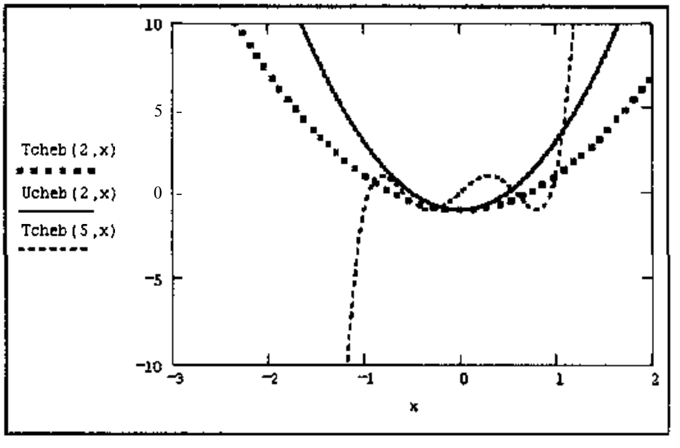


Рис. 10.13. Полиномы Чебышева

10.7. Строковые функции (String)

Приведем перечень функций, благодаря которым пользователь может оперировать со строковыми переменными, подобно операциям с числами:

- concat(S1,S2, ...) — строковая переменная, полученная объединением строковых переменных или констант S1, S2,... (листинг 10.9);
- error(s) — возвращает строку s как сообщение об ошибке (рис. 10.14);
- IsString(x) — возвращает 1, если x строковая переменная, и 0 — в остальных случаях (листинг 10.10);

- num2str(z) — возвращает строку, чьи знаки соответствуют десятичному значению числа z (листинг 10.10);

Примечание

Функция `num2str(z)` используется, когда проще манипулировать с числом как со строкой, нежели как с математической переменной.

- `search(S,Subs,m)` — стартовая позиция подстроки Subs в строке S при поиске, начиная с позиции m, при неуспешном поиске возвращает -1 (листинг 10.9);
- `str2num(S)` — преобразование строкового представления числа s (в любой форме) в число (листинг 10.10);
- str2vec(s) — преобразование в вектор ASCII-кодов строки s (листинг 10.10);
- `strlen(S)` — количество знаков в строке s (листинги 10.9, 10.10);
- `substr(S,m,n)` — подстрока, полученная из строки s выделением п знаков, начиная с позиции m в строке s (листинг 10.9);
- vec2str(v) — строковое представление элементов вектора v ASCII-кодов;
  - s — строка;
  - v — вектор ASCII-кодов (целых чисел, $0<v<255$).

Листинг 10.9. Примеры использования строковых функций

```
concat ! "Hello, " , " " , "World" ,"!")= "Hello, World!"
substr ( "Hello, World!" ,4,8) = "o, World"
substr ( "Hello, World!" ,0,5) ="Hello"
search ( "Hello, World! " , "Wo" ,1) =7
search { "Hello, World!" , "wo" ,1) =-1
strlen ("hello") =5
```

Листинг 10.10. Функции взаимных преобразований чисел и строк

```
IsString(1)=0
IsString ("!") =1
strlen ("Hello, World!" ) = 13
num2str (579 + 3i) = "579 + 3i"
num2str(12.345) ="12.345"
str2num ( "123.4567" ) =123.457
```

$$\mathrm{str2vec}("17") = \begin{pmatrix} 49 \\ 55 \end{pmatrix}$$

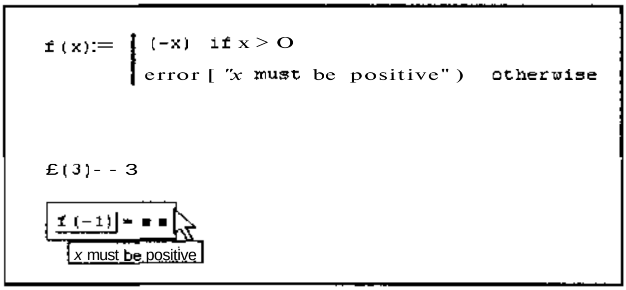


Рис. 10.14. Использование функции создания сообщения об ошибке

10.8. Функции сокращения и округления (Truncation and Round-Off)

- `ceil(z)` — наименьшее целое, не меньшее `z` (листинг 10.11);
- `floor(z)` — наибольшее целое число, меньшее или равное `z` (листинг 10.11);
- `round(z,n)` — при n>0 возвращает округленное значение z с точностью до `n` знаков после десятичной точки, при `n<0` — округленное значение z с n цифрами слева от десятичной точки, при n=0 — округленное до ближайшего целого значение z (листинг 10.12);
- `trunc(z)` — целая часть числа (листинг 10.11);
  - z — действительный или комплексный скаляр.

Примечание

Начиная с версии Mathcad 11, функции округления и сокращения чисел поддерживают также и комплексные аргументы (последние строки листингов 10.11 и 10.12).

Листинг 10.11. Функции сокращения и округления

```
ceil(3.7) = 4      floor(3.7) = 3     trunc(3.7) = 3
ceil(-3.7) = -3    floor(-3.7) = -4   trunc(-3.7) = -3

ceil(3.7 - 2.1·i) = 4 - 2i

floor(3.7 - 2.1·i) = 3 - 3i

trunc(3.7 - 2.1·i) = 3 - 2i
```

Листинг 10.12. Округление чисел

```
round (1.23456789 ,0) = 1            round ( 12 . 3456789 , 0 ] = 12
round (12.3456789 ,1) = 12.3         round { 12 . 3456789 , -1) = 10
round (12. 3456789, 2) =12.35        round (12. 345678 9, -2 ) = 0
round (12.3456789 , 5) =12.34568
round (1.2345 + 6.789i , 1) = 1.2 + 6.8i
```

Примечание

При округлении не забывайте о принципах представления чисел в Mathcad. Чтобы отобразить нужное количество знаков после десятичной точки, воспользуйтесь диалогом **Result Format** (Формат результата) *(см. гл. 4).*

10.9. Кусочно-непрерывные функции (Piecewise Continuous)

- heaviside step(x) — функция Хевисайда, возвращает 1, если x≥o, и o — в остальных случаях (рис. 10.15);
  - x — действительный скаляр.
- `if` (cond,x,y) — возвращает x, если логическое условие cond верно (не ноль), и y в остальных случаях (листинг 10.13);
- `Kronecker delta(x, y)` — дельта-символ Кронекера: возвращает 1, если x=y, и o в остальных случаях (рис, 10.15);
- sign(x) — возвращает o, если x=o, 1, если x>o, и -1 — в остальных случаях (листинг 10.13);
  - x — действительное число.

Примечание

В документах символ Кронекера обозначается греческой буквой 5, а функция Хевисайда — буквой Ф.

Листинг 10.13. Функции условия и знака

```
sign (-4) =-1
sign (1.3) = 1
if{1 > 3 , 1 , 3) = 3
if(1 > 3 , "Yes" , "No" ) = "No"
if(2 + 5 ≥0 , "Yes" , "No" ) = "Yes"
```

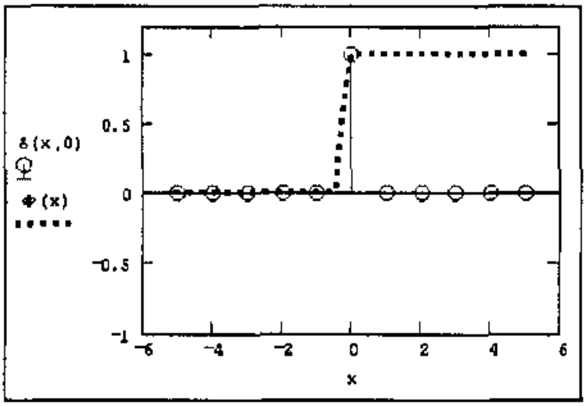


Рис. 10.15. Функции Кронекера и Хевисайда

10.10. Функции преобразования координат (Vector and Matrix)

В Mathcad 2001 появилось семейство новых функций, позволяющих перейти от одних координат к другим, как на плоскости, так и в пространстве:

- xy2pol(x,y) — преобразование прямоугольных координат в полярные;
- pol2xy(r,θ) — преобразование полярных координат в прямоугольные;
- angle(x,y) — угол между точкой (x,y) и осью ox (*см. разд. 10.4*);
- atan2(x,y) — угол, отсчитываемый от оси ох до точки (x,y) (*см. разд. 10.4*);
- xyz2cyl(x,y,z) — преобразование прямоугольных координат в цилиндрические;
- cyl2xyz(r,θ,z) — преобразование цилиндрических координат в прямоугольные;
- xyz2sph(x,y,z) — преобразование прямоугольных координат в сферические;
- sph2xyz(r,θ,φ) — преобразование сферических координат в прямоугольные;
  - x,y — прямоугольные координаты на плоскости;
  - x,y,z — прямоугольные координаты в пространстве;
  - r,θ — полярные координаты на плоскости;
  - r,θ,z — цилиндрические координаты;
  - r,θ,φ — сферические координаты.

Несколько примеров преобразования координат приведены в листингах 10.14 и 10.15. Обратите внимание на возможность ввода аргументов этих функций как в виде списка, так и в виде вектора.

Листинг 10.14. Функции преобразования координат на плоскости

$$\mathrm{xy2pol}\begin{pmatrix} 1 \\ 7 \end{pmatrix} = \begin{pmatrix} 7.071 \\ 1.429 \end{pmatrix} \qquad \mathrm{xy2pol}(1, 7) = \begin{pmatrix} 7.071 \\ 1.429 \end{pmatrix}$$

$$\mathrm{pol2xy}\begin{pmatrix} 7.071 \\ 1.429 \end{pmatrix} = \begin{pmatrix} 0.999 \\ 7 \end{pmatrix} \qquad \mathrm{pol2xy}(7.071, 1.429) = \begin{pmatrix} 0.999 \\ 7 \end{pmatrix}$$

Листинг 10.15. Функции преобразования координат в пространстве

$$\mathrm{xyz2cyl}(1, 1, 1) = \begin{pmatrix} 1.414 \\ 0.785 \\ 1 \end{pmatrix} \qquad \mathrm{cyl2xyz}(1, \pi, 3.93) = \begin{pmatrix} -1 \\ 0 \\ 3.93 \end{pmatrix}$$

$$\mathrm{xyz2sph}(1, 1, 1) = \begin{pmatrix} 1.732 \\ 0.7S5 \\ 0.955 \end{pmatrix} \qquad \mathrm{sph2xyz}\left(\sqrt{2}, \frac{\pi}{4}, \frac{\pi}{2}\right) = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

10.11. Финансовые функции (Finance)

Начиная с версии 2000, в Mathcad появились функции, облегчающие финансовый анализ. Приведем список этих функций, не вдаваясь в пояснения и надеясь на то, что заинтересованный читатель найдет подробное описание и практические примеры их применения в справочной системе Mathcad.

- `cnper(rate,pv,fv)` — отвечает числу составных периодов, необходимых для получения будущего значения вклада при заданных текущем значении вклада и проценте начислений;
  - `rate` — фиксированный процент по вкладу; должен быть действительным скаляром, `rate>-1`;
  - pv — текущее значение вклада, pv>o;
  - fv — будущее значение вклада, `fv>0`.
- `crate(nper,pv,fv)` — отвечает фиксированному проценту начислений по вкладу на период, необходимый для прироста от текущего значения вклада до будущего значения при заданном числе составных периодов;
  - `nper` — число составных периодов; должно быть целым числом, `nper≥1`;

- pv — текущее значение вклада, pv>o;
- fv — будущее значение вклада, fv>o.

❐ cumint(rate,nper,pv,start,end,[type]) — отвечает совокупному проценту по заему между начальным и конечным периодами при фиксированном проценте, общем числе составных периодов и текущем значении заема;

❐ cumprn(rate,nper,pv,start,end, [type]> — отвечает совокупной сумме по заему между начальным и конечным периодами при фиксированном проценте, общем числе составных периодов и текущем значении заема;

- rate — фиксированный процент по вкладу; должен быть действительным скаляром, rate≥o;
- nper — общее число составных периодов; должно быть положительным целым числом;
- pv — текущее значение заема, pv>o;
- start — начальный период накопления; должен быть положительным целым числом;
- end— конечный период накопления; должен быть положительным целым числом; start≤ end;
- type=0 для платежа, сделанного в конце периода, или 1 для платежа, сделанного в начале периода.

❐ eff(rate,nper) — отвечает эффективной ежегодной процентной ставке при данной номинальной ежегодной процентной ставке и числе составных периодов в год;

- rate — номинальная процентная ставка; должна быть действительным скаляром;
- nper — общее число составных периодов в год, nper >o.

О fv(rate,nper,pmt, [[pv] , [type]]) — соответствует будущему значению вклада или заема через особое число составных периодов, установленных периодически, при постоянных платежах и фиксированной процентной ставке;

- rate — фиксированная процентная ставка за период; должна быть действительным скаляром, rate>o;
- nper— общее число составных периодов в год, nper >o;
- pv — текущее значение заема;
- type=0 для платежа, сделанного в конце периода, или 1 для платежа, сделанного в начале периода.

- fvadj(prin,v) — соответствует будущему значению ежегодной общей суммы капитала, на который начисляются проценты, при применении серии составных процентных ставок;
  - prin — ежегодная общая сумма;
  - v — вектор процентных ставок, каждая из которых применяется с той же самой основной суммой и процентами с нее за период времени.
- fvc(rate,v) — соответствует будущему значению серии денежных потоков, происходящих с регулярными интерватами, и приносящими специальную процентную ставку;
  - rate — фиксированная процентная ставка за период; должна быть действительным скаляром;
  - v — вектор регулярных денежных потоков.
- ipmt(rate,per,nper,pv, [[fv], type]]) — соответствует процентному платежу по вкладу или заему за данный период, основанному на периодичности, постоянных платежах через данное число составных периодов, использующих фиксированную процентную ставку и особое текущее значение;
  - rate — фиксированная процентная ставка за период, rate≥o;
  - per — период, за который Вы хотите найти ставку; должен быть положительным целым числом;
  - nper — общее число составных периодов, per≤ nper;
  - pv - текущее значение;
  - fv — будущее значение;
  - type=o для платежа, сделанного в конце периода, или 1 для для платежа, сделанного в начале периода.
- irr(v, [guess]) — отвечает внутренней ставке возврата для серии денежных потоков, происходящих с регулярными интервалами;
  - v — вектор денежных потоков, определяемых за регулярные интервалы; должен состоять по крайней мере из одного положительного и отрицательного числа;
  - guess — численное значение, которым Вы предполагаете аппроксимировать ответ, если им пренебрегается, то guess=0 . 1 (10%).
- mirr(v, fin_rate, rein_rate) — соответствует модифицированной процентной ставке возврата для серии денежных потоков с регулярными интервалами при условии, что ставка финансирования подлежит оплате в соответствии с суммой заимствования, а ставка реинвестирования приносит доход с суммы, которую Вы повторно инвестируете;

- v — вектор денежных потоков, определяемых за регулярные интервалы; он должен состоять по крайней мере из одного положительного и отрицательного числа;
- fin_rate — финансовая ставка платежа по заимствованным денежным потокам;
- rein_rate — ставка реинвестирования.

- nom(rate,nper) — соответствует номинальной процентной ставке, включающей эффективную ежегодную процентную ставку и число составных периодов за год;
  - rate — эффективная ежегодная процентная ставка; должна быть действительным скаляром, rate>-1;
  - nper — общее число составных периодов за год, nper>0.
- npv{rate,v) — вычисляет чистое текущее значение вклада, включающее скидки и регулярные денежные потоки;
  - rate — фиксированная процентная ставка, с которой вклад зарабатывает процент за период; должна быть действительным скаляром;
  - v — вектор регулярных денежных потоков.
- nper(rate,pmt,pv, [[fv], [type]]) — отвечает числу периодов для вклада или заема, основанных на периодичности, постоянных платежах, использующих фиксированную процентную ставку и особое текущее значение;
- pmt(rate,nper,pv, [[fv],[type]]) — соответствует платежу по вкладу или заему, основанному на периодичности, постоянных платежах через данное число составных периодов, использующих фиксированную процентную ставку и особое текущее значение;
- ppmt(rate,per,nper,pv, [[fv],[type]]} — соответствует платежу по общей сумме вклада или заема, основанному на периодичности, постоянных платежах через данное число составных периодов, использующих фиксированную процентную ставку и особое будущее значение;
- pv(rate,nper,pmt, [[fv], [type]]) — соответствует текущему значению вклада или заема, основанному на периодичности, постоянных платежах через данное число составных периодов, использующих фиксированную процентную ставку и особый взнос;
- rate(nper,pmt,pv, [[fv], [type], [guess]]) — соответствует процентной ставке на период вклада или заема при особом числе периодических составных периодов, постоянных платежах и особом текущем значении;
  - rate — фиксированная процентная ставка;
  - per — период;

- nper — общее число составных периодов за год; должно быть положительным целым числом;
- pmt — платеж, делаемый каждый период;
- pv — текущее значение вклада;
- fv — будущее значение вклада;
- type=o для платежа, сделанного в конце периода, или 1 для платежа, сделанного в начале периода;
- guess — численное значение, которым Вы предполагаете аппроксимировать ответ; если им пренебрегается, то guess=0.01(10%).

ГЛАВА 11

Обыкновенные дифференциальные уравнения

Дифференциальные уравнения — это уравнения, в которых неизвестными являются не переменные (т. е. числа), а функции одной или нескольких переменных. Эти уравнения (или системы) включают соотношения между искомыми функциями и их производными. Если в уравнения входят производные только по одной переменной, то они называются *обыкновенными дифференциальными уравнениями* (далее чаше используется сокращение *ОДУ*). В противном случае говорят об *уравнениях в частных производных (см. гл. 13)*. Таким образом, решить (иногда говорят *проинтегрировать*) дифференциальное уравнение — значит определить неизвестную функцию на определенном интервале изменения ее переменных.

Как известно, одно обыкновенное дифференциальное уравнение *(см. разд. 11.1—11.2)* или система ОДУ *(см. разд. 11.3)* имеет единственное решение, если помимо уравнения определенным образом заданы *начальные* или *граничные* условия. В соответствующих курсах высшей математики доказываются теоремы о существовании и единственности решения в зависимости от тех или иных условий. Имеются два типа задач, которые возможно решать с помощью Mathcad 11:

- *задачи Коши* — для которых определены начальные условия на искомые функции, т. е. заданы значения этих функций в начальной точке интервала интегрирования уравнения;
- *краевые задачи* — для которых заданы определенные соотношения сразу на обеих границах интервала *(они рассматриваются в гл. 12)*.

Как правило, решение задач Коши для ОДУ и их систем — задача хорошо разработанная и с вычислительной точки зрения не слишком сложная. Большое значение здесь имеет представление результатов и анализ зависимостей решения от различных параметров системы *(см. разд. 11.4)*. Между тем, имеется целый класс ОДУ, называемых *жесткими,* который не поддается решению стандартными методами, типа методов Рунге-Кутты. Для них в Mathcad имеются специальные возможности *(см. разд. 11.5)*.

11.1. ОДУ первого порядка

Дифференциальное уравнение первого порядка может по определению содержать, помимо самой искомой функции y(t), только ее первую производную y'(t). В подавляющем большинстве случаев дифференциальное уравнение можно записать в *стандартной форме (форме Коши):*

$$y'(t)=f(y(t),t), \quad (1)$$

и только с такой формой умеет работать вычислительный процессор Mathcad. Правильная с математической точки зрения постановка соответствующей задачи Коши для ОДУ первого порядка должна, помимо самого уравнения, содержать одно начальное условие — значение функции $y(t_0)$ в некоторой точке t_0. Требуется явно определить функцию y(t) на интервале от t_0 до t_1. По характеру постановки задачи Коши называют еще *задачами с начальными условиями* (initial value problem), в отличие от краевых задач.

Для численного интегрирования одного ОДУ у пользователя Mathcad 11 (начиная с версии Mathcad 2000 Pro) имеется выбор — либо использовать вычислительный блок Given/Odesolve, либо встроенные функции, как в прежних версиях Mathcad. Первый путь предпочтительнее из соображений наглядности представления задачи и результатов, а второй дает пользователю больше рычагов воздействия на параметры численного метода. Рассмотрим последовательно оба варианта решения.

11.1.1. Вычислительный блок *Given/Odesolve*

Вычислительный блок для решения одного ОДУ, реализующий численный метод Рунге-Кутты, состоит из трех частей:

- Given — ключевое слово;
- ОДУ и начальное условие, записанное с помощью логических операторов, причем начальное условие должно быть в форме y(to)=b;
- `Odesolve(t,t1)` — встроенная функция для решения ОДУ относительно переменной t на интервале (t0,t1).

Примечание

Допустимо, и даже часто предпочтительнее, задание функции Odesolve (t, t1, step) с тремя параметрами, где **step** — внутренний параметр численного метода, определяющий количество шагов, в которых метод Рунге-

> Кутты, будет рассчитывать решение дифференциального уравнения. Чем больше step, тем с лучшей точностью будет получен результат, но тем больше времени будет затрачено на его поиск. Помните, что подбором этого параметра можно заметно (в несколько раз) ускорить расчеты без существенного ухудшения их точности.

Пример решения задачи Коши для ОДУ первого порядка $y'=y-y^2$ посредством вычислительного блока приведен в листинге 11.1.

Листинг 11.1. Решение задачи Коши для ОДУ первого порядка

Given

$$\frac{d}{dt}y(t) = y(t) - y(t)^2$$

$y(0) = 0.1$

$y := \mathrm{Odesolve}(t, 10)$

Не забывайте о том, что вставлять логические операторы следует при помощи панели инструментов **Boolean** (Булевы операторы). При вводе с клавиатуры помните, что логическому знаку равенства соответствует сочетание клавиш <Ctrl>+<=>. Символ производной можно ввести как средствами панели **Calculus** (Вычисления), как это сделано в листинге 11.1, так и в виде штриха, набрав его с помощью сочетания клавиш <Ctrl>+<F7> (соответствующий пример будет приведен ниже в листинге 11.3.) Выбирайте тот или иной способ представления производной из соображений наглядности представления результатов — на ход расчетов он не влияет.

Mathcad требует, чтобы конечная точка интегрирования ОДУ лежала правее начальной: `t0<t1` (в листинге 11.1 `t0=0,t1=10`), иначе будет выдано сообщение об ошибке. Как можно заметить, результатом применения блока `Given/Odesolve` является функция `y(t)`, определенная на промежутке `(t0,t1)`. Следует воспользоваться обычными средствами Mathcad, чтобы построить ее график или получить значение функции в какой-либо точке указанного интервала, например: `y(3)`=0.691.

Пользователь имеет возможность выбирать между двумя модификациями численного метода Рунге-Кутты. Для смены метода необходимо нажатием правой кнопки мыши на области функции odesolve вызвать контекстное меню и выбрать в нем один из двух пунктов: **Fixed** (Фиксированный шаг) или **Adaptive** (Адаптивный). По умолчанию применяется первый из них, т. е. метод Рунге-Кутты с фиксированным шагом. *Подробнее о различии этих методов сказано в разд. 11.3.*

11.1.2. Встроенные функции *rkfixed, Rkadapt, Bulstoer*

Альтернативный метод решения ОДУ перешел из прежних версий Mathcad. Он заключается в использовании одной из встроенных функций rkfixed, Rkadapt или Bulstoer. Этот способ несколько проигрывает первому и в простоте, и в наглядности. Поэтому я советую предпочесть вычислительный блок `Given/Odesolve`. Однако иногда приходится решать ОДУ первого порядка с помощью второго способа, например, при следующих обстоятельствах:

- одно ОДУ решается в контексте решения более сложных задач, в которые входят системы дифференциальных уравнений (для которых вычислительный блок неприменим) — в этом случае может потребоваться единый стиль программирования;
- ответ предпочтительнее получить в виде вектора, а не функции;
- Вы привыкли к записи ОДУ в старых версиях Mathcad, у Вас много документов, созданных с их помощью и т. п.

Поскольку решение вторым способом одного ОДУ мало чем отличается от решения систем ОДУ *(см. разд. 11.3)*, приведем пример его использования в задаче из листинга 11.1 практически без комментариев (см. листинг 11.2) и с помощью одной из трех существующих для этих целей встроенных функций rkfixed. Обратите внимание только на необходимость явного задания количества точек интегрирования ОДУ `M=100` в третьей строке листинга, а также на получение результата, в отличие от вычислительного блока, не в виде функции, а в виде матрицы размерности `M×2`. Она состоит из двух столбцов: в одном находятся значения аргумента t (от t0 до `t1` включительно), а в другом соответствующие значения искомой функции `y(t)`.

Листинг 11.2. Решение задачи Коши для ОДУ первого порядка вторым способом

$y := 0.1$

$D(t, y) := y - y^2$

$M := 100$

$y := \mathrm{rkfixed}(y, 0, 10, M, D)$

Совет

В листинге 11.2 приведен пример не лучшего стиля Mathcad-программирования. Сначала переменной y присвоено значение скаляра `y=0.1`, а затем этой же переменной присвоено матричное значение (результат решения ОДУ). Старайтесь избегать такого стиля, который ухудшает читаемость программы и может приводить, в более сложных случаях, к трудно опознаваемым ошибкам. Неплохим решением было бы назвать результат по-другому, например и.

График решения рассматриваемого уравнения показан на рис. 11.1. Обратите внимание, что он соответствует получению решения в матричном виде (листинг 11.2), поэтому по осям отложены соответствующие столбцы, выделенные из матрицы у оператором <>.

Примечание

Пример, решенный в листингах 11.1—11.2, взят из области математической экологии и описывает динамику популяций с внутривидовой конкуренцией. Сначала происходит рост численности популяции, близкий к экспоненциальному, а затем выход на стационарное состояние.

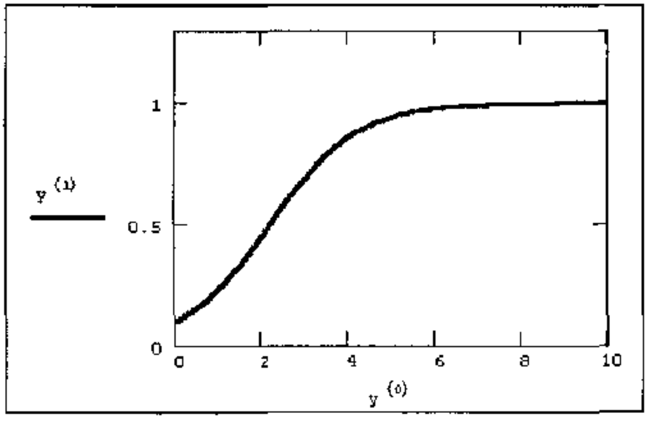


Рис. 11.1. Решение уравнения $y'=y-y^2$ (листинг 11.2)

11.2. ОДУ высшего порядка

Обыкновенное дифференциальное уравнение с неизвестной функцией y(t), в которое входят производные этой функции вплоть до $y^{(N)}(t)$, называется ОДУ N-ГО порядка. Если имеется такое уравнение, то для корректной постановки задачи Коши требуется задать N начальных условий на саму функцию y(t) и ее производные от первого до (N-1)-го порядка включительно. В Mathcad 11 можно решать ОДУ высших порядков как с помощью вычислительного блока `Given/Odesolve`, так и путем сведения их к системам уравнений первого порядка.

Внутри вычислительного блока:

- ОДУ должно быть линейно относительно старшей производной, т. е. фактически должно быть поставлено в стандартной форме;
- начальные условия должны иметь форму $y(t)=b$ или $y^{(N)}(t)=b$, а не более сложную (как, например, встречающаяся в некоторых математических приложениях форма $y(t)+y'(t)=b$).

В остальном, решение ОДУ высших порядков ничем не отличается от решения уравнений первого порядка *(см. разд. 11.1),* что иллюстрируется листингом 11.3. Как Вы помните, допустимо написание производной как в виде знака дифференциала (так в листинге 11.3 введено само уравнение), так и с помощью штриха (так введено начальное условие для первой производной). Не забывайте пользоваться булевыми операторами при вводе уравнений и начальных условий. Полученное решение y(t) показано на рис. 11.2.

Листинг 11.3. Решение задачи Коши для ОДУ второго порядка

Given

$$\frac{d^2}{dt^2}y(t) + 0.1\cdot\frac{d}{dt}y(t) + 1\cdot y(t) = 0$$

$y(0) = 0.1$

$y'(0) = 0$

y := Odesolve (t , 50)

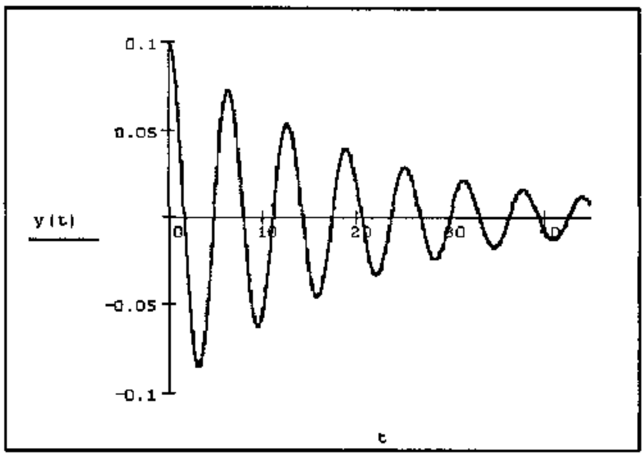


Рис. 11.2. Решение уравнения осциллятора (листинг 11.3)

Примечание

В листинге 11.3 решено уравнение затухающего гармонического осциллятора, которое описывает, например, колебания маятника. Для модели маятника y(t) описывает изменения угла его отклонения от вертикали, y'(t) — угловую скорость маятника, y"(t) — ускорение, а начальные условия, соответственно, начальное отклонение маятника y(0)=0.1 и начальную скорость y'(0)= 0.

Второй способ решения ОДУ высшего порядка связан со сведением его к эквивалентной системе ОДУ первого порядка. Покажем на том же примере из листинга 11.3, как это делается. Действительно, если формально обозначить $y_0(t) \equiv y(t)$, а $y_1(t) \equiv y'(t) = y_0'(t)$, то исходное уравнение запишется через функции $y_0(t)$ и $y_1(t)$ в виде системы двух ОДУ:

$$\begin{cases} y_0' = y_1, \\ y_1' + 0.1 \cdot y_1 + 1 \cdot y_0 = 0. \end{cases} \qquad (1)$$

Именно эта система решается в качестве примера в *разд. 11.3.* Таким образом, любое ОДУ N-ГО порядка, линейное относительно высшей производной, можно свести к эквивалентной системе N дифференциальных уравнений.

11.3. Системы ОДУ первого порядка

Mathcad требует, чтобы система дифференциальных уравнений была представлена в стандартной форме:

$$\begin{aligned} &y_0'(t) = f_0(y_0(t), y_1(t), \ldots, y_{N-1}(t), t), \\ &y_1'(t) = f_1(y_0(t), y_1(t), \ldots, y_{N-1}(t), t), \\ &\ldots \\ &y_{N-1}'(t) = f_{N-1}(y_0(t), y_1(t), \ldots, y_{N-1}(t), t) \end{aligned} \qquad (0$$

Задание системы (1) эквивалентно следующему векторному представлению:

$$Y'(t) = F(Y(t), t), \qquad (2)$$

где Y и Y' — соответствующие неизвестные векторные функции переменной t размера N×1, а F — векторная функция того же размера и {N+1} количества переменных (N компонент вектора и, возможно, t). Именно векторное представление (2) используется для ввода системы ОДУ в среде Mathcad.

Для того чтобы определить задачу Коши для системы ОДУ, следует определить еще N начальных условий, задающих значение каждой из функций $y_i(t0)$ в начальной точке интегрирования системы to. В векторной форме они могут быть записаны в виде

$$Y(t0) = B, \qquad (3)$$

где в — вектор начальных условий размера N×1, составленный из $y_i(t0)$.

Примечание

Как Вы заметили, задача сформулирована для систем ОДУ первого порядка. Однако если в систему входят и уравнения высших порядков, то ее можно свести к системе большего числа уравнений первого порядка, подобно тому как это было сделано на примере уравнения осциллятора *(см. разд. 11.2)*.

Обратите внимание на необходимость векторной записи как самого уравнения, так и начального условия. В случае одного ОДУ соответствующие векторы имеют только один элемент, а в случае системы N>1 уравнений — N.

11.3.1. Встроенные функции для решения систем ОДУ

В Mathcad 11 имеются три встроенные функции, которые позволяют решать поставленную в форме (2—3) задачу Коши различными численными методами.

- `rkfixed(y0,to,t1,M,D)` — метод Рунге-Кутты с фиксированным шагом;
- `Rkadapt(y0,to,t1,M,D)` — метод Рунге-Кутты с переменным шагом;
- `Bulstoer(y0,t0,t1,M,D)` — метод Булирша-Штера;
  - `y0` — вектор начальных значений в точке to размера `N×1`;
  - to — начальная точка расчета;
  - `t1` — конечная точка расчета;
  - `M` — число шагов, на которых численный метод находит решение;
  - D — векторная функция размера `N×1` двух аргументов — скалярного t и векторного y. При этом y — искомая векторная функция аргумента t того же размера `N×1`.

Внимание!

Соблюдайте регистр первой буквы рассматриваемых функций, поскольку это влияет на выбор алгоритма счета, в отличие от многих других встроенных функций Mathcad, например Find≡find *(см. разд. 11.3.2).*

Каждая из приведенных функций выдает решение в виде матрицы размера `(M+1)×(N+1)`. В ее левом столбце находятся значения аргумента t, делящие интервал на равномерные шаги, а в остальных `N` столбцах — значения искомых функций $y_0(t), y_1(t), \ldots, y_{N-1}(t)$, рассчитанные для этих значений аргумента. Поскольку всего точек (помимо начальной) `M`, то строк в матрице решения будет всего `M+1`.

В подавляющем большинстве случаев достаточно использовать первую функцию rkfixed, как это показано и листинге 11.4 на примере решения системы ОДУ модели осциллятора с затуханием *(см. разд. 11.2).*

Листинг 11.4. Решение системы двух ОДУ

$$D(t,y) := \begin{pmatrix} \\ -y_0 - 0.1 \cdot y_1 \end{pmatrix}$$

$$y0 := \begin{pmatrix} 0.1 \\ 0 \end{pmatrix}$$

M := 100

и := rkfixed (y0, 0, 50 , M, D)

Самая важная — это первая строка листинга, в которой, собственно, определяется система ОДУ. Сравните рассматриваемую систему *(разд. 11.2,1)*, записанную в стандартной форме, с формальной ее записью в Mathcad, чтобы не делать впоследствии ошибок. Во-первых, функция D, входящая в число параметров встроенных функций для решения ОДУ, должна быть функцией обязательно двух аргументов. Во-вторых, второй ее аргумент должен быть вектором того же размера, что и сама функция в. В-третьих, точно такой же размер должен быть и у вектора начальных значений уо (он определен во второй строке листинга).

Не забывайте, что векторную функцию D(t,y) следует определять через компоненты вектора у с помощью кнопки нижнего индекса **(Subscript)** с наборной панели **Calculator** (Калькулятор) или нажатием клавиши <[>. В третьей строке листинга определено число шагов, на которых рассчитывается решение, а его последняя строка присваивает матричной переменной и результат действия функции `rkfixed`. Решение системы ОДУ будет осуществлено на промежутке `(0,50)`.

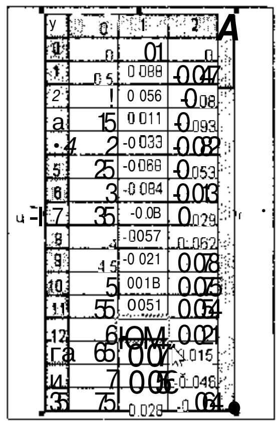


Рис. 11.3. Матрица решений системы уравнений (листинг 11.4)

Как выглядит все решение, показано на рис. 11.3. Размер полученной матрицы будет равен `(M+1)×(N+1)`, т. е. `101×3`. Просмотреть все компоненты матрицы `u`, которые не помещаются на экране, можно с помощью вертикальной полосы прокрутки. Как нетрудно сообразить, на этом рисунке отмечено выделением расчетное значение первого искомого вектора y_0 на 12-м шаге $u_{12,1}=0.07$. Это соответствует, с математической точки зрения, найденному значению $y_0(6.0)=0$ .07. Для вывода элементов решения в последней точке интервала используйте выражения типа $u_{M,1}$=7 .523хю~^{3}.

Внимание!

Обратите внимание на некоторое разночтение в обозначении индексов вектора начальных условий и матрицы решения. В ее первом столбце собраны значения нулевой компоненты искомого вектора, во втором столбце — первой компоненты и т. д.

Чтобы построить график решения, надо отложить соответствующие компоненты матрицы решения по координатным осям: значения аргумента $u^{<0>}$ — вдоль оси x, а $u^{<1>}$ и $u^{<2>}$ — вдоль оси у (рис. 11.4). Как известно, решения обыкновенных дифференциальных уравнений часто удобнее изображать не в таком виде, а в фазовом пространстве, по каждой из осей которого откладываются значения каждой из найденных функций. При этом аргумент входит в них лишь параметрически. В рассматриваемом случае двух ОДУ такой график — *фазовый портрет* системы — является кривой на фазовой плоскости и поэтому особенно нагляден. Он изображен на **рис.** 11.5 (слева), и можно заметить, что для его построения потребовалось лишь поменять метки осей на $u^{<1>}$ и $u^{<2>}$, соответственно.

Примечание

Фазовый портрет типа, изображенного на рис. 11.5, имеет одну стационарную точку *(аттрактор),* на которую "накручивается" решение. В теории динамических систем аттрактор такого типа называется *фокусом.*

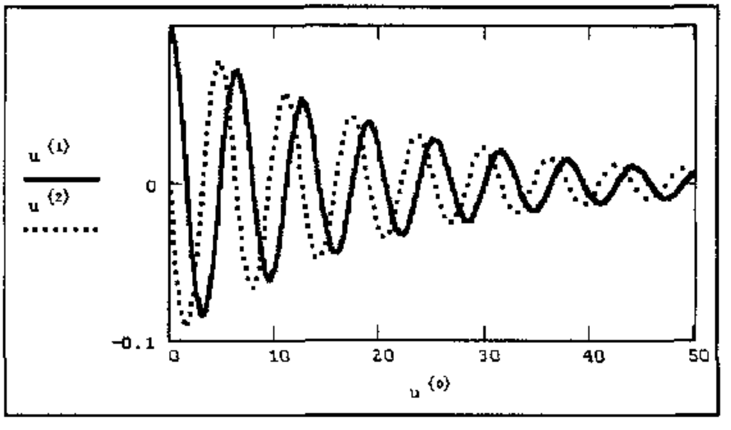


Рис. 11.4. График решения системы ОДУ (11.2—1) (листинг 11.4)

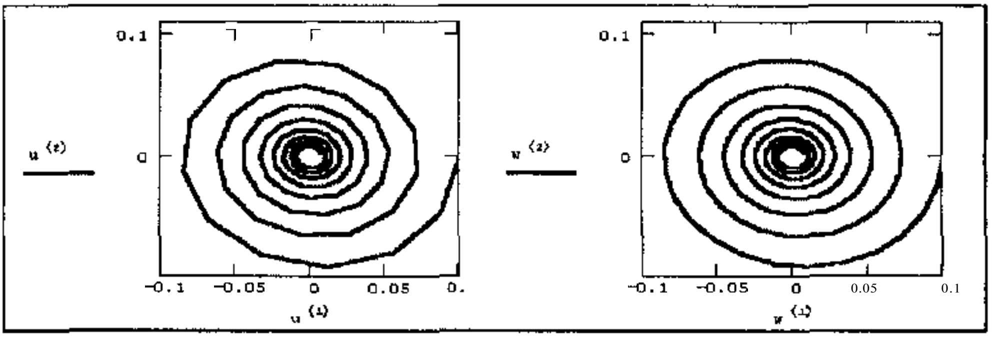


Рис. 11.5. Фазовый портрет решения системы ОДУ при M=100 (слева) и M=200 (справа) (листинг 11.4)

В общем случае, если система состоит из N ОДУ, то фазовое пространство является N-мерным. При N>3 наглядность теряется, и для визуализации фазового портрета приходится строить его различные проекции.

На том же рис. 11.5, справа, показан для сравнения результат расчета фазового портрета с большим числом шагов. Видно, что в этом случае обеспечивается лучшая точность и, в результате, решение получается более гладким. Конечно, при этом увеличивается и время расчетов.

Внимание!

При решении систем ОДУ многие проблемы могут быть устранены при помощи простой попытки увеличить число шагов численного метода. В частности, сделайте так при неожиданном возникновении ошибки **"Found a number with a magnitude greater than 10^307"** (Найдено число, превышающее значение 10307). Данная ошибка может означать не то, что решение в действительности расходится, а просто недостаток шагов для корректной работы численного алгоритма.

В заключение следует сказать несколько слов об особенностях различных численных методов. Все они основаны на аппроксимации дифференциальных уравнений разностными аналогами. В зависимости от конкретной формы аппроксимации, получаются алгоритмы различной точности и быстродействия. В Mathcad использован наиболее популярный алгоритм Рунге-Кутты четвертого порядка, описанный в большинстве книг по методам вычислений. Он обеспечивает малую погрешность для широкого класса систем ОДУ за исключением жестких систем. Поэтому в большинстве случаев стоит применять функцию rkfixed. Если по различным причинам время расчетов становится критичным или точность неудовлетворительна, стоит попробовать вместо `rkfixed` другие функции, благо сделать это очень просто, благодаря одинаковому набору параметров. Для этого нужно только поменять имя функции в программе.

Функция Rkadapt может быть полезна в случае, когда известно, что решение на рассматриваемом интервале меняется слабо, либо существуют участки медленных и быстрых его изменений. Метод Рунге-Кутты с переменным шагом разбивает интервал не на равномерные шаги, а более оптимальным способом. Там, где решение меняется слабо, шаги выбираются более редкими, а в областях его сильных изменений — частыми. В результате, для достижения одинаковой точности требуется меньшее число шагов, чем для rkfixed. Метод Булирша-Штера `Bulstoer` часто оказывается более эффективным для поиска гладких решений.

11.3.2. Решение систем ОДУ в одной заданной точке

Зачастую при решении дифференциальных уравнений требуется определить значения искомых функций не на всем интервале `(t0,t1)`, а только в одной его последней точке. Например, весьма распространены задачи поиска аттракторов динамических систем. Известно, что для широкого класса ОДУ одна и та же система при разных (или даже любых, как рассмотренный выше пример осциллятора с затуханием) начальных условиях при $t \to \infty$ приходит в одну и ту же точку (аттрактор). Поэтому часто нужно определить именно эту точку.

Такая задача требует меньше ресурсов компьютера, чем решение системы ОДУ на всем интервале, поэтому в Mathcad имеются модификации встроенных функций rkadapt и Bulstoer. Они имеют несколько другой набор параметров и работают быстрее своих аналогов.

- rkadapt(y0, to,t1,acc,D,k, s) — метод Рунге-Кутты с переменным шагом;
- bulstoer(y0,t0,t1,acc,D,k,s) — метод Булирша-Штера;
  - y0 — вектор начальных значений в точке to;
  - t0,t1 — начальная и конечная точки расчета;
  - acc — погрешность вычисления (чем она меньше, тем с лучшей точностью будет найдено решение; рекомендуется выбирать значения погрешности в районе 0.001);
  - D — векторная функция, задающая систему ОДУ;
  - к — максимальное число шагов, на которых численный метод будет находить решение;
  - s — минимально допустимая величина шага.

Как легко заметить, вместо числа шагов на интервале интегрирования ОДУ, в этих функциях необходимо задать точность расчета численным методом значения функций в последней точке. В этом смысле параметр acc похож на константу TOL, которая влияет на большинство встроенных численных алгоритмов Mathcad. Количество шагов и их расположение определяется численным методом автоматически, чтобы обеспечить эту точность. Два последних параметра нужны для того, чтобы пользователь мог искусственно повлиять на разбиение интервала на шаги. Параметр к служит для того, чтобы шагов не было чрезмерно много, причем, нельзя сделать к>юоо. Параметр s — для того чтобы ни один шаг не был слишком малым для появления больших погрешностей при разностной аппроксимации дифференциальных уравнений внутри алгоритма. Эти параметры следует задавать явно, исходя из свойств конкретной системы ОДУ. Как правило, проведя ряд тестовых расчетов, можно подобрать их оптимальный набор для каждого конкретного случая.

Пример использования функции bulstoer для того же примера (11.2—1) приведен в листинге 11.5. В его первых двух строках, как обычно, определяется система уравнений и начальные условия; в следующей строке матрице u присваивается решение, полученное с помощью bulstoer. Структура этой матрицы в точности такая же, как и в случае решения системы ОДУ посредством уже рассмотренных нами ранее встроенных функций *(см. разд. 11.3.1)*. Однако в данном случае нам интересна только последняя точка интервала. Поскольку сделанное численным методом количество шагов, т. е. размер матрицы и, заранее неизвестно, то его необходимо предварительно определить. Это сделано в четвертой строке листинга, присваивающей это число переменной M, в этой же строке оно выведено на экран. В предпоследней строке листинга осуществлен вывод решения системы ОДУ на конце интервала, т. е. в точке t = 50 в виде вектора. В последней строке для примера еще раз выводится искомое значение первой функции из системы ОДУ (сравните его с соответствующим местом вектора из предыдущей строки).

Листинг 11.5. Решение системы двух ОДУ

$$D(t,y) := \begin{pmatrix} y_1 \\ -y_0 - 0.1 \cdot y_1 \end{pmatrix}$$

$$y0 := \begin{pmatrix} 0.1 \\ 0 \end{pmatrix}$$

$$u := \text{buistoer}(y0, 0, 50, 10^{-5}, D, 300, 0.0001)$$

$$M := \text{length}(u^{\langle 1 \rangle}) - 1 \qquad M = 21$$

$$\left(u^T\right)^{\langle M \rangle} = \begin{pmatrix} 50 \\ 7.638 \times 10^{-3} \\ 2.648 \times 10^{-3} \end{pmatrix}$$

$$u_{M,1} = 7.638 \times 10^{-3}$$

Чтобы попробовать альтернативный численный метод, достаточно в листинге 11.5 заменить имя функции buistoer на rkadapt.

Внимание!

Функции buistoer и rkadapt (те, что пишутся с маленькой буквы) не предназначены для нахождения решения в промежуточных точках интервала, хотя они и выдают их в матрице-результате. На рис. 11.6 показаны фазовые портреты рассматриваемой системы ОДУ, полученные с помощью buistoer (результат листинга 11.5) и с помощью rkadapt (при соответствующей замене третьей строки листинга 11.5). Видно, что несмотря на высокую точность (10-5) и верный результат на конце интервала, левый график мало напоминает правильный фазовый портрет (см. рис. 11.5 или правый график на рис. 11.6), начиная быть приемлемым только при предельно допустимом для обсуждаемых функций значении acc=10-16.

В заключение остановимся на влиянии выбора параметра acc на расчеты. Для этого воспользуемся простой программой, представленной на листинге 11.6. В ней из матрицы решения все той же задачи Коши взято лишь полученное значение одной из функций на правой фанице интервала. Но зато этот результат оформлен в виде функции пользователя y(г), в качестве аргумента которой выбран параметр acc функции bulstoer.

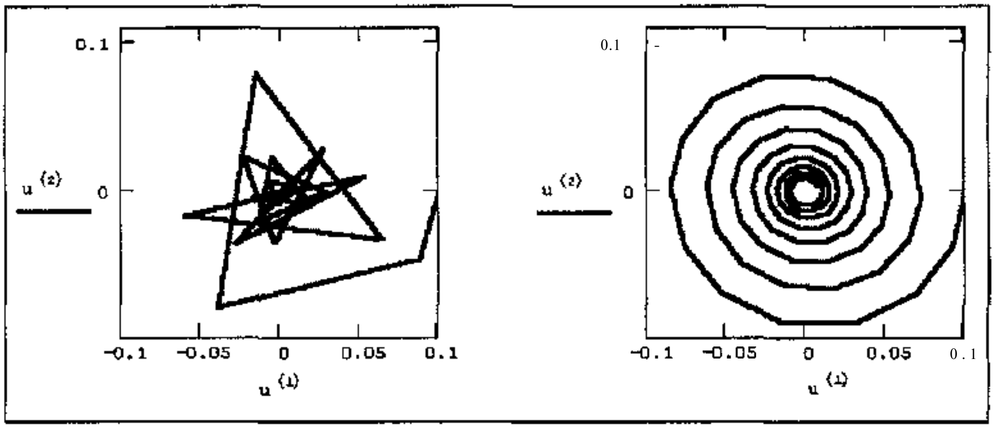


Рис. 11.6. Фазовый портрет, полученный buistoer (слева) и rkadapt (справа) (листинг 11.5)

Листинг 11.6. Использование решения ОДУ для определения функции пользователя

$$D(t, y) := \begin{pmatrix} y_1 \\ -y_0 - 0.1 \cdot y_1 \end{pmatrix} \qquad y0 := \begin{pmatrix} 0.1 \\ 0 \end{pmatrix}$$

$$y(\varepsilon) := \left| \begin{array}{l} u \leftarrow \mathrm{buistoer}\,(y0, 0, 50, \varepsilon, D, 30, 0.0001) \\ M \leftarrow \mathrm{length}\,(u^{\langle 1 \rangle}) - 1 \\ u_{M,1} \end{array} \right.$$

Вычисленный вид y(е) показан на рис. 11.7 вместе с аналогичным результатом для функции rkadapt. Как видно, в данном примере численные методы работают несколько по-разному. Метод Рунге-Кутты дает результат тем ближе к истинному, чем меньше выбирается ε=acc. Метод Булирша-Штера демонстрирует менее естественную зависимость y(е): даже при относительно больших ε реальная точность остается хорошей (намного лучше метода Рунге-Кутты). Поэтому для экономии времени расчетов (подчеркнем еще раз: для данной конкретной задачи) в функции buistoer можно выбирать и большие acc.

Чтобы обеспечить заданную точность, алгоритмы, реализованные во встроенных функциях, могут изменять как количество шагов, разбивающих интервал (t0,t1), так и их расположение вдоль интервала. Чтобы выяснить, на сколько шагов разбивался интервал при расчетах y(e) на рис. 11.7 для каждого ε, следует вычислить размер получающейся матрицы. Для этого можно, например, определить подобные функции:

$$M(\varepsilon) := \mathrm{length}\left(\mathrm{rkadapt}(y0, 0, 50, e, D, 100, 0.0001)^{\langle 1 \rangle}\right).$$

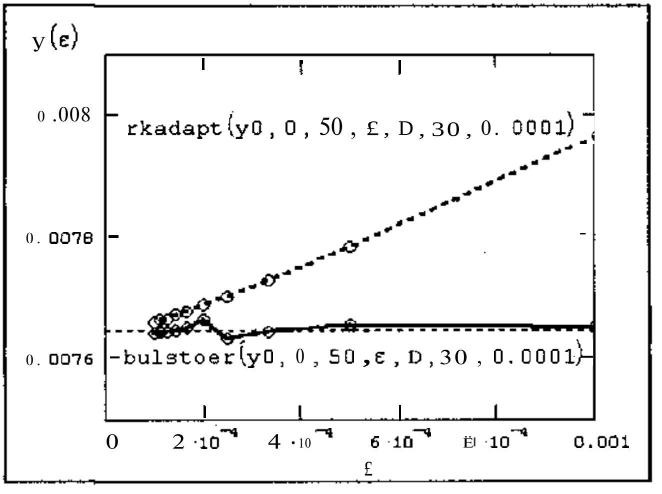


Рис. 11.7. Зависимость расчетного значения одного из уравнений системы ОДУ на конце интервала от параметра acc (листинг 11.6)

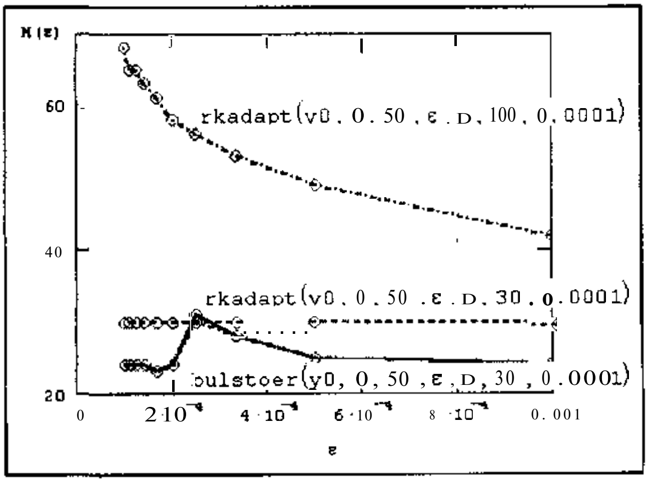


Рис. 11.8. Зависимость числа шагов от параметра acc численных методов

Сравнив два результата применения rkadapt для `k=30` и `k=100`, обратите внимание (рис. 11.8), как еще один параметр — максимальное число шагов к, влияет на вид $M(\varepsilon)$. Заметим, что такие же изменения параметра `k` на расчет $M(\varepsilon)$ посредством функции `bulstoer` влияют слабо.

Таким образом, проводя тестовые расчеты для различных задач и подбирая наилучший набор параметров, можно существенно сэкономить ресурсы компьютера. Конечно, проводить подобный анализ стоит в случаях, когда время расчетов играет важную роль.

11.3.3. Некоторые примеры

В предыдущих разделах были использованы примеры исключительно *линейных* уравнений, т. е. содержащих только первую степень неизвестных функций и их производных. Между тем, многие *нелинейные* уравнения демонстрируют совершенно удивительные свойства, причем решение большинства из них можно получить лишь численно. Рассмотрим несколько наиболее известных классических примеров систем ОДУ, имея в виду, что читателю они могут пригодиться как в познавательных, так и в практических целях. Это модели динамики популяций (Вольтерры), генератора автоколебаний (Ван дер Поля), турбулентной конвекции (Лоренца) и химической реакции с диффузией (Пригожина). Все они (впрочем, как и уже приведенные выше в этой главе) содержат производные по времени t и описывают динамику различных физических параметров. Задачи Коши для таких моделей называют *динамическими системами,* и для их изучения центральным моментом является анализ фазовых портретов, т. е. решений, получающихся при выборе всевозможных начальных условий.

Примечание

В большинстве примеров, изложенных ниже, для построения фазового портрета рассчитывается несколько решений для разных начальных условий.

Ограничимся в дальнейшем минимальными комментариями и приведем листинги и графики решений без подробного обсуждения.

Модель "хищник—жертва"

Модель взаимодействия "хищник—жертва" независимо предложили в 1925—1927 гг. Лотка и Вольтерра. Два дифференциальных уравнения (листинг 11.7) моделируют временную динамику численности двух биологических популяций жертвы y_0 и хищника y_1. Предполагается, что жертвы размножаются с постоянной скоростью c, а их численность убывает вследствие поедания хищниками. Хищники же размножаются со скоростью, пропорциональной количеству пиши (с коэффициентом `r`), и умирают естественным образом (смертность определяется константой D). В листинге рассчитываются три решения D, G, *P* для разных начальных условий.

Листинг 11.7. Модель "хищник—жертва"

$C := 0.1 \qquad D := 1 \qquad r := 0.1$

$$F(t,y) := \begin{pmatrix} C \cdot y_0 - r \cdot y_0 \cdot y_1 \\ -D \cdot y_1 + r \cdot y_0 \cdot y_1 \end{pmatrix}$$

$M := 500$

$t0 := 0 \qquad t1 := 100$

$$D := \mathrm{rkfixed}\left[\begin{pmatrix} 10 \\ 8 \end{pmatrix}, t0, t1, M, F\right]$$

$$G := \mathrm{rkfixed}\left[\begin{pmatrix} 10 \\ 4 \end{pmatrix}, t0, t1, M, F\right]$$

$$P := \mathrm{rkfixed}\left[\begin{pmatrix} 10 \\ 1.5 \end{pmatrix}, t0, t1, M, F\right]$$

Модель замечательна тем, что в такой системе наблюдаются циклическое увеличение и уменьшение численности и хищника (рис. 11.9), и жертвы, так часто наблюдаемое в природе. Фазовый портрет системы представляет собой концентрические замкнутые кривые, окружающие одну стационарную точку, называемую *центром*. Как видно, модельные колебания численности обеих популяций существенно зависят от начальных условий — после каждого периода колебаний система возвращается в ту же точку. Динамические системы с таким поведением называют *негрубыми*.

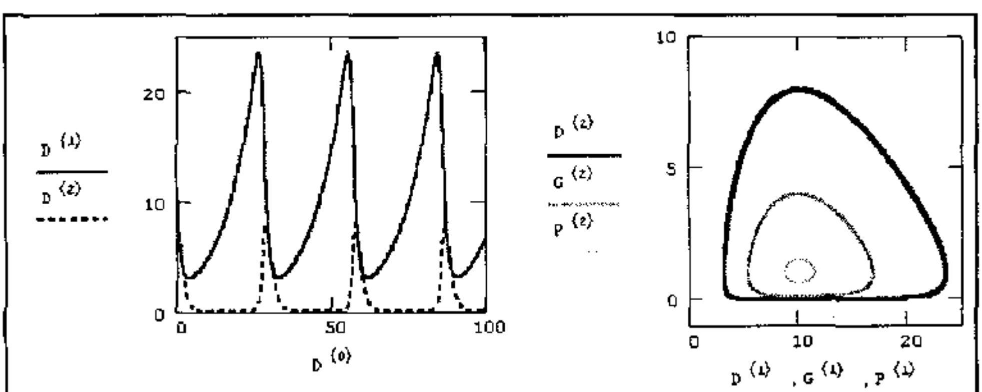


Рис. 11.9. График решения (слева) и фазовый портрет (справа) системы "хищник—жертва" (листинг 11.7)

Автоколебания

Рассмотрим решение уравнения Ван дер Поля, описывающего электрические колебания в замкнутом контуре, состоящем из соединенных последовательно конденсатора, индуктивности, нелинейного сопротивления и элементов, обеспечивающих подкачку энергии извне (листинг 11.8). Неизвестная функция времени y(t) имеет смысл электрического тока, а в параметре μ заложены количественные соотношения между составляющими электрической цепи, в том числе и нелинейной компонентой сопротивления.

Листинг 11.8. Модель Ван дер Поля (μ=1)

$\mu := 1$

Given

$$\frac{d^2}{dt^2}y(t)-\mu\cdot\left(1-y(t)^2\right)\cdot\frac{d}{dt}y(t)+y(t)=0$$

$y(0)=0.01$

$y'(0)=0$

$y := \mathrm{Odesolve}(t, 30)$

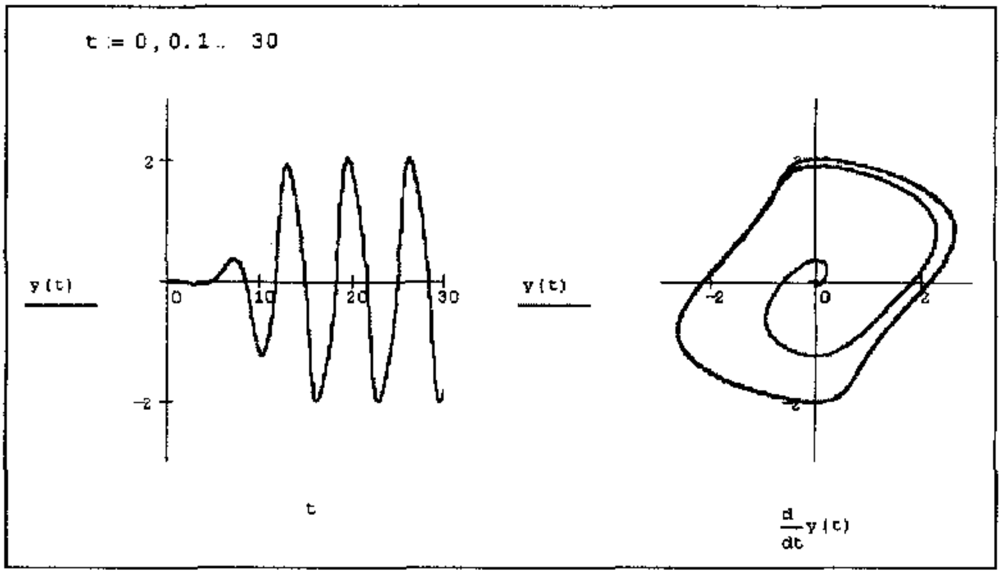


Рис. 11.10. График решения (слева) и фазовый портрет (справа) уравнения Ван дер Поля (листинг 11.8)

Решением уравнения Ван дер Поля являются колебания, вид которых для $\mu = 1$ показан на рис. 11.10. Они называются *автоколебаниями* и принципиально отличаются от рассмотренных нами ранее *(например, колебаний маятника в разд. 11.3.2)* тем, что их характеристики (амплитуда, частота, спектр) не зависят от начальных условий, а определяются исключительно свойствами самой динамической системы. Через некоторое время расчетов после выхода из начальной точки решение выходит на один и тот же цикл колебаний, называемый *предельным циклом*. Аттрактор типа предельного цикла является замкнутой кривой на фазовой плоскости. К нему асимптотически притягиваются все окрестные траектории, выходящие из различных начальных точек, как изнутри (рис. 11.10), так и снаружи (рис. 11.11) предельного цикла.

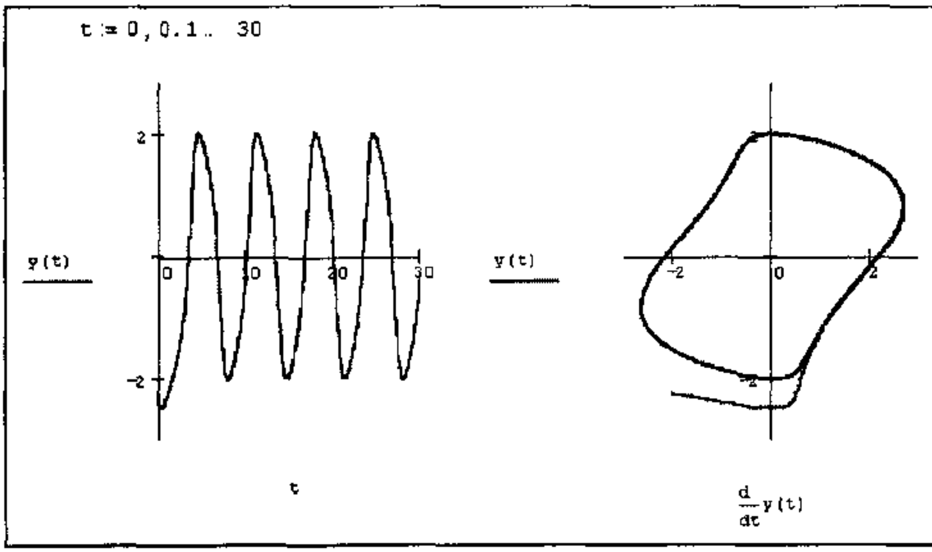


Рис. 11.11. Решение уравнения Ван дер Поля при других начальных условиях y=-2, y'=-3

Примечание

Если компьютер у Вас не самый мощный, то расчет фазового портрета с рис. 11.10—11.11 в Mathcad может занять относительно продолжительное время, что связано с численным определением сначала решения y(t), а потом его производной. Время расчетов можно было бы существенно сократить, если использовать вместо вычислительного блока **Given/Odesolve** одну из встроенных функций, которые выдают решение в виде матрицы, например `rkfixed`.

Аттрактор Лоренца

Одна из самых знаменитых динамических систем предложена в 1963 г. Лоренцем в качестве упрощенной модели конвективных турбулентных движений жидкости в нагреваемом сосуде тороидальной формы. Система состоит из трех ОДУ и имеет три параметра модели (листинг 11.9). Поскольку неизвестных функций три, то фазовый портрет системы должен определяться не на плоскости, а в трехмерном пространстве.

Листинг 11.9. Модель Лоренца

$a := 10 \qquad r := 27 \qquad b := \frac{8}{3}$

$$y0 := \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix}$$

$$F(t,y) := \begin{pmatrix} \sigma \cdot y_1 - \sigma \cdot y_0 \\ -y_0 \cdot y_2 + r \cdot y_0 - y_1 \\ y_1 \cdot y_0 - b \cdot y_2 \end{pmatrix}$$

$t0 := 0 \qquad t1 := 30$

$N := 1000$

$D := \mathrm{rkfixed}(y0, t0, t1, N, F)$

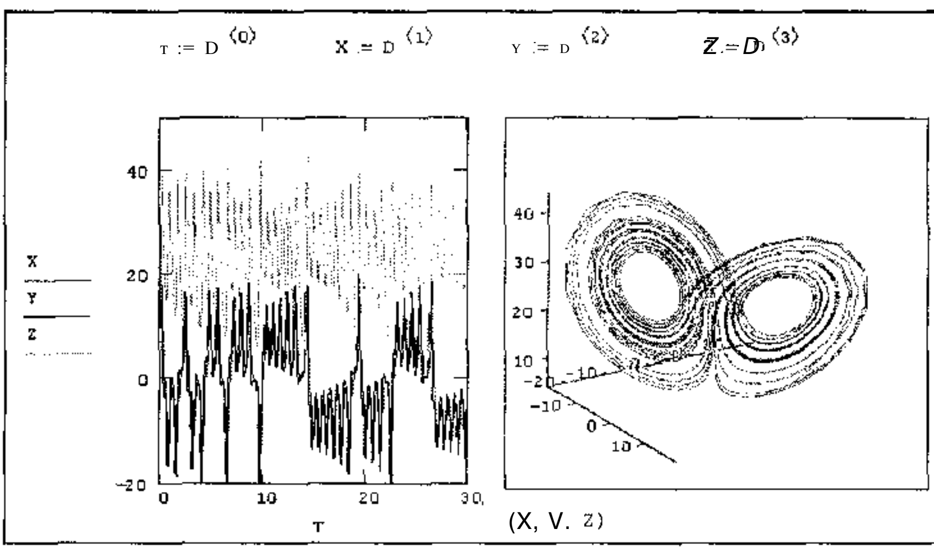


Рис. 11.12. Аттрактор Лоренца (листинг 11.9)

Решением системы Лоренца при определенном сочетании параметров (рис. 11.12) является *странный аттрактор* (или *аттрактор Лоренца)* — притягивающее множество траекторий на фазовом пространстве, которое по виду идентично случайному процессу. В некотором смысле, аттрактор Лоренца является стохастическими автоколебаниями, которые поддерживаются в динамической системе за счет внешнего источника.

Решение в виде странного аттрактора появляется только при некоторых сочетаниях параметров. В качестве примера на рис. 11.13 приведен результат для r=10 и тех же значений остальных параметров. Как видно, аттрактором в этом случае является фокус. Перестройка типа фазового портрета происходит в области промежуточных r. Критическое сочетание параметров, при которых фазовый портрет системы качественно меняется, называется в теории динамических систем точкой *бифуркации*. Физический смысл бифуркации в модели Лоренца, согласно современным представлениям, описывает переход ламинарного движения жидкости к турбулентному.

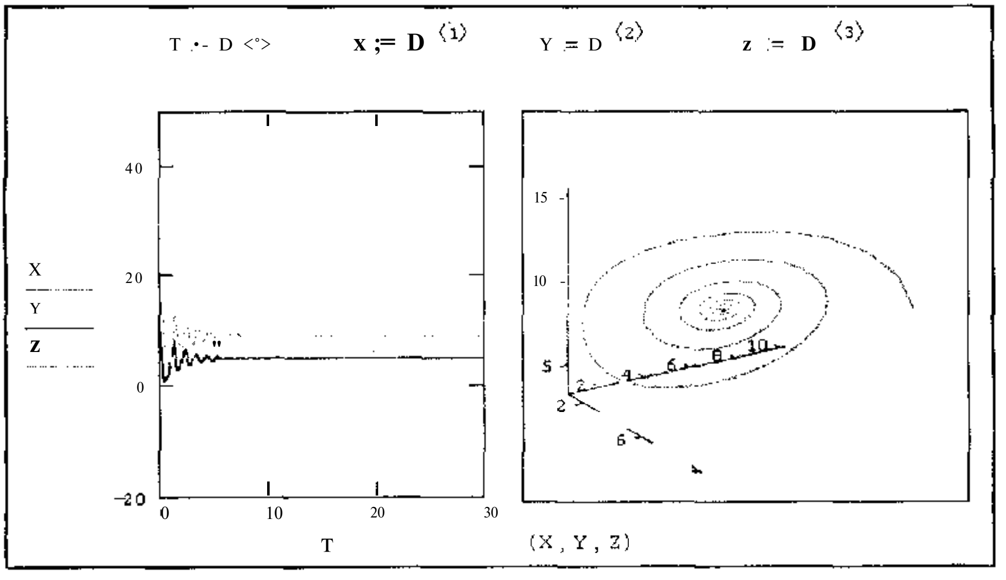


Рис. 11.13. Решение системы Лоренца с измененным параметром r=10

Замечательно, что решение подобных нелинейных динамических систем можно получить только численно, поэтому их изучение стало бурно развиваться с ростом возможностей вычислительной техники в последние полвека.

11.4. Фазовый портрет динамической системы

До сих пор в этой главе в качестве примеров расчета динамических систем мы приводили графики траекторий на фазовой плоскости. Однако для надежного исследования фазового портрета необходимо решить систему ОДУ большое количество раз с самыми разными начальными условиями (и, возможно, с разным набором параметров модели), чтобы посмотреть, к каким аттракторам сходятся различные траектории. В Mathcad можно реализовать эту задачу, например, в форме алгоритма, приведенного в листинге 11.10 для решения системы уравнений автокаталитической химической реакции с диффузией. Эта модель, называемая моделью *брюсселятора,* предложена в 1968 г. Лефевром и Пригожиным. Неизвестные функции отражают динамику концентрации промежуточных продуктов некоторой реальной химической реакции. Параметр модели в равен исходной концентрации катализатора.

Листинг 11.10. Построение фазового портрета для модели брюсселятора

$$V := \begin{pmatrix} 0 & 0 & 2.5 & 1.5 & 0.5 & 1 & 1 & 1.5 & 0.1 & 0.5 \\ 0.5 & 1.5 & 0 & 0 & 1 & 0 & 1 & 2 & 0.1 & 0.2 \end{pmatrix}$$

$B := 0.5$

$$D(t, y) := \begin{bmatrix} -(B+1)\cdot y_0 + (y_0)^2 \cdot y_1 + 1 \\ B \cdot y_0 - (y_0)^2 \cdot y_1 \end{bmatrix}$$

$t0 := 0 \qquad t1 := 10$

$M := 100$

```
U:= | y ← V⟨0⟩
    | Z ← rkfixed(y, t0, t1, M, D)
    | Z1⟨0⟩ ← Z⟨0⟩
    | Z1⟨1⟩ ← Z⟨1⟩
    | Z1⟨2⟩ ← Z⟨2⟩
    | for k ∈ 1 .. last[(V^T)⟨1⟩]
    |     | y ← V⟨k⟩
    |     | Z ← rkfixed(y, t0, t1, M, D)
    |     | Z2⟨0⟩ ← Z⟨0⟩
    |     | Z2⟨1⟩ ← Z⟨1⟩
    |     | Z2⟨2⟩ ← Z⟨2⟩
    |     | Z1 ← stack(Z1, Z2)
    | Z1
```

Предложенный алгоритм формирует из отдельных матриц решений системы ОДУ с разными начальными условиями объединенную матрицу U. Пары начальных условий задаются в первой строке листинга в виде матрицы v размера 2×10. Последнее означает построение десяти траекторий. Для того чтобы поменять количество траекторий, измените соответствующим образом размер этой матрицы. Затем (рис. 11.14) элементы матрицы U выводятся на график в виде отдельных точек. Отсутствие соединения точек линиями является недостатком алгоритма, но это плата за возможность представить в Mathcad несложным образом сразу большое количество траекторий на фазовой плоскости.

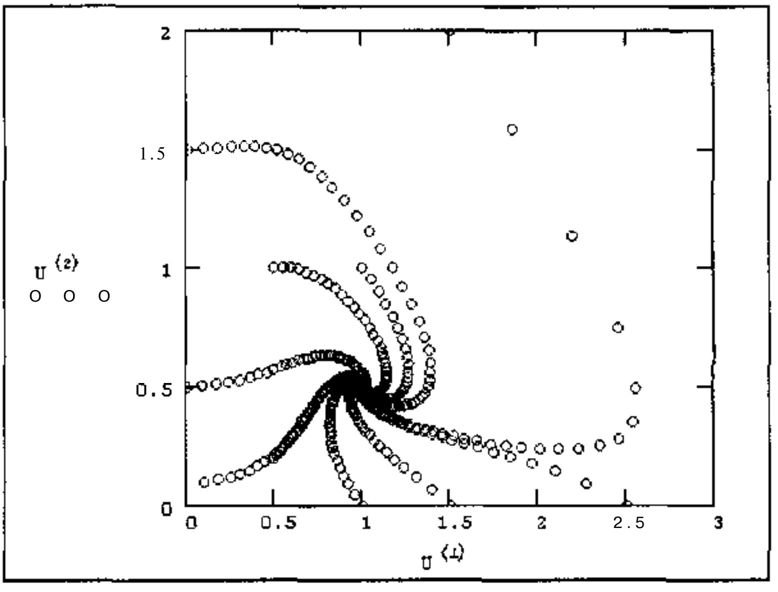


Рис. 11.14. Фазовый портрет брюсселятора при B=0 . 5 (листинг **11.10**)

Как видно из рис. 11.14, все траектории, вышедшие из разных точек, асимптотически стремятся к одному и тому же аттрактору (1,0.5). Из теории динамических систем нам известно, что такой аттрактор называется *узлом (с узлом мы уже встречались в примерах разд. 11.1)*. Конечно, в общем случае при анализе фазового портрета желательно "прощупать" большее число траекторий, задавая более широкий диапазон начальных условий. Не исключено, что в других областях фазовой плоскости траектории будут сходиться к другим аттракторам.

Эволюцию фазового портрета брюсселятора можно наблюдать, проводя расчеты с различным параметром в. При его увеличении узел будет сначала постепенно смещаться в точку с координатами (1,в), пока не достигнет бифуркационного значения в=2. В этой точке происходит качественная перестройка портрета, выражающаяся в рождении предельного цикла. При дальнейшем увеличении в происходит лишь количественное изменение параметров этого цикла. Решение, полученное при B=2.5, показано на рис. 11.15.

Примечание

Чтобы найти аттракторы динамической системы, как известно, нужно решить систему алгебраических уравнений, получающуюся из системы ОДУ заменой нулями их левых частей. Эти задачи также удобно решать средствами Mathcad *(см. гл. 8).* В частности, исследование зависимости фазового портрета от параметров системы ОДУ и поиск бифуркаций можно проводить методами продолжения *(см. разд. "Метод продолжения по параметру" гл. 8).*

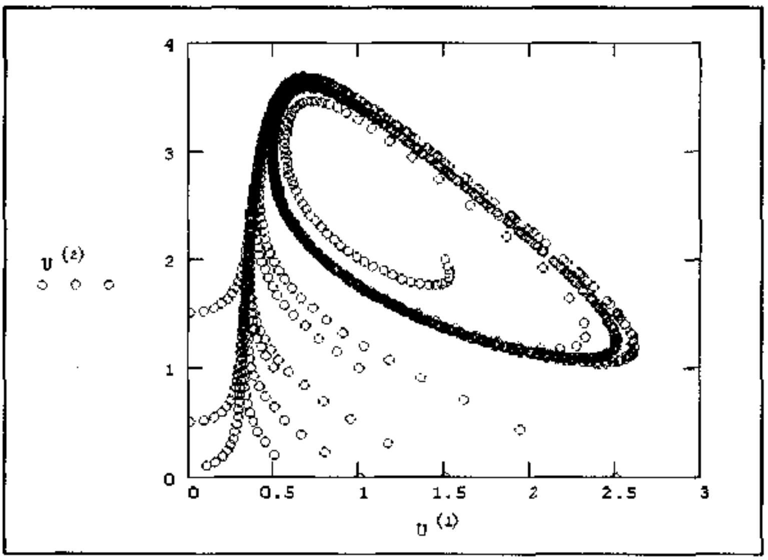


Рис. 11.15. Фазовый портрет брюсселятора при B=2.5

Читатели, сталкивающиеся с расчетом динамических систем, несомненно оценят возможности Mathcad по построению фазовых портретов и исследованию бифуркаций. Возможно также, что они найдут лучшие программные решения этой задачи, чем алгоритм, предложенный в данном разделе автором.

11.5. Жесткие системы ОДУ

До сих пор мы имели дело с "хорошими" уравнениями, которые надежно решались численными методами Рунге-Кутты. Однако имеется класс так называемых *жестких* (stiff) систем ОДУ, для которых стандартные методы практически неприменимы, поскольку их решение требует исключительно малого значения шага численного метода. Некоторые из специальных алгоритмов, разработанных для этих систем, реализованы в Mathcad.

11.5.1. Что такое жесткие ОДУ?

Строгого общепринятого математического определения жестких ОДУ нет. В рамках этой книги будем считать, что жесткие системы — это те уравнения, решение которых получить намного проще с помощью определенных неявных методов, чем с помощью явных методов (типа тех, что были рассмотрены в предыдущих разделах). Примерно такое определение было предложено в 1950-х годах классиками в этой области Кертиссом и Хиршфельдером. Начнем разговор о жестких ОДУ с примера нежесткого уравнения (листинг 11.11), решение которого показано на рис. 11.16. На том же графике показано решение подобного ОДУ с другим коэффициентом при правой части, равным не -10, а -30. Решение обоих уравнений не составило труда, и численный метод Рунге-Кутты дал правильный результат.

Листинг 11.11. Решение нежесткого ОДУ

```
Given
d
—y(t) = -10·(y(t) - cos(t))
dt
y(0) = 1
y := Odesolve(t, 1)
```

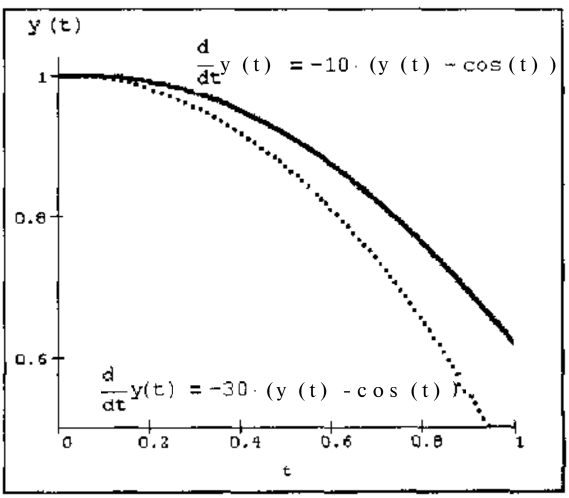


Рис. 11.16. Решение нежестких ОДУ методом Рунге-Кутты (листинг 11.11)

На рис. 11.17 показано решение того же ОДУ с коэффициентом -50. Вас, несомненно, должен насторожить результат, выданный Mathcad. Характерная "разболтка" решения говорит о неустойчивости алгоритма. Первое, что можно сделать, — увеличить количество шагов в методе Рунге-Кутты. Для этого достаточно добавить третий параметр step в функцию Odesolve(t,1,step). После нескольких экспериментов можно подобрать такое значение step, которое будет обеспечивать устойчивость решения. Читатель может самостоятельно убедиться, что при step>20 "разболтка" пропадает, и решение становится похожим на графики, показанные на рис. 11.16.

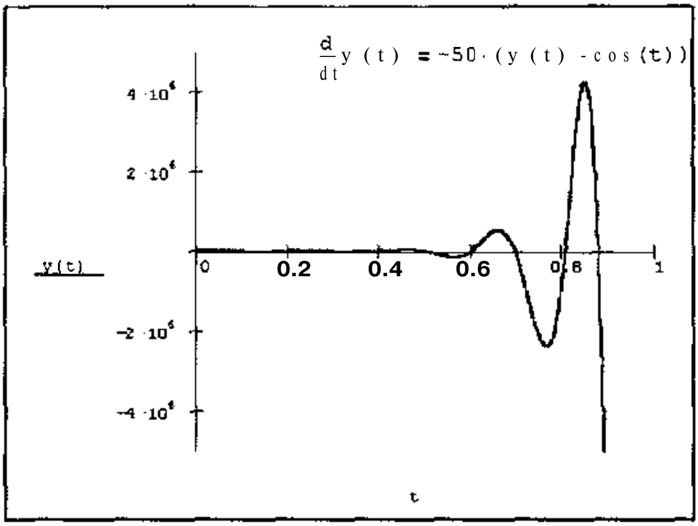


Рис. 11.17. Неверное решение более жесткого ОДУ методом Рунге-Кутты

Таким образом, во-первых, мы выяснили, что одни и те же уравнения с разными параметрами могут быть как жесткими, так и нежесткими. Во-вторых, чем жестче уравнение, тем больше шагов в обычных численных методах требуется для его устойчивого решения. С классическим примером ОДУ из листинга 11.11 все получилось хорошо, т. к. оно было не очень жестким, и небольшое увеличение числа шагов разрешило все проблемы. Для решения обычными методами более жестких уравнений требуются миллионы, миллиарды и даже большее число шагов.

Примечание

Некоторые ученые замечают, что в последние годы методы Рунге-Кутты стали уступать свое главенствующее положение среди алгоритмов решения ОДУ методам, способным решать жесткие задачи.

Исторически, интерес к жестким системам возник в середине XX века при изучении уравнений химической кинетики с одновременным присутствием очень медленно и очень быстро протекающих химических реакций. Тогда неожиданно оказалось, что считавшиеся исключительно надежными методы

Рунге-Кутты стали давать сбой при расчете этих задач. Рассмотрим классическую модель взаимодействия трех веществ (Робертсон, 1966), которая как нельзя лучше передает смысл понятия жесткости ОДУ.

Пусть вещество "O" медленно превращается в "1": "0"→"1" (со скоростью 0.1), вещество "1" при каталитическом воздействии самого себя превращается очень быстро в вещество "2": "1"+"1"→"2"+ "1" (10^3). И, наконец, подобным образом (но со средней скоростью) реагируют вещества "2" и "1": "1"+"2"→"0"+ "2" (10^2). Система ОДУ, описывающая динамику концентрации реагентов, с попыткой решения методом Рунге-Кутты, приведена в листинге 11.12.

Листинг 11.12. Жесткая система ОДУ химической кинетики

$$F(t,y) := \begin{pmatrix} -0.1\cdot y_0 + 10^2\cdot y_1\cdot y_2 \\ 0.1\cdot y_0 - 10^2\cdot y_1\cdot y_2 - 10^3\cdot y_1 \\ 10^3\cdot y_1 \end{pmatrix} \qquad y0 := \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

D := rkfixed (yO , 0 , 50 , 20000 , F)

Бросается в глаза сильно различающийся порядок коэффициентов при разных слагаемых. Именно степень этого различия чаще всего и определяет жесткость системы ОДУ. В качестве соответствующей характеристики выбирают матрицу Якоби (якобиан) векторной функции F(t,y), т.е. функциональную матрицу, составленную из производных F(t,y) *(см. разд. "Частные производные" гл. 7)*. Чем вырожденнее матрица Якоби, тем жестче система уравнений. В приведенном примере определитель якобиана и вовсе равен нулю при любых значениях y_0, y_1 и y_2 (листинг 11.13, вторая строка). В первой строке листинга 11.13 приведено напоминание способа вычисления якобиана средствами Mathcad на примере определения элементов его первой строки.

Листинг 11.13. Якобиан рассматриваемой системы ОДУ химической кинетики

$$\frac{\partial}{\partial x}F\left[t,\begin{pmatrix} x \\ y_1 \\ y_2 \end{pmatrix}\right]_0 \to -.1 \qquad \frac{\partial}{\partial x}F\left[t,\begin{pmatrix} y_0 \\ x \\ y_2 \end{pmatrix}\right]_0 \to 100\cdot y_2 \qquad \frac{\partial}{\partial x}F\left[t,\begin{pmatrix} y_0 \\ y_1 \\ x \end{pmatrix}\right]_0 \to 100\cdot y_1$$

$$\left|\begin{pmatrix} -0.1 & 10^2\cdot y_2 & 10^2\cdot y_1 \\ 0.1 & -10^2\cdot y_2 - 10^3 & -10^2\cdot y_1 \\ 0 & 10^3 & 0 \end{pmatrix}\right| \to 0$$

Для примера, приведенного в листинге 11.12, стандартным методом Рунге-Кутты все-таки удается найти решение (оно показано на рис. 11.18). Однако для этого требуется очень большое число шагов, M=20000, что делает расчеты очень медленными. При меньшем числе шагов численному алгоритму не удается найти решение. В процессе работы алгоритма оно расходится, и Mathcad вместо результата выдает ошибку о превышении предельно большого числа.

Еще один факт, на который стоит обратить внимание, — это различие в порядке величины получающегося решения. Как видно из рис. 11.18, концентрация первого реагента y_1 существенно (в тысячи раз) превышает концентрацию остальных. Это свойство также очень характерно для жестких систем.

Примечание

В принципе, можно было бы снизить жесткость системы "вручную", применяя масштабирование. Для этого нужно искусственно уменьшить искомую функцию y1, к примеру, в тысячу раз, разделив все слагаемые в системе ОДУ, содержащие y1, на 1000. После масштабирования для решения полученной системы методом Рунге-Кутты будет достаточно взять всего M=20 шагов.

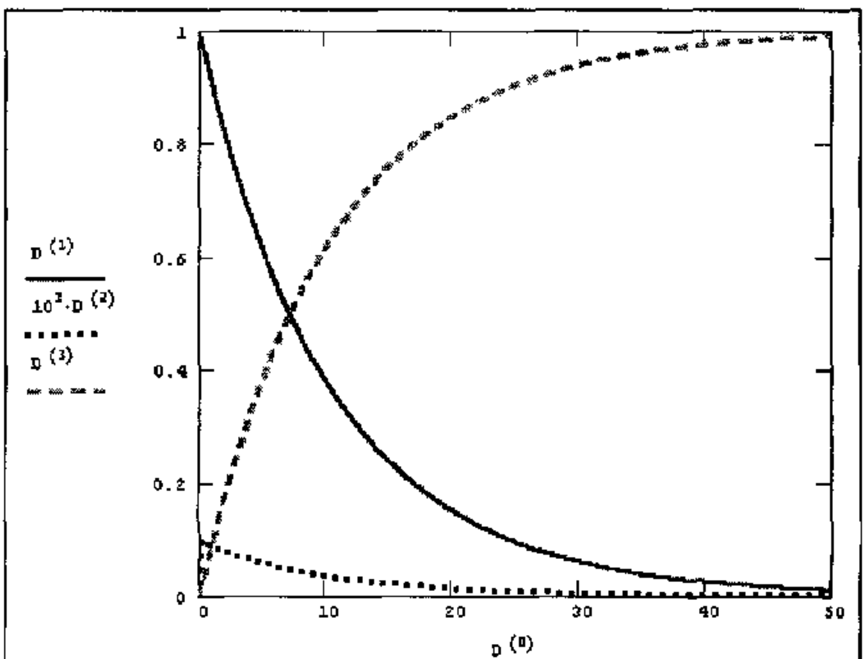


Рис. 11.18. Решение жесткой системы ОДУ химической кинетики методом Рунге-Кутты (листинг 11.12)

11.5.2. Функции для решения жестких ОДУ

Решение жестких систем дифференциальных уравнений можно осуществить только с помощью встроенных функций, аналогичных по действию семейству рассмотренных выше функций для обычных ОДУ.

- `Radau(y0,t0,t1,M,F)` — алгоритм RADAUS для жестких систем ОДУ;
- `Stiffb(yo,t0,t1,M,F,J)` — алгоритм Булирша-Штера для жестких систем ОДУ;
- `Stiffr(y0,t0,t1,M,F,J)` — алгоритм Розенброка для жестких систем ОДУ;
  - `y0` — вектор начальных значений в точке to;
  - `t0,t1` — начальная и конечная точки расчета;
  - м — число шагов численного метода;
  - F — векторная функция F(t,y) размера `1×N`, задающая систему ОДУ;
  - `J` - матричная функция `J(t,y)` размера `(N+1)×N`, составленная из вектора производных функции F(t,y) по t (правый столбец) и ее якобиана (N левых столбцов).

Как Вы можете заметить, для двух последних функций серьезным отличием от функций, решающих нежесткие системы, является добавление к стандартному набору параметров дополнительной матричной функции, задающей якобиан системы ОДУ. Решение выдается в виде матрицы, по форме идентичной аналогичным функциям решения нежестких задач Коши.

Покажем действие этих алгоритмов на том же примере жесткой системы ОДУ химической кинетики (листинг 11.14). Обратите внимание, как следует представлять в данном случае якобиан, сравнив задание матричной функции в предпоследней строке листинга 11.14 с заданием якобиана из листинга 11.13.

Листинг 11.14. Решение жесткой системы ОДУ химической кинетики

$$y0 := \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$F(t,y) := \begin{pmatrix} -0.1 \cdot y_0 + 10^2 \cdot y_1 \cdot y_2 \\ 0.1 \cdot y_0 - 10^2 \cdot y_1 \cdot y_2 - 10^3 \cdot y_1 \\ 10^3 \cdot y_1 \end{pmatrix}$$

$$J(t,y) := \begin{pmatrix} 0 & -0.1 & 10^2\cdot y_2 & 10^2\cdot y_1 \\ 0 & 0.1 & -10^2\cdot y_2 - 10^3 & -10^2\cdot y_1 \\ 0 & 0 & 10^3 & 0 \end{pmatrix}$$

D:= stiffb (yO . 0 , 50 , 20 , F, J)

Расчеты показывают, что для получения того же результата (см. рис. 11.18) оказалось достаточно в тысячу раз меньшего количества шагов численного алгоритма, чем для стандартного метода Рунге-Кутты! Примерно во столько же раз требуется меньше компьютерного времени на проведение расчетов. Стоит ли говорить, что, если Вы имеете дело с жесткими (в той или иной степени) системами, применение описанных специальных алгоритмов просто необходимо.

Важно заметить, что до сих пор мы имели дело с примером не очень жесткой системы. Попробуйте вместо скоростей упомянутых химических реакций *(см. разд. 11.5.1),* 0.1, 10^3 и 10^2 взять другие числа, например о.О5, ю^4 и ю^7, соответственно. Заметим, что такое соотношение скоростей часто встречается в прикладных задачах химической кинетики и определяет куда более жесткую систему ОДУ. Ее уже никак не удается решить стандартными методами, поскольку число шагов численного метода должно быть просто гигантским. А между тем, алгоритмы для жестких ОДУ справляются с этой задачей с легкостью (рис. 11.19), причем практически при тех же значениях шага, что были взяты в листинге 11.14. Обратите внимание, что порядки величины решений для концентраций различных веществ на рис. 11.19 различаются еще сильнее, чем в предыдущем (менее жестком) примере.

Примечание

Это еще раз доказывает, что одна и та же система ОДУ с различными коэффициентами может быть жесткой в разной степени. В частности, приведенный выше пример генератора Ван дер Поля с параметром μ=5000 — это уже пример жесткой задачи.

В заключение приведем соответствующие встроенные функции, которые применяются для решения жестких систем ОДУ не на всем интервале, а только в одной заданной точке t1.

- radau(y0,t0,t1,acc, F, k,s) — метод RADAUS
- stiffb(y0,to,t1,acc,F,J,k,s) — метод Булирша-Штера
- stiffr(y0,t0,t1,acc,F,J,k,s) — метод Розенброка

Имена этих функций пишутся с маленькой буквы, а их действие и набор параметров полностью аналогичны рассмотренным нами ранее для функций, относящихся к решению в заданной точке нежестких систем *(см. разд. 11.3.2)*. Отличие заключается в специфике применяемого алгоритма и необходимости задания матричной функции якобиана `J(t,y)`.

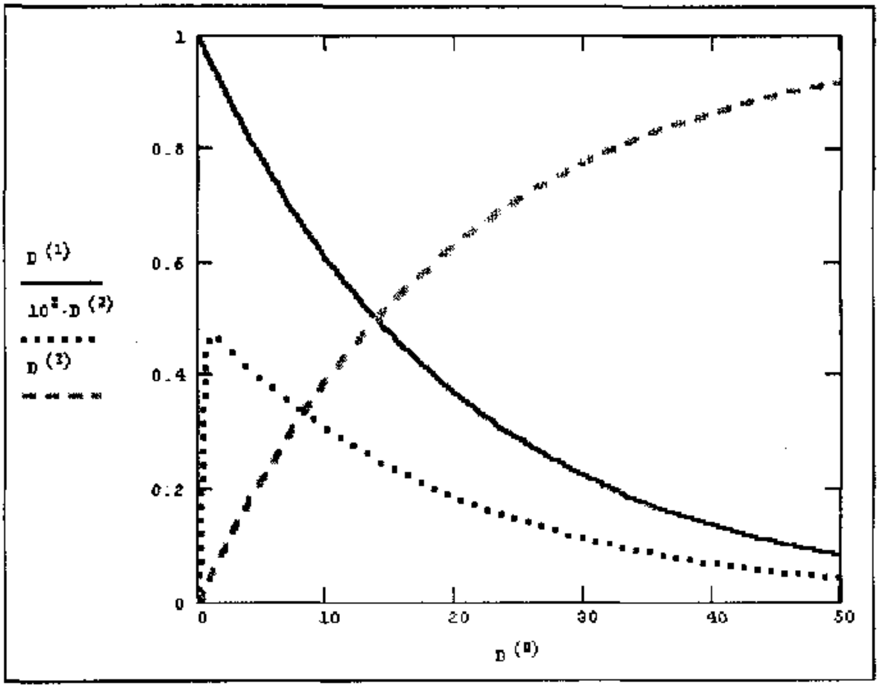


Рис. 11.19. Решение более жесткой системы ОДУ химической кинетики методом Розенброка

ГЛАВА 12

Краевые задачи

В этой главе рассматриваются краевые задачи для обыкновенных дифференциальных уравнений (ОДУ). Средства Mathcad позволяют решать краевые задачи для систем ОДУ, в которых часть граничных условий поставлена в начальной точке интервала, а остальная часть — в его конечной точке *(см. разд. 12.1)*. Также возможно решать уравнения с граничными условиями, поставленными в некоторой внутренней точке. Для решения краевых задач в Mathcad предусмотрены соответствующие встроенные функции, реализующие алгоритм пристрелки *(см. разд. 12.1.2)*.

Краевые задачи во множестве практических приложений часто зависят от некоторого числового параметра. При этом решение существует не для всех его значений, а лишь для счетного их числа. Такие задачи называют задачами на собственные значения *(см. разд. 12.2)*.

Несмотря на то, что, в отличие от задач Коши для ОДУ, в Mathcad не предусмотрены встроенные функции для решения жестких краевых задач, с ними все-таки можно справиться, применив программирование разностных схем *(см. разд. 12.3)*.

12.1. Краевые задачи для ОДУ

Постановка краевых задач для ОДУ отличается от задач Коши, рассмотренных в *главе 11,* тем, что граничные условия для них ставятся не в одной начальной точке, а на обеих границах расчетного интервала. Если имеется система N обыкновенных дифференциальных уравнений первого порядка, то часть из N условий может быть поставлена на одной границе интервала, а оставшиеся условия — на противоположной границе.

Примечание

Дифференциальные уравнения высших порядков можно свести к эквивалентной системе ОДУ первого порядка *(см. гл. 11)*.

12.1.1. О постановке краевых задач

Чтобы лучше понять, что из себя представляют краевые задачи, рассмотрим их постановочную часть на конкретном физическом примере модели взаимодействия встречных световых пучков. Предположим, что надо определить распределение интенсивности оптического излучения в пространстве между источником (лазером) и зеркалом, заполненном некоторой средой (рис. 12.1). Будем считать, что от зеркала отражается R-я часть падающего излучения (т. е. его коэффициент отражения равен R), а среда как поглощает излучение с коэффициентом ослабления a(x), так и рассеивает его. Причем коэффициент рассеяния назад равен r(x). В этом случае закон изменения интенсивности $y_0(x)$ излучения, распространяющегося вправо, и интенсивности $y_1(x)$ излучения влево определяется системой двух ОДУ первого порядка:

$$\begin{aligned} \frac{0}{dx} &= -a(x) \cdot y_0(x) + r(x) \cdot y_1(x) \\ \frac{dy_1(x)}{dx} &= a(x) \cdot y_1(x) - r(x) \cdot y_0(x) \end{aligned} \qquad (1)$$

Для правильной постановки задачи требуется, помимо уравнений, задать такое же количество граничных условий. Одно из них будет выражать известную интенсивность излучения ю, падающего с левой границы x=o, а второе — закон отражения на его правой границе x=1:

$$\begin{aligned} y_0(0) &= I0 \\ y_1(1) &= R \cdot y_0(1) \end{aligned} \qquad \textbf{(2)}$$

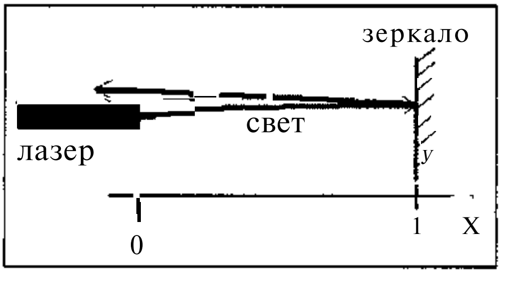


Рис. 12.1. Модель для постановки краевой задачи

Полученную задачу называют *краевой* (boundary value problem), поскольку условия поставлены не на одной, а на обеих границах интервала {0,1). И, в связи с этим, их не решить методами предыдущей главы, предназначенными для задач с начальными условиями. Далее для показа возможностей Mathcad будем использовать этот пример с R=1 И конкретным видом a(x)=const=1 и r(x)=const=0.1, описывающим случай *изотропного* (не зависящего от координаты x) рассеяния.

Примечание

Модель рис. 12.1 привела к краевой задаче для системы *линейных* ОДУ. Она имеет аналитическое решение в виде комбинации экспонент. Более сложные, *нелинейные* задачи, возможно решить только численно. Нетрудно сообразить, что модель станет нелинейной, если сделать коэффициенты ослабления и рассеяния зависящими от интенсивности излучения. Физически это будет соответствовать изменению оптических свойств среды под действием мощного излучения.

Примечание

Модель встречных световых пучков привела нас к системе уравнений (1), в которые входят производные только по одной переменной x. Если бы мы стали рассматривать более сложные эффекты рассеяния в стороны (а не только вперед и назад), то в уравнениях появились бы частные производные по другим пространственным переменным y и z. В этом случае получилась бы краевая задача для уравнений в частных производных, решение которой во много раз сложнее ОДУ.

12.1.2. Алгоритм стрельбы

Для решения краевых задач в Mathcad реализован наиболее популярный алгоритм, называемый методом *стрельбы* или *пристрелки* (shooting method). Он, по сути, сводит решение краевой задачи к решению серии задач Коши с различными начальными условиями. Рассмотрим здесь его основной принцип на примере модели (рис. 12.1), а встроенные функции, реализующие этот алгоритм, приведем в следующем разделе.

Суть метода стрельбы заключается в пробном задании недостающих граничных условий на левой границе интервала и решении затем полученной задачи Коши хорошо известными методами *(см. гл. 11)*. В нашем примере не хватает начального условия для $y_1(0)$, поэтому сначала зададим ему произвольное значение, например $y_1(0)=10$. Конечно, такой выбор не совсем случаен, поскольку из физических соображений ясно, что, во-первых, интенсивность излучения — величина заведомо положительная, и, во-вторых, отраженное излучение должно быть намного меньше падающего. Решение задачи Коши с помощью функции `rkfixed` приведено в листинге 12.1.

Листинг 12.1. Решение пробной задачи Коши для модели (12.1,1)

$r(x) := .1 \qquad a(x) := 1$

$R := 1$

$$D(x, y) := \begin{pmatrix} -a(x)\cdot y_0 + r(x)\cdot y_1 \\ a(x)\cdot y_1 - r(x)\cdot y_0 \end{pmatrix}$$

$I0 := 100$ $\quad y := \begin{pmatrix} I0 \\ 10 \end{pmatrix}$

$M := 10$

$S := \mathrm{rkfixed}(y, 0, 1, M, D)$

График полученных решений показан на рис. 12.2 (слева). Из него видно, что взятое наугад второе начальное условие не обеспечило выполнение граничного условия при x=1. И понятно, что для лучшего выполнения этого граничного условия следует взять большее значение $y_1(0)$. Возьмем, например, $y_1(0)=15$, и вновь решим задачу Коши. Результат показан на том же рис. 12.2 (в центре). Граничное условие выполняется с лучшей точностью, но опять-таки оказалось недостаточным. Для еще одного значения $y_1(0)=20$ получается решение, показанное на рис. 12.2 (справа). Из сравнения двух правых графиков легко заключить, что недостающее начальное условие больше 15, но меньше 20. Продолжая подобным образом "пристрелку" по недостающему начальному условию, возможно отыскать правильное решение краевой задачи.

В этом и состоит принцип алгоритма стрельбы. Выбирая пробные начальные условия (проводя пристрелку) и решая соответствующую серию задач Коши, можно найти то решение системы ОДУ, которое (с заданной точностью) удовлетворит фаничному условию (или, в общем случае, условиям) на другой фанице расчетного интервала.

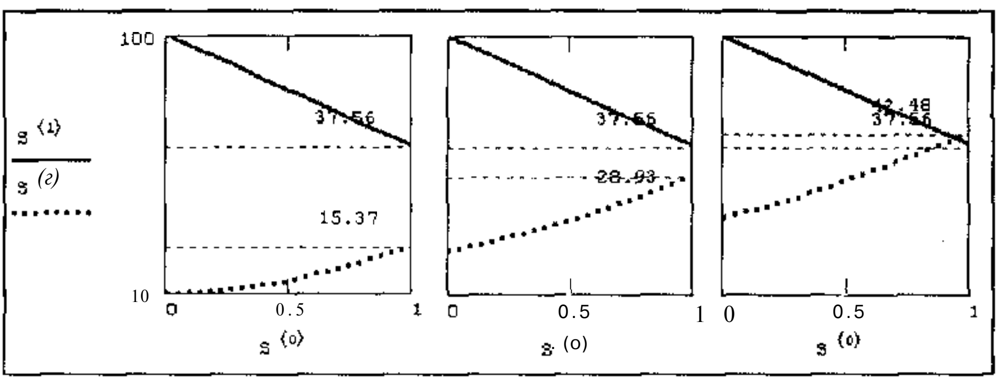


Рис. 12.2. Иллюстрация метода стрельбы (листинг **12.1**)

Конечно, описанный алгоритм несложно запрограммировать самому, оформив его как решение системы заданных алгоритмически уравнений, выражающих граничные условия на второй границе, относительно неизвестных пристрелочных начальных условий. Но делать этого нет необходимости, поскольку он оформлен в Mathcad в виде встроенных функций.

12.1.3. Решение двухточечных краевых задач

Решение краевых задач для систем ОДУ методом стрельбы в Mathcad достигается применением двух встроенных функций. Первая предназначена для двухточечных задач с краевыми условиями, заданными на концах интервала.

- sbval(z,x0,x1,D,load,score) — поиск вектора недостающих L начальных условий для двухточечной краевой задачи для системы N ОДУ;
  - z — вектор размера L×1, присваивающий недостающим начальным условиям (на левой границе интервала) начальные значения;
  - x0 — левая граница расчетного интервала;
  - x1 — правая граница расчетного интервала;
  - load(x0,z) — векторная функция размера N×1 левых граничных условий, причем недостающие начальные условия именуются соответствующими компонентами векторного аргумента z;
  - score(x1,y) — векторная функция размера L×1, выражающая L правых граничных условий для векторной функции y в точке x1;
  - D(x,y) — векторная функция, описывающая систему N ОДУ, размера N×1 и двух аргументов — скалярного x и векторного y. При этом y — это неизвестная векторная функция аргумента x того же размера N×1.

Внимание!

Решение краевых задач в Mathcad, в отличие от большинства остальных операций, реализовано не совсем очевидным образом. В частности, помните, что число элементов векторов D и load равно количеству уравнений N, а векторов z, score и результата действия функции sbval — количеству правых граничных условий L. Соответственно, левых граничных условий в задаче должно быть (N-L).

Как видно, функция sbval предназначена не для поиска собственно решения, т. е. неизвестных функций $y_i(x)$, а для определения недостающих начальных условий в первой точке интервала, т. е. $y_i(x0)$. Чтобы вычислить $y_i(x)$ на всем интервале, требуется дополнительно решить задачу Коши.

Разберем особенности использования функции sbval на конкретном примере (листинг 12.2), описанном выше *(см. разд. 12.1.1)*. Краевая задача состоит из системы двух уравнений (N=2), одного левого (L=1) и одного правого (N-L=2-1=1) граничного условия.

Листинг 12.2. Решение краевой задачи

```
r(x) := .1        a(x) := 1
R := 1            I0 := 100
```

$$D(x, y) := \begin{pmatrix} -a(x) \cdot y_0 + r(x) \cdot y_1 \\ a(x) \cdot y_1 - r(x) \cdot y_0 \end{pmatrix}$$

го := Ю

$$load(x0, z) := \begin{pmatrix} 100 \\ z_0 \end{pmatrix}$$

$score(x1, y) := R \cdot y_0 - y_1$

$I1 := sbval(z, 0, 1, D, load, score)$

$I1 = (18.555)$

$$S := rkfixed\left[\begin{pmatrix} I0 \\ I1_0 \end{pmatrix}, 0, 1, 10, D\right]$$

Первые три строки листинга задают необходимые параметры задачи и саму систему ОДУ. В четвертой строке определяется вектор z. Поскольку правое граничное условие всего одно, то недостающее начальное условие тоже одно, соответственно, и вектор z имеет только один элемент z_0. Ему необходимо присвоить начальное значение (мы приняли $z_0=10$, как в листинге 12.1), чтобы запустить алгоритм стрельбы *(см. разд 12.1.2).*

Примечание

Начальное значение фактически является параметром численного метода и поэтому может сильно повлиять на решение краевой задачи.

В следующей строке листинга векторной функции `load(x,z)` присваиваются левые граничные условия. Эта функция аналогична векторной переменной, определяющей начальные условия для встроенных функций, решающих задачи Коши. Отличие заключается в записи недостающих условий. Вместо конкретных чисел на соответствующих местах пишутся имена искомых элементов вектора z. В нашем случае вместо второго начального условия стоит аргумент z_0 функции load. Первый аргумент функции load — это точка, в которой ставится левое граничное условие. Ее конкретное значение определяется непосредственно в списке аргументов функции sbval.

Следующая строка листинга определяет правое граничное условие, для введения которого используется функция `score(x,y)`. Оно записывается точно так же, как система уравнений в функции D. Аргумент x функции score аналогичен функции load и нужен для тех случаев, когда граничное условие явно зависит от координаты x. Вектор score должен состоять из такого же числа элементов, что и вектор z.

Реализованный в функции sbval алгоритм стрельбы ищет недостающие начальные условия таким образом, чтобы решение полученной задачи Коши делало функцию `score(x,y)` как можно ближе к нулю. Как видно из листинга, результат применения sbval для интервала (0,1) присваивается векторной переменной и. Этот вектор похож на вектор z, только в нем содержатся искомые начальные условия вместо приближенных начальных значений, заданных в z. Вектор I1содержит, как и z, всего один элемент $I1_0$. С его помощью можно определить решение краевой задачи y(x) (последняя строка листинга). Тем самым, функция sbval сводит решение краевых задач к задачам Коши. График решения краевой задачи показан на рис. 12.3,

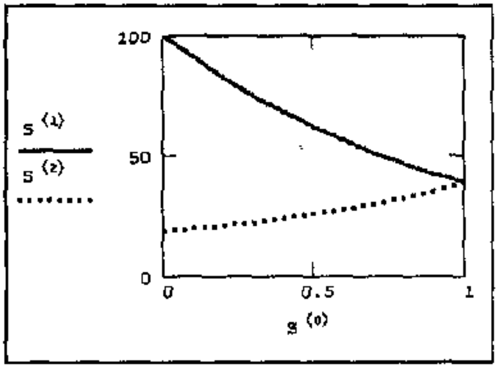


Рис. 12.3. Решение краевой задачи для R=1 (листинг 12.2)

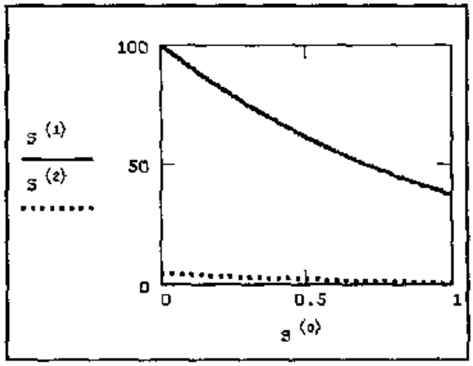


Рис. 12.4. Решение краевой задачи для R=0

На рис. 12.4 показано решение той же самой краевой задачи, но с другим правым граничным условием, соответствующим R=0, Т. е. без зеркала на правой границе. В этом случае слабый обратный пучок света образуется исключительно за счет обратного рассеяния излучения от лазера. Конечно, многие из читателей уже обратили внимание, что реальная физическая среда не может создавать такого большого рассеяния назад. Иными словами, более реальны значения r(x)<<a(x). Однако, когда коэффициенты в системе ОДУ при разных y_i очень сильно (на порядки) различаются, система ОДУ становится жесткой, и функция sbval не может найти решения, выдавая вместо него сообщение об ошибке("Could not find a solution").

Внимание!

Метод стрельбы не годится для решения жестких краевых задач. Поэтому алгоритмы решения жестких ОДУ в Mathcad приходится программировать самому *(см. разд. 12.3).*

12.1.4. Решение краевых задач с дополнительным условием в промежуточной точке

Иногда дифференциальные уравнения определяются с граничными условиями не только на концах интервала, но и с дополнительным условием в некоторой промежуточной точке расчетного интервала. Чаще всего такие задачи содержат данные о негладких в некоторой внутренней точке интервала решениях. Для них имеется встроенная функция bvaifit, также реализующая алгоритм стрельбы.

- `bvalfit(z1,z2,x0,x1,xf,D,load1,load2,score)` — поиск вектора недостающих граничных условий для краевой задачи с дополнительным условием в промежуточной точке для системы N ОДУ;
  - `z1` — вектор, присваивающий недостающим начальным условиям на левой границе интервала начальные значения;
  - `z2` — вектор того же размера, присваивающий недостающим начальным условиям на правой границе интервала начальные значения;
  - `x0` — левая граница расчетного интервала;
  - `x1` — правая граница расчетного интервала;
  - xf — точка внутри интервала;
  - `D(x,y)` — векторная функция, описывающая систему N ОДУ, размера N×1 и двух аргументов — скалярного x и векторного y. При этом y — это неизвестная векторная функция аргумента x того же размера N×1;
  - `load1(x0,z)` — векторная функция размера N×1 левых граничных условий, причем недостающие начальные условия поименовываются соответствующими компонентами векторного аргумента z;
  - `load2(x1,z)` — векторная функция размера N×1 правых граничных условий, причем недостающие начальные условия поименовываются соответствующими компонентами векторного аргумента z;
  - `score(xf,y)` — векторная функция размера N×1, выражающая внутреннее условие для векторной функции y в точке xf.

Обычно функция bvaifit применяется для задач, в которых производная решения имеет разрыв во внутренней точке xf. Некоторые из таких задач не удается решить обычным методом пристрелки, поэтому приходится вести пристрелку сразу из обеих граничных точек. В этом случае дополнительное внутреннее условие в точке xf является просто условием сшивки в ней левого и правого решений. Поэтому для данной постановки задачи оно записывается в форме `score(xf,y) :=y`.

Рассмотрим действие функции bvaifit на знакомом примере модели взаимодействия пучков света (см. рис. 12.1), предположив, что в промежутке между `xf=0.5` и `x1=1.0` находится другая, оптически более плотная среда с другим коэффициентом ослабления излучения a(x)=з. Соответствующая краевая задача решена в листинге 12.3, причем разрывный показатель ослабления определяется в его второй строке.

Листинг 12.3. Краевая задача с граничным условием в промежуточной точке

```
I0 := 100    r(x) := .1    R := 1

a(x) := | 1  if x < 0.5
        | 3  otherwise

D(x,y) := ( -a(x)·y0 + r(x)·y1 )
          (  a(x)·y1 - r(x)·y0 )

z10 := 20

load1(x0, z1) := ( 100 )
                 ( z10 )

z20 := 30

load2(x1, z2) := (  z20  )
                 ( R·z20 )

score(xf, y) := y

I1 := bvalfit(z1, z2, 0, 1, 0.5, D, load1, load2, score)^T

I1 = (  5.618 )
     ( 13.801 )
```

Система уравнений и левое краевое условие вводится так же, как и в предыдущем листинге для функции sbval. Обратите внимание, что таким же образом записано и правое краевое условие. Для того чтобы ввести условие отражения на правой границе, пришлось определить еще один неизвестный пристрелочный параметр $z2_0$. Строка листинга, в которой определена функция score, задает условие стрельбы — сшивку двух решений в точке xf. В самой последней строке листинга выдан ответ — определенные численным методом значения обоих пристрелочных параметров, которые объединены в вектор I1(мы применили в предпоследней строке операцию транспонирования, чтобы результат получился в форме вектора, а не матрицы-строки).

Для корректного построения фафика решения лучше составить его из двух частей — решения задачи Коши на интервале (x0,xf) и другой задачи Коши для интервала (xf,x1). Реализация этого способа приведена в листинге 12.4, который является продолжением листинга 12.3. В последней строке листинга 12.4 выведено значение второй искомой функции на правой границе интервала. Всегда полезно проконтролировать, что оно совпадает с соответствующим пристрелочным параметром (выведенном в последней строке листинга 12.3).

Листинг 12.4. Решение краевой задачи *(продолжение листинга 12.3)*

$$M := 10$$

$$S0 := \mathrm{rkfixed}\left[\begin{pmatrix} I0 \\ I1_0 \end{pmatrix}, 0, 0.5, M, D\right]$$

$$S1 := \mathrm{rkfixed}\left[\begin{pmatrix} S0_{M,1} \\ S0_{M,2} \end{pmatrix}, 0.5, 1, M, D\right]$$

$$S1_{M,2} = 13.801$$

Решение краевой задачи приведено на рис. 12.5. С физической точки зрения естественно, что интенсивность света уменьшается быстрее по мере распространения в более плотной среде в правой половине расчетного интервала. В средней точке `xf=0.5`, как и ожидалось, производные обоих решений имеют разрыв.

Примечание

Еще один пример решения краевых задач с разрывными коэффициентами ОДУ приведен в справочной системе Mathcad.

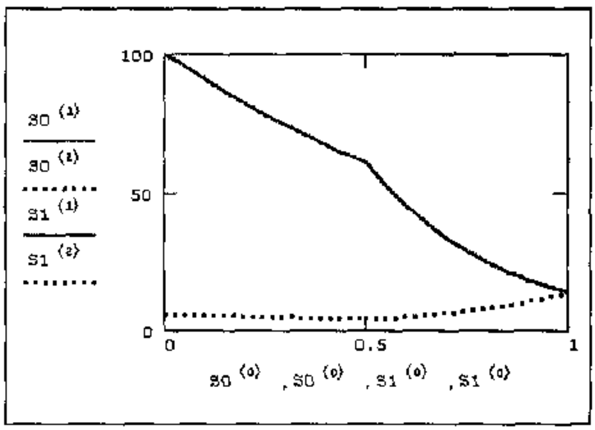


Рис. 12.5. Решение краевой задачи с разрывом в xf=0.5 (листинги 12.3—12.4)

Ради справедливости необходимо заметить, что разобранную краевую задачу легко решить и с помощью функции `sbval`, заменив в листинге 12.2 зависимость a(x) на третью строку листинга 12.3. В этом случае листинг 12.2 даст в точности тот же ответ, что показам на рис. 12.5. Однако в определенных случаях (в том числе из соображений быстроты расчетов) удобнее использовать функцию `bvalfit`, т. е. вести пристрелку с обеих границ интервата.

Совет

Если Вы имеете дело с подобными уравнениями, попробуйте сначала решить их как обычную краевую задачу с помощью более надежной и легкой в применении функции `sbval`.

12.2. Задачи на собственные значения для ОДУ

Задачи на собственные значения — это краевые задачи для системы ОДУ, в которой правые части зависят от одного или нескольких параметров X. Значения этих параметров неизвестны, а решение краевой задачи существует только при определенных λ_k, которые называются *собственными значениями* (eigenvalues) задачи. Решения, соответствующие этим λ_k, называют *собственными функциями* (eigenfunctions) задачи. Правильная постановка таких задач требует формулировки количества граничных условий, равного сумме числа уравнений и числа собственных значений. Физическими примерами задач на собственные значения являются, например, уравнение колебаний струны, уравнение Шредингера в квантовой механике, уравнения волн в резонаторах и многие другие.

С вычислительной точки зрения, задачи на собственные значения очень похожи на рассмотренные выше краевые задачи. В частности, для многих из них также применим метод стрельбы *(см. разд. 12.1.2)*. Отличие заключается в пристрелке не только по недостающим левым граничным условиям, но еще и по искомым собственным значениям. В Mathcad для решения задач на собственные значения используются те же функции sbval и `bvalfit`. В их первый аргумент, т. е. вектор, присваивающий начальные значения недостающим начальным условиям, следует включить и начальное приближение для собственного значения.

Рассмотрим методику решения на конкретном примере определения собственных упругих колебаний струны. Профиль колебаний струны y(x) описывается линейным дифференциальным уравнением второго порядка

$$\frac{d}{dx}\left[p(x)\frac{\qquad}{dx}\right] + \lambda \cdot q(x) \cdot y(x) = 0,$$

где `p(x)` и q(x) — жесткость и плотность, которые, вообще говоря, могут меняться вдоль струны. Если струна закреплена на обоих концах, то граничные условия задаются в виде `y(0)=y(1)=0`. Сформулированная задача является частным случаем *задачи Штурма-Лиувилля.* Поскольку решается система двух ОДУ, содержащая одно собственное значение X, то по идее задача требует задания трех (2+1) условий. Однако, как легко убедиться, уравнение колебаний струны — линейное и однородное, поэтому в любом случае решение y(x) будет определено с точностью до множителя. Это означает, что производную решения можно задать произвольно, например y (0)=1, что и будет третьим условием. Тогда краевую задачу можно решать как задачу Коши, а пристрелку вести только по одному параметру — собственному значению.

Процедура поиска первого собственного значения представлена в листинге 12.5.

Листинг 12.5. Решение задачи о собственных колебаниях струны

$p(x) := 1 \qquad q(x) := 1 \qquad p'(x) := 0$

$a := 0 \qquad b := 1$

$\lambda_0 := 0.5$

$$D(x, y) := \begin{bmatrix} y_1 \\ -\frac{1}{p(x)}\cdot(p'(x)\cdot y_1 + y_2\cdot q(x)\cdot y_0) \\ 0 \end{bmatrix}$$

$$\mathrm{load}(a, \lambda) := \begin{pmatrix} 0 \\ 1 \\ \lambda_0 \end{pmatrix}$$

$\mathrm{score}(b, y) := y_0$

$\lambda := \mathrm{sbval}(\lambda, a, b, D, \mathrm{load}, \mathrm{score})$

$\lambda = (\ 9.87\ ) \qquad 1^2\cdot\pi^2 = 9.87$

В первых двух строках листинга определяются функции, входящие в задачу, в том числе p'(x):=0, и границы расчетного интервала (0,1). В третьей строке дается начальное приближение к собственному значению λ_0, в четвертой вводится система ОДУ. Обратите внимание, что она состоит не из двух, а из трех уравнений. Первые два из них определяют эквивалентную (I) систему ОДУ первого порядка, а третье необходимо для задания собственного значения в виде еще одного компонента y_2 искомого вектора y. Поскольку, по определению, собственное значение постоянно при всех x, то его производная должна быть приравнена нулю, что отражено в последнем уравнении. Важно также, что во втором из уравнений собственное значение записано как y_2, поскольку является одним из неизвестных.

В следующих двух строках листинга задается левое граничное условие, включающее и недостающее условие на собственное значение для третьего уравнения, и правое граничное условие $y_0=0$. В предпоследней строке листинга обычным образом применяется функция sbval, а в последней выводится результат ее работы вместе с известным аналитически собственным значением $n^2\cdot\pi^2$. Как легко убедиться, мы нашли первое собственное значение для n=1, а чтобы найти другие собственные значения, необходимо задать другие начальные приближения к ним (в третьей строке листинга 12.5). Например, выбор λ_0=50 приводит ко второму собственному значению $2^2 \cdot \pi^2$, а λ_0=100 — к третьему $3^2 \cdot \pi^2$.

Чтобы построить график соответствующей собственной функции, надо добавить в листинг строку, программирующую решение задачи Коши, например, такую: U:=rkfixed(load(a,Λ),a,b,100,D). Полученные кривые показаны на рис. 12.6 в виде коллажа трех графиков, рассчитанных для трех собственных значений.

Примечание

Примеры решения нескольких задач на собственные значения можно найти в разделе Mathcad Resources.

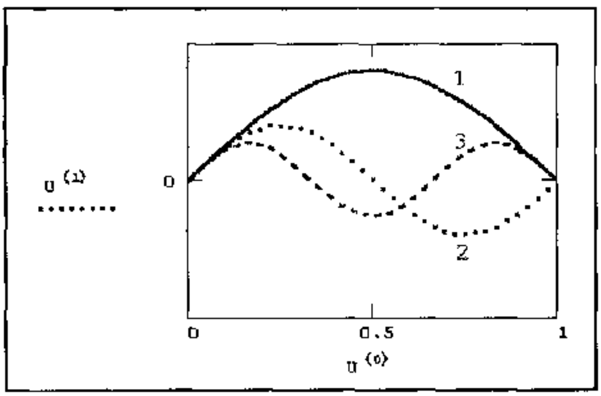


Рис. 12.6. Первые три собственные функции задачи колебаний струны (коллаж трех графиков)

12.3. Разностные схемы для ОДУ

Многие краевые задачи не поддаются решению методом стрельбы. Однако в Mathcad 11 других встроенных алгоритмов нет. Тем не менее, это не означает, что по-другому решать краевые задачи невозможно, ведь другие численные алгоритмы несложно запрограммировать самому пользователю. Рассмотрим возможную реализацию наглядного метода, называемого разностным, которым можно решать краевые задачи как для ОДУ, так и для дифференциальных уравнений в частных производных.

12.3.1. О разностном методе решения ОДУ

Разберем идею разностного метода решения краевых задач на примере взаимодействия световых пучков (см. рис. 12.1), переобозначив в системе (12.1,1) интенсивность излучения вправо на y, а интенсивность излучения

11 Зак 984

влево на у (просто в целях удобства, чтобы не писать индекс). Суть метода заключается в покрытии расчетного интервала сеткой из N точек. Тем самым определяются (N-1) шагов (рис. 12.7). Затем надо заменить дифференциальные уравнения исходной краевой задачи аппроксимирующими их уравнениями в конечных разностях, выписав соответствующие разностные уравнения для каждого i-го шага. В нашем случае достаточно просто заменить первые производные из (12.1,1) их разностными аналогами (такой метод называется еще *методом Эйлера):*

$$\frac{Y_{i+1}-Y_i}{Д} = -a(x_i)\cdot Y_i + r(x_i)\cdot y_i$$
$$\frac{y_{i+1}-y_i}{\Delta} = a(x_i)\cdot y_i - r(x_i)\cdot y_i \qquad (1)$$

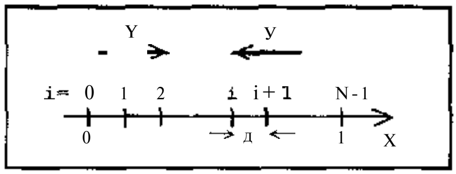


Рис. 12.7. Сетка, покрывающая расчетный интервал

Примечание

Существует множество способов аппроксимации дифференциальных уравнений разностными. От выбора конкретного варианта зависит не только простота, быстрота и удобство вычислений, но и сама возможность получения правильного ответа.

Получилась система (по числу шагов) $2\cdot(N-1)$ разностных линейных алгебраических уравнений с $2\cdot N$ неизвестными Y_i и y_i. Для того чтобы она имела единственное решение, надо дополнить число уравнений до $2\cdot N$. ЭТО МОЖНО сделать, записав в разностном виде оба граничных условия:

$$Y_0=10, \quad y_N=R\cdot Y_N. \qquad (2)$$

Сформированная полная система алгебраических уравнений называется *разностной схемой,* аппроксимирующей исходную краевую задачу. Обратите внимание, что правые части разностных уравнений системы (1) на каждом шаге записаны для левой границы шага. Такие разностные схемы называют *явными,* т. к. все значения Y_{i+1} и y_{i+1} находятся в левой части уравнений. Полученную явную разностную схему легко записать в матричной форме

$$A\cdot z=B, \qquad (3)$$

где z — неизвестный вектор, получающийся объединением векторов Y и y. Решив систему (3), мы получим решение краевой задачи.

Примечание

На самом деле, все несколько сложнее, поскольку, вообще говоря, необходимо еще доказать, что, во-первых, разностная схема действительно аппроксимирует дифференциальные уравнения и, во-вторых, при $N \to \infty$ разностное решение действительно сходится к дифференциальному.

Процесс решения системы разностных уравнений называют также *реализацией* разностной схемы. Программа, которая решает рассматриваемую краевую задачу разностным методом, приведена в листинге 12.6.

Листинг 12.6. Реализация явной разностной схемы

$a(x) := 1 \qquad r(x) := 0.01 \qquad I0 := 100 \qquad R := 1$

$N := 5$

$N - 1$

$i := 1 .. N - 1$

$A_{i,i} := a(i \cdot \Delta - \Delta) \cdot \Delta - 1 \qquad A_{i,i+1} := 1 \qquad A_{i,i+N} := -\Delta \cdot r(i \cdot Д - \Delta)$

$i := N + 1 .. 2 \cdot N - 1$

$A_{i,i} := -a(i \cdot \Delta - \Delta) \cdot \Delta - 1 \qquad A_{i,i+1} := 1 \qquad A_{i,i-N} := Д \cdot r(i \cdot Д - \Delta)$

$A_{0,1} := 1$

$A_{N,N-2} := 1 \qquad A_{N,N} := -R$

$i := 1 .. 2 - N - 1$

$B_i := 0$

$B_0 := I0$

$A := \mathrm{submatrix}(A, 0, 2 \cdot N - 1, 1, 2 \cdot N)$

$U := \mathrm{lsolve}(A, B)$

$U_N = 13.453 \qquad U_{N-1} = 31.78$

Дадим минимальные комментарии, надеясь, что заинтересовавшийся читатель с карандашом в руках разберется в порядке индексов и соответствии матричных элементов, а возможно, составит и более удачную программу.

В первой строке листинга определяются функции и константы, входящие в модель, во второй задается число точек сетки N=5 и ее равномерный шаг. Следующие две строки определяют матричные коэффициенты, аппроксимирующие уравнения для Y_i, а пятая и шестая — для y_i. Седьмая и восьмая строки листинга задают, соответственно, левое и правое граничное условие, а строки с девятой по одиннадцатую — правые части системы (3). В следующей строке завершается построение матрицы A вырезанием из нее левого нулевого столбца. В предпоследней строке листинга применена встроенная функция lsolve для решения системы (3), а в последней выведены рассчитанные ею неизвестные граничные значения. Графики решения приведены на рис. 12.8, причем первые N элементов итогового вектора есть вычисленное излучение вперед, а последние N элементов — излучение назад.

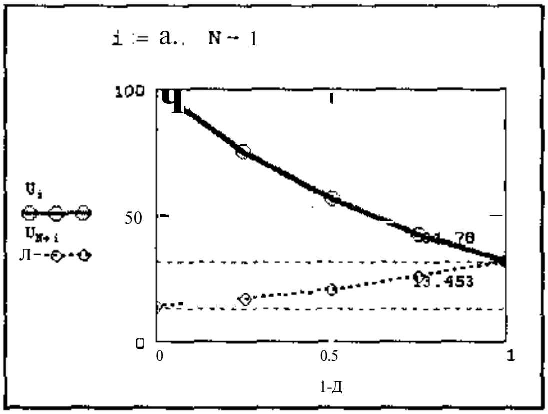


Рис. 12.8. Решение краевой задачи разностным методом (листинг 12.6)

Как мы увидели, реализация в Mathcad разностных схем вполне возможна и не слишком трудоемка — предложенная программа состоит всего из двух десятков математических выражений. Конечно, для их написания требуется и время, и часто кропотливые расчеты, но, собственно, в этом и состоит работа математика. Кстати говоря, при небольшом числе шагов, расчеты по разностным схемам не требуют существенного времени (программа, приведенная в листинге 12.6, работает быстрее, чем метод стрельбы, встроенный в функцию `sbval`). Существуют, кроме того, весьма очевидные для многих читателей пути ускорения расчетов, связанные с применением более подходящих методов решения систем линейных уравнений с разреженной матрицей.

12.3.2. Жесткие краевые задачи

Один из случаев, когда применение разностных схем может быть очень полезным, связан с решением жестких краевых задач *(подробнее о жестких ОДУ читайте в гл. 11)*. В частности, рассматриваемая задача о встречных световых пучках становится жесткой при увеличении коэффициента ослабления a(x) в несколько десятков раз. Например, при попытке решить ее с a(x) :=100 с помощью листинга 12.2 вместо ответа выдается сообщение об ошибке "Can't converge to a solution. Encountered too many integration steps" (Не сходится к решению. Слишком много шагов интегрирования). Это и неудивительно, поскольку жесткие системы характерны тем, что требуют исключительно малого значения шага в стандартных алгоритмах.

Для жестких задач неприменимы и явные разностные схемы, о которых рассказывалось в предыдущем разделе *(см. разд. 12.3.1)*. Результат расчетов по программе листинга 12.6, например с `a(x):=20` (рис. 12.9), дает характерную для неустойчивых разностных схем "разболтку" — колебания нарастающей амплитуды, не имеющие ничего общего с реальным решением.

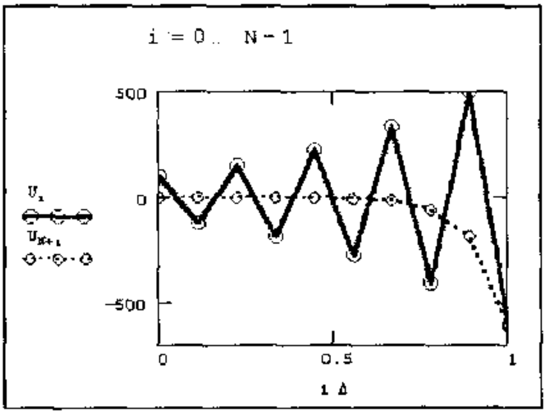


Рис. 12.9. Неверное решение жесткой краевой задачи по неустойчивой явной разностной схеме

Выходом из положения будет использование *неявных* разностных схем. Применительно к нашей задаче достаточно заменить правые части уравнений (1) значениями не на левой, а на правой границе каждого шага:

$$\frac{Y_{i+1}-Y_i}{\Delta} = -a(x_{i+1})\cdot Y_{i+1} + r(x_{i+1})\cdot y_{i+1}$$
$$\frac{y_{i+1}-y_i}{\Delta} = a(x_{i+1})\cdot y_{i+1} - r(x_{i+1})\cdot Y_{i-1} \qquad (4)$$

Граничные условия, конечно, можно оставить в том же виде (2). Поскольку мы имеем дело с линейными дифференциальными уравнениями, то и схему (4) легко будет записать в виде матричного равенства (3), перегруппировывая соответствующим образом выражение (4) и приводя подобные слагаемые. Разумеется, полученная матрица А будет иной, нежели матрица А ДЛЯ явной схемы (1). Поэтому и решение (реализация неявной схемы) может отличаться от изображенного на рис. 12.9 результата расчетов по явной схеме. Программа, составленная для решения системы (4), приведена в листинге 12.7.

Листинг 12.7. Реализация неявной разностной схемы для жесткой краевой задачи

$a(x) := 20 \qquad r(x) := 0.01 \qquad I0 := 100 \qquad R := 1$

$N := 20 \qquad Д := \dfrac{1}{N-1}$

$i := 1 .. N-1$

$A_{i,\,i} := 0 - 1$ $\quad A_{i,\,i+N+1} := -Д \cdot r(i-Д - д)$ $\quad A_{i,\,i+1} := a(i \cdot \Delta - \Delta) \cdot \Delta + 1$

$i := N + 1 \,..\, 2 \cdot N - 1$

$A_{i,\,i} := 0 - 1$ $\quad A_{i,\,i+1} := 1 - a(i \cdot \Delta - \Delta) \cdot \Delta$ $\quad A_{i,\,i-N+1} := Д \cdot r(i \cdot \Delta - \Delta)$

$A_{0,\,1} := 1$

$A_{N,\,N \cdot 2} := 1$ $\quad A_{N,\,N} := -R$

$i := 1 \,..\, 2 \cdot N - 1$

$B_i := 0$

$B_0 := 10$

$A := \mathrm{submatrix}(A,\, 0,\, 2 \cdot N - 1,\, 1,\, 2 \cdot N)$

$U := \mathrm{lsolve}(A,\, B)$

$U_N = 0.017$ $\quad U_{N-1} = 1.523 \times 10^{-6}$

Не будем специально останавливаться на обсуждении листинга 12.7, поскольку он почти в точности повторяет предыдущий листинг. Отличие заключается лишь в формировании матрицы А другим способом, согласно неявной схеме. Решение, показанное на рис. 12.10, демонстрирует, что произошло "небольшое чудо": "разболтка" исчезла, а распределение интенсивностей стало физически предсказуемым. Обратите внимание, что (из-за взятого нами слишком большого коэффициента ослабления излучения) отраженный пучок света имеет очень маленькую интенсивность, и ее пришлось построить на графике с увеличением в тысячу раз.

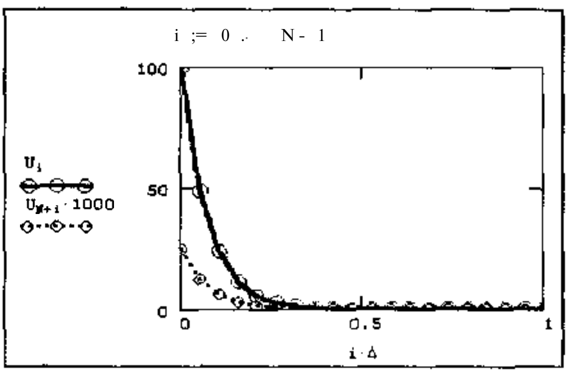


Рис. 12.10. Решение краевой задачи разностным методом (листинг 12.7)

ГЛАВА 13

Дифференциальные уравнения в частных производных

Дифференциальные уравнения в частных производных представляют собой одну из наиболее сложных и одновременно интересных задач вычислительной математики. Эти уравнения характеризуются тем, что для их решения не существует единого универсального алгоритма и большинство задач требует своего собственного особого подхода. Уравнениями в частных производных описывается множество разнообразных физических явлений, и с их помощью можно с успехом моделировать самые сложные явления и процессы (диффузия, гидродинамика, квантовая механика, экология и т. д.).

Дифференциальные уравнения в частных производных требуют нахождения функции не одной, как для ОДУ, а нескольких переменных, например f(x,y) или f(x,t). Постановка задач *(см. разд. 13.1)* включает в себя само уравнение (или систему уравнений), содержащее производные неизвестной функции по различным переменным (частные производные), а также определенное количество краевых условий на границах расчетной области.

Несмотря на то, что Mathcad обладает довольно ограниченными возможностями по отношению к уравнениям в частных производных, в нем имеется несколько встроенных функций, количество и возможности которых увеличились в новой 11-й версии *(см. разд. 13.3)*. Решать уравнения в частных производных можно и путем непосредственного программирования пользовательских алгоритмов *(см. разд. 13.2)*. Автор совершенно сознательно сначала рассматривает численные методы для решения уравнений в частных производных, а уже затем описывает предназначенные для этого встроенные функции, чтобы читатель ясно осознавал, каким образом Mathcad 11 производит расчеты. «Слепое» использование встроенных функций для решения уравнений в частных производных не всегда бывает успешным, и ответственность за верный выбор их параметров часто ложится на пользователя, которому необходимо четко представлять основные принципы функционирования численных алгоритмов, примененных во встроенных функциях.

Рассмотрение уравнений в частных производных построено на анализе различных вариаций уравнения диффузии тепла, которое допускает наглядную физическую интерпретацию.

13.1. Постановка задач

13.1.1. Классификация уравнений в частных производных

Постановка задач для уравнений в частных производных включает определение самого уравнения (или системы нескольких уравнений), а также необходимого количества краевых условий (число и характер задания которых определяется спецификой уравнения). По своему названию уравнения должны содержать частные производные неизвестной функции u (или нескольких функций, если уравнений несколько) по различным аргументам, например пространственной переменной x и времени t. Соответственно, для решения задачи требуется вычислить функцию нескольких переменных, например u(x,t) в некоторой области определения аргументов $0 \le x \le L$ и $0 \le t \le T$. Граничные условия определяются как заданные временные зависимости функции u, или производных этой функции на границах расчетной области 0 и L, а начальные — как заданная u(x, 0).

Сами уравнения в частных производных (несколько условно) можно разделить на три основных типа:

- *параболические* — содержащие первую производную по одной переменной и вторую — по другой, причем все эти производные входят в уравнение с одинаковым знаком;
- *гиперболические* — содержащие первую производную по одной переменной и вторую — по другой, входящие в уравнение с разными знаками;
- *эллиптические* — содержащие только вторые производные, причем одного знака.

Некоторые более сложные уравнения нельзя однозначно подогнать под приведенную классификацию, тогда говорят о гибридных типах уравнений.

13.1.2. Пример: уравнение диффузии тепла

На протяжении всей главы мы будем использовать в качестве примера очень наглядное и имеющее различные, от очевидных до самых неожиданных, решения уравнение теплопроводности.

Двумерное динамическое уравнение

Рассмотрим следующее параболическое уравнение в частных производных, зависящее от трех переменных — двух пространственных x и y, а также от времени t:

$$\frac{\partial u(x,y,t)}{\partial t} = D\left(\frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2}\right) + \phi(x,y,t,u) \qquad (1)$$

Примечание

Выражение в скобках в правой части уравнения (сумму вторых пространственных производных функции и часто, ради краткости, обозначают при помощи *оператора Лапласа:* Δu).

Это уравнение называется *двумерным уравнением теплопроводности* или, по-другому, *уравнением диффузии тепла.* Оно описывает динамику распределения температуры u(x,y,t) на плоской поверхности (например, на металлической пластине) в зависимости от времени (рис. 13.1). Физический смысл коэффициента D, который, вообще говоря, может быть функцией как координат, так и самой температуры, заключается в задания скорости перетекания тепла от более нагретых областей в менее нагретые. Функция ϕ(x,y,t,u) описывает приток тепла извне, т.е. источники тепла, которые также могут зависеть как и от пространственных координат (что задает локализацию источников), так и от времени и температуры и.

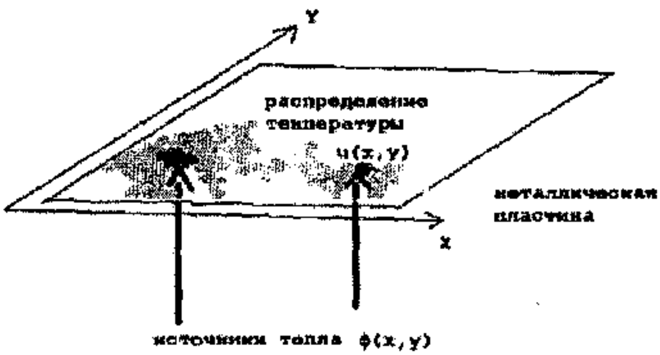


Рис. 13.1. Физическая модель, описываемая двумерным уравнением теплопроводности

Для того чтобы правильно поставить краевую задачу для двумерного уравнения теплопроводности, следует определить следующие дополнительные условия:

- граничные условия, т. е. динамику функции u(x,y,t) и / или ее производных на границах расчетной области;
- начальное условие, т. е. функцию u(x,y,t).

Примечание

Если рассматривается не одно уравнение в частных производных, а система уравнений, то соответствующие начальные и граничные условия должны быть поставлены для каждой из неизвестных функций.

Стационарное двумерное уравнение

Частный случай уравнения теплопроводности определяет стационарную, т. е. не зависящую от времени задачу. Стационарное уравнение описывает физическую картину распределения температуры по пластине, не изменяющуюся с течением времени. Такая картина может возникнуть при условии, что стационарный источник тепла действует довольно продолжительное время, и переходные процессы, вызванные его включением, прекратились. Пример численного решения такого уравнения показан на рис. 13.2 в виде поверхности u(x,y).

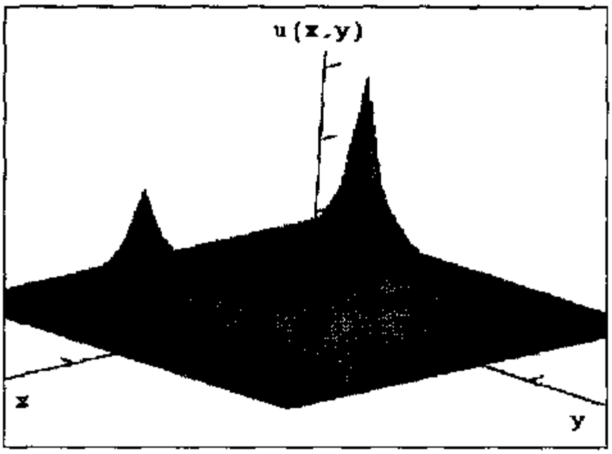


Рис. 13.2. Решение стационарного двумерного уравнения теплопроводности (см. листинг 13.7 ниже)

Как несложно сообразить, если искомая функция не зависит от времени, то частная производная по времени в левой части уравнения равна нулю, и само уравнение можно переписать (переобозначив ради упрощения $\phi \leftarrow \phi/D$) следующим образом:

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = -\phi(x,y,u). \qquad (2)$$

Полученное уравнение, согласно классификации предыдущего раздела, является эллиптическим. Его называют *уравнением Пуассона,* а для его решения в Mathcad предусмотрены две встроенные функции. Если, к тому же, источники равны нулю, то уравнение (2), принимающее вид $\Delta u=0$, называют *уравнением Лапласа.*

Одномерное динамическое уравнение

Предположим, что мы рассматриваем задачу распределения тепла не по плоской поверхности, а по удлиненному телу типа металлического стержня (рис. 13.3). В этом случае зависимость от координаты у в общем уравнении теплопроводности пропадает, и получается одномерное уравнение:

$$\frac{\partial u(x,t)}{\partial t} = D\frac{\partial^2 u(x,t)}{\partial x^2} + \phi(x,t,u) \qquad (3)$$

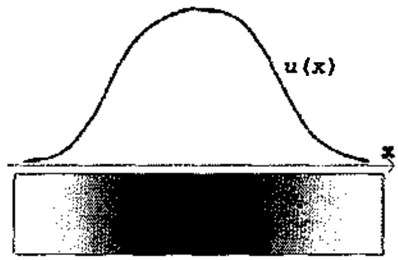


Рис. 13.3. *Физическая* модель одномерного уравнения теплопроводности

Одномерное уравнение намного проще двумерного, поскольку объем вычислений для реализации алгоритма его численного решения не так велик. Типичное решение одномерного уравнения диффузии тепла с коэффициентом диффузии D=2, нулевым источником ф=о и начальным распределением температуры в форме нагретой центральной области стержня показано (в виде графика поверхности) на рис. 13.4.

Начиная с новой версии Mathcad 11, для решения одномерных параболических и гиперболических уравнений можно применять новую встроенную ФУНКЦИЮ `pdesolve`.

Линейное и нелинейное уравнения

Если присмотреться к уравнению диффузии тепла внимательнее, то можно условно разделить практические случаи его использования на два типа.

- Линейная задача — если коэффициент диффузии D не зависит от температуры и и, кроме того, если источник тепла ф либо также не зависит от и, либо зависит от `u` линейно. В этом случае неизвестная функция `u(x,t)` и все ее производные входят в уравнение только в первой степени (линейно).
- Нелинейная задача — если уравнение имеет нелинейную зависимость от u(x,t), т. е. или коэффициент диффузии зависит от и, и / или источник тепла нелинейно зависит от и.

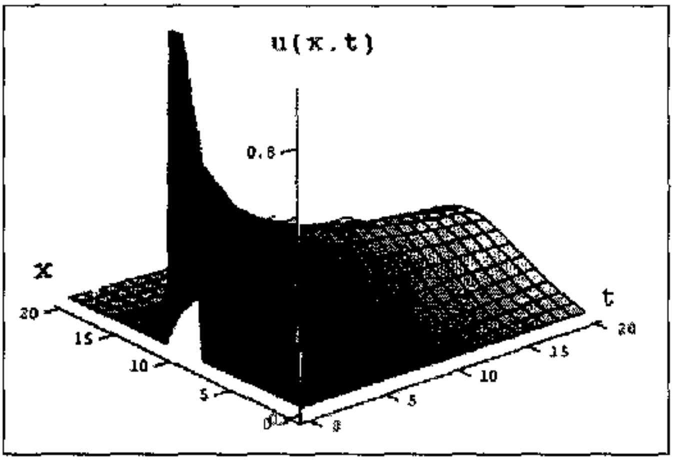


Рис. 13.4. Решение одномерного уравнения теплопроводности (см. листинг 13.1 ниже)

Решение линейных уравнений в частных производных, как правило, получаются вполне предсказуемыми и их часто можно получить аналитически (этим проблемам посвящены соответствующие главы науки, называемой математической физикой). В случае уравнения теплопроводности линейная задача описывает физически ожидаемое решение, выражающее остывание пластины или стержня в форме перетекания тепла от нагретого центра к холодной периферии.

Нелинейные уравнения, напротив, могут демонстрировать самые неожиданные решения, причем в подавляющем большинстве практических задач их можно получить только численно, а никак не аналитически.

Примечание

Различные линейные и нелинейные варианты рассматриваемого уравнения теплопроводности описывают различные модели физических сред, которые характеризуются определенными зависимостями D(U) и $\phi(u)$. В частности, для металлов в большинстве случаев можно считать, что D=const, в то время как для плазмы имеется специфическая зависимость коэффициента диффузии от температуры.

Обратное уравнение теплопроводности

Замечательными свойствами обладает так называемое обратное уравнение диффузии тепла, которое получается путем замены в исходном (прямом) уравнении переменной t на -t. Согласно постановке задачи, обратное уравнение теплопроводности описывает реконструкцию динамики профиля температуры остывающего стержня, если известно начальное условие в виде профиля температуры в некоторый момент времени после начала остывания. Таким образом, требуется определить, как происходило остывание стержня. Мы ограничимся самым простым линейным уравнением с D=const без источников тепла:

$$\frac{\partial u(x,t)}{\partial t} = -D\frac{\partial^2 u(x,t)}{\partial x^2}. \qquad (4)$$

Это уравнение гиперболического типа и оно, несмотря на кажущуюся близость к рассмотренным вариантам уравнения теплопроводности, обладает весьма замечательными свойствами.

Если попробовать осуществить расчет обратного уравнения диффузии тепла по тем же самым алгоритмам, что и для обычных уравнений (для этого достаточно в листинге 13.1 или 13.2 заменить значение коэффициента диффузии на отрицательное число, например `D=-1`), то мы получим заведомо нефизичное решение. Оно показано на рис. 13.5 в виде профилей распределения температуры для нескольких последовательных моментов времени. Как видно, решение выражается в появлении все более быстрых пространственных осцилляций профиля температуры для каждого нового момента времени. Очень существенно, что такое поведение решение является не проявлением неустойчивости численного алгоритма (*см. разд. «Устойчивость» этой главы),* а определяется спецификой самой задачи.

Оказывается, что обратное уравнение теплопроводности принадлежит к довольно широкому классу задач, называемых *некорректными.* Некорректные задачи нельзя решать стандартными методами, а для того чтобы с ними справиться (т. е., чтобы получить осмысленное физическое решение) приходится несколько менять саму их постановку, вводя в нее дополнительную априорную информацию о строении решения.

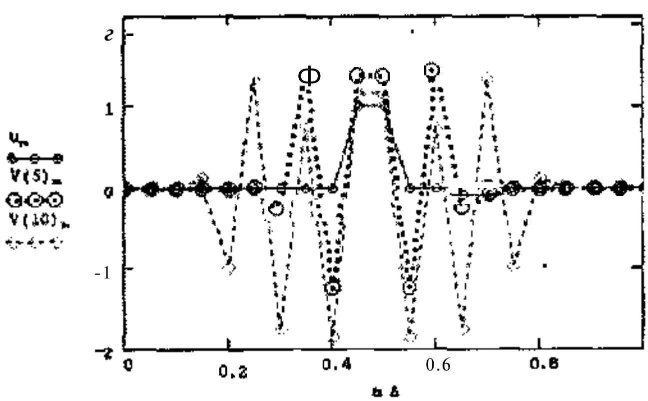


Рис. 13.5. Численное решение обратного уравнения теплопроводности дает совершенно нефизичную картину динамики температуры (см. листинги 13.1, 2 ниже с параметром D=-1)

13.2. Разностные схемы

Рассмотрим одномерное уравнение теплопроводности (3) и на его примере разберем наиболее часто использующийся для численного решения уравнений в частных производных *метод сеток.* Выпишем еще раз само уравнение

$$\frac{\partial u(x,t)}{\partial t} = D\frac{\partial^2 u(x,t)}{\partial x^2} + \phi(x,t,U)\,, \tag{5}$$

а также и начальное

$$u(x,0) = u_0(x) \tag{6}$$

и граничные условия

$$u(0,t) = f_0(t)\,, \qquad u(L,t) = f_1(t)\,, \tag{7}$$

которые необходимы для правильной с математической точки зрения постановки задачи.

Основная идея численного решения уравнений в частных производных очень похожа на метод решения краевых задач для ОДУ, рассмотренный нами в предыдущей главе. Основным отличием от ОДУ является необходимость дискретизации уравнения не по одной, а по нескольким переменным (в зависимости от размерности задачи).

Таким образом, сначала следует покрыть расчетную область <x,t) *сеткой* и использовать затем узлы этой сетки для разностной аппроксимации уравнения. В результате, вместо поиска непрерывных зависимостей u{x,t) достаточно будет отыскать значения функции в узлах сетки (а ее поведение в промежутках между узлами может быть получено при помощи построения какой-либо интерполяции). По этой причине дискретное представление функции и часто называют *сеточной функцией.*

Поскольку уравнения в частных производных по определению зависят от производных неизвестных функций по нескольким переменным, то способов дискретизации этих уравнений , может быть, как правило, несколько. Конфигурацию узлов, используемую для разностной записи уравнений в частных производных на сетке, называют *шаблоном,* а полученную систему разностных уравнений - *разностной схемой.* О принципах построения разностных схем, и, в частности, о классах явных и неявных схем, мы уже подробно говорили на примере краевых задач для ОДУ *(см. разд. 12.3.1),* поэтому, излишне не повторяясь, перейдем к рассмотрению типичных особенностей уравнений в частных производных, которые возникают при разработке и реализации разностных схем.

13.2.1. Явная схема Эйлера

Рассмотрим сначала математические аспекты построения разностной схемы для уравнения диффузии тепла, а затем приведем примеры работы разработанного алгоритма применительно к линейному и нелинейному уравнениям.

Построение разностной схемы

Используем для решения уравнения теплопроводности шаблон, изображенный на рис. 13.6. Для дискретизации второй производной по пространственной координате необходимо использовать три последовательных узла, в то время как для разностной записи первой производной по времени достаточно двух узлов. Записывая на основании данного шаблона дискретное представление для (i,k)-го узла, получим разностное уравнение:

$$\frac{u_{i,k+1} - u_{i,k}}{\Delta_t} = D\frac{u_{i-1,k} - 2u_{i,k} + u_{i+1,k}}{\Delta_x^2} + \phi_{i,k}. \qquad (8)$$

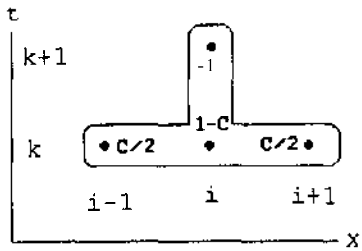


Рис. 13.6. Шаблон аппроксимации явной схемы для уравнения теплопроводности

Приведем в разностной схеме (8) подобные слагаемые, перенеся в правую часть значения сеточной функции с индексом k (как часто говорят, с предыдущего *слоя* по времени), а в левую — с индексом k+1 (т. е. со следующего временного слоя). Кроме этого, введем коэффициент c , который будет характеризовать отношение шагов разностной схемы по времени и пространству $c_{i,k} = 2\frac{\Delta_t}{\Delta_x^2}D_{i,k}$. Несколько забегая вперед, заметим, что значение параметра c, называемого *коэффициентом Куранта,* имеет большое значение для анализа устойчивости разностной схемы. С учетом этих замечаний, разностная схема (8) запишется в виде:

$$u_{i,k+1} = \frac{c_{i,k}}{2}u_{i-1,k} + (1 - c_{i,k})u_{i,k} + \frac{c_{i,k}}{2}u_{i+1,k} + \phi_{i,k}, \qquad (9)$$

Множители для каждого из значений сеточной функции в узлах шаблона, соответствующие разностному уравнению (9), приведены рядом с каждой точкой шаблона на рис. 13.6. Фактически, геометрия шаблона и эти множители задают построенную нами разностную схему.

Несложно убедиться в том, что для получения замкнутой системы разностных алгебраических уравнений систему (9) необходимо дополнить дискретным представлением начального и граничных условий (6) и (7). Тогда число неизвестных будет в точности равно числу уравнений, и процесс формирования разностной схемы будет окончательно завершен.

Примечание

Важно подчеркнуть, что возможная нелинейность полученной системы алгебраических уравнений определяется зависимостями от температуры функций D(u) и ф(и), т. е. как коэффициент диффузии, так и источник тепла могут быть функциями сеточной функции u_{ik}.

Если присмотреться к разностным уравнениям (9) повнимательнее, то можно сразу предложить несложный алгоритм реализации этой разностной схемы. Действительно, каждое неизвестное значение сеточной функции со следующего временного слоя, т. е. левая часть соотношения (9) явно выражается через три ее значения с предыдущего слоя (правая часть), которые уже известны. Таким образом, в случае уравнения теплопроводности нам очень повезло — для расчета 1-го слоя по времени следует попросту подставить в (9) начальное условие (известные значения u с нулевого слоя в узлах сетки), для расчета 2-го слоя достаточно использовать вычисленный таким образом набор и с 1-го слоя и т. д. Из-за того, что разностная схема сводится к такой явной подстановке, ее и называют *явной,* а благодаря пересчету значений с текущего слоя через ранее вычисленные слои - *схемой бегущего счета.*

Линейное уравнение

Сделанные замечания относительно реализации явной схемы для уравнения диффузии тепла сразу определяют алгоритм ее программирования в Mathcad. Для решения задачи нужно аккуратно ввести в листинг соответствующие формулы при помощи элементов программирования.

Решение системы разностных уравнений (9) для модели без источников тепла, т. е. ф(x,T,t)=0 и постоянного коэффициента диффузии D=const приведена в листинге 13.1. В его первых трех строках заданы шаги по временной и пространственной переменным т и Д, а также коэффициент диффузии D, равный единице. В следующих двух строках заданы начальные (нагретый центр области) и граничные (постоянная температура на краях) условия, соответственно. Затем приводится возможное программное решение разностной схемы, причем функция пользователя v(t) задает вектор распределения искомой температуры в каждый момент времени (иными словами, на каждом слое), задаваемый целым числом t.

Листинг 13.1. Явная схема для линейного уравнения теплопроводности

$\tau := 0.0005 \qquad M := 20$

$\frac{1}{M}$

$D(u) := 1$

$\phi(x, u) := 0$

$\mathrm{Init}(x) := \Phi(x - 0.45) - \Phi(x - 0.55)$

$\mathrm{Border}(t) := 0$

$m := 0 .. M$

$u_m := \mathrm{Init}(m \cdot \Delta)$

$$F(v) := \left| \begin{array}{l} v1_0 \leftarrow \mathrm{Border}(\tau \cdot \tau) + 0 \cdot (v_0 + v_1) \\ v1_M \leftarrow \mathrm{Border}(t \cdot \tau) + 0 \cdot (v_M + v_{M-1}) \\ \text{for } m \in 1 .. M-1 \\ \quad v1_m \leftarrow \phi(m \cdot \Delta, v_m) \cdot \tau + v_{m-1} \cdot \dfrac{D(v_{m-1}) \cdot \tau}{\Delta^2} + v_m \cdot \left(1 - \dfrac{2 \cdot D(v_m) \cdot \tau}{\Delta^2}\right) + v_{m+1} \cdot \dfrac{D(v_{m+1}) \cdot \tau}{\Delta^2} \\ v1 \end{array} \right.$$

$$V(t) := \left| \begin{array}{ll} u & \text{if } t = 0 \\ F(V(t-1)) & \text{otherwise} \end{array} \right.$$

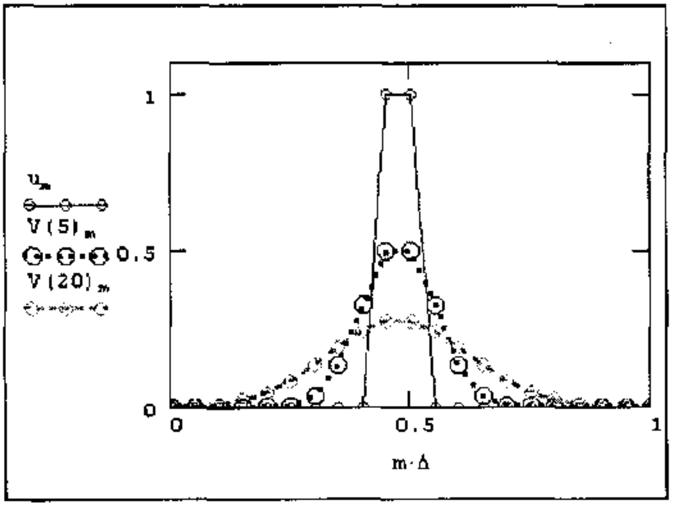


Рис. 13.7. Решение линейного уравнения теплопроводности (листинг 13.1)

Начальное распределение температуры вдоль расчетной области и решение для двух моментов времени показано на рис. 13.7 сплошной, пунктирной и штриховой линиями, соответственно. Физически такое поведение вполне естественно — с течением времени тепло из более нагретой области перетекает в менее нагретую, а зона изначально высокой температуры остывает и размывается.

Нелинейное уравнение

Намного более интересные решения можно получить для нелинейного уравнения теплопроводности, например с нелинейным источником тепла $\phi(u)=10^3\cdot(u-u^3)$. Заметим, что в листинге 13.1 мы предусмотрительно определили коэффициент диффузии и источник тепла в виде пользовательских функций, зависящих от аргумента u, т. е. от температуры (если бы собирались моделировать явную зависимость их от координат, то следует ввести в пользовательскую функцию в качестве аргумента переменную x, как это сделано для источника тепла ф). Поэтому нет ничего проще замены определения этих функций с констант $D(u)=1$ и $\phi(x,u)=0$ на новые функции, которые станут описывать другие модели диффузии тепла. Начнем с того, что поменяем четвертую строку листинга 13.1 на $\phi(x,u)=10^3\cdot(u-u^3)$, не изменяя пока постоянного значения коэффициента диффузии.

Примечание

С физической точки зрения зависимость коэффициента диффузии и функции источника тепла от температуры означает, что эти параметры будут меняться от точки к точке среды, определяясь локальными значениями текущей температуры в этих точках. Ввод ненулевого источника тепла означает, что среда получает определенное количество тепла, тем большее, чем больше локальная температура. Можно догадаться, что введение такой зависимости может моделировать, в частности, горение среды.

Если осуществить расчеты с упомянутым источником (имеющим кубическую нелинейность), то получится очень интересное решение уравнения теплопроводности, имеющее профиль тепловых фронтов. С течением времени граница раздела высокой и низкой температуры распространяется в обе стороны от зоны первичного нафева, оставаясь весьма четко выделенной (рис. 13.8).

Еше более неожиданные решения возможны при нелинейности также и коэффициента диффузии. Например, если взять квадратичный коэффициент диффузии $D(x,u)=u^2$ (что с учетом его умножения на неизвестные функции создаст кубическую нелинейность уравнения), а также $\phi(x,u)=10^3\cdot u^{3.5}$, то Вы сможете наблюдать совсем иной режим горения среды. В отличие от рассмотренного эффекта распространения тепловых фронтов, горение оказывается локализованным в области первичного нагрева среды, причем, температура в центре нафева со временем возрастает до бесконечной величины (рис. 13.9). Такое решение описывает так называемый режим горения «с обострением».

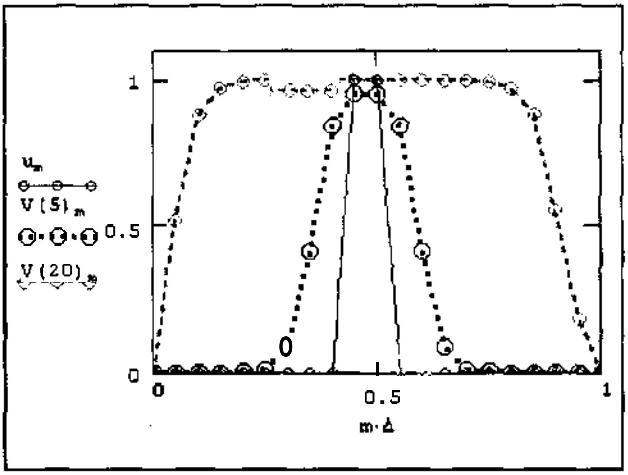


Рис. 13.8. Решение уравнения теплопроводности с нелинейным источником (тепловой фронт)

Читателю предлагается поэкспериментировать с этим и другими нелинейными вариантами уравнения теплопроводности. Существенно, что такие интересные результаты удается получить лишь численно, а в Mathcad только с применением элементов программирования.

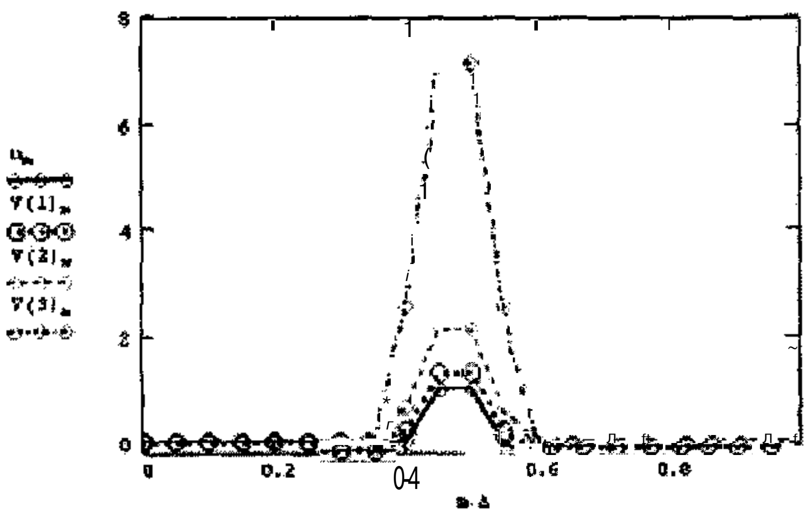


Рис. 13.9. Решение уравнения теплопроводности с нелинейным источником и коэффициентом диффузии (режим локализации горения)

Устойчивость

Как мы убедились, явная разностная схема Эйлера дает вполне разумные результаты и вполне может использоваться для практического моделирования задач, связанных с решением уравнений в частных производных. Однако теперь пришло время сказать об очень важной характеристике разностных схем, которая называется их *устойчивостью*. Не вдаваясь в детали, заметим, что производить расчеты можно только при помощи устойчивых разностных схем, а чтобы пояснить это понятие, обратимся вновь к листингу 13.1, реализующему явную схему для линейного уравнения диффузии.

Слегка изменим соотношение шагов по времени и пространственной координате, произведя расчеты сначала с τ=0.0005 (эти результаты показаны на рис. 13.7 выше) и также с τ=0.0010 и τ=0.0015. Результат применения разностной схемы Эйлера для τ=0.0010 примерно тот же, что и для меньшего значения τ, приведенного на рис. 13.7. А вот следующее (казалось бы, незначительное) увеличение шага по времени приводит к катастрофе (рис. 13.10). Вместо ожидаемого решения получается совершенно неожиданные профили температуры, которые быстро осциллируют вдоль пространственной координаты, причем амплитуда и число пиков этих осцилляций быстро увеличиваются от шага к шагу. Совершенно ясно, что полученное решение не имеет ничего общего с физикой моделируемого явления, а является следствием внутренних свойств самой разностной схемы, которые до этого были для нас скрыты.

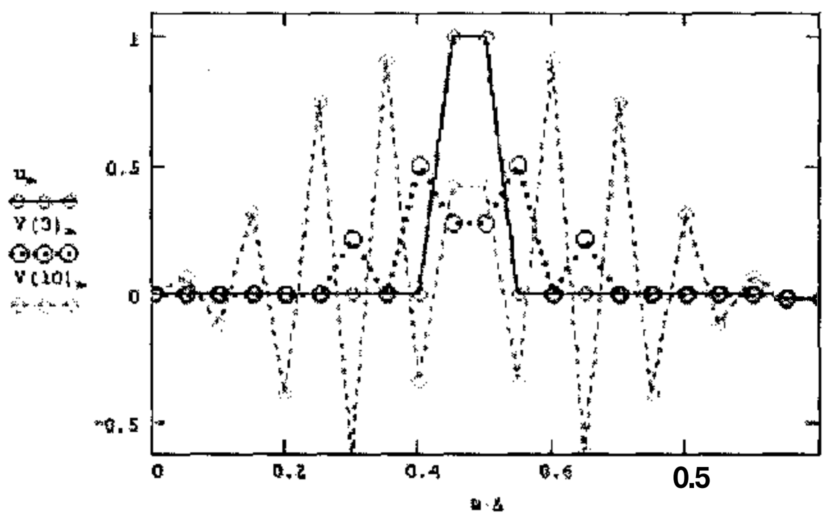


Рис. 13.10. Численное решение уравнения теплопроводности при помощи явной схемы Эйлера (см. листинг 13.1 ниже с временным шагом τ=0.0015)

Характерная «разболтка» решения как раз и является проявлением неустойчивости явной схемы Эйлера для выбранного соотношения шагов по времени и пространству. В теории численных методов показывается, что явная схема Эйлера для уравнения теплопроводности устойчива при значениях коэффициента Куранта, меньших 1, и неустойчива в противоположном случае. Иными словами, существует ограничение для выбора соотношения шагов, заключающееся в том, что для расчета на более частых пространственных сетках необходимо использовать также и малые шаги по времени.

Примечание

Как несложно убедиться, для τ=0.0005 коэффициент Куранта c=0.4, для τ=0.0010 он все еще меньше единицы: C=0.8, а для τ=0.0015 решение уже больше единицы: C=1.2, в связи с чем схема становится неустойчивой (рис. 13.10).

13.2.2. Неявная схема Эйлера

В отличие от явной схемы Эйлера, неявная является безусловно устойчивой (т. е. не выдающей «разболтки» ни при каких значениях коэффициента Куранта). Однако, ценой устойчивости является необходимость решения на каждом шаге по времени системы алгебраических уравнений.

Построение неявной разностной схемы

Чтобы построить неявную разностную схему для уравнения диффузии, используем шаблон, изображенный на рис. 13.11, т. е. для дискретизации пространственной производной будем брать значения сеточной функции с верхнего (неизвестного) слоя по времени. Таким образом, разностное уравнение для (i,k)-го узла будет отличаться от уравнения для явной схемы (8) только индексами по временной координате в правой части:

$$\frac{u_{i,k+1}-u_{i,k}}{\Delta_t}=D\frac{u_{i-1,k+1}-2u_{i,k+1}+u_{i+1,k+1}}{\Delta_x^2}+\phi_{i,k}, \tag{10}$$

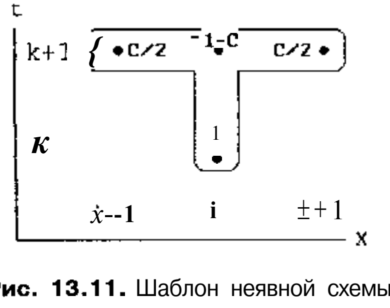


Рис. 13.11. Шаблон неявной схемы для уравнения теплопроводности

Если привести подобные слагаемые, то получится система уравнений, связывающая для каждого i-го узла три неизвестных значения сеточной функции (в самом этом узле и в соседних с ним слева и справа узлах). Множители при неизвестных значениях сеточной функции в узлах шаблона показаны на рис. 13.11 в виде подписей, подобно тому как это было сделано для явной схемы (см. рис.13.6).

Очень важно, что если само уравнение теплопроводности линейно, то с в левой части разностного уравнения является константой, а ф в его правой части может зависеть только от первой степени и. Поэтому система уравнений (10) для всех пространственных узлов i=1...M-1 является линейной системой, что существенно упрощает ее решение (поскольку известно, что для линейных систем с ненулевым определителем решение существует и является единственным). Напомним, что для получения замкнутой системы линейных уравнений необходимо дополнить данный набор разностных уравнений граничными условиями, т. е. известными значениями сеточной функции для i=0 и i=M.

Примечание

Если рассматривать нелинейный случай, то на каждом шаге по времени пришлось бы решать систему нелинейных уравнений, число решений которых могло бы быть большим, и среди них требовалось бы отыскать нужное, а не паразитное решение.

Для реализации неявной схемы, таким образом, можно использовать комбинацию средств программирования Mathcad и встроенной функции решения системы линейных уравнений lsolve. Один из возможных способов решения предложен в листинге 13.2. Большая часть этого листинга является вводом параметров задачи (шагов, начальных и граничных условий), и только в последней его строке определяется функция пользователя, вычисляющая сеточную функцию на каждом временном слое (при помощи встроенной функции решения системы линейных уравнений lsoive). В нескольких предыдущих строках листинга (после расчета коэффициента Куранта сои) формируется матрица системы уравнений А, которая записывается в подходящем для Mathcad виде, как это сделано в листинге 13.2. Как несложно убедиться, столбец правых частей разностных уравнений выражается вычисленными значениями сеточной функции с предыдущего слоя.

Результаты расчетов по неявной схеме показаны на рис. 13.12 и, как видно, они дают примерно те же результаты, что и в случае применения явной схемы (см. рис. 13.7). Обратите внимание, что решение устойчиво при любых значениях коэффициента Куранта (в том числе, и больших 1), поскольку, как следует из соответствующих положений теории численных методов, неявная схема является безусловно устойчивой.

Листинг 13.2. Неявная схема для линейного уравнения теплопроводности

```
D:=1                         ϕ(x,u) :=0

Border(t) :=0

Init(x) := Φ(x - 0.45) - Φ(x - 0.55)

т:= 0.005

M := 20
```

$Д:=\frac{1}{M}$ $\qquad\qquad$ $\Delta = 0.05$

m:=0.. M

$u_m := \mathrm{Init}(m \cdot \Delta)$

$Cou := \frac{2 \cdot D \cdot \tau}{\Delta^2}$ $\qquad\qquad$ $Cou = 4$

m:=1.. M − 1

$A_{m,m} := -Cou - 1$

$A_{m,m-1} := \frac{Сои}{2}$ $\qquad\qquad$ $A_{m,m+1} := \frac{Сои}{2}$

$\bar{A}_{0,0} := 1$ $\qquad\qquad$ $A_{M,M} := 1$

$V(n) := \left| \begin{array}{l} u \quad \text{if } n = О \\ \mathrm{lsolve}(A, -V(n-1)) \quad \text{otherwise} \end{array} \right.$

Алгоритм прогонки

Приведем в данном разделе описание чрезвычайно популярного алгоритма реализации неявных разностных схем, который называется методом *прогонки*. Этот алгоритм имел историческое значение для становления технологий расчетов уравнений в частных производных, и мы просто не можем не упомянуть о нем в этой книге.

Примечание

Сразу оговоримся, что его применение для решения уравнений в частных производных в среде Mathcad может быть оправдано, только если Вы работаете с очень частыми сетками, которые приводят к системам разностных уравнений большой размерности и, соответственно, очень долгому времени вычислений.

Основным вычислительным ядром программы, реализующей на Mathcad неявную разностную схему, было решение (на каждом временном слое) системы линейных алгебраических уравнений, задаваемых матрицей А. Заметим, что эта матрица, как говорят, имеет *диагональное преобладание,* а точнее, является *трехдиагональной* (рис. 13.13). Все ее элементы, кроме элементов на главной диагонали и двух соседних диагоналях, равны нулю. С точки зрения оптимизации быстродействия алгоритма, применение встроенной функции lsolve является весьма расточительным, поскольку основной объем арифметических операций, выполняемых компьютером (а он составляет, как нетрудно убедиться величину порядка M^2), сводится к непроизводительному перемножению нулей.

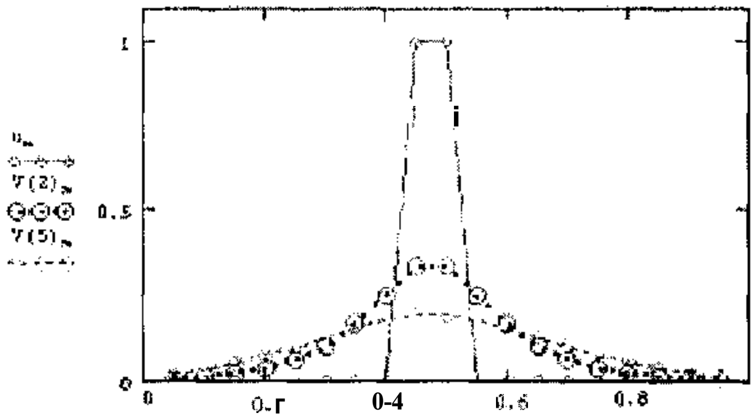


Рис. 13.12. Решение линейного уравнения теплопроводности при помощи неявной схемы на первом слое по времени (листинг 13.2)

Для отыскания решения линейных систем алгебраических уравнений имеется чрезвычайно эффективный алгоритм, называемый прогонкой, который позволяет снизить число арифметических операций на целый порядок, т. е. до значения порядка M. Это означает, что при использовании пространственных сеток с юоо узлами выигрыш во времени вычислений составит величину порядка ю^3! Реализация данного алгоритма приведена в листинге 13.3, который является продолжением листинга 13.2, используя определенные в нем коэффициенты матрицы A, а также начальное условие.

Программа листинга 13.3 осуществляет пересчет одного шага по времени, т. е. заменяет содержимое столбца u с предыдущего временного слоя вычисленными значениями неизвестной функции со следующего слоя. Первые пять строк листинга 13.3 представляют так называемый *обратный* ход прогонки, а последние две строки — ее *прямой* ход. Заинтересовавшемуся читателю предлагается самому оформить представленный алгоритм прогонки в виде программы решения разностных уравнений для вычисления произвольного временного слоя по примеру листингов 13.1 и 13.2. Заметим, что описание этого знаменитого алгоритма можно отыскать практически в любом современном учебнике по численным методам.

	0	1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	0	0	0	0	0	0	0
1	0.5	-2	0.5	0	0	0	0	0	0	0	0
2	0	0.5	-2	0.5	0	0	0	0	0	0	0
3	0	0	0.5	-2	0.5	0	0	0	0	0	0
A = 4	0	0	0	0.5	-2	0.5	0	0	0	0	0
5	0	0	0	0	0.5	-2	0.5	0	0	0	0
6	0	0	0	0	0	0.5	-2	0.5	0	0	0
7	0	0	0	0	0	0	0.5	-2	0.5	0	0
8	0	0	0	0	0	0	0	0.5	-2	0.5	
9	0	0	0	0	0	0	0	0	0.5	-2	0.5
10	0	0	0	0	0	0	0	0	0	0	1

Рис. 13.13. Матрица системы линейных разностных уравнений для неявной схемы (листинг 13.2 для M=10)

Листинг 13.3. Алгоритм прогонки (продолжение листинга 13.2)

$\alpha_{M-1} := 0$ $\qquad\qquad\qquad\qquad$ $\beta_{M-1} := 0$

$m := M-1\,..\,1$

$$\gamma_m := \frac{1}{A_{m,m} + A_{m,m+1}\cdot\alpha_m}$$

$\alpha_{m-1} := \gamma_m \cdot A_{m,m-1}$

$\beta_{m-1} := \gamma_m \cdot \left(A_{m,m+1}\cdot\beta_m - b_m\right)$

$m := 0\,..\,M-1$

$u_{m+1} := \alpha_m \cdot u_m + \beta_m$

13.2.3. О возможности решения многомерных уравнений

Все, что было сказано до сих пор, касалось исключительно способов решения одномерных (в смысле пространственных координат) уравнений. И алгоритмы разностных схем, и встроенные функции, включая появившиеся в 11-й версии *(см. следующий разд.)*, относились к уравнениям, зависящим от одной пространственной координаты.

Можно ли при помощи Mathcad решать двумерные или трехмерные (пространственные) уравнения? С точки зрения программирования пользователем численных алгоритмов типа метода сеток, принципиальных ограничений нет. Разумеется, если сначала аккуратно выписать разностную схему соответствующего многомерного дифференциального уравнения, то вполне возможно запрограммировать ее при помощи описанных нами средств. Самым главным противодействием будет существенное увеличение времени расчетов. Простая оценка необходимого количества операций показывает, что ввод зависимости уравнения от второй пространственной координаты многократно увеличивает число разностных уравнений, которые должны решаться при реализации каждого шага по времени. К примеру, если используется пространственная сетка из 100 узлов по каждой координате, то вместо 10^2 разностных уравнений на каждом шаге придется решать уже 10^4 уравнений, т. е. объем вычислений сразу же возрастает в 100 раз. Вообще говоря, пакет Mathcad не является экономичной средой вычислений, и бороться с их сильно возрастающим объемом пользователю следует еще на этапе разработки алгоритма. Хорошим примером такой борьбы может служить применение специфических алгоритмов, типа метода прогонки *(см. разд. «Алгоритм прогонки» этой главы)*.

Приведем некоторые дополнительные замечания, связанные с возможностью осуществить редукцию громоздких (в смысле организации вычислений) двумерных задач к более простым. Рассмотрим ради примера двумерное уравнение теплопроводности (1) без источника с нулевыми граничными условиями и некоторым начальным двумерным распределением температуры по расчетной поверхности:

$$\frac{\partial u(x,y,t)}{\partial t} = \mu\left(\frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2}\right).$$

Произведем дискретизацию данного уравнения по временной координате, заменяя первую производную ее разностным аналогом и несколько перегруппировывая слагаемые и множители:

$$\frac{\partial^2 u_{i+1}}{\partial x^2} + \frac{\partial^2 u_{i+1}}{\partial y^2} = \frac{u_{i+1} - u_i}{\Delta\tau}$$

Как Вы видите, мы используем неявную разностную схему, заранее заботясь о том, чтобы разностная задача была более устойчивой. Здесь $u_i(x,y)$ — известная с предыдущего шага по времени функция двух пространственных переменных, а $u_{i+1}(x,y)$ — функция, подлежащая определению при реализации каждого шага по времени.

Можно посмотреть на полученную задачу с несколько другой стороны — а именно как на дифференциальное уравнение относительно неизвестной функции двух переменных $u_{i+1}(x,y)$. Подчеркнем, что такое уравнение получается для каждого шага по времени, т. е. для реализации всей разностной схемы требуется решить большое число таких уравнений.

С предложенной точки зрения, на каждом временном шаге необходимо решить некоторое двумерное эллиптическое линейное уравнение, причем его граничные условия определяются граничными условиями исходной задачи.

Это уравнение очень похоже на уравнение Пуассона с той лишь разницей, что в его правую часть, описывающую источник, входит неизвестная функция (к счастью, линейно). Таким образом, зависимость от найденного на предыдущем шаге по времени решения определяется зависимостью от него источника, т. е. правой части.

Суммируя сказанное, можно констатировать, что если Вы имеете запрограммированный алгоритм решения выписанного эллиптического уравнения, чуть более сложного, чем уравнение Пуассона, то его с легкостью можно использовать в качестве подпрограммы реализации разностной схемы двумерного уравнения теплопроводности. Забегая вперед, приходится отметить, что, к сожалению, встроенные функции Mathcad для решения уравнения Пуассона в данном случае не годятся в качестве такой подпрограммы, поскольку предполагают независимость источника от самой неизвестной функции и могут справляться лишь с правой частью, зависящей только от пространственных координат *(см. разд. 13.3.2).*

Не будем далее останавливаться на способах решения многомерных уравнений, ограничившись этими замечаниями относительно путей оптимизации алгоритмов их решения.

13.3. Встроенные функции для решения уравнений в частных производных

Как видно из предыдущего раздела, с уравнениями в частных производных вполне можно справиться, и не прибегая к специфическим средствам Mathcad. Между тем, в пакете Mathcad 11 имеется несколько встроенных функций, при помощи которых можно автоматизировать процесс решения дифференциальных уравнений в частных производных. Рассмотрим в данном разделе основные аспекты их применения, отмечая не только инструкции по их применению, описанные разработчиками Mathcad, но также и некоторые «скрытые» возможности этих функций.

13.3.1. Параболические и гиперболические уравнения

В новой версии Mathcad 11 разработчики впервые применили встроенную функцию `pdesolve` для решения уравнений в частных производных, отлично осознавая значимость этих задач для современного исследователя и инженера. Эта функция применяется в рамках вычислительного блока, начинающегося ключевым словом Given и пригодна для решения различных гиперболических и параболических уравнений.

Встроенная функция для решения одномерного уравнения (или системы уравнений) в частных производных (того, которое определит пользователь в рамках вычислительного блока Given), зависящего от времени t и пространственной координаты x, имеет целый набор различных аргументов и работает следующим образом.

- Pdesolve(u, x, xrange, t, trange, [xpts], [tpts])) — возвращает скалярную (для единственного исходного уравнения) или векторную (для системы уравнений) функцию двух аргументов (x,t), являющуюся решением дифференциального уравнения (или системы уравнений) в частных производных. Результирующая функция получается интерполяцией сеточной функции, вычисляемой согласно разностной схеме.

  - и — явно заданный вектор имен функций (без указания имен аргументов), подлежащих вычислению. Эти функции, а также граничные условия (в форме Дирихле или Неймана) должны быть определены пользователем перед применением функции pdesolve в вычислительном блоке после ключевого слова Given. Если решается не система уравнений в частных производных, а единственное уравнение, то, соответственно, вектор u должен содержать только одно имя функции и вырождается в скаляр.
  - x —пространственная координата (имя аргумента неизвестной функции).
  - xrange — пространственный интервал, т. е. вектор значений аргумента x для граничных условий. Этот вектор должен состоять из двух действительных чисел (представляющих левую и правую границу расчетного интервала).
  - t — время (имя аргумента неизвестной функции).
  - trange — расчетная временная область: вектор значений аргумента t, который должен состоять из двух действительных чисел (представляющих левую и правую границу расчетного интервала по времени).
  - xpts — количество пространственных точек дискретизации (может не указываться явно, в таком случае будет подобрано программой автоматически).
  - tpts — количество временных слоев, т. е. интервалов дискретизации по времени (также может не указываться пользователем явно).

Примечание

Помимо этой функции для решения параболических и гиперболических уравнений, начиная с новой версии Mathcad 11, можно использовать еще одну встроенную функцию numol(). Функция numol() имеет еще большее число аргументов и позволяет управлять дополнительными параметрами метода сеток. Однако пользоваться ею намного сложнее, чем функцией PdesolveO, и поэтому в нашей книге мы не будем на ней особо останавливаться.

В качестве примера использования этой новой функции Mathcad 11 (листинг 13.4) используем то же самое одномерное уравнение теплопроводности (5) с граничными и начальными условиями (6) и (7).

Листинг 13.4. Решение одномерного уравнения теплопроводности

$D := 0.1$

$L := 1$ $\qquad\qquad\qquad\qquad T := 10$

Given

$u_t(x, t) \bullet D \cdot u_{xx}(x, t)$

$u(x, 0) = \Phi(x - 0.45) - \Phi(x - 0.55)$

$u(0, t) \bullet 0$ $\qquad\qquad\qquad\qquad u(L, t) = 0$

$$u := \text{Pdesolve}\left[u, x, \begin{pmatrix} 0 \\ L \end{pmatrix}, t, \begin{pmatrix} 0 \\ T \end{pmatrix}, 100, 10\right]$$

Для корректного использования функции `Pdesolve` предварительно, после ключевого слова Given, следует записать само уравнение и граничные условия при помощи логических операторов (для их ввода в Mathcad существует специальная панель). Обратите внимание, что уравнение должно содержать имя неизвестной функции `u(x,t)` вместе с именами аргументов (а не так, как она записывается в пределах встроенной функции `Pdesolve`). Для идентификации частных производных в пределах вычислительного блока следует использовать нижние индексы, например, $u_{xx}(,t)$ для обозначения второй производной функции `u` по пространственной координате x..

Как видно из рис. 13.14, на котором изображены результаты расчетов по листингу 13.4, встроенная функция с успехом справляется с уравнением диффузии, отыскивая уже хорошо знакомое нам решение. Заметим, что использование встроенной функции pdesolve связано с довольно громоздкими вычислениями, которые могут отнимать существенное время.

Примечание

Как Вы можете заметить, выбирать величину шага по пространственной и временной переменным может как сам алгоритм, так и пользователь (неявным образом, через число узлов сетки). Читателю предлагается повторить вычисления листинга 13.4 для различных комбинаций параметров (главным образом, числа узлов сетки), чтобы проверить, в каких случаях алгоритм встроенной функции справляется с задачей, выдавая верное решение, а в каких дает сбой.

Приведем еще один пример применения функции Pdesolve для решения уравнений в частных производных. Рассмотрим одномерное волновое уравнение, которое описывает, например, свободные колебания струны музыкального инструмента:

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2}. \qquad (11)$$

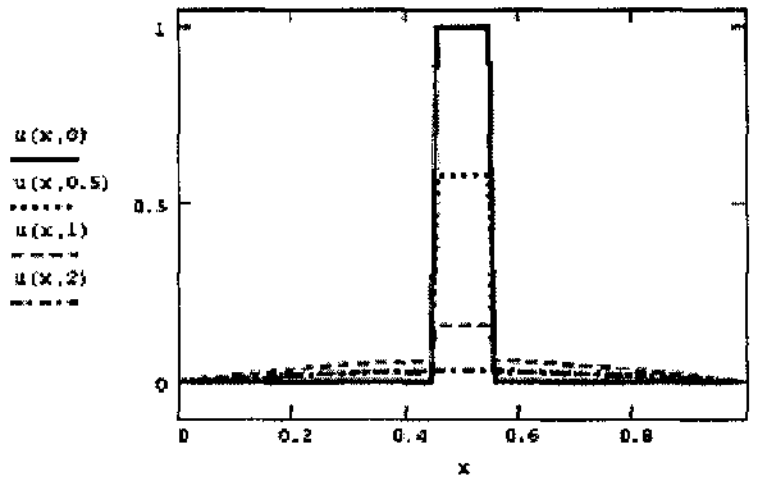


Рис. 13.14. Решение уравнения диффузии тепла при помощи встроенной функции Pdesolve (листинг 13.4)

Здесь неизвестная функция u(x,t) описывает динамику смещения профиля струны относительно невозмущенного (прямолинейного) положения, а параметр с характеризует материал, из которого изготовлена струна.

Как Вы видите, уравнение (И) содержит производные второго порядка как по пространственной координате, так и по времени. Для того чтобы можно было использовать встроенную функцию pdesolve, необходимо переписать волновое уравнение в виде системы двух уравнений в частных производных, введя вторую неизвестную функцию $v=u_t$. Программа для решения волнового уравнения приведена в листинге 13.5, а результат — на рис. 13.15.

Листинг 13.5. Решение волнового уравнения

c:=1

L:=2 · л T:=1

Given

$v_t(x, t) = c^2 \cdot u_{xx}(x, t)$

$u_t(x, t) = v(x, t)$

u (x, 0) s $\sin\left(\frac{\pi \cdot x}{L}\right)$ v(x, 0) = 0

u(0, t) = 0 u(L, t) и 0

$$\begin{pmatrix}u\\v\end{pmatrix} := \mathrm{Pdesolve}\left[\begin{pmatrix}u\\v\end{pmatrix}, x, \begin{pmatrix}0\\L\end{pmatrix}, N, \begin{pmatrix}0\\T\end{pmatrix}\right]$$

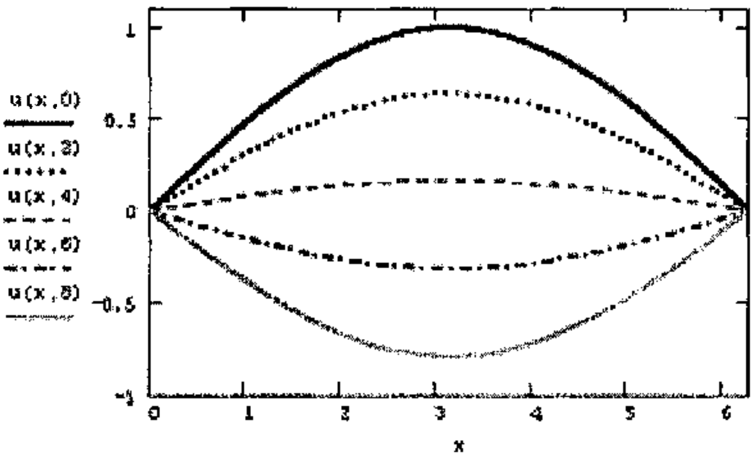


Рис. 13.15. Решение волнового уравнения (листинг 13.5)

13.3.2. Эллиптические уравнения

Решение эллиптических уравнений в частных производных реализовано только для единственного типа задач — двумерного уравнения Пуассона. Это уравнение содержит вторые производные функции u(x,y) по двум пространственным переменным:

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = -f(x,y) \qquad (12)$$

Уравнение Пуассона описывает, например, распределение электростатического поля u(x,y) в двумерной области с плотностью заряда f(x,y) или *(см. разд. 13.1.2)* стационарное распределение температуры u(x,y) на плоскости, в которой имеются источники (или поглотители) тепла с интенсивностью f(x,y).

Примечание

Несмотря на то, что применение встроенных функций, описанных в данном разделе, анонсировано разработчиками Mathcad только для уравнения Пуассона, их можно применять и для решения других уравнений, даже необязательно эллиптического типа. О *том, как осуществить такие расчеты, написано в конце данного раздела.*

Уравнение Пуассона с нулевыми граничными условиями

Корректная постановка краевой задачи для уравнения Пуассона требует задания граничных условий. В Mathcad решение ищется на плоской квадратной области, состоящей из (M+1)×(M+1) точек. Поэтому граничные условия должны быть определены пользователем для всех четырех сторон упомянутого квадрата. Самый простой вариант — это нулевые граничные условия, т. е. постоянная температура по всему периметру расчетной области. В таком случае можно использовать встроенную функцию `multigrid`.

- `multigrid(F,ncycle)` — матрица решения уравнения Пуассона размера (M+1)×(M+1) на квадратной области с нулевыми граничными условиями;
  - `F` — матрица размера (M+1)×(M+1), задающая правую часть уравнения Пуассона;
  - `ncycle` — параметр численного алгоритма (количество циклов в пределах каждой итерации).

Внимание!

Сторона квадрата расчетной области должна включать точно $M=2^n$ шагов, т.е. 2^n+1 узлов, где n — целое число.

Параметр численного метода ncycie в большинстве случаев достаточно взять равным 2. Листинг 13.6 содержит пример использования функции muitigrid для расчета краевой задачи на области ззхзз точки и точечным источником тепла в месте, задаваемом координатами (15,20) внутри этой области.

Листинг 13.6. Решение уравнения Пуассона с нулевыми граничными условиями

$M := 32$

$F_{M,M} := 0$

$F_{15,20} := 10^4$

$G := \mathrm{multigrid}(-F, 2)$

В первой строке листинга задается значение M=32, в двух следующих строках создается матрица правой части уравнения Пуассона, состоящая из всех нулевых элементов, за исключением одного, задающего расположение источника. В последней строке матрице G присваивается результат действия функции muitigrid. Обратите внимание, первый ее аргумент сопровождается знаком "минус", что соответствует записи правой части уравнения Пуассона (11). Графики решения показаны на рис. 13.16 и 13.17 в виде трехмерной поверхности и линий уровня, соответственно.

Уравнение Пуассона с произвольными граничными условиями

В более сложных случаях, например для решения краевой задачи с ненулевыми условиями на границах, следует использовать другую встроенную функцию relax, имеющуюся в Mathcad.

- `relax(a,b,c,d,e,F,v,rjac)` — матрица решения дифференциального уравнения в частных производных на квадратной области, полученного с помощью алгоритма релаксации для метода сеток;
  - a,b,c,d,e — квадратные матрицы коэффициентов разностной схемы, аппроксимирующей дифференциальное уравнение;
  - F — квадратная матрица, задающая правую часть дифференциального уравнения;
  - v — квадратная матрица граничных условий и начального приближения к решению;
  - `rjac` — параметр численного алгоритма (спектральный радиус итераций Якоби).

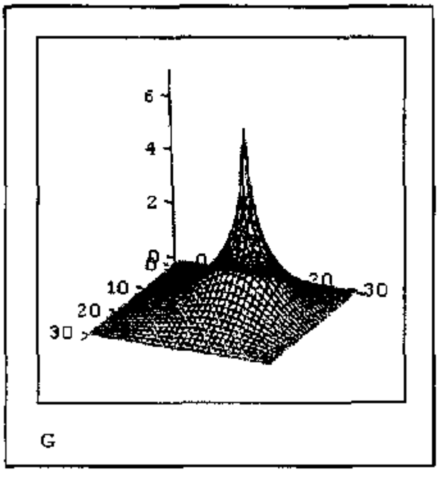


Рис. 13.16. График поверхности решения уравнения Пуассона (листинг 13.6)

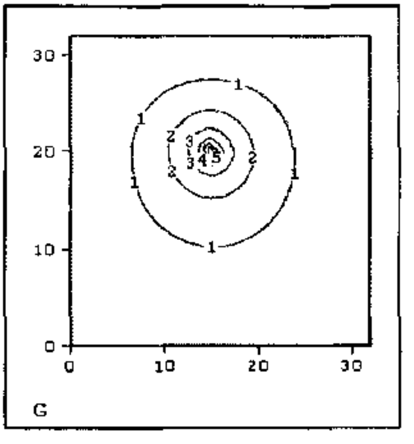


Рис. 13.17. График линий уровня решения уравнения Пуассона (листинг 13.6)

Параметр численного алгоритма характеризует скорость сходимости итераций. Он должен быть числом от о до 1. В матрице граничных условий v необходимо задать только граничные элементы, исходя из значения краевых условий по периметру расчетной области. Прочие (внутренние) элементы этой матрицы служат для задания начального приближения к решению. Суть алгоритма релаксации сводится к тому, что в ходе итераций происходит проверка уравнений и соответствующая коррекция значений искомой функции в каждой точке. Если начальное приближение выбрано удачно, то можно надеяться, что алгоритм сойдется ("срелаксирует") к правильному решению.

Внимание!

Все матрицы, задающие как коэффициенты разностной схемы a,b,c,d,e, граничные условия v, так и само решение F, должны иметь одинаковый размер (M+1)×(M+1), соответствующий размеру расчетной области. При этом целое число M обязательно должно быть степенью двойки: $M=2^n$.

Решение уравнения Пуассона с тремя источниками разной интенсивности при помощи функции relax приведено в листинге 13.7.

Листинг 13.7. Решение уравнения Пуассона с помощью функции relax

$M := 32$

$F_{M,M} := 0$

$F_{15,20} := 10 \qquad F_{25,10} := 5 \qquad F_{10,10} := -5$

$i := 0 .. M \qquad k := 0 .. M$

$a_{i,k} := 1$

$b := a$

$c := a$

$d := a$

$e := -4 \cdot a$

$v_{i,k} := 0$

$G := relax(a, b, c, d, e, -F, v, .95)$

Первые три строки имеют тот же смысл, что и в предыдущем листинге. Только вместо одного источника тепла взято их другое распределение — один сильный источник, один более слабый и один сток тепла. В следующих шести строках задаются коэффициенты разностной схемы. Отложим их обсуждение до последнего раздела этой главы, ограничившись утверждением, что для решения уравнения Пуассона коэффициенты должны быть взяты именно такими, как показано в листинге 13.7. В предпоследней строке задана матрица нулевых граничных условий и нулевых начальных приближений, а в последней матрице G присваивается результат действия функции relax. График полученного решения в виде линий уровня показан на рис. 13.18.

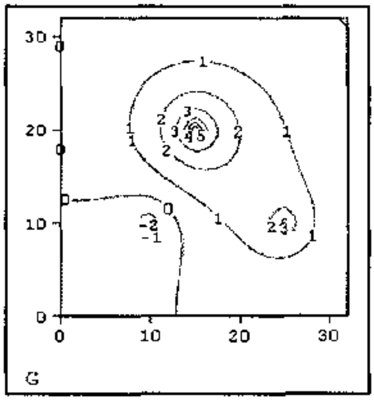


Рис. 13.18. Решение уравнения Пуассона с помощью функции relax (листинг 13.7)

Разностная схема для решения уравнения Пуассона

Несмотря на отсутствие сведений в справочной системе Mathcad о решении других линейных дифференциальных уравнений в частных производных, кроме уравнения Пуассона, сделать это возможно с помощью той же функции relax *(см. предыдущий разд.)*. Для этого нужно правильным образом задать коэффициенты разностной схемы.

Начнем с пояснения выбора этих коэффициентов (см. листинг 13.7) для уравнения Пуассона. Согласно основным идеям метода сеток *(см. разд. «Разностные схемы» этой главы)*, для дискретизации обеих пространственных производных в уравнении (12) следует использовать по три соседних узла вдоль каждой из координат. Поэтому уравнение Пуассона (12) может быть записано в разностной форме при помощи шаблона типа "крест" (рис. 13.19). В этом случае после приведения подобных слагаемых в разностных уравнениях коэффициенты разностной схемы будут такими, как показано возле узлов шаблона на этом рисунке (аналогичные коэффициенты для явной и неявных схем решения уравнения теплопроводности см. на рис. 13.6 и 13.11, соответственно).

Теперь если Вы сравните полученные числа с константами, которые присвоены элементам матриц-аргументов функции relax (см. листинг 13.7), то увидите, что они как раз и описывают вычисленные нами только что коэффициенты разностной схемы "крест". Таким образом, нетрудно сообразить, что с помощью встроенной функции relax можно решать и другие линейные дифференциальные уравнения в частных производных, которые можно аппроксимировать схемой типа "крест" или схемой, являющейся ее составной частью. Конечно, для того чтобы использовать эту встроенную функцию для другого уравнения, необходимо будет составить соответствующую разностную схему.

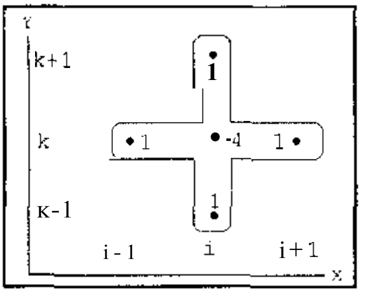


Рис. 13.19. Шаблон аппроксимации уравнения Пуассона "крест"

Решение уравнения диффузии тепла при помощи функции *relax*

Приведем пример применения встроенной функции relax для решения другого уравнения в частных производных (т. е. не уравнения Пуассона, для которого она изначально предназначена). Вычислим при помощи этой функции решение уже хорошо нам знакомого однородного линейного уравнения теплопроводности *(см. разд. 13.1.2)*. Будем использовать явную разностную схему, шаблон которой изображен на рис. 13.6. Для того чтобы «приспособить» для явной схемы функцию relax, требуется только задать ее аргументы в соответствии с коэффициентами, показанными на шаблоне (см. тот же рис. 13.6). Программа, реализующая таким способом явную схему, представлена на листинге 13.8. Число Куранта в этом листинге обозначено переменной c, как и положено явной разностной схеме, она выдает устойчивое решение только для c<1.

Листинг 13.8. Решение уравнения теплопроводности при помощи функции relax

$M := 16$

$C := 0.9$

$i := 0..M$ $\qquad$ $k := 0..M$

$a_{i,k} := \frac{C}{2}$

$b_{i,k} := \frac{C}{2}$

$c_{i,k} := 0$

$d_{i,k} := 1 - c$

$e_{i,k} := -1$

$F_{i,k} := 0$

$v_{i,k} := 0$ $v_{0,5} := 1$

G := relax (a, b, c, d, e, F, v, .5)

Результат действия программы листинга 13.8 показан на рис. 13.20 в виде трехмерной поверхности. Если сравнить рис. 13.20 с рис. 13.4, полученным при расчетах по запрограммированной разностной схеме, то в графиках рис. 13.4 нетрудно узнать сечения этой поверхности плоскостями t=const. Еще раз подчеркнем, что использовать встроенную функцию можно только для тех уравнений, которые допускают построение разностной схемы типа «крест» (см. рис. 13.16) или составного фрагмента этой схемы.

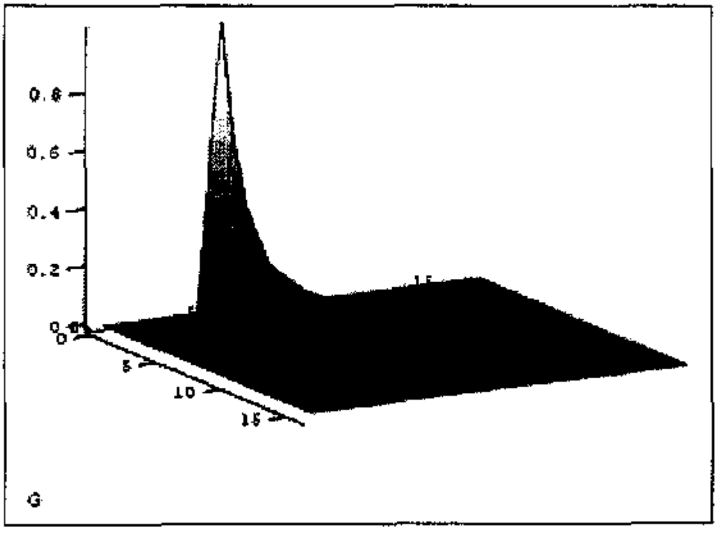


Рис. 13.20. Решение уравнения теплопроводности с помощью функции relax (листинг 13.8)

В заключение разговора об уравнениях в частных производных, нельзя не сказать несколько слов об их визуализации. Результат решения динамических уравнений (зависящих от времени t) выглядит намного эффектнее, если будет представлен в виде анимации. Для создания анимационных роликов расчетное время следует выразить через константу FRAME и затем применить команду **View / Animate** (Вид / Анимация) *(как об этом рассказано в разд. "Создание анимации" гл. 16).*

ГЛАВА 14

Математическая статистика

Mathcad имеет развитый аппарат работы с задачами математической статистики. С одной стороны, имеется большое количество встроенных специальных функций, позволяющих рассчитывать плотности вероятности и другие характеристики основных законов распределения случайных величин *(см. разд. 14.1)*. Наряду с этим в Mathcad запрограммировано соответствующее количество генераторов псевдослучайных чисел для каждого закона распределения, что позволяет эффективно проводить моделирование методами Монте-Карло. Имеется возможность строить гистограммы и рассчитывать статистические характеристики выборок случайных чисел и случайных процессов, таких, как средние, дисперсии, корреляции и т. п. *(см. разд. 14.2)*. При этом случайные последовательности могут как создаваться генераторами случайных чисел, так и вводиться пользователем из файлов.

В заключение главы рассматривается решение средствами Mathcad нескольких типичных примеров из области анализа случайных процессов *(см. разд. 14.3)* математической статистики *(см. разд. 14.4)*.

14.1. Случайные величины

Для моделирования различных физических, экономических и прочих эффектов широко распространены методы, называемые *методами Монте-Карло*. Их основная идея состоит в создании определенной последовательности *случайных* чисел, моделирующей тот или иной эффект, например, шум в физическом эксперименте, случайную динамику биржевых индексов и т. п. Для этих целей в Mathcad имеется ряд встроенных функций, реализующих различные типы генераторов псевдослучайных чисел.

Согласно определению, случайная величина принимает то или иное значение, но какое конкретно, зависит от случайных обстоятельств опыта и заранее точно предсказано быть не может. Можно лишь говорить о *вероятности* $P(x_K)$ принятия *случайной дискретной* величиной того или иного значения x_K, или о вероятности попадания *непрерывной* случайной величины в тот или иной числовой интервал (х,х+Дх). Вероятность $P(x_K)$ ИЛИ P(X)-(ДX), соответственно, может принимать значения от о (такое значение случайной величины совершенно невероятно) до 1 (случайная величина заведомо примет значение от х до х+Дх). Соотношение $P(x_K)$ называют *законом распределения* случайной величины, а зависимость р(х) между возможными значениями непрерывной случайной величины и вероятностями попадания в их окрестность называется ее *плотностью вероятности* (probability density).

В Mathcad имеется ряд встроенных функций, задающих используемые в математической статистике законы распределения. Они вычисляют как значение плотности вероятности различных распределений по значению случайной величины х, так и некоторые сопутствующие функции. Все они, по сути, являются либо встроенными аналитическими зависимостями, либо специальными функциями. Большой интерес представляет наличие генераторов случайных чисел, создающих *выборку* псевдослучайных данных с соответствующим законом распределения. Рассмотрим подробно возможности Mathcad на нескольких наиболее популярных законах распределения, а затем приведем перечень всех распределений, встроенных в Mathcad.

14.1.1. Нормальное (Гауссово) распределение

В теории вероятности доказано, что сумма различных независимых случайных слагаемых (независимо от их закона распределения) оказывается случайной величиной, распределенной согласно нормальному закону (т. н. *центральная предельная теорема)*. Поэтому нормальное распределение хорошо моделирует самый широкий круг явлений, для которых известно, что на них влияют несколько независимых случайных факторов.

Перечислим встроенные функции, имеющиеся в Mathcad для описания нормального распределения вероятностей:

- dnorm(x,μ,σ) — плотность вероятности нормального распределения;
- pnorm(x,μ,σ) — функция нормального распределения;
- cnorm{x} — функция нормального распределения для μ=0,σ=1;
- qnorm(P,μ,σ) — обратная функция нормального распределения;
- rnorm(M,μ,σ) — вектор м независимых случайных чисел, каждое из которых имеет нормальное распределение;
  - х — значение случайной величины;
  - р — значение вероятности;
  - μ — математическое ожидание;
  - σ — среднеквадратичное отклонение.

Математическое ожидание и дисперсия являются, по сути, параметрами распределения. Плотность распределения для трех пар значений параметров показана на рис. 14.1. Напомним, что плотность распределения dnorm задает вероятность попадания случайной величины x в малый интервал от x до x+Дх. Таким образом, например, для первого графика (сплошная линия) вероятность того, что случайная величина x примет значение в окрестности нуля, приблизительно в три раза больше, чем вероятность того, что она примет значение в окрестности x=2. А значения случайной величины, большие 5 и меньшие -5, и вовсе маловероятны.

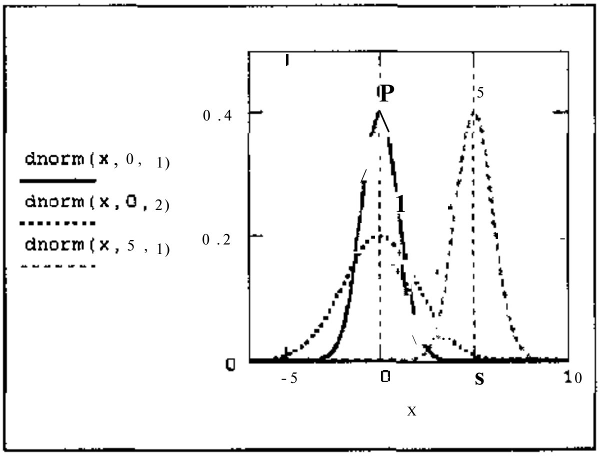


Рис. **14.1.** Плотность вероятности нормальных распределений

Функция распределения F(X) (cumulative probability) — это вероятность того, что случайная величина примет значение меньшее или равное x. Как следует из математического смысла, она является интегралом от плотности вероятности в пределах от $-\infty$ до x. Функции распределения для упомянутых нормальных законов изображены на рис. 14.2. Функция, обратная F(x) (inverse cumulative probability), называемая еще *квантилем распределения,* позволяет по заданному аргументу p определить значение x, причем случайная величина будет меньше или равна x с вероятностью p.

Примечание

Здесь и далее графики различных статистических функций, показанные на рисунках, получены с помощью Mathcad без каких-либо дополнительных выражений в рабочей области.

Приведем несколько примеров, позволяющих почувствовать математический смысл рассмотренных функций на примере случайной величины x, распределенной по нормальному закону с $\mu=0$ и $\sigma=1$ (листинги 14.1 — 14.5).

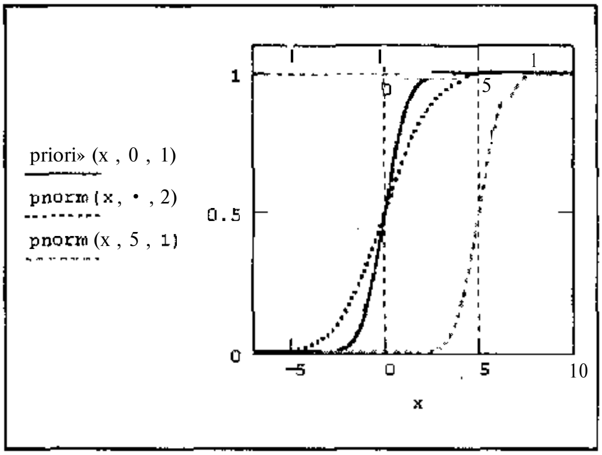


Рис. 14.2. Нормальные функции распределения

Листинг 14.1. Вероятность того, что x будет меньше 1.881

```
pnorm (1.881 ,0,1) = 0.97
```

Листинг 14.2. 97%-ный квантиль нормального распределения

```
qnorm (0.97 , 0, 1) = 1.881
```

Листинг 14.3. Вероятность того, что x будет больше 2

```
1 - pnorm(2 , 0 , 1) = 0.02275
```

Листинг 14.4. Вероятность того, что x будет находиться в интервале (2,3)

```
pnorm (3,0,1) - pnorm (2, 0, 1) = 0.021
```

$$\frac{1}{2}\cdot\left(\mathrm{erf}\left(\frac{3}{\sqrt{2}}\right)-\mathrm{erf}\left(\frac{2}{\sqrt{2}}\right)\right)=0.021$$

Листинг 14.5. Вероятность того, что |x|<2

```
pnorm (2,0,1) -pnorm (-2,0,1) = 0.954
```

$$\mathrm{erf}\left(\frac{2}{\sqrt{2}}\right)=0.954$$

Обратите внимание, что задачи двух последних листингов решаются двумя разными способами. Второй из них связан с еще одной встроенной функцией erf, называемой *функцией ошибок* (или *интегралом вероятности*, или *функцией Крампа)*.

* erf (x) — функция ошибок;
* erfc(x)≡1-erf(x).

Математический смысл функции ошибок ясен из листинга 14.5. Интеграл вероятности имеет всего один аргумент, в отличии от функции нормального распределения. Исторически, последняя пересчитывалась через табулированный интеграл вероятности по формулам, приведенным в листинге !4.6 для произвольных значений параметров μ и о (листинг 14.6).

Листинг 14.6. Вероятность того, что x будет в интервале (2,3)

$\mu := 5 \qquad \sigma := 2$

$\mathrm{pnorm}(3,\mu,\sigma)-\mathrm{pnorm}(2,\mu,\sigma)=0.092$

$$\frac{1}{2}\cdot\left(\mathrm{erf}\left(\frac{3-\mu}{\sigma\cdot\sqrt{2}}\right)-\mathrm{erf}\left(\frac{2-\mu}{\sigma\cdot\sqrt{2}}\right)\right)=0.092$$

Если Вы имеете дело с моделированием методами Монте-Карло, то в качестве генератора случайных чисел с нормальным законом распределения применяйте встроенную функцию rnorm. В листинге 14.7 ее действие показано на примере создания двух векторов по M=500 элементов в каждом с независимыми псевдослучайными числами $x1_i$ и $x2_i$, распределенными согласно нормальному закону. О характере распределения случайных элементов векторов можно судить по рис. 14.3. В дальнейшем мы будем часто сталкиваться с генерацией случайных чисел и расчетом их различных средних характеристик.

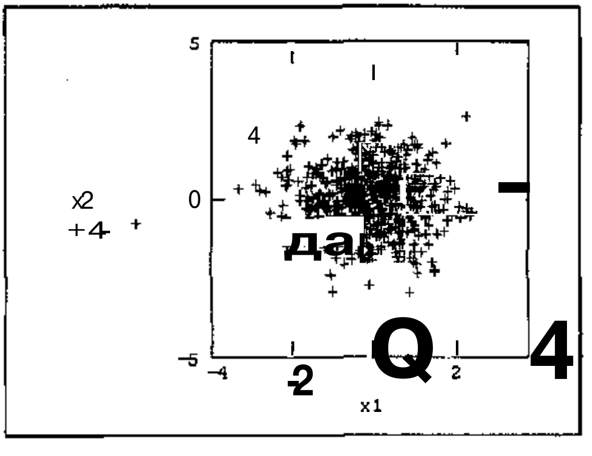


Рис. 14.3. Псевдослучайные числа с нормальным законом распределения (листинг 14.7)

Листинг 14.7. Генерация двух векторов с нормальным законом распределения

о := 1 μ := О

м := 500

x1 := rnorm (м , μ , о)

x2 := rnorm(м , μ , а)

14.1.2. Равномерное распределение

Самое простое распределение случайной величины — это распределение с постоянной вероятностью. Вероятность P=const=1/(b-a) при x∈ (a,Ь) и р=о, для х вне интервала (a,b). Эту плотность вероятности, наряду с прочими статистическими характеристиками, задают следующие встроенные функции:

- dunif (x,a,b) — плотность вероятности равномерного распределения;
- punif(x,a,b) — функция равномерного распределения;
- qunif(p,a,b) — квантиль равномерного распределения;
- runif(M,a,b) — вектор м независимых случайных чисел, каждое из которых имеет равномерное распределение;
- rnd (x) — случайное число, имеющее равномерную плотность распределения на интервале (0,x);
  - x — значение случайной величины;
  - p — значение вероятности;
  - (a,b) — интервал, на котором случайная величина распределена равномерно.

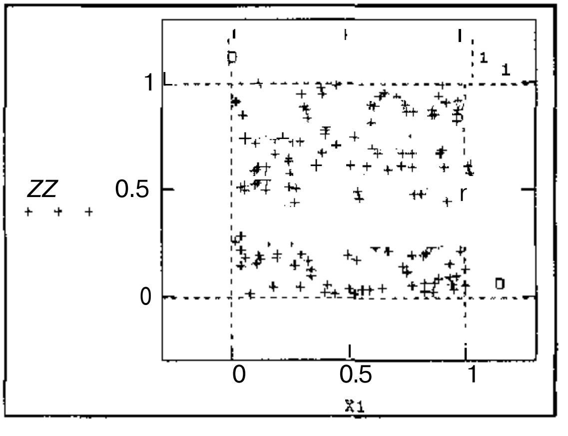


Рис. 14.4. Псевдослучайные числа с равномерным законом распределения

Чаще всего в несложных программах применяется последняя функция, которая приводит к генерации одного псевдослучайного числа. Наличие такой встроенной функции в Mathcad — дань традиции, применяемой в большинстве сред программирования. Пример использования генератора вектора из `M` случайных чисел показан на рис. 14.4, который получен заменой в двух последних строках листинга 14.7 генератора нормальных чисел на `runif(M,0,1)`. Плотность вероятности и функция равномерного распределения показаны на рис. 14.5.

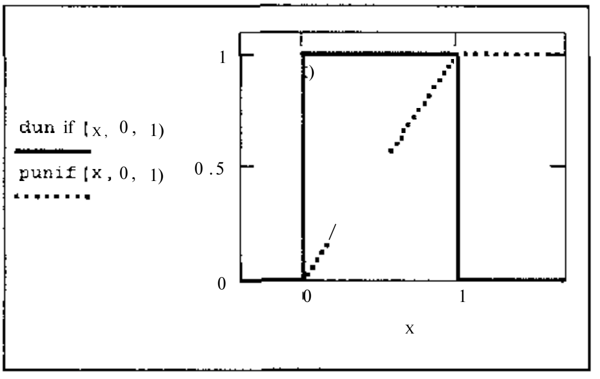


Рис. 14.5. Плотность вероятности и функция равномерного распределения

14.1.3. Биномиальное распределение

Приведем встроенные функции, описывающие еще одно распределение случайной величины, которая, в отличие от двух предыдущих, является не непрерывной, а может принимать лишь дискретные значения. Биномиальное распределение описывает последовательность независимых испытаний, каждое из которых может приводить к генерации определенного события с постоянной вероятностью p.

- `dbinom(k,n,p)` — плотность вероятности биномиального распределения (рис. 14.6);
- pbinom(k,n,p) — функция биномиального распределения;
- qbinom(p,n,p) — квантиль биномиального распределения;
- `rbinom(M,n,p)` — вектор `M` независимых случайных чисел, каждое из которых имеет биномиальное распределение;
  - `k` — дискретное значение случайной величины;
  - p — значение вероятности;
  - п — параметр распределения (количество независимых испытаний);
  - p — параметр распределения (вероятность единичного случайного события).

Примером биномиального распределения может служить n-кратное подбрасывание монеты. Вероятность выпадения орла или решки в каждом испытании равна p=o.5, а суммарное количество выпадений, например орла, задается биномиальной плотностью вероятности. Приведем простой пример: если монета подбрасывалась 50 раз, то наиболее вероятное количество выпадений орла, как видно по максимуму плотности вероятности на рис. 14.6, составляет 25. Вероятность того, что орел выпадет 25 раз, составляет dbinom(25,50,0.5)=0.112, а, скажем, вероятность того, что 15 раз dbinom(15,50,0.5)=0.002.

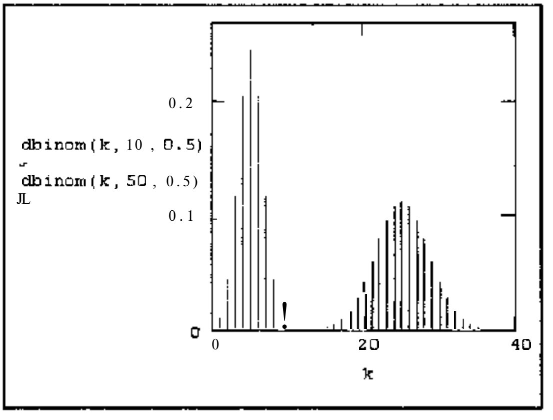


Рис. 14.6. Плотность вероятности биномиального распределения

14.1.4. Другие статистические распределения

Как легко заметить по рассмотренным трем распределениям, Mathcad имеет четыре основные категории встроенных функций. Они различаются написанием их первой литеры, а оставшаяся часть имени функций (ниже в списке функций она условно обозначена звездочкой) идентифицирует тот или иной тип распределения.

- d*(x,par) — плотность вероятности;
- p*(x,par) — функция распределения;
- q*(P,par) — квантиль распределения;
- r*(M,par) — вектор M независимых случайных чисел, каждое из которых имеет соответствующее распределение;
  - x — значение случайной величины (аргумент функции);
  - p — значение вероятности;
  - par — список параметров распределения.

Чтобы получить функции, относящиеся, например, к равномерному распределению, вместо * надо поставить unif и ввести соответствующий список параметров par. Он будет состоять в данном случае из двух чисел a,ь — интервала распределения случайной величины.

Перечислим все типы распределения, реализованные в Mathcad, вместе с их параметрами, на этот раз обозначив звездочкой * недостающую первую букву встроенных функций. Некоторые из плотностей вероятности показаны на рис. J4.7.

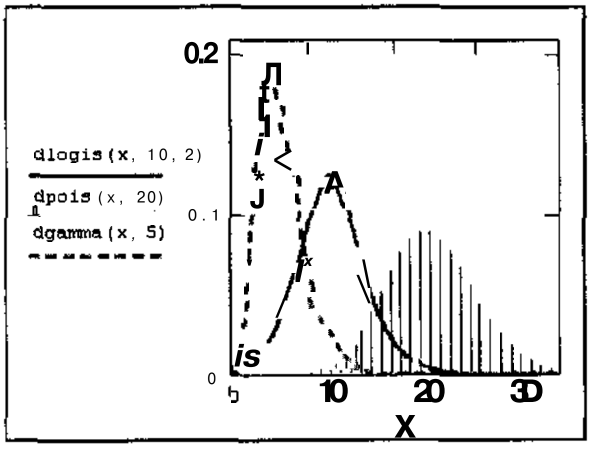


Рис. 14.7. Плотность вероятности некоторых распределений

- *beta(x,s1,s2) — бета-распределение (s1,s2>0 — параметры, 0<x<1).
- *binom(k,n,p) — биномиальное распределение (п — целый параметр, o≤k≤n и o≤p≤1 — параметр, равный вероятности успеха единичного испытания).
- *cauchy(x,l,s) — распределение Коши (1 — параметр разложения, s>o — параметр масштаба).
- *chisq(x,d) — χ^2 ("хи-квадрат") распределение (d>0 — число степеней свободы).
- *exp(x,r) — экспоненциальное распределение (r>0 — показатель экспоненты).
- *F(x,d1,d2) — распределение Фишера (d1,d2>0 — числа степеней свободы).
- *gamma(x,s) — гамма-распределение (s>o — параметр формы).
- *geom(k,p) — геометрическое распределение (0≤p≤1 — параметр, равный вероятности успеха единичного испытания).

- `*hypergeom(k,a,b,n)` — гипергеометрическое распределение (`a,b,n` — целые параметры).
- `*lnorm(x,μ,σ)` — логарифмически нормальное распределение (μ — натуральный логарифм математического ожидания, σ>0 — натуральный логарифм среднеквадратичного отклонения).
- `*logis(x,l,s)` — логистическое распределение (l — математическое ожидание, s>o — параметр масштаба).
- `*nbinom(k,n,p)` — отрицательное биномиальное распределение (п>о — целый параметр, 0<p≤1).
- `*norm(x,μ,σ)` — нормальное распределение (μ — среднее значение, σ>0 — среднеквадратичное отклонение).
- `*pois(k,λ)` — распределение Пуассона (λ>0 — параметр).
- `*t(x,d)` — распределение Стьюдента (d>o — число степеней свободы).
- *unif(x,a,b) — равномерное распределение (a<ь — границы интервала).
- `*weibull(x,s)` — распределение Вейбулла (s>o — параметр).

Вставку рассмотренных статистических функций в программы удобно осуществлять с помощью диалогового окна **Insert Function** (Вставка функции). Для этого необходимо выполнить следующие действия:

1. Установите курсор на место вставки функции в документе.
2. Вызовите диалоговое окно **Insert Function** нажатием кнопки f(x) на стандартной панели инструментов или командой меню **Insert** / **Function** (Вставка / Функция), или нажатием клавиш <Ctrl>+<E>.

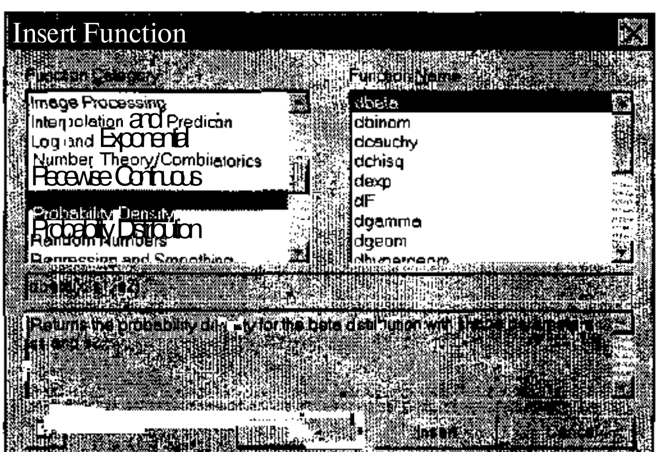


Рис. 14.8. Диалоговое окно Insert Function

3. В списке **Function Category** (Категория функции) (рис. 14.8) выберите одну из категорий статистических функций. Категория **Probability** Density (Плотность вероятности) содержит встроенные функции для плотности вероятности, **Probability Distribution** (Функция распределения) — для вставки функций или квантилей распределения, **Random Numbers** (Случайные числа) — для вставки функции генерации случайных чисел.
4. В списке **Function Name** (Имя функции) выберите функцию, в зависимости от требующегося закона распределения. При выборе того или иного элемента списка в текстовых полях в нижней части окна будет появляться информация о назначении выбранной функции.
5. Нажмите кнопку **OK** для вставки функции в документ.

14.2. Статистические характеристики

В большинстве статистических расчетов Вы имеете дело либо со случайными данными, полученными в ходе какого-либо эксперимента (которые выводятся из файла или печатаются непосредственно в документе), либо с результатами генерации случайных чисел, рассмотренными в предыдущих разделах встроенными функциями, моделирующими то или иное явление методом Монте-Карло. Рассмотрим возможности Mathcad по оценке функций распределения и расчету числовых характеристик случайных данных.

14.2.1. Построение гистограмм

Гистограммой называется график, аппроксимирующий по случайным данным плотность их распределения. При построении гистограммы область значений случайной величины (a,b) разбивается на некоторое количество bin сегментов, а затем подсчитывается процент попадания данных в каждый сегмент. Для построения гистограмм в Mathcad имеется несколько встроенных функций. Рассмотрим их, начиная с самой сложной по применению, чтобы лучше разобраться в возможностях каждой из функций.

Гистограмма с произвольными сегментами разбиения

- hist(intvls,x) — вектор частоты попадания данных в интервалы гистограммы;
  - intvls — вектор, элементы которого задают сегменты построения гистограммы в порядке возрастания $a \le intvls_i < b$;
  - x — вектор случайных данных.

Если вектор intvls имеет bin элементов, то и результат hist имеет столько же элементов. Построение гистограммы иллюстрируется листингом 14.8 и рис. 14.9.

Листинг 14.8. Построение гистограммы

```
N:=1000
bin:=30
x:=rnorm(N,0,1)
lower:=floor(min(x))
upper:=ceil(max(x))
```

$$h := \frac{upper - lower}{bin}$$

$j := 0 .. bin$

$int_j := lower + h \cdot j$

$f := \frac{1}{N \cdot h} \cdot hist(int, x)$

Для анализа взято `N=1000` данных с нормальным законом распределения, созданных генератором случайных чисел (третья строка листинга). Далее определяются границы интервала `(upper, lower)`, содержащего внутри себя все случайные значения, и осуществляется его разбиение на количество (bin) одинаковых сегментов, начальные точки которых записываются в вектор `int` (предпоследняя строка листинга).

Внимание!

В векторе `int` можно задать произвольные границы сегментов разбиения так, чтобы они имели разную ширину.

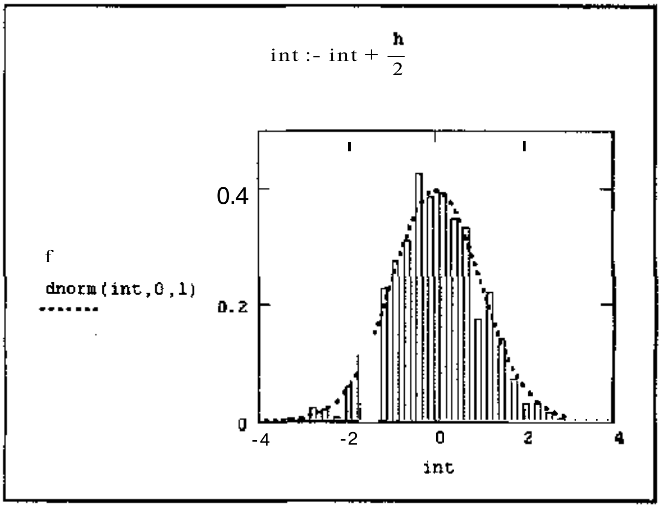


Рис. 14.9. Построение гистограммы (листинг 14.8)

Обратите внимание, что в последней строке листинга осуществлена нормировка значений гистограммы, с тем чтобы она правильно аппроксимировала плотность вероятности, также показанную на графике. Очень важно переопределение вектора int в самом верху рис. 14.9, которое необходимо для перехода от левой границы каждого элементарного сегмента к его центру.

Гистограмма с разбиением на равные сегменты

Если нет необходимости задавать сегменты гистограммы разной ширины, то удобнее воспользоваться упрощенным вариантом функции hist.

- hist(bin,x) — вектор частоты попадания данных в интервалы гистограммы;
  - bin — количество сегментов построения гистограммы;
  - x — вектор случайных данных.

Для того чтобы использовать этот вариант функции hist вместо предыдущего, достаточно заменить первый из ее аргументов в листинге 14.8 следующим образом:

$$f := \frac{1}{N \cdot h} \cdot \mathrm{hist}(\mathrm{int}, x)$$

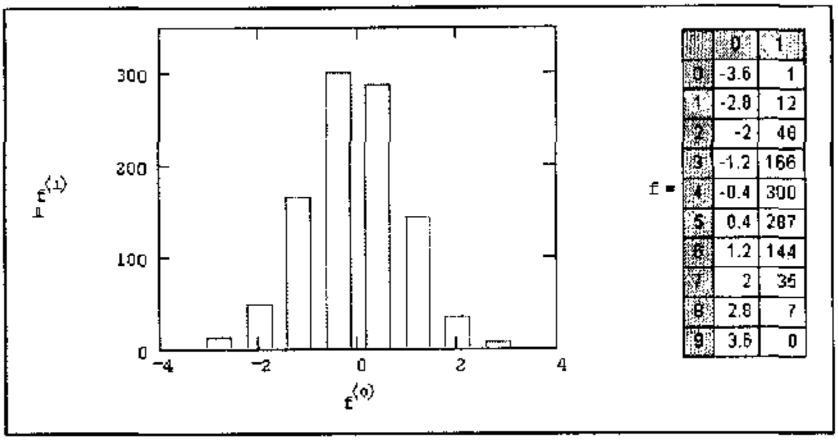
Недостаток упрощенной формы функции hist в том, что по-прежнему необходимо дополнительно определять вектор сегментов построения гистограммы. От этого недостатка свободна появившаяся в Mathcad 2001 функция histogram.

- histogram(bin,x) — матрица гистограммы размера binx2, состоящая из столбца сегментов разбиения и столбца частоты попадания в них данных;
  - bin — количество сегментов построения гистограммы;
  - x — вектор случайных данных.

Примеры использования функции histogram приведены в листинге 14.9 и рис. 14.10. Сравнение с предыдущим листингом подчеркивает простоту построения гистограммы этим способом (стоит отметить, что в листинге 14.9, в отличие от предыдущего, мы не нормировали гистограмму).

Листинг 14.9. Упрощенный вариант построения гистограммы

```
N:=1000
bin :=10
x:=rnorm(N , 0 , 1)
f:=histogram (bin , x)
```



	0	1
0	-3.6	1
1	-2.8	12
2	-2	48
3	-1.2	166
4	-0.4	300
5	0.4	287
6	1.2	144
7	2	35
8	2.8	7
9	3.6	0

Рис. 14.10. График и матрица гистограммы (листинг 14.9}

Создание графика гистограммы

Для того чтобы создать график в виде гистограммы:

1. Постройте двумерный график, задайте переменные по осям и пределы оси x (в примере из листинга 14.9 это числа lower и upper).
2. Войдите в диалоговое окно **Formatting Currently Selected Graph** (Форматирование) выбранного графика (например, двойным щелчком мыши) и перейдите на вкладку **Traces** (Графики).

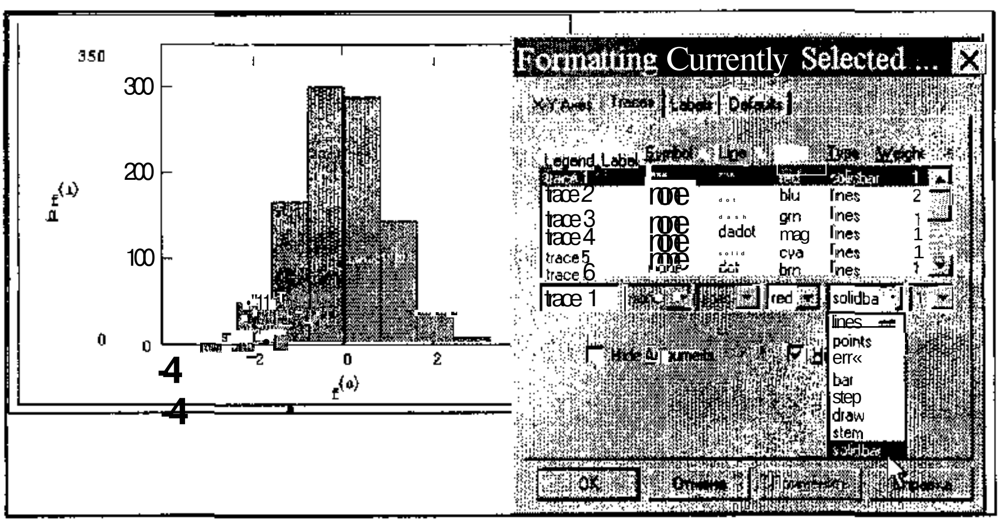


Рис. 14.11. Установка типа графика для построения гистограммы

3. Установите для серии данных гистограммы в поле **Type** (Тип) элемент списка **bar** (столбцы) или **solidbar** (гистограмма) (рис. 14.11).
4. Нажмите кнопку OK.

На рис. 14.9 и 14.10 были применены установки графика **bar** (столбцы). В Mathcad 2001 появилась новая возможность построения гистограммы в более привычном виде — закрашенными столбиками **(solidbar).** Такой тип графика иллюстрируется рис. 14.11.

14.2.2. Среднее значение и дисперсия

В Mathcad 11 имеется ряд встроенных функций для расчетов числовых статистических характеристик рядов случайных данных.

- mean(x) — выборочное среднее значение;
- median (x) — выборочная *медиана* (median) — значение аргумента, которое делит гистограмму плотности вероятностей на две равные части;
- var{x) — выборочная *дисперсия* (variance);
- stdev(x) — *среднеквадратичное (или "стандартное") отклонение* (standard deviation);
- max(x), min(x) — максимальное и минимальное значения выборки;
- mode(x) — наиболее часто встречающееся значение выборки;
- Var(x), Stdev(x) — выборочная дисперсия и среднеквадратичное отклонение в другой нормировке;
  - x — вектор (или матрица) с выборкой случайных данных.

Пример использования первых четырех функций приведен в листинге 14.10.

Листинг 14.10. Расчет числовых характеристик случайного вектора

```
x := rweibull (1000 ,1.5)
N := length(x)          N = 1x 10^3
mean (x) = 0.917
median (x) = 0.781
var (x) = 0.402
stdev (x) =0.634              √var(x) = 0.634
hi := mean(x) + stdev (x)            lo := mean (x) - stdev (x)
```

На рис. 14.12 приведена гистограмма выборки случайных чисел, распределенных согласно закону Вейбулла. Пунктирные вертикальные прямые, показанные на графике, рассчитаны в последней строке листинга и обозначают стандартное отклонение от среднего значения. Гистограмма получена с помощью листинга 14.8, рассмотренного в предыдущем разделе. Обратите внимание, что поскольку распределение **Вейбулла,** в отличие, например, от Гауссова, несимметричное, то медиана не совпадает со средним значением.

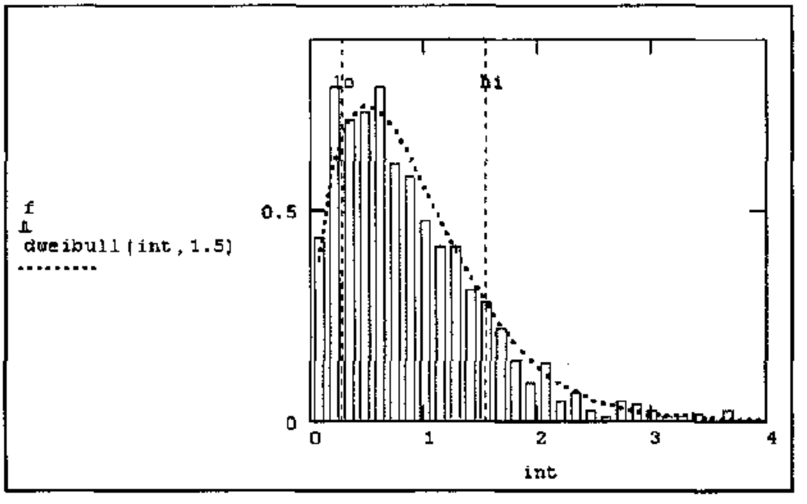


Рис. 14.12. Гистограмма распределения Вейбулла (листинг 14.10)

Определение статистических характеристик случайных величин приведено в листинге 14.11 на еще одном примере обработки выборки малого объема (по пяти данным). В том же листинге иллюстрируется применение еще двух функций, которые имеют смысл дисперсии и стандартного отклонения в несколько другой нормировке. Сравнивая различные выражения, Вы без труда освоите связь между встроенными функциями.

Внимание!

Осторожно относитесь к написанию первой литеры в этих функциях, особенно при обработке малых выборок (листинг **14.11**).

Листинг 14.11. К определению статистических характеристик

$x := (5\ \ 2\ \ 14\ \ 3\ \ 2)^T$

$N := \mathrm{length}(x)$ $\qquad N = 5$

$\frac{1}{N} \cdot \sum_{i=0}^{N-1} x_i = 5.2$ $\qquad m := \mathrm{mean}(x)$ $\qquad m = 5.2$

$\mathrm{median}(x) = 3$ $\qquad \mathrm{mode}(x) = 2$ $\qquad \max(x) = 14$ $\qquad \min(x) = 2$

$$\sum_{i=0}^{N-1} (x_i - m)^2 = 20.56$$

$var(x) = 20.56$ $\qquad Var(x) \cdot \frac{N-1}{N} = 20.56$

$\sqrt{var(x)} = 4.534$ $\qquad stdev(x) = 4.534$

$$Stdev(x) \cdot \sqrt{\frac{N-1}{N}} = 4.534$$

$$\frac{1}{N-1} \cdot \sum_{i=0}^{N-1} (x_i - m)^2 = 25.7$$

$Var(x) = 25.7$ $\qquad var(x) \cdot \frac{N}{N-1} = 25.7$

$\sqrt{Var(x)} = 5.07$ $\qquad Stdev(x) = 5.07$

$$stdev(x) \cdot \sqrt{\frac{N}{N-1}} = 5.07$$

14.2.3. Генерация коррелированных случайных чисел

До сих пор мы рассматривали наиболее простой случай применения генераторов независимых случайных чисел. В методах Монте-Карло часто требуется создавать случайные числа с определенной *корреляцией*. Приведем пример программы, создающей два вектора x1 и x2 одинакового размера и одним и тем же распределением, случайные элементы которых попарно коррелированы с коэффициентом корреляции R (листинг 14.12),

Листинг 14.12. Генерация попарно коррелированных случайных чисел

$a := 3$ $\qquad N := 1000$

$R := 0.4$

$x1 := rnorm(N, 0, \sigma)$

$x2 := R \cdot x1 + \sqrt{1 - R^2} \cdot rnorm(N, 0, a)$

$corr(x1, x2) = 0.415$

Результат действия программы для R=0.4 показан на рис. 14.13 (слева). Сравните полученную выборку с правым графиком, полученным для высокой корреляции (R=0.9) и с рис. 14.3 *(см. разд. 14.1.1)* для независимых данных, т. е. R=0.

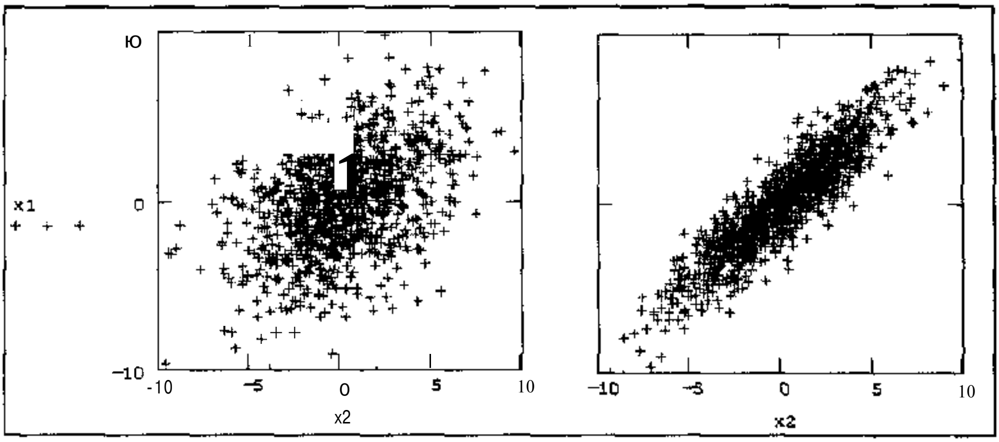


Рис. 14.13. Псевдослучайные числа с корреляцией R=0 .4 (листинг 14.12) и R=0 .9

14.2.4. Ковариация и корреляция

Функции, устанавливающие связь между парами двух случайных векторов, называются *ковариацией* и корреляцией (или, по-другому, *коэффициентом корреляции*). Они различаются нормировкой, как следует из их определения (листинг 14.13).

- corr(x) — коэффициент корреляции двух выборок;
- cvar(x) — ковариация двух выборок;
  - x1, x2 — векторы (или матрицы) одинакового размера с выборками случайных данных.

Листинг 14.13. Расчет ковариации и корреляции (продолжение листинга 14.12)

m1 := mean (x1) m2 := mean (x2)

σ1 := stdev (x1) σ2 := stdev (x2)

$$\frac{1}{N}\cdot\sum_{i=0}^{N-1}\left(x1_i - m1\right)\cdot\left(x2_i - m2\right) = 3.823$$

cvar (x1 , x2) = 3.823

$$\frac{\mathrm{cvar}(x1, x2)}{\sigma1\cdot\sigma2} = 0.415$$

corr (x1 , x2) = 0 . 415

14.2.5. Коэффициенты асимметрии и эксцесса

Коэффициент асимметрии задает степень асимметричности плотности вероятности относительно оси, проходящий через ее центр тяжести. Коэффициент асимметрии определяется третьим центральным моментом распределения. В любом симметричном распределении с нулевым математическим ожиданием, например нормальным, все нечетные моменты, в том числе и третий, равны нулю, поэтому коэффициент асимметрии тоже равен нулю.

Степень сглаженности плотности вероятности в окрестности главного максимума задается еще одной величиной — *коэффициентом эксцесса.* Он показывает, насколько острую вершину имеет плотность вероятности по сравнению с нормальным распределением. Если коэффициент эксцесса больше нуля, то распределение имеет более острую вершину, чем распределение Гаусса, если меньше нуля, то более плоскую.

Для расчета коэффициентов асимметрии и эксцесса в Mathcad имеются две встроенные функции.

- `kurt(x)` — коэффициент эксцесса *(kurtosis)* выборки случайных данных x;
- `skew(x)` — коэффициент асимметрии *(skewness)* выборки случайных данных x.

Примеры расчета коэффициентов асимметрии и эксцесса для распределения Вейбулла (см. рис. 14.10) приведены в листинге 14.14.

Листинг 14.14. Расчет выборочных коэффициентов асимметрии и эксцесса

```
x := rweibull (1000 ,1.5)
skew (x) = 1.216
kurt (x) = 1.89
```

14.2.6. Другие статистические характеристики

В предыдущих разделах были рассмотрены встроенные функции, рассчитывающие наиболее часто используемые статистические характеристики выборок случайных данных. Иногда в статистике встречаются и иные функции, например, помимо арифметического среднего, применяются другие средние значения.

- `gmean(x)` — геометрическое среднее выборки случайных чисел;
- `hmean(x)` — гармоническое среднее выборки случайных чисел.

Математическое определение этих функций и пример их использования в Mathcad приведены в листинге 14.15.

Листинг 14.15. Вычисление различных средних значений

$N := 10$

$x := \mathrm{runif}(N, 0, 1)$

$\frac{1}{N} \cdot \sum_{i=0}^{N-1} x_i = 0.338$ mean (x) = 0.338

$\left(\frac{1}{N} \bullet \sum_{i=0}^{N-1} \frac{1}{x_i}\right)^{-1} = 0.012$ hmean (x) = 0.012

$\sqrt[N]{\prod_{i=0}^{N-1} x_i} = 0.171$ gmean (x) = 0.171

14.2.7. Действие статистических функций на матрицы

Все рассмотренные примеры работы статистических функций относились к векторам, элементы которых были случайными числами. Но точно так же все эти функции применяются и по отношению к выборкам случайных данных, сгруппированных в матрицы. При этом статистические характеристики рассчитываются для совокупности всех элементов матрицы, без разделения ее на строки и столбцы. Например, если матрица имеет размерность M×N, то и объем выборки будет равен мы.

Соответствующий пример вычисления среднего значения приведен в листинге 14.16. В его первой строке определяется матрица данных x размера 4x2. Действие встроенной функции mean матричного аргумента (последняя строка листинга) иллюстрируется явным суммированием элементов матрицы x (предпоследняя строка). Действие прочих встроенных функций на матрицы совершенно аналогично действию их на векторы (листинг 14.17).

Листинг 14.16. Вычисление среднего значения элементов матрицы

$x := \begin{pmatrix} 1.0 & 4 \\ 1.5 & 7 \\ 0.9 & 1.2 \\ 1.2 & 12 \end{pmatrix}$

$M := \mathrm{rows}(x)$ $M = 4$

$N := \mathrm{cols}(x)$ $N = 2$

$$\frac{1}{M \cdot N} \cdot \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} x_{i,j} = 3.6$$

mean (x) = 3.6

Листинг 14.17. Действие различных статистических функций на матрицу

median (x) = 1.35

mode (x) = 1.2

var (x) = 14.033

Var (x) = 16.037

stdev (x) = 3.746

Stdev (x) = 4.005

Примечание

Некоторые статистические функции (например, вычисления ковариации) имеют два аргумента. Они также могут быть матрицами, но, в соответствии со смыслом функции, должны иметь одинаковую размерность.

Большинству статистических функций позволяется иметь в качестве аргументов даже не одну матрицу, а любое количество матриц, векторов и скаляров. Числовые характеристики будут рассчитаны для всей совокупности значений аргументов функции. Соответствующий пример приведен в листинге 14.18.

Листинг 14.18. Статистические функции нескольких аргументов

$x := (1 \ \ -1 \ \ 4)$

$y := \begin{pmatrix} 3 \\ 8 \end{pmatrix}$

$z := \begin{pmatrix} 3 & 7 \\ 11 & 4 \end{pmatrix}$

mean (x, y, z) = 4.444

stdev (x, 5, 77) = 29.976

median (2) = 2

mode (y, z) = 3

14.3. Случайные процессы

Встроенные функции для генерации случайных чисел создают выборку из случайных данных A_i. Часто требуется создать непрерывную или дискретную случайную функцию A(t) одной или нескольких переменных *(случайный процесс* или *случайное поле),* значения которой будут упорядочены относительно своих переменных. Создать псевдослучайный процесс можно способом, представленным в листинге 14.19.

Листинг 14.19. Генерация псевдослучайного процесса

```
N := 20
T :=0.5
Tmax := (N-1) • г
j := 0 .. N - 1
Tj := j • т
x := rnorm(N, 0, 1)
KS1 := cspline(T, x)
A(t) := interp(KS1, T, x, t)
```

В первой строке листинга 14.19 определено количество N независимых случайных чисел, которые будут впоследствии сгенерированы, и радиус временной корреляции т. В следующих трех строках определяются моменты времени T_j, которым будут отвечать случайные значения $A(t_j)$. Создание нормального случайного процесса сводится к генерации обычным способом вектора независимых случайных чисел x и построению интерполяционной зависимости в промежутках между ними. В листинге 14.19 используется сплайн-интерполяция *(см. гл. 15).*

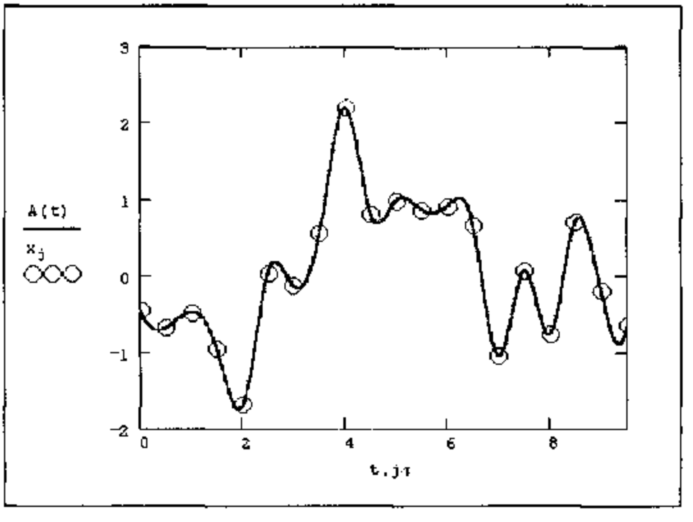


Рис. 14.14. Псевдослучайный процесс (листинг 14.19)

В результате получается случайный процесс A(t), радиус корреляции которого определяется расстоянием т между точками, для которых строится интерполяция. График случайного процесса A(t) вместе с исходными случайными числами показан на рис. 14.14. Случайное поле можно создать несколько более сложным способом с помощью многомерной интерполяции.

К случайным процессам, сгенерированным таким способом, как и к данным эксперимента, применяются любые статистические методы обработки, например корреляционный или спектральный анализ. Приведем в качестве примера листинг 14.20, показывающий, как организовать расчет корреляционной функции случайного процесса.

Листинг 14.20. Дискретизация случайного процесса и вычисление корреляционной функции *(продолжение листинга 14.19)*

$\Delta := 0.02$

$M := 20$

$n := \text{floor}\left(\dfrac{\text{Tmax}}{\Delta}\right) \qquad n = 475$

$j := 0 .. n$

$Y_j := A(\Delta \cdot j)$

$m := \text{mean}(Y)$

$D := \text{var}(Y) \qquad D = 0.785$

$$R(j) := \frac{1}{(n - 2 \cdot M)} \cdot \sum_{i = M}^{n-M} \frac{(Y_{i+j} - m) \cdot (Y_i - m)}{D}$$

$R(0) = 1.025$

Дискретизация интервала (0,ттах) для случайного процесса A(t) произведена с различным элементарным интервалом Д (первая строка листинга). В зависимости от значения Д, получается различный объем п выборки случайных чисел Y_i, являющихся значениями случайной функции A(t) в точках дискретизации. В последних четырех строках определяются различные характеристики случайной величины Y, являющиеся, по сути, характеристиками случайного процесса A(t). График рассчитанной в $2 \cdot M+1$ точках корреляционной функции R(j) показан на рис. 14.15.

Примечание

Внимательному читателю предлагается самостоятельно ответить на вопрос: почему при таком расчете корреляционной функции ее значение R(0) не равно 1, как должно быть по определению?

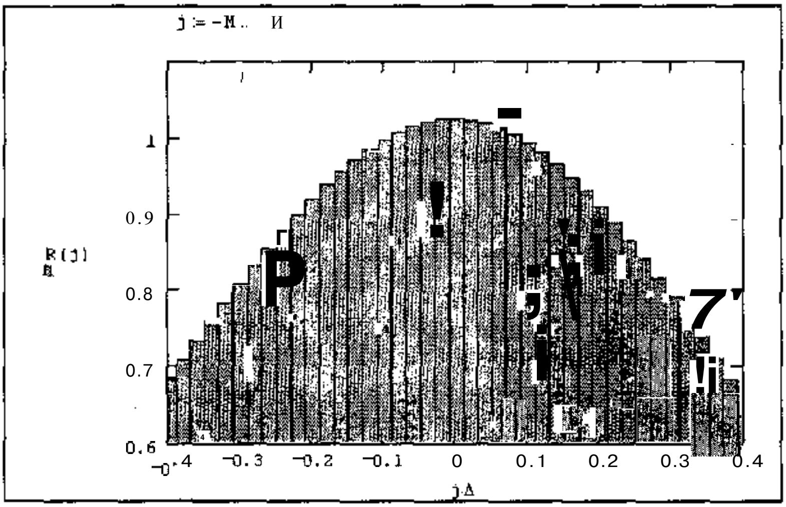


Рис. 14.15. Корреляционная функция (листинги 14.19—14.20)

14.4. Некоторые примеры

Приведем два характерных статистических примера, которые легко решаются с помощью Mathcad.

14.4.1. Интервальная оценка дисперсии

Требуется определить числовой интервал (L,U), внутри которого будет лежать с вероятностью 1-α=75% дисперсия нормальной случайной величины, исходя из объема выборки в N чисел. Эта задача решается в статистике с помощью χ^2-распределения (листинг 14.21).

Листинг 14.21. Интервальное оценивание дисперсии

$N := 50$

$x := rnorm(N, 0, 1)$

$\alpha := 0.25$ $\qquad 1 - \alpha = 0.75$

$\chi2_0 := qchisq\left(\frac{\alpha}{2}, N-1\right)$ $\qquad \chi2_0 = 37.901$

$\chi2_1 := qchisq\left(1 - \frac{\alpha}{2}, N-1\right)$ $\qquad \chi2_1 = 60.53$

$L := \frac{(N-1) \cdot Stdev(x)^2}{\chi2_1}$ $\qquad U := \frac{(N-1) \cdot Stdev(x)^2}{\chi2_0}$

$L = 0.662$ $\qquad U = 1.057$

Указанный интервал называется (1-α)% *доверительным интервалом.* Обратите внимание на использование при решении данной задачи функции stdev (с прописной буквы) для расчета выборочного стандартного отклонения. В статистике часто встречаются выражения, которые более удобно записывать через функции в такой нормировке, именно для этого они и появились в Mathcad.

14.4.2. Проверка статистических гипотез

В статистике рассматривается огромное число задач, связанных с проверкой тех или иных гипотез н. Разберем пример простой гипотезы. Пусть имеется выборка N чисел с нормальным законом распределения и неизвестными дисперсией и математическим ожиданием. Требуется принять или отвергнуть гипотезу н о том, что математическое ожидание закона распределения равно некоторому числу μ0=0 .2.

Задачи проверки гипотез требуют задания *уровня критерия* проверки гипотезы а, который описывает вероятность ошибочного отклонения истинной н. Если взять а очень малым, то гипотеза, даже если она ложная, будет почти всегда приниматься; если, напротив, взять а близким к 1, то критерий будет очень строгим, и гипотеза, даже верная, скорее всего, будет отклонена. В нашем случае гипотеза состоит в том, что μ0=0.2, а альтернатива — что μ0≠0.2. Оценка математического ожидания, как следует из курса классической статистики, решается с помощью распределения Стьюдента с параметром N-1 (этот параметр называется *степенью свободы распределения).*

Для проверки гипотезы (листинг 14.22) рассчитывается (а/2) — квантиль распределения Стьюдента т, который служит критическим значением для принятия или отклонения гипотезы. Если соответствующее выборочное значение t по модулю меньше т, то гипотеза принимается (считается верной). В противном случае гипотезу следует отвергнуть.

Листинг 14.22. Проверка гипотезы о математическом ожидании при неизвестной дисперсии

$N := 50$

$x := \mathrm{rnorm}(N, 0, 1)$

$a := 0.1 \qquad 1 - a = 0.9$

$\mu 0 := 0.2$

$$t := \frac{\mathrm{mean}(x) - \mu 0}{\left(\frac{\mathrm{Stdev}(x)}{\sqrt{N}}\right)} \qquad t = -1.883$$

$$T := qt\left(1-\frac{\alpha}{2}, N-1\right) \qquad T = 1.677$$

$$|t| < T = 0$$

В последней строке листинга вычисляется истинность или ложность условия, выражающего решение задачи. Поскольку условие оказалось ложным (равным не 1, а 0), то гипотезу необходимо отвергнуть.

На рис. 14.16 показано распределение Стьюдента с `N-1` степенью свободы вместе с критическими значениями, определяющими соответствующий интервал. Если t (оно тоже показано на графике) попадает в него, то гипотеза принимается; если не попадает (как произошло в данном случае) — отвергается. Если увеличить a, ужесточив критерий, то границы интервала будут сужаться, по сравнению с показанными на рисунке.

В листинге 14.23 приводится альтернативный способ проверки той же самой гипотезы, связанный с вычислением значения не квантиля, а самого распределения Стьюдента.

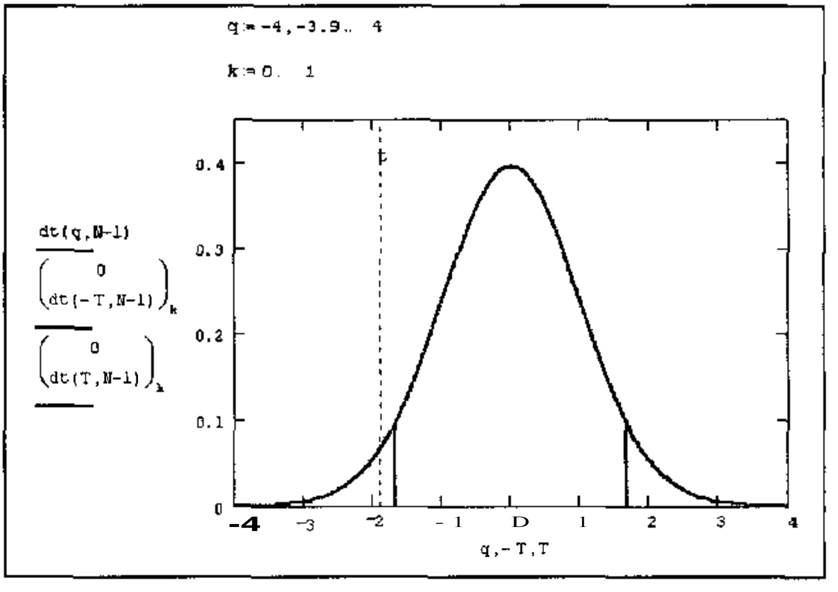


Рис. 14.16. К задаче проверки статистических гипотез (листинг 14.22)

Листинг 14.23. Другой вариант проверки гипотезы (продолжение листинга 14.22)

$$pt(t, N-1) = 0.033$$

$$\frac{\alpha}{2} < pt(t, N-1) < 1-\frac{\alpha}{2} = 0$$

Мы разобрали только два характерных примера статистических вычислений. Однако с помощью Mathcad легко решаются самые разнообразные задачи математической статистики.

Примечание

Большое количество задач разобрано в Ресурсах в рубрике **Statistics** (Статистика) справочной системы Mathcad **11**.

13 Зак. 984

ГЛАВА 15

Обработка данных

Когда Вы имеете дело с выборкой экспериментальных данных, то они, чаще всего, представляются в виде массива, состоящего из пар чисел (x_i, y_i). Поэтому возникает задача аппроксимации дискретной зависимости $y(x_i)$ непрерывной функцией f(x). Функция f(x), в зависимости от специфики задачи, может отвечать различным требованиям:

- f(x) должна проходить через точки (x_i, y_i), т.е. $f(x_i) = y_i, i = 1 \ldots n$. В этом случае *(см. разд. 15.1)* говорят об *интерполяции* данных функцией f (x) во внутренних точках между x_i, или *экстраполяции* за пределами интервала, содержащего все x_i;
- f (x) должна некоторым образом (например в виде определенной аналитической зависимости) приближать $y(x_i)$, не обязательно проходя через точки (x_i, y_i). Такова постановка задачи *регрессии (см. разд. 15.2),* которую во многих случаях также можно назвать *сглаживанием* данных;
- f (x) должна приближать экспериментальную зависимость $y(x_i)$, учитывая к тому же, что данные (x_i, y_i) получены с некоторой погрешностью, выражающей шумовую компоненту измерений. При этом функция f (x), с помощью того или иного алгоритма, уменьшает погрешность, присутствующую в данных (x_i, y_i). Такого типа задачи называют *задачами фильтрации (см. разд. 15.3).* Сглаживание — частный случай фильтрации.

Различные виды построения аппроксимирующей зависимости f (x) иллюстрирует рис. 15.1. На нем исходные данные обозначены кружками, интерполяция отрезками прямых линий — пунктиром, линейная регрессия — наклонной прямой линией, а фильтрация — жирной гладкой кривой. Эти зависимости приведены в качестве примера и отражают лишь малую часть возможностей Mathcad по обработке данных. Вообще говоря, в Mathcad имеется целый арсенал встроенных функций, позволяющий осуществлять самую различную регрессию, интерполяцию-экстраполяцию и сглаживание данных.

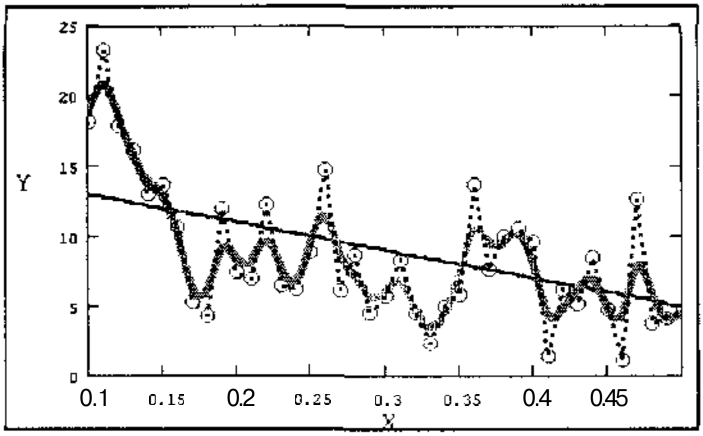


Рис. 15.1. Разные задачи аппроксимации данных

Как в целях подавления шума, так и для решения других проблем обработки данных, широко применяются различные интегральные преобразования. Они ставят в соответствие всей совокупности данных y(x} некоторую функцию другой координаты (или координат) F(ω). Примерами интегральных преобразований являются преобразование Фурье *(см. разд. 15.4.1)* и вейвлетное преобразование *(см. разд. 15.4.2)*. Напомним, что некоторые преобразования, например Фурье и Лапласа, можно осуществить в режиме символьных вычислений *(см. гл. 5)*. Каждое из интегральных преобразований эффективно для решения своего круга задач анализа данных.

15.1. Интерполяция

Для построения интерполяции-экстраполяции в Mathcad имеются несколько встроенных функций, позволяющих "соединить" точки выборки данных (x_i, y_i) кривой разной степени гладкости. По определению интерполяция означает построение функции A(X), аппроксимирующей зависимость y{x) в промежуточных точках (между x_i). Поэтому интерполяцию еще по-другому называют *аппроксимацией.* В точках x_i значения интерполяционной функции должны совпадать с исходными данными, т. е. $A(x_i)=y(x_i)$.

Примечание

Везде в этом разделе при рассказе о различных типах интерполяции будем использовать вместо обозначения A(X) другое имя ее аргумента A(t), чтобы не путать вектор данных x и скалярную переменную t.

15.1.1. Линейная интерполяция

Самый простой вид интерполяции — линейная, которая представляет искомую зависимость A{X} В виде ломаной линии. Интерполирующая функция A(x) состоит из отрезков прямых, соединяющих точки (рис. 15.2).

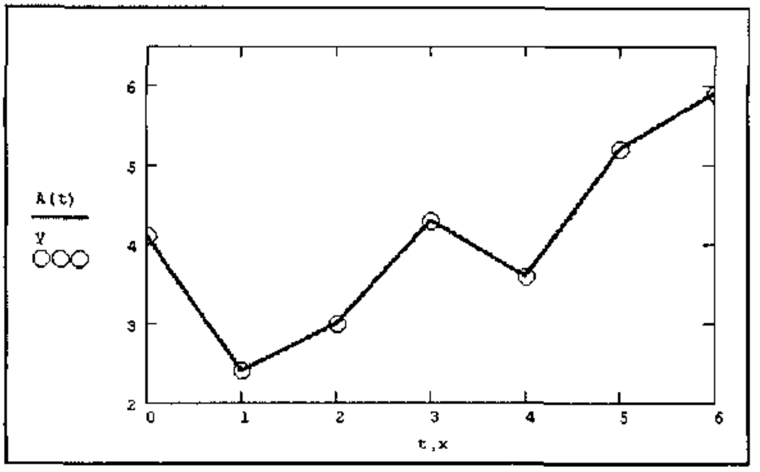


Рис. **15.2.** Линейная интерполяция (листинг 15.1)

Для построения линейной интерполяции служит встроенная функция `linterp` (листинг 15.1).

- `linterp(x,y,t)` — функция, аппроксимирующая данные векторов x и y кусочно-линейной зависимостью;
  - x — вектор действительных данных аргумента;
  - y — вектор действительных данных значений того же размера;
  - t — значение аргумента, при котором вычисляется интерполирующая функция.

Внимание!

Элементы вектора x должны быть определены в порядке возрастания, т. е. $x_1<x_2<x_3<\ldots<x_N$.

Листинг 15.1. Линейная интерполяция

```
x:=(0 1 2 3 4 5 6)^T
y:=(4.1 2.4 3 4.3 3.6 5.2 5.9)^T
A(t):=linterp(x,y,t)
```

Как видно из листинга, чтобы осуществить линейную интерполяцию, надо выполнить следующие действия:

1. Введите векторы данных x и y (первые две строки листинга).
2. Определите функцию `linterp(x,y,t)`.

3. Вычислите значения этой функции в требуемых точках, например `linterp(x,y,2.4)=3.52` или `linterp(x,y,6)=5.9`, или постройте ее график, как показано на рис. 15.2.

Примечание

Обратите внимание, что функция A(t) на графике имеет аргумент t, а не x. Это означает, что функция A(t) вычисляется не только при значениях аргумента (т. е. в семи точках), а при гораздо большем числе аргументов в интервале (0,6), что автоматически обеспечивает Mathcad. Просто в данном случае эти различия незаметны, т. к. при обычном построении графика функции A(x) от векторного аргумента x (рис. 15.3) Mathcad, по умолчанию, соединяет точки графика прямыми линиями (т. е. скрытым образом осуществляет их линейную интерполяцию).

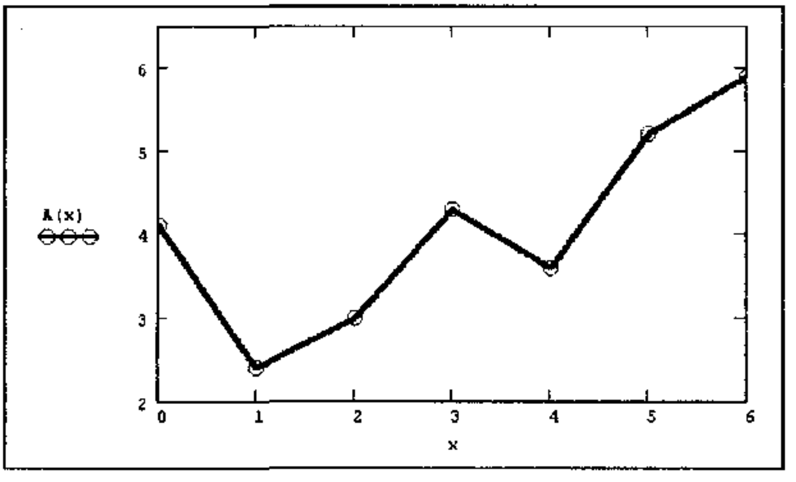


Рис. 15.3. Обычное построение графика функции от векторной переменной x (листинг 15.1)

15.1.2. Кубическая сплайн-интерполяция

В большинстве практических приложений желательно соединить экспериментальные точки не ломаной линией, а гладкой кривой. Лучше всего для этих целей подходит интерполяция кубическими сплайнами, т. е. отрезками кубических парабол (рис. 15.4).

- `interp(s,x,y,t)` — функция, аппроксимирующая данные векторов x и y кубическими сплайнами;
  - s — вектор вторых производных, созданный одной из сопутствующих функций `cspline`, `pspline` или `lspline`;
  - x — вектор действительных данных аргумента, элементы которого расположены в порядке возрастания;

- y — вектор действительных данных значений того же размера;
- t — значение аргумента, при котором вычисляется интерполирующая функция.

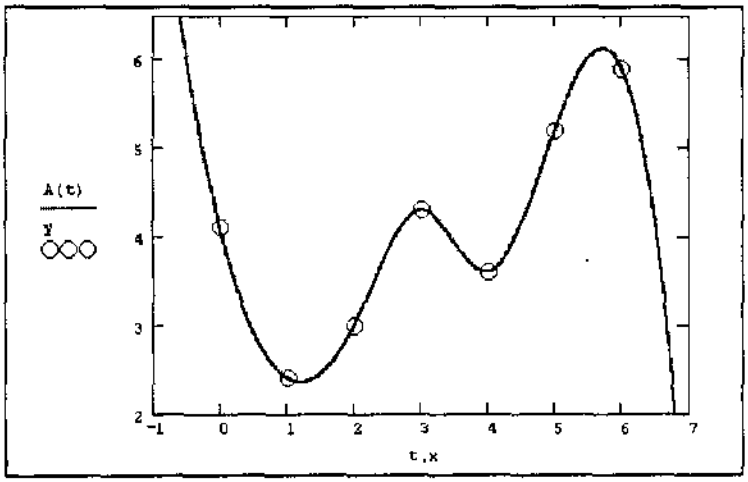


Рис. 15.4. Сплайн-интерполяция {см. листинг 15.2)

Сплайн-интерполяция в Mathcad реализована чуть сложнее линейной. Перед применением функции `interp` необходимо предварительно определить первый из ее аргументов — векторную переменную s. Делается это при помощи одной из трех встроенных функций тех же аргументов (x,y).

- `lspline(x,y)` — вектор значений коэффициентов линейного сплайна;
- `pspline(x,y)` — вектор значений коэффициентов квадратичного сплайна;
- `cspline(x,y)` — вектор значений коэффициентов кубического сплайна;
  - x,y — векторы данных.

Выбор конкретной функции сплайновых коэффициентов влияет на интерполяцию вблизи конечных точек интервала. Пример сплайн-интерполяции приведен в листинге 15.2.

Листинг 15.2. Кубическая сплайн-интерполяция

```
x := ( 0  1  2  3  4  5  6 )^T
y := ( 4.1  2.4  3  4.3  3.6  5.2  5.9 )^T
s := cspline (x, y)
A(t) := interp (s, x, y, t)
```

Смысл сплайн-интерполяции заключается в том, что в промежутках между точками осуществляется аппроксимация в виде зависимости $A(t)= =a \cdot t^3+b \cdot t^2+c \cdot t+d$. Коэффициенты a,b,c,d рассчитываются независимо для каждого промежутка, исходя из значений y_i в соседних точках. Этот процесс скрыт от пользователя, поскольку смысл задачи интерполяции состоит в выдаче значения A(t) в любой точке t (рис. 15.4).

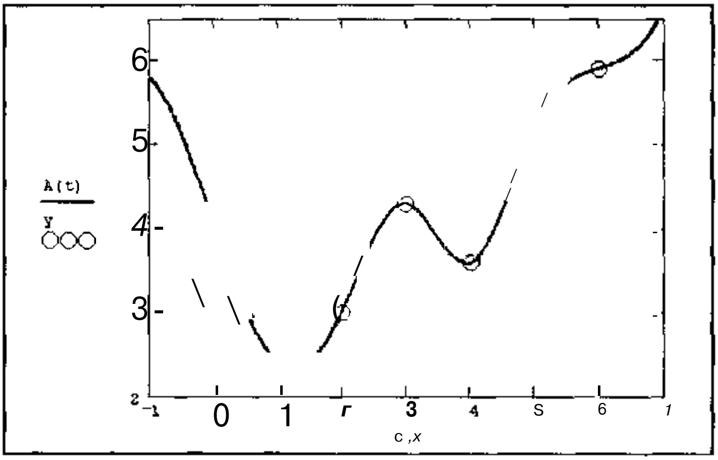


Рис. 15.5. Сплайн-интерполяция с выбором коэффициентов линейного сплайна lspline

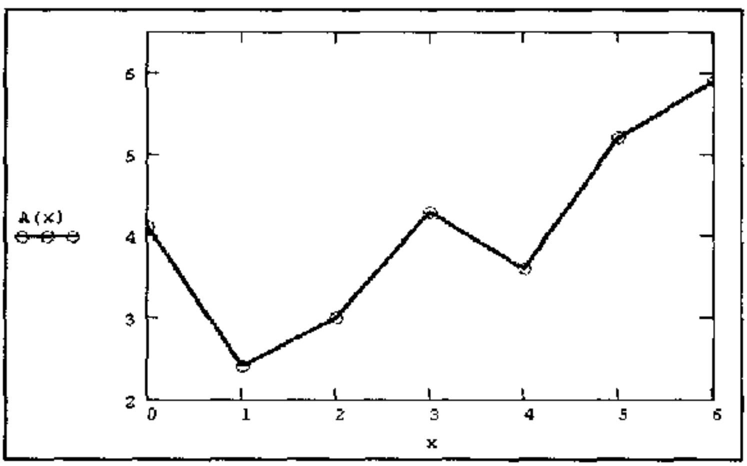


Рис. 15.6. Ошибочное построение графика сплайн-интерполяции (см. листинг 15.2)

Чтобы подчеркнуть различия, соответствующие разным вспомогательным функциям cspline, pspline, lspline, покажем результат действия листинга 15.2 при замене функции cspline в предпоследней строке на линейную lspline (рис. 15.5). Как видно, выбор вспомогательных функций существенно влияет на поведение A(t) вблизи граничных точек рассматриваемого интервала (0,6) и особенно разительно меняет результат экстраполяции данных за его пределами.

В заключение остановимся на уже упоминавшейся в предыдущем разделе распространенной ошибке при построении графиков интерполирующей функции (см. рис. 15.3). Если на графике, например являющемся продолжением листинга 15.2, задать построение функции A(X) вместо A(t), то будет получено просто соединение исходных точек ломаной (рис. 15.6). Так происходит потому, что в промежутках между точками вычисления интерполирующей функции не производятся.

15.1.3. Полиномиальная сплайн-интерполяция

Более сложный тип интерполяции — так называемая *интерполяция **B-сплайнами.*** В отличие от обычной сплайн-интерполяции *(см. разд. 15.1.2),* сшивка элементарных B-сплайнов производится не в точках x_i, а в других точках u_i, координаты которых предлагается ввести пользователю. Сплайны могут быть полиномами 1, 2 или з степени (линейные, квадратичные или кубические). Применяется интерполяция B-сплайнами точно так же, как и обычная сплайн-интерполяция, различие состоит только в определении вспомогательной функции коэффициентов сплайна.

- interp(s,x,y,t) — функция, аппроксимирующая данные векторов x и y с помощью B-сплайнов;
- bspline(x,y,u,n) — вектор значений коэффициентов B-сплайна;
  - s — вектор вторых производных, созданный функцией bspline;
  - x — вектор действительных данных аргумента, элементы которого расположены в порядке возрастания;
  - y — вектор действительных данных значений того же размера;
  - t — значение аргумента, при котором вычисляется интерполирующая функция;
  - u — вектор значений аргумента, в которых производится сшивка B-сплайнов;
  - п — порядок полиномов сплайновой интерполяции (1, 2 или з).

Примечание

Размерность вектора и должна быть на 1, 2 ИЛИ 3 меньше размерности векторов x и y. Первый элемент вектора u должен быть меньше ИЛИ равен первому элементу вектора x, а последний элемент ц — больше или равен последнему элементу x.

Интерполяция B-сплайнами иллюстрируется листингом 15.3 и рис. 15.7.

Листинг 15.3. Интерполяция B-сплайнами

```
x := ( 0  1  2  3  4  5  6 )T
y := ( 4.1  2.4  3  4.3  3.6  5.2  5.9 )T
и := <-0.5  2.2  3.3  4.1  5.5  7 )T
s := bspline ( x , y , и , 2 )
A ( t ) := interp( s , x , y , t )
```

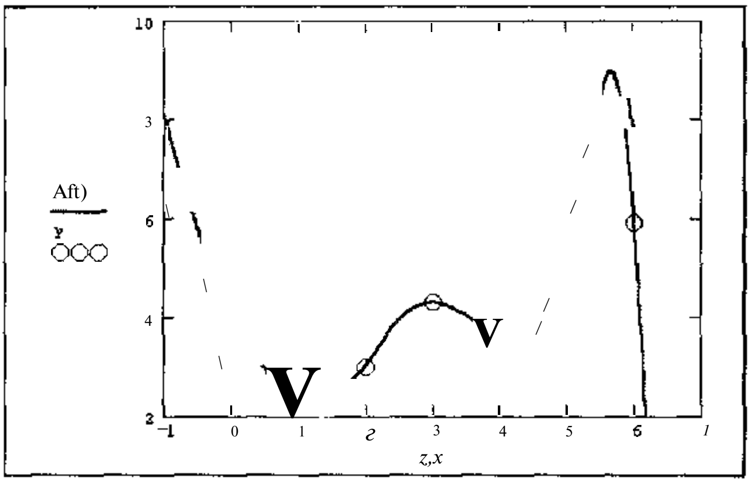


Рис. 15.7. B-сплайн-интерполяция (листинг 15.3)

15.1.4. Экстраполяция функцией предсказания

Все рассмотренные выше *(см. разд. 15.1.1—15.1.3)* функции осуществляли экстраполяцию данных за пределами их интервала с помощью соответствующей зависимости, основанной на анализе расположения нескольких исходных точек на границах интервала. В Mathcad имеется более развитый инструмент экстраполяции, который учитывает распределение данных вдоль всего интервала. В функцию predict встроен линейный алгоритм предсказания поведения функции, основанный на анализе, в том числе осцилляций.

- predict(y,m,n) — функция предсказания вектора, экстраполирующего выборку данных;
  - y — вектор действительных значений, взятых через равные промежутки значений аргумента;

- m — количество последовательных элементов вектора y, согласно которым строится экстраполяция;
- n — количество элементов вектора предсказаний.

Пример использования функции предсказания на примере экстраполяции осциллирующих данных y_j с меняющейся амплитудой приведен в листинге 15.4. Полученный график экстраполяции, наряду с самой функцией, показан на рис. 15.8. Аргументы и принцип действия функции predict отличаются от рассмотренных выше встроенных функций интерполяции-экстраполяции. Значений аргумента для данных не требуется, поскольку по определению функция действует на данные, идущие друг за другом с равномерным шагом. Обратите внимание, что результат функции predict вставляется "в хвост" исходных данных.

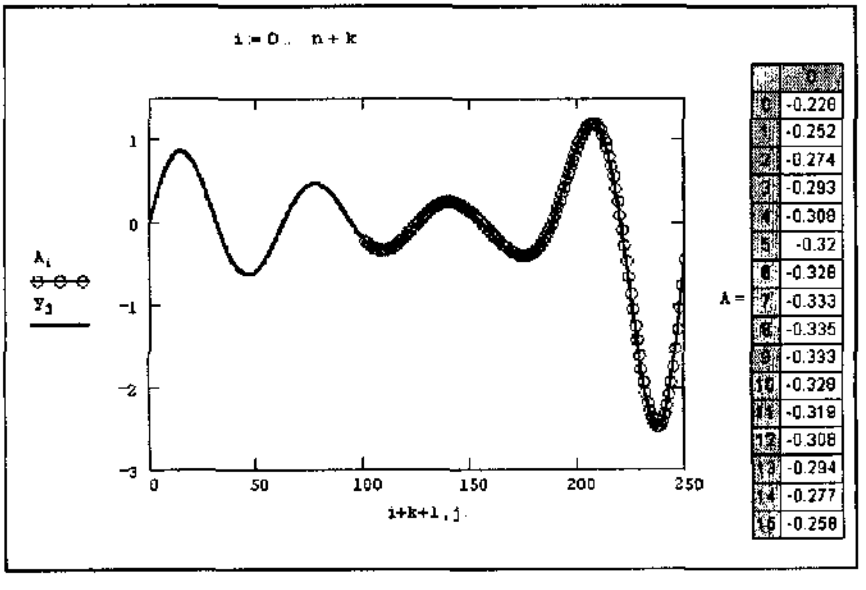
Листинг 15.4. Экстраполяция при помощи функции предсказания

$k := 100$ $\qquad$ $j := 0 .. k$

$$y_j := e^{\frac{-j}{100}} \cdot \sin\left(\frac{i}{10}\right)$$

$m := 50$ $\qquad$ $n := 150$

A := predict (y, m, n)



	0
0	-0.228
1	-0.252
2	-0.274
3	-0.293
4	-0.308
5	-0.32
6	-0.328
7	-0.333
8	-0.335
9	-0.333
10	-0.328
11	-0.319
12	-0.308
13	-0.294
14	-0.277
15	-0.258

Рис. 15.8. Экстраполяция при помощи функции предсказания (листинг 15.4)

Как видно из рис. 15.9, функция предсказания может быть полезна при экстраполяции данных на небольшие расстояния. Вдали от исходных данных результат часто бывает неудовлетворительным. Кроме того, функция predict хорошо работает в задачах анализа подробных данных с четко прослеживающейся закономерностью (типа рис. 15.8), в основном осциллирующего характера.

Если данных мало, то предсказание может оказаться бесполезным. В листинге 15.5 приведена экстраполяция небольшой выборки данных (из примеров, рассмотренных в предыдущих разделах). Соответствующий результат показан на рис. 15.9 для различных крайних точек массива исходных данных, для которых строится экстраполяция.

Листинг 15.5. Экстраполяция при помощи функции предсказания

```
y := ( 4.1  2.4  3  4.3  3.6  5.2  5.9 )^T
k := rows (y) - 1
га:=5                                    n := 10
A := predict {y, m, п}
```

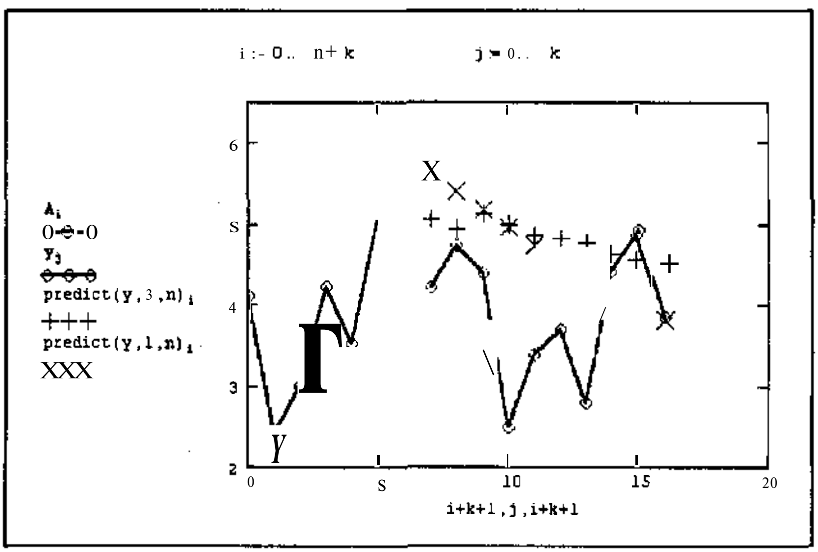


Рис. 15.9. Работа функции предсказания в случае малого количества данных (листинг 15.5)

15.1.5. Многомерная интерполяция

Двумерная сплайн-интерполяция приводит к построению поверхности z(x,y), проходящей через массив точек, описывающий сетку на координатной плоскости (x,y). Поверхность создается участками двумерных кубических сплайнов, являющихся функциями (x,y) и имеющих непрерывные первые и вторые производные по обеим координатам.

Многомерная интерполяция строится с помощью тех же встроенных функций, что и одномерная *(см. разд. 15.1.2),* но имеет в качестве аргументов не векторы, а соответствующие матрицы. Существует одно важное ограничение, связанное с возможностью интерполяции только квадратных NXN массивов данных.

- interp(S,X,Z,V) — скалярная функция, аппроксимирующая данные выборки двумерного поля по координатам x и y кубическими сплайнами;
  - s — вектор вторых производных, созданный одной из сопутствующих функций cspline, pspline или lspline;
  - x — матрица размерности NX2, определяющая диагональ сетки значений аргумента (элементы обоих столбцов соответствуют меткам x и y и расположены в порядке возрастания);
  - z — матрица действительных данных размерности NXN;
  - v — вектор из двух элементов, содержащий значения аргументов x и y, для которых вычисляется интерполяция.

Примечание

Вспомогательные функции построения вторых производных имеют те же матричные аргументы, что и interp: lspline(X,Y), pspline(X,Y), cspline(X,Y).

Пример исходных данных приведен на рис. 15.10 в виде графика линий уровня, программная реализация двумерной интерполяции показана в листинге 15.6, а ее результат — на рис. 15.11.

Листинг 15.6. Двумерная интерполяция

$$X := \begin{pmatrix} 0 & 0 \\ 1 & 10 \\ 2 & 20 \\ 3 & 30 \\ 4 & 40 \end{pmatrix} \qquad Y := \begin{pmatrix} 1 & 1 & 0 & 1.1 & 1.2 \\ 1 & 2 & 3 & 2.1 & 1.5 \\ 1.3 & 3.3 & 5 & 1.7 & 2 \\ 1.3 & 3 & 3.7 & 2.1 & 2.9 \\ 1.5 & 2 & 2.5 & 2.8 & 4 \end{pmatrix}$$

S:= cspline (X, Y)

$$V := \begin{pmatrix} 3.7 \\ 2.2 \end{pmatrix} \qquad \text{interp}(S, X, Y, V) = 1.636$$

k := 30

$i := 0 .. k \qquad j := 0 .. k$

$$A_{i,j} := \mathrm{interp}\left[S,X,Y,\begin{pmatrix}\dfrac{i}{k}\cdot 4\\ \dfrac{j}{k}\cdot 40\end{pmatrix}\right]$$

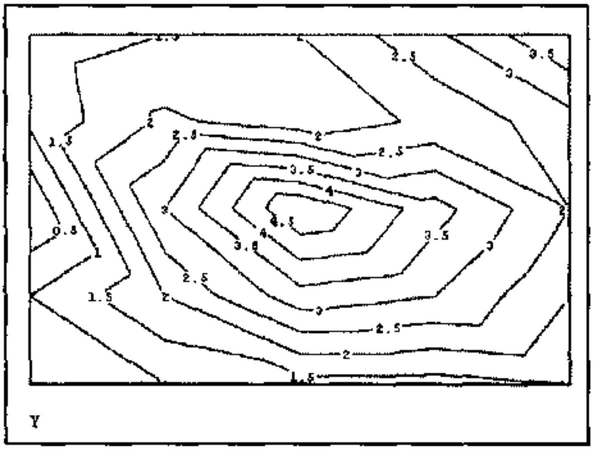


Рис. 15.10. Исходное двумерное поле данных (листинг 15.6)

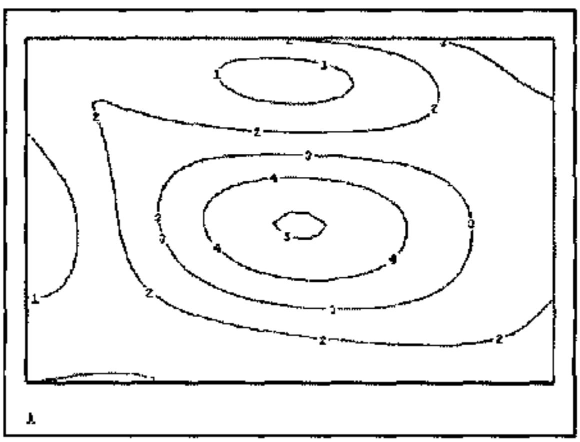


Рис. 15.11. Результат двумерной интерполяции (листинг 15.6)

15.2. Регрессия

Задачи математической рефессии имеют смысл приближения выборки данных (x_i, y_i) некоторой функцией $f(x)$, определенным образом минимизирующей совокупность ошибок $|f(x_i)-y_i|$. Регрессия сводится к подбору неизвестных коэффициентов, определяющих аналитическую зависимость f(x). В силу производимого действия большинство задач регрессии являются частным случаем более общей проблемы сглаживания данных.

Как правило, регрессия очень эффективна, когда заранее известен (или, по крайней мере, хорошо угадывается) закон распределения данных (x_i, y_i).

15.2.1. Линейная регрессия

Самый простой и наиболее часто используемый вид регрессии — линейная. Приближение данных (x_i, y_i) осуществляется линейной функцией $y(x)=b+a\cdot x$. На координатной плоскости (x,y) линейная функция, как известно, представляется прямой линией (рис. 15.12). Еще линейную регрессию часто называют *методом наименьших квадратов,* поскольку коэффициенты a и b вычисляются из условия минимизации суммы квадратов ошибок $|b+a\cdot x_i-y_i|$.

Примечание

Чаще всего такое же условие ставится и в других задачах регрессии, т. е. приближения массива данных (x_i, y_i) другими зависимостями y(x). Исключение рассмотрено в листинге 15.9.

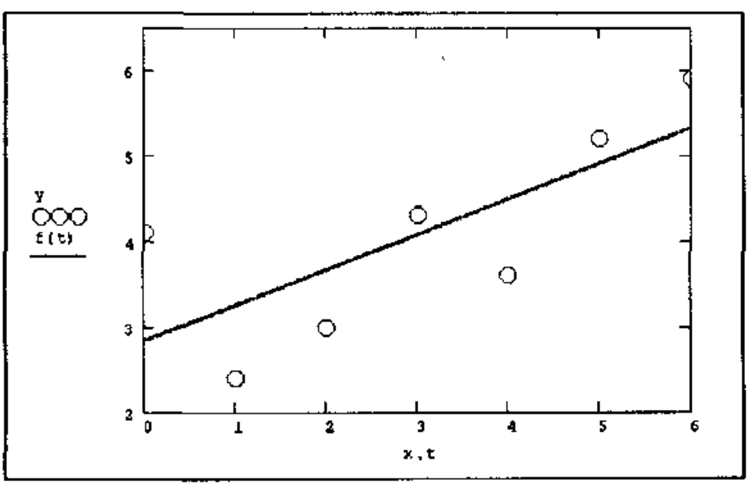


Рис. 15.12. Линейная регрессия (листинг 15.7 или 15.8)

Для расчета линейной регрессии в Mathcad имеются два дублирующих друг друга способа. Правила их применения представлены в листингах 15.7 и 15.8. Результат обоих листингов получается одинаковым (рис. 15.12).

- `line(x,y)` — вектор из двух элементов (b,a) коэффициентов линейной регрессии $b+a\cdot x$;

- intercept(x,y) — коэффициент b линейной регрессии;
- slope(x,y) — коэффициент a линейной регрессии;
  - x — вектор действительных данных аргумента;
  - y — вектор действительных данных значений того же размера.

Листинг 15.7. Линейная регрессия

```
x:= ( 0  1  2  3  4  5  6 )^T
y:= ( 4.1  2.4  3  4.3  3.6  5.2  5.9 )^T
```

$$\mathrm{line}(x,y) = \begin{pmatrix} 2.829 \\ 0.414 \end{pmatrix}$$

$$f(t) := \mathrm{line}(x,y)_0 + \mathrm{line}(x,y)_1 \cdot t$$

Листинг 15.8. Другая форма записи линейной регрессии

```
x := ( 0  1  2  3  4  5  6 )^T
y := ( 4.1  2.4  3  4.3  3.6  5.2  5.9 )^T
intercept (x, y) = 2.829
slope (x, y) = 0.414
f(t) := intercept (x, y) + slope (x, y) · t
```

В Mathcad имеется альтернативный алгоритм, реализующий не минимизацию суммы квадратов ошибок, а медиан-медиэнную линейную регрессию для расчета коэффициентов a и b (листинг 15.9).

- medfit(x,y) — вектор из двух элементов (b,a) коэффициентов линейной медиан-медианной регрессии b+a·x;
  - x,y — векторы действительных данных одинакового размера.

Листинг 15.9. Построение линейной регрессии двумя разными методами (*продолжение листинга 15.7*)

$$\mathrm{medfit}(x,y) = \begin{pmatrix} 2.517 \\ 0.55 \end{pmatrix}$$

$$g(t) := \mathrm{medfit}(x,y)_0 + \mathrm{medfit}(x,y)_1 \cdot t$$

Различие результатов среднеквадратичной и медиан-медианной регрессии иллюстрируется рис. 15.13.

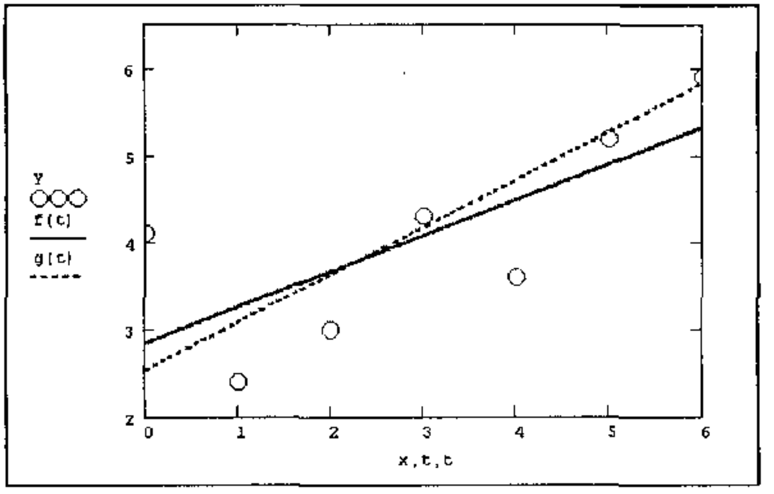


Рис. 15.13. Линейная регрессия по методу наименьших квадратов и методу медиан (листинги 15.7 и 15.9)

15.2.2. Полиномиальная регрессия

В Mathcad реализована регрессия одним полиномом, отрезками нескольких полиномов, а также двумерная регрессия массива данных.

Полиномиальная регрессия

Полиномиальная регрессия означает приближение данных (x_i, y_i) полиномом k-й степени $A(x)=a+b\cdot x+c\cdot x^2+d\cdot x^3+...+h\cdot x^k$ (рис. 15.14). При к=1 полином является прямой линией, при к=2 — параболой, при к=з — кубической параболой и т. д. Как правило, на практике применяются k<5.

> **Примечание**
>
> Для построения регрессии полиномом к-й степени необходимо наличие по крайней мере (k+1) точек данных.

В Mathcad полиномиальная регрессия осуществляется комбинацией встроенной функции regress и полиномиальной интерполяции *(см. разд. 15.1.2).*

- `regress(x,y,k)` — вектор коэффициентов для построения полиномиальной регрессии данных;
- interp(s,x,y, t) — результат полиномиальной регрессии;
  - `s=regress(x,y,k)`;
  - x — вектор действительных данных аргумента, элементы которого расположены в порядке возрастания;
  - y — вектор действительных данных значений того же размера;

- к — степень полинома регрессии (целое положительное число);
- t — значение аргумента полинома регрессии.

Внимание!

Для построения полиномиальной регрессии после функции regress Вы обязаны использовать функцию interp.

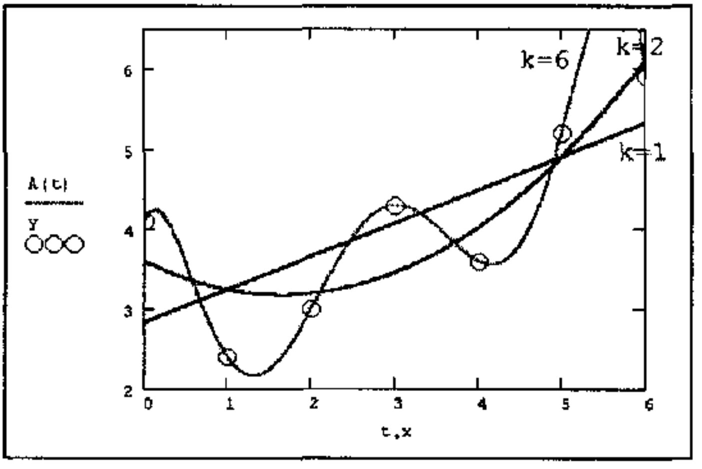


Рис. 15.14. Регрессия полиномами разной степени (коллаж результатов листинга 15.10 для разных к)

Пример полиномиальной регрессии квадратичной параболой приведен в листинге 15.10.

Листинг 15.10. Полиномиальная регрессия

```
x := (0   1   2   3   4   5   6)T
y := (4.1   2.4   3   4.3   3.6   5.2   5.9)T
к:=2
s := regress (x, y, к)
A(t) :=interp (s, x, y, t)
```

Регрессия отрезками полиномов

Помимо приближения массива данных одним полиномом имеется возможность осуществить регрессию сшивкой отрезков (точнее говоря, участков, т. к. они имеют криволинейную форму) нескольких полиномов. Для этого имеется встроенная функция loess, применение которой аналогично функции regress (листинг 15.11 и рис. 15.15).

- loess(x,y,span) — вектор коэффициентов для построения регрессии данных отрезками полиномов;
- interp(s,x,y,t) — результат полиномиальной регрессии;
  - s=loess(x,y,span);
  - x — вектор действительных данных аргумента, элементы которого расположены в порядке возрастания;
  - y — вектор действительных данных значений того же размера;
  - span — параметр, определяющий размер отрезков полиномов (положительное число, хорошие результаты дает значение порядка span=0.75).

Параметр span задает степень сглаженности данных. При больших значениях span регрессия практически не отличается от регрессии одним полиномом (например, span=2 дает почти тот же результат, что и приближение точек параболой).

Листинг 15.11. Регрессия отрезками полиномов

$x:=(0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6)^T$

$y:=(4.1\ \ 2.4\ \ 3\ \ 4.3\ \ 3.6\ \ 5.2\ \ 5.9)^T$

s := loess (x, y, 0.9)

A (t) := interp (s,x,y, t)

Совет

Регрессия одним полиномом эффективна, когда множество точек выглядит как полином, а регрессия отрезками полиномов оказывается полезной в противоположном случае.

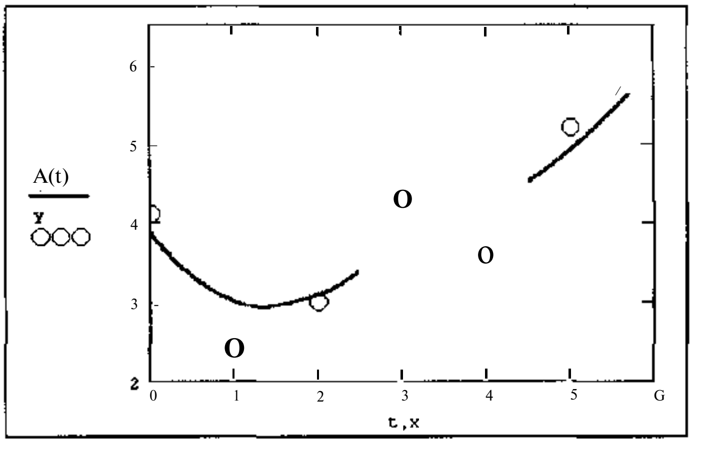


Рис. 15.15. Регрессия отрезками полиномов (листинг 15.11)

Двумерная полиномиальная регрессия

По аналогии с одномерной полиномиальной регрессией и двумерной интерполяцией *(см. разд. 15.1.5)* Mathcad позволяет приблизить множество точек $z_{i,j}(x_i, y_j)$ поверхностью, которая определяется многомерной полиномиальной зависимостью. В качестве аргументов встроенных функций для построения полиномиальной регрессии должны стоять в этом случае не векторы, а соответствующие матрицы.

- `regress(X,Z,k)` — вектор коэффициентов для построения полиномиальной регрессии данных;
- `loess(X,Z,span)` — вектор коэффициентов для построения регрессии данных отрезками полиномов;
- `interp(S,X,Z,V)` — скалярная функция, аппроксимирующая данные выборки двумерного поля по координатам x и y кубическими сплайнами;
  - s — вектор вторых производных, созданный одной из сопутствующих функций loess или regress;
  - x — матрица размерности N×2, определяющая пары значений аргумента (столбцы соответствуют меткам x и y);
  - z — вектор действительных данных размерности N;
  - span — параметр, определяющий размер отрезков полиномов;
  - k — степень полинома регрессии (целое положительное число);
  - v — вектор из двух элементов, содержащий значения аргументов x и y, для которых вычисляется интерполяция.

Внимание!

Для построения регрессии не предполагается никакого предварительного упорядочивания данных (как, например, для двумерной интерполяции, которая требует их представления в виде матрицы NXN). В связи с этим данные представляются как вектор.

Двумерная полиномиальная регрессия иллюстрируется листингом 15.12 и рис. 15.16. Сравните стиль представления данных для двумерной регрессии с представлением тех же данных для двумерной сплайн-интерполяции (см. листинг 15.6) и ее результаты с исходными данными (см. рис. 15.10) и их сплайн-интерполяцией (см. рис. 15.11).

Листинг 15.12. Двумерная полиномиальная регрессия

$$X:=\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 \\ 0 & 10 & 20 & 30 & 40 & 0 & 10 & 20 & 30 & 40 & 0 & 10 & 20 & 30 & 40 & 0 & 10 & 20 & 30 & 40 & 0 & 10 & 20 & 30 & 40 \end{pmatrix}^T$$

```
Z:=(1 1 0 1.11.21 2 3 2.11.51.33.35 1.72 1.33 3.72.12.91.52 2.52.84)^T
S := regress (X, Z , 3)
k := 30
i := 0 .. k                    j := 0 .. k
```

$$A_{i,j} := \mathrm{interp}\left[S, X, Z, \begin{pmatrix} \frac{i}{K} \cdot 4 \\ \frac{j}{K} \cdot 40 \end{pmatrix}\right]$$

Примечание

Обратите внимание на знаки транспонирования в листинге. Они применены для корректного представления аргументов (например z в качестве вектора, а не строки).

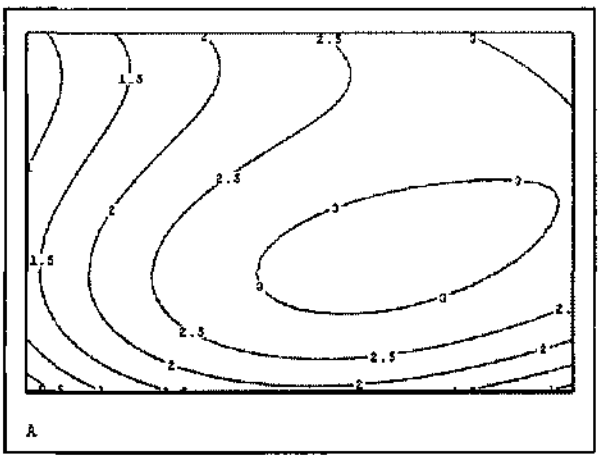


Рис. 15.16. Двумерная полиномиальная регрессия (листинг 15.12)

15.2.3. Регрессия специального вида

Кроме рассмотренных, в Mathcad встроено еще несколько видов трехпараметрической регрессии. Их реализация несколько отличается от приведенных выше вариантов регрессии тем, что для них, помимо массива данных, требуется задать некоторые начальные значения коэффициентов `a,b,c`. Используйте соответствующий вид регрессии, если хорошо представляете себе, какой зависимостью описывается Ваш массив данных. Когда тип регрессии плохо отражает последовательность данных, то ее результат часто бывает неудовлетворительным и даже сильно различающимся в зависимости от выбора начальных значений. Каждая из функций выдает вектор уточненных параметров a,b,c.

- expfit(x,y,g) — регрессия экспонентой $f(x)=a \cdot e^{bx}+c$;
- lgsfit(x,y,g) — регрессия логистической функцией $f(x)=a/(1+b \cdot e^{-cx})$;
- sinfit(x,y,g) — регрессия синусоидой $f(x)=a \cdot \sin(x+b)+c$;
- pwfit(x,y,g) — регрессия степенной функцией $f(x)=a \cdot x^b+c$;
- logfit(x,y,g) — регрессия логарифмической функцией $f(x)=a \cdot \ln(x+b)+c$;
- lnfit(x,y) — регрессия двухпараметрической логарифмической функцией $f(x)=a \cdot \ln(x)+b$;
  - x — вектор действительных данных аргумента;
  - y — вектор действительных значений того же размера;
  - g — вектор из трех элементов, задающий начальные значения a,b,c.

Примечание

Правильность выбора начальных значений можно оценить по результату регрессии — если функция, выданная Mathcad, хорошо приближает зависимость y(x), значит они были подобраны удачно.

Пример расчета одного из видов трехпараметрической регрессии (экспоненциальной) приведен в листинге 15.13 и на рис. 15.17. В предпоследней строке листинга выведены в виде вектора вычисленные коэффициенты a,b,c, а в последней строке через эти коэффициенты определена искомая функция f(x).

Листинг 15.13. Экспоненциальная регрессия

$x := (0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6)^T$

$y := (4.1\ \ 2.4\ \ 3\ \ 4.3\ \ 3.6\ \ 5.2\ \ 5.9)^T$

$$g := \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

C := expfit(x, y, д)

$$C = \begin{pmatrix} 0.111 \\ 0.544 \\ 3.099 \end{pmatrix}$$

$f(t) := C_0 \cdot \exp(C_1 \cdot t) + C_2$

Примечание

Многие задачи регрессии данных различными двухпараметрическими зависимостями y(x) можно свести к более надежной, с вычислительной точки зрения, линейной регрессии. Делается это с помощью соответствующей замены переменных.

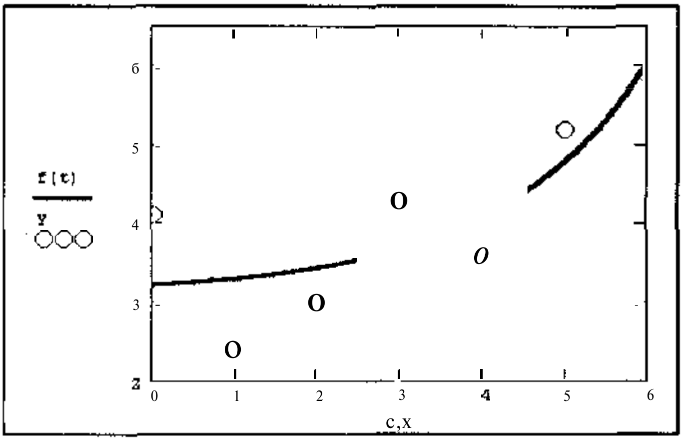


Рис. 15.17. Экспоненциальная регрессия (листинг 15.13)

15.2.4. Регрессия общего вида

В Mathcad можно осуществить регрессию в виде линейной комбинации $C_1f_1(x)+C_2f_2(x)+\ldots$, где $f_i(x)$ — любые функции пользователя, а C_i — подлежащие определению коэффициенты. Кроме того, имеется путь проведения регрессии более общего вида, когда комбинацию функций и искомых коэффициентов задает сам пользователь.

Приведем встроенные функции для регрессии общего вида и примеры их использования (листинги 15.14 и 15.15), надеясь, что читатель при необходимости найдет более подробную информацию об этих специальных возможностях в справочной системе и **Mathcad Resources.**

- `linfit(x,y,F)` — вектор параметров линейной комбинации функций пользователя, осуществляющей регрессию данных;
- `genfit(x,y,g,G)` — вектор параметров, реализующих регрессию данных с помощью функций пользователя общего вида;
  - x — вектор действительных данных аргумента, элементы которого расположены в порядке возрастания;
  - y — вектор действительных значений того же размера;
  - F(X) — пользовательская векторная функция скалярного аргумента;
  - g — вектор начальных значений параметров регрессии размерности N;

* G(x,C) — векторная функция размерности N+1, составленная из функции пользователя и ее N частных производных по каждому из параметров c.

Листинг 15.14. Регрессия линейной комбинацией функций пользователя

$x := (0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6)^T$

$y := (4.1\ \ 2.4\ \ 3\ \ 4.3\ \ 3.6\ \ 5.2\ \ 5.9)^T$

$$F(x) := \begin{pmatrix} \frac{1}{x+1} \\ x \\ e^x \end{pmatrix}$$

$C := \mathrm{linfit}(x, y, F)$

$$C = \begin{pmatrix} 3.957 \\ 0.854 \\ 5.605 \times 10^{-4} \end{pmatrix}$$

$$f(x) := \frac{C_0}{x+1} + C_1 \cdot x + C_2 \cdot e^x$$

Листинг 15.15. Регрессия общего вида

$x := (0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6)^T$

$y := (4.1\ \ 2.4\ \ 3\ \ 4.3\ \ 3.6\ \ 5.2\ \ 5.9)^T$

$$g := \begin{pmatrix} 0.1 \\ 1 \end{pmatrix} \qquad G(x, C) := \begin{pmatrix} C_0 \cdot x^{C_1 \cdot x} \\ x^{C_1 \cdot x} \\ C_0 \cdot x^{C_1 \cdot x + 1} \end{pmatrix}$$

$C := \mathrm{genfit}(x, y, g, G)$

$$C = \begin{pmatrix} 3.168 \\ 0.057 \end{pmatrix}$$

$$f(x) := C_0 \cdot x^{C_1 \cdot x}$$

15.3. Сглаживание и фильтрация

При анализе данных часто возникает задача их фильтрации, заключающаяся в устранении одной из составляющих зависимости $y(x_i)$. Наиболее часто целью фильтрации является подавление быстрых вариаций $y(x_i)$, которые чаще всего обусловлены шумом. В результате из быстроосциллирующей зависимости $y(x_i)$ получается другая, сглаженная зависимость, в которой доминирует более низкочастотная составляющая.

Наиболее простыми и эффективными рецептами сглаживания (smoothing) можно считать регрессию различного вида *(см. разд. 15.2)*. Однако регрессия часто уничтожает информативную составляющую данных, оставляя лишь наперед заданную пользователем зависимость.

Часто рассматривают противоположную задачу фильтрации — устранение медленно меняющихся вариаций в целях исследования высокочастотной составляющей. В этом случае говорят о задаче устранения тренда. Иногда интерес представляют смешанные задачи выделения среднемасштабных вариаций путем подавления как более быстрых, так и более медленных вариаций. Одна из возможностей решения связана с применением полосовой фильтрации.

Несколько примеров программной реализации различных вариантов фильтрации приведены в данном разделе.

15.3.1. Встроенные функции для сглаживания

В Mathcad имеется несколько встроенных функций, реализующих различные алгоритмы сглаживания данных.

- `medsmooth(y,b)` — сглаживание алгоритмом "бегущих медиан";
- `ksmooth(x,y,b)` — сглаживание на основе функции Гаусса;
- `supsmooth(x,y)` — локальное сглаживание адаптивным алгоритмом, основанное на анализе ближайших соседей каждой пары данных;
  - `x` — вектор действительных данных аргумента (для `supsmooth` его элементы должны быть расположены в порядке возрастания);
  - у — вектор действительных значений того же размера, что и x;
  - `b` — ширина окна сглаживания.

Все функции имеют в качестве аргумента векторы, составленные из массива данных, и выдают в качестве результата вектор сглаженных данных того же размера. Функция medsmooth предполагает, что данные расположены равномерно.

Примечание

Подробную информацию об алгоритмах, заложенных в функции сглаживания, Вы найдете в справочной системе Mathcad в статье **Smoothing** (Сглаживание), находящейся в разделе **Statistics** (Статистика)..

Часто бывает полезным совместить сглаживание с последующей интерполяцией или регрессией. Соответствующий пример приведен в листинге 15.16

для функции supsmooth. Результат работы листинга показан на рис. 15.18 (кружки обозначают исходные данные, крестики — сглаженные, пунктирная кривая — результат сплайн-интерполяции). Сглаживание тех же данных при помощи "бегущих медиан" и функции Гаусса с разным значением ширины окна пропускания показаны на рис. 15.19 и 15.20, соответственно.

Листинг 15.16. Сглаживание с последующей сплайн-интерполяцией

```
x:=(0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16)^T
y:=(4.1  2.4  3  4.3  3.6  5.2  5.9  5  4.7  4  3.5  3.9  3  2.7  3.7  4.8  5.4)^T
z := supsmooth (x, y)
s := cspline (x, z)
A(t) := interp (s, x, z, t)
```

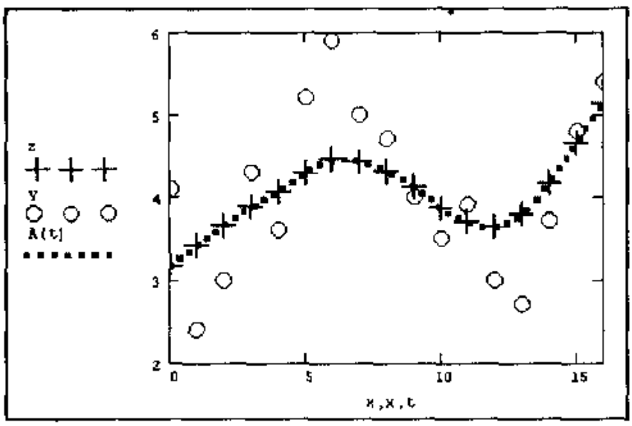


Рис. 15.18. Адаптивное сглаживание (листинг 15.16)

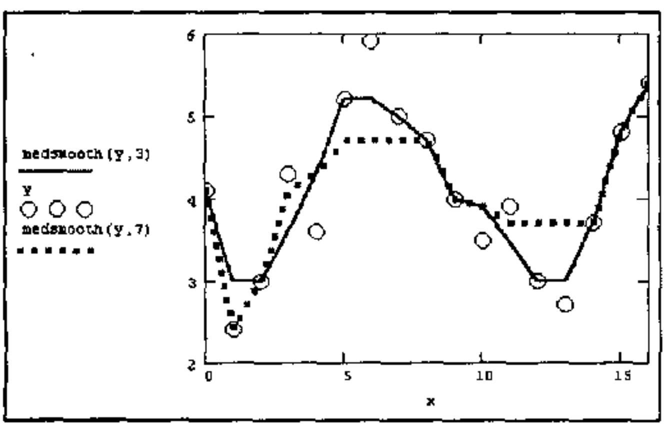


Рис. 15.19. Сглаживание "бегущими медианами"

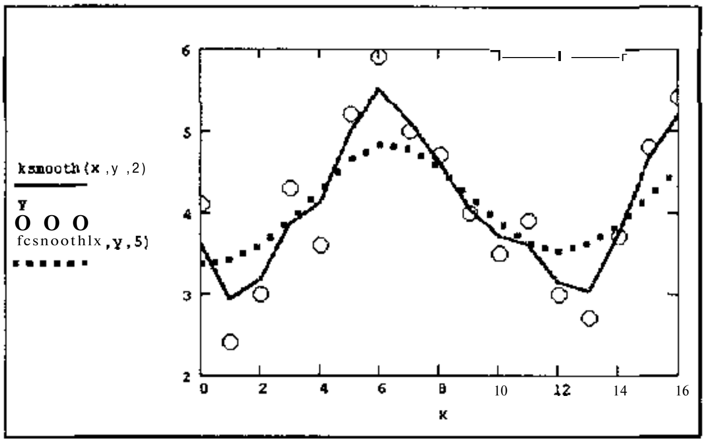


Рис. 15.20. Сглаживание при помощи функции ksmooth

15.3.2. Скользящее усреднение

Помимо встроенных в Mathcad, существует несколько популярных алгоритмов сглаживания, на одном из которых хочется остановиться особо. Самый простой и очень эффективный метод — это скользящее усреднение. Его суть состоит в расчете для каждого значения аргумента среднего значения по соседним w данным. Число w называют *окном* скользящего усреднения; чем оно больше, тем больше данных участвуют в расчете среднего, тем более сглаженная кривая получается. На рис. 15.21 показан результат скользящего усреднения одних и тех же данных (кружки) с разным окном w=3

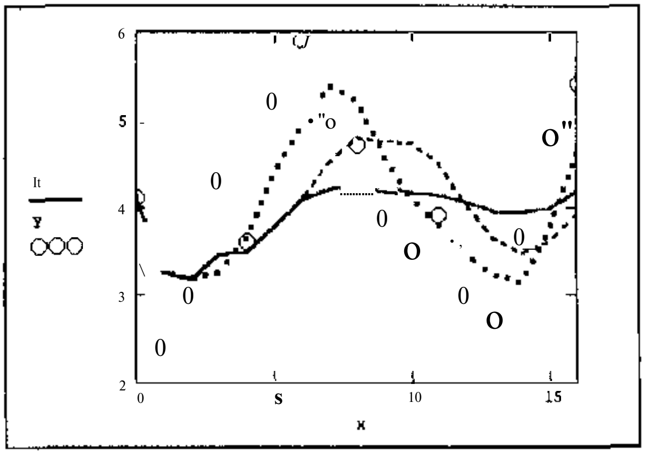


Рис. 15.21. Скользящее усреднение с разными w=3 , 5,15
(листинг 15.17, коллаж трех графиков)

(пунктир), w=5 (штрихованная кривая) и w=15 (сплошная кривая). Видно, что при малых w сглаженные кривые практически повторяют ход изменения данных, а при больших w — отражают лишь закономерность их медленных вариаций.

Чтобы реализовать в Mathcad скользящее усреднение, достаточно очень простой программы, приведенной в листинге 15.17. Она использует только значения y, оформленные в виде вектора, неявно предполагая, что они соответствуют значениям аргумента x, расположенным через одинаковые промежутки. Вектор x применялся лишь для построения графика результата (рис. 15.21).

Листинг 15.17. Сглаживание скользящим усреднением

$x := (0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8\ \ 9\ \ 10\ \ 11\ \ 12\ \ 13\ \ 14\ \ 15\ \ 16)^T$

$y := (4.1\ \ 2.4\ \ 3\ \ 4.3\ \ 3.6\ \ 5.2\ \ 5.9\ \ 5\ \ 4.7\ \ 4\ \ 3.5\ \ 3.9\ \ 3\ \ 2.7\ \ 3.7\ \ 4.8\ \ 5.4)^T$

$w := 15$

$N := \mathrm{rows}(y) \qquad N = 17$

$i := 0 .. N-1$

$$m_i := \mathrm{if}\left(i < w,\ \frac{\sum_{j=0}^{i} y_j}{i+1},\ \frac{\sum_{j=i-w+1}^{i} y_j}{w}\right)$$

Примечание

Приведенная программная реализация скользящего усреднения самая простая, но не самая лучшая. Возможно, Вы обратили внимание, что все кривые скользящего среднего на рис. 15.21 слегка "обгоняют" исходные данные. Почему так происходит, понятно: согласно алгоритму, заложенному в последнюю строку листинга 15.17, скользящее среднее для каждой точки вычисляется путем усреднения значений предыдущих w точек. Чтобы результат скользящего усреднения был более адекватным, лучше применить центрированный алгоритм расчета по w/2 предыдущим и w/2 последующим значениям. Он будет немного сложнее, поскольку придется учитывать недостаток точек не только в начале (как это сделано в программе с помощью функции условия if), но и в конце массива исходных данных.

15.3.3. Устранение тренда

Еще одна типичная задача возникает, когда интерес исследований заключается не в анализе медленных (или *низкочастотных)* вариаций сигнала y(x) (для чего применяется сглаживание данных), а в анализе быстрых его изменений. Часто бывает, что быстрые (или *высокочастотные)* вариации накладываются определенным образом на медленные, которые обычно называют *трендом.* Часто тренд имеет заранее предсказуемый вид, например линейный. Чтобы устранить тренд, можно предложить последовательность действий, реализованную в листинге 15.18.

1. Вычислить регрессию f{x}, например линейную, исходя из априорной информации о тренде (предпоследняя строка листинга).
2. Вычесть из данных y (x) тренд ffx) (последняя строка листинга).

Листинг 15.18. Устранение тренда

```
x := (0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16)ᵀ
y := (5.1 4.4 5.7 5.3 5.6 5.2 5.9 7.7 6.7 7 6.5 5.9 8.1 8.7 9.7 9.8 10.4)ᵀ
N := rows(y)                    N = 17
i := 0 .. N - 1
f(t) := line(x,y)0 + line(x,y)1 · t
mi := yi - f(xi)
```

На рис. 15.22 показаны исходные данные (кружками), выделенный с помощью регрессии линейный тренд (сплошной прямой линией) и результат устранения тренда (пунктир, соединяющий крестики).

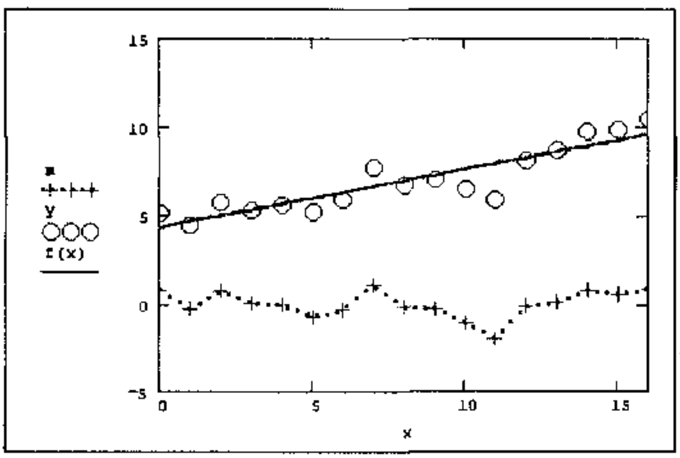


Рис. 15.22. Устранение тренда (листинг 15.18)

15.3.4. Полосовая фильтрация

В предыдущих разделах была рассмотрена фильтрация быстрых вариаций сигнала (сглаживание) и его медленных вариаций (снятие тренда). Иногда требуется выделить среднемасштабную составляющую сигнала, уменьшив как более быстрые, так и более медленные его компоненты. Одна из возможностей решения этой задачи связана с применением полосовой фильтрации на основе последовательного скользящего усреднения.

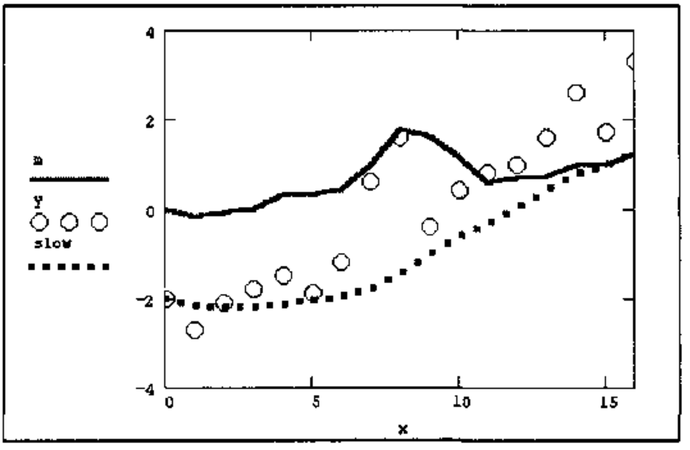


Рис. 15.23. Результат полосовой фильтрации (листинг 15.19)

Алгоритм полосовой фильтрации приведен в листинге 15.19, а результат его применения показан на рис. 15.23 сплошной кривой. Алгоритм реализует такую последовательность операций:

1. Приведение массива данных y к нулевому среднему значению путем его вычитания из каждого элемента y (третья и четвертая строки листинга).
2. Устранение из сигнала y высокочастотной составляющей, имеющее целью получить сглаженный сигнал middle, например с помощью скользящего усреднения с малым окном w (в листинге 15.19 w=3).
3. Выделение из сигнала middle низкочастотной составляющей slow, например, путем скользящего усреднения с большим окном w (в листинге 15.19 w=7) либо с помощью снятия тренда *(см. разд. 15.3.3)*.
4. Вычитание из сигнала middle тренда slow (последняя строка листинга), тем самым выделяя среднемасштабную составляющую исходного сигнала y.

Листинг 15.19. Полосовая фильтрация

```
x:=(0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)T
y:=(5.1 4.4 5 5.3 5.6 5.2 5.9 7.7 8.7 6.7 7.5 7.9 8.1 8.7 9.7 8.8 10.4)T
meanY :=mean (y)
```

$y := y - meanY$

$N := rows(y)$ $N=17$

$i := 0 .. N - 1$

$w := 3$

$$middle_i := if\left(i < w, \frac{\sum_{j=0}^{i} y_j}{i+1}, \frac{\sum_{j=i-w+1}^{i} y_j}{w}\right)$$

$w := 7$

$$slow_i := if\left(i < w, \frac{\sum_{j=0}^{i} middle_j}{i+1}, \frac{\sum_{j=i-w+1}^{i} middle_j}{w}\right)$$

$m := middle - slow$

15.4. Интегральныепреобразования

Интегральные преобразования массива сигнала y(x) ставят в соответствие всей совокупности данных y(x) некоторую функцию другой координаты F(V). Рассмотрим встроенные функции для расчета интегральных преобразований, реализованных в Mathcad.

15.4.1. Преобразование Фурье

Математический смысл преобразования Фурье состоит в представлении сигнала y(x) в виде бесконечной суммы синусоид вида Ffv) sin{vx). Функция Ffv) называется *преобразованием Фурье* или *интегралом Фурье,* или *Фурье-спектром* сигнала. Ее аргумент v имеет смысл частоты соответствующей составляющей сигнала. Обратное преобразование Фурье переводит спектр F(V) в исходный сигнал y(x}. Согласно определению,

$$F(v) = \int_{-\infty}^{\infty} y(x) \cdot exp(-ivx)dx.$$

Как видно, преобразование Фурье является существенно комплексной величиной, даже если сигнал действительный.

Преобразование Фурье действительных данных

Преобразование Фурье имеет огромное значение для различных математических приложений, и для него разработан очень эффективный алгоритм, называемый *БПФ* (быстрым преобразованием Фурье). Этот алгоритм реализован в нескольких встроенных функциях Mathcad, различающихся нормировками.

- `fft(y)` — вектор прямого преобразования Фурье;
- `FFT(y)` — вектор прямого преобразования Фурье в другой нормировке;
- `ifft(v)` — вектор обратного преобразования Фурье;
- `IFFT(V)` — вектор обратного преобразования Фурье в другой нормировке;
  - y — вектор действительных данных, взятых через равные промежутки значений аргумента;
  - v — вектор действительных данных Фурье-спектра, взятых через равные промежутки значений частоты.

Внимание!

Аргумент прямого Фурье-преобразования, т. е. вектор y, должен иметь ровно 2^n элементов (n — целое число). Результатом является вектор с $1+2^{n-1}$ элементами. И наоборот, аргумент обратного Фурье-преобразования должен иметь $1+2^{n-1}$ элементов, а его результатом будет вектор из 2^n элементов. Если число данных не совпадает со степенью 2, то необходимо дополнить недостающие элементы нулями.

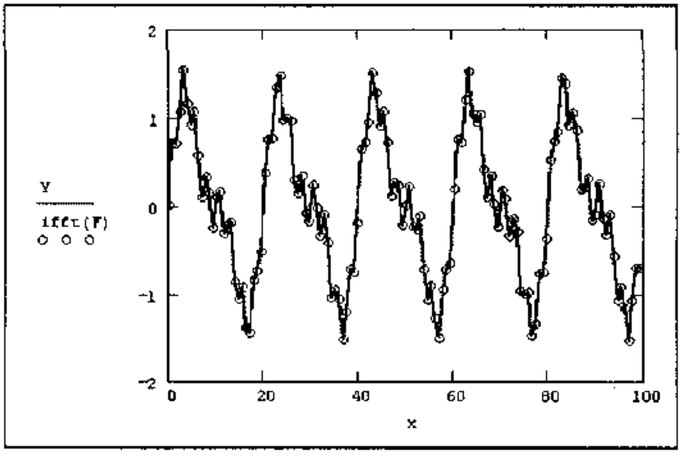


Рис. 15.24. Исходные данные и обратное преобразование Фурье (листинг 15.20)

Пример расчета Фурье-спектра для суммы трех синусоидальных сигналов разной амплитуды (показанных в виде сплошной кривой на рис. 15.24), приведен в листинге 15.20. Расчет проводится по N=128 точкам, причем предполагается, что интервал дискретизации данных y_i равен Д. В предпоследней строке листинга применяется встроенная функция ifft, а в последней корректно определяются соответствующие значения частот Ω_i. Обратите внимание, что результаты расчета представляются в виде модуля Фурье-спектра (рис. 15.25), поскольку сам спектр является комплексным. Очень полезно сравнить полученные амплитуды и местоположение пиков спектра с определением синусоид в листинге 15.20.

Листинг 15.20. Быстрое преобразование Фурье

$N := 128$

$xMAX := 100 \qquad \Delta := \frac{xMAX}{N}$

$i := 0 .. N-1$

$x_i := i \cdot \Delta$

$y_i := \sin(2 \cdot \pi \cdot 0.05 \cdot x_i) + 0.5 \cdot \sin(2 \cdot \pi \cdot 0.1 \cdot x_i) + 0.25 \cdot \sin(2 \cdot \pi \cdot 0.4 \cdot x_i)$

$F := fft(y)$

$\Omega_i := (i+1) \cdot \frac{1}{xMAX}$

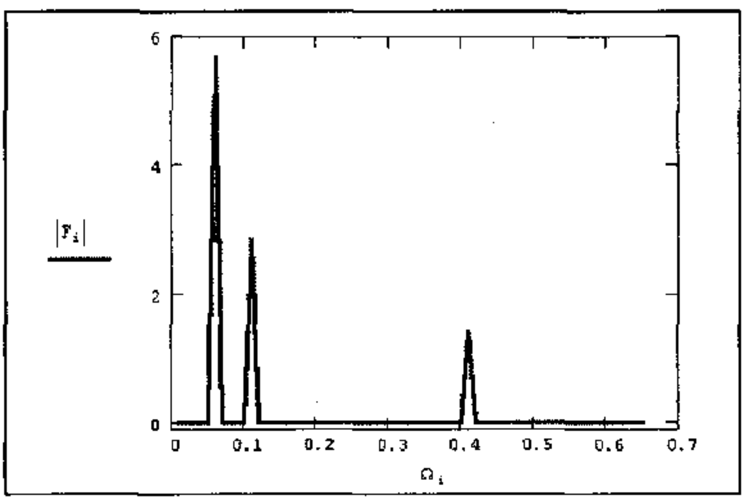


Рис. 15.25. Преобразование Фурье {листинг 15.20}

Результат обратного преобразования Фурье показан в виде кружков на том же рис. 15.24, что и исходные данные. Видно, что в рассматриваемом случае сигнал y(x) восстановлен с большой точностью, что характерно для плавного изменения сигнала.

14 Зак. 984

Преобразование Фурье комплексных данных

Алгоритм быстрого преобразования Фурье для комплексных данных встроен в соответствующие функции, в имя которых входит литера "с".

- `cfft(y)` — вектор прямого комплексного преобразования Фурье;
- CFFT(y) — вектор прямого комплексного преобразования Фурье в другой нормировке;
- `icfft(y)` — вектор обратного комплексного преобразования Фурье;
- `ICFFT(v)` — вектор обратного комплексного преобразования Фурье в другой нормировке;
  - y — вектор данных, взятых через равные промежутки значений аргумента;
  - v — вектор данных Фурье-спектра, взятых через равные промежутки значений частоты.

Функции действительного преобразования Фурье используют тот факт, что в случае действительных данных спектр получается симметричным относительно нуля, и выводят только его половину *(см. выше разд. "Преобразование Фурье действительных данных" этой главы)*. Поэтому, в частности, по 128 действительным данным получалось всего 65 точек спектра Фурье. Если к тем же данным применить функцию комплексного преобразования Фурье (рис. 15.26), то получится вектор из 128 элементов. Сравнивая рис. 15.25 и 15.26, можно уяснить соответствие между результатами действительного и комплексного Фурье-преобразования.

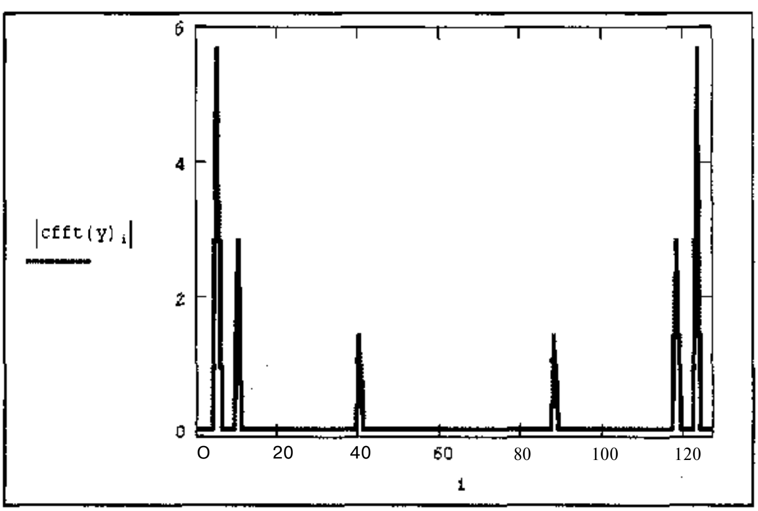


Рис. 15.26. Комплексное преобразование Фурье (продолжение листинга 15.20)

Двумерное преобразование Фурье

В Mathcad имеется возможность применять встроенные функции комплексного преобразования Фурье не только к одномерным, но и к двумерным массивам, т. е. матрицам. Соответствующий пример приведен в листинге 15.21 и на рис. 15.27 в виде графика линий уровня исходных данных и рассчитанного Фурье-спектра.

Листинг 15.21. Двумерное преобразование Фурье

$N := 64$

$i := 0 .. N - 1 \qquad j := 0 .. N - 1$

$$y_{i,j} := \sin\left(\frac{i+j}{10}\right) + \sin\left(\frac{i-j}{10}\right)$$

$F := CFFT(y)$

$F := submatrix(F, 7, N-7, 7, N-7)$

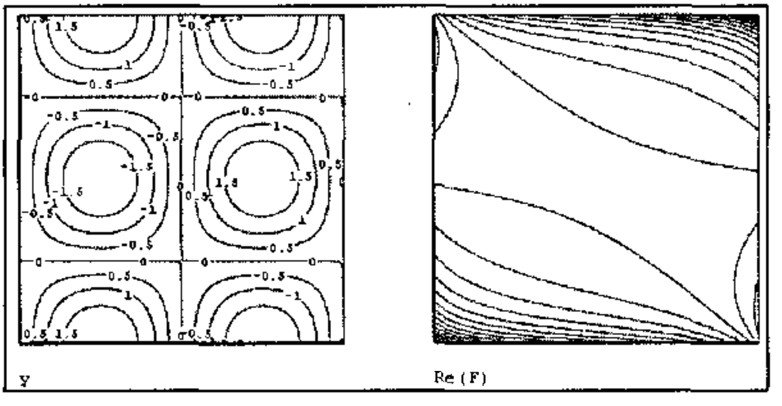


Рис. 15.27. Данные (слева) и их Фурье-спектр (справа) (листинг 15.21)

15.4.2. Вейвлетное преобразование

В последнее время возрос интерес к другим интегральным преобразованиям, в частности *вейвлетному* (или *дискретному волновому)* преобразованию. Оно применяется, главным образом, для анализа нестационарных сигналов и для многих задач подобного рода оказывается более эффективным, чем преобразование Фурье. Основным отличием вейвлетного преобразования является разложение данных не по синусоидам (как для преобразования Фурье), а по другим функциям, называемым *вейвлетобразующими*. Вейвлетобразуюшие функции, в противоположность бесконечно осциллирующим синусоидам, локализованы в некоторой ограниченной области своего аргумента, а вдали от нее равны нулю или ничтожно малы. Пример такой функции, называемой "мексиканской шляпой", показан на рис. 15.28.

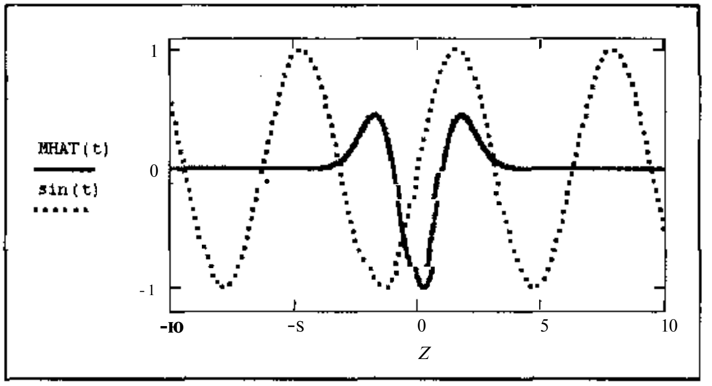


Рис. 15.28. Сравнение синусоиды и вейвлетобразующей функции

Из-за своего математического смысла вейвлет-спектр имеет не один аргумент, а два. Помимо частоты, вторым аргументом ь является место локализации вейвлетобразующей функции. Поэтому Ь имеет ту же размерность, что и х.

Встроенная функция вейвлет-преобразования

Mathcad имеет одну встроенную функцию для расчета вейвлет-преобразования на основе вейвлетобразующей функции Даубечи.

- wave{y) — вектор прямого вейвлет-преобразования;
- `iwave(v)` — вектор обратного вейвлет-преобразования;
  - у — вектор данных, взятых через равные промежутки значений аргумента;
  - v — вектор данных вейвлет-спектра.

Аргумент функции вейвлет-преобразования, т. е. вектор у, должен так же, как и в преобразовании Фурье, иметь ровно 2^n элементов (п — целое число). Результатом функции wave является вектор, скомпонованный из нескольких коэффициентов с двухпараметрического вейвлет-спектра. Использование функции wave объясняется на примере анализа суммы двух синусоид в листинге 15.22, а три семейства коэффициентов вычисленного вейвлет-спектра показаны на рис. 15.29.

Листинг 15.22. Поиск вейвлет-спектра Даубечи

$$f(t) := \sin\left(2\cdot\pi\frac{t}{50\ j}\right)+0.3\cdot\sin\left(2\cdot\pi\frac{t}{10}\right)$$

$Nmax := 256$ $\qquad$ $i := 0\,..\,Nmax - 1$

$y_i := f(i)$

$W := wave(y)$

$Nlevels := \frac{\ln(Nmax)}{\ln(2)} - 1 \qquad Nlevels = 7$

$k := 1, 2 .. Nlevels$

$coeffs(level) := submatrix\left(W, 2^{level}, 2^{level+1} - 1, 0, 0\right)$

$c_{i,k} := coeffs(k)_{floor\left[\frac{i}{\left(\frac{Nmax}{2^k}\right)}\right]}$

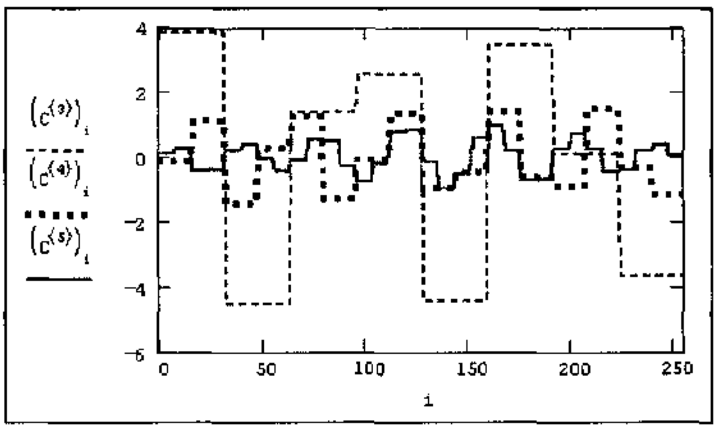


Рис. 15.29. Вейвлет-спектр на основе функции Даубечи (листинг 15.22)

Программирование других вейвлет-преобразований

Помимо встроенной функции вейвлет-спектра Даубечи и возможностей пакета расширения Mathcad 11, возможно непосредственное программирование алгоритмов пользователя для расчета вейвлет-спектров. Оно сводится к аккуратному расчету соответствующих семейств интегралов. Один из примеров такой программы приведен в листинге 15.23, а ее результат на рис. 15.30. Анализу подвергается та же функция, составленная из суммы двух синусов, а график двухпараметрического спектра c{a,Ъ) выведен в виде привычных для вейвлет-анализа линий уровня на плоскости (a,Ь).

Листинг **15.23.** Поиск вейвлет-спектра на основе **"мексиканской шляпы"**

$f(t) := \sin\left(2\cdot\pi\frac{t}{50}\right) + 0.3\cdot\sin\left(2\cdot\pi\frac{t}{10}\right)$

$$MHAT(t) := \frac{d^2}{dt^2}\exp\left(\frac{-t^2}{2}\right)$$

$Nmax := 256$

$$\psi(a, b, t) := MHAT\left(\frac{t-b}{a}\right)$$

$$W(a, b) := \int_{-Nmax}^{Nmax} \psi(a, b, t)\cdot f(t)\, dt$$

$i := 0 .. 10$ $\qquad$ $b := 0, 1 .. \frac{Nmax}{10}$

$$a_i := \frac{(i+15)^4}{2\times 10^4}$$

$$N_{i,b} := W\left[(a_i), 2\cdot b - \frac{Nmax}{10}\right]$$

Примечание

Программа листинга очень проста, но исключительно далека от хорошей в смысле быстродействия. Каждый интеграл вычисляется независимо, без использования методов ускорения, типа применяемых в алгоритме БПФ.

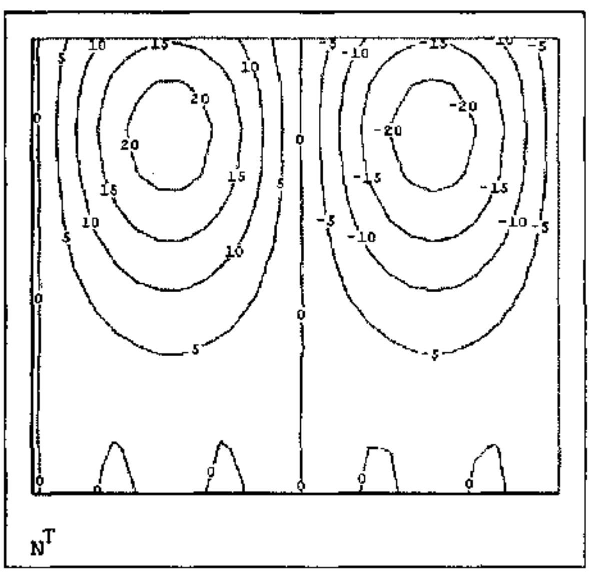
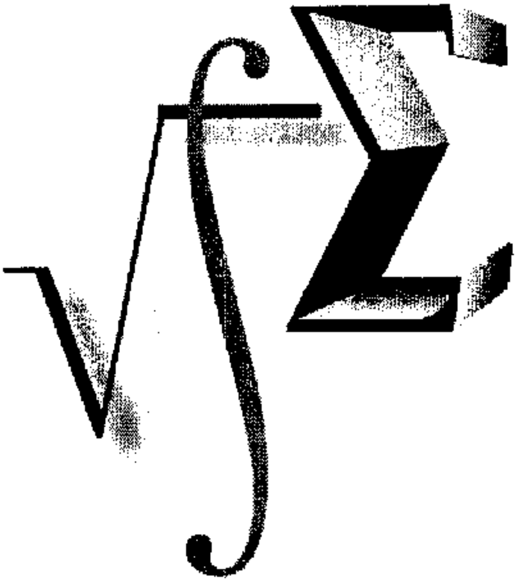


Рис. 15.30. Вейвлет-спектр на основе "мексиканской шляпы" (листинг 15.23}



ЧАСТЬ IV

ОФОРМЛЕНИЕ РАСЧЕТОВ

ГЛАВА 16

Ввод-вывод данных

В данной главе рассматриваются вопросы ввода входных данных в документы Mathcad и вывода результатов вычислений. В начале главы приведено краткое напоминание о числовом вводе-выводе значений, перечисляются типы данных, которые применимы в среде Mathcad *(см. разд. 16.1)* при определении переменных и функций.

Наиболее мощными средствами вывода результатов в Mathcad являются графики, и именно их эффективному использованию посвящено основное содержание главы *(см. разд. 16.2—16.4)*.

На применении динамической смены графиков основан аппарат создания видеофайлов анимации *(см. разд. 16.5),* который делает результаты работы в Mathcad особенно эффектными. К тому же, Mathcad обладает целым спектром возможностей по вводу-выводу данных во внешние текстовые и графические файлы *(см. разд. 16.6)*.

16.1. Числовой ввод-вывод

Наиболее простой и распространенный ввод-вывод данных в Mathcad реализован присваиванием и (либо численным, либо символьным) выводом непосредственно в документе. Фактически документ Mathcad является одновременно и кодом программы и результатом ее выполнения. Поэтому самый простой и распространенный способ ввода-вывода — это непосредственное присвоение и вывод вычисленных значений в документах. Числовому вводу и выводу данных посвящена практически вся глава 4 (о вводе данных *см. разд. "Типы данных", "Размерные переменные", "Массивы" гл. 4,* об управлении формой вывода — *разд. "Формат вывода числовых данных" гл. 4),* поэтому ограничимся напоминанием об этом важном элементе системы Mathcad (листинги 16.1, 16.2).

Листинг 16.1. Числовой ввод данных

$i := 0 .. 4$

$x := 1.5257285$

$y := 1234.567890$

$$A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

$f(x) := x^2$

Листинг 16.2. Числовой вывод данных *(продолжение листинга 16.1)*

$x = 1.526$ $\qquad f(x) = 2.328$

$y = 1234.568$ $\qquad y = 1.235 \times 10^3$

$$A = \begin{pmatrix} \; & 2 \\ \; & 4 \end{pmatrix} \qquad f(A) = \begin{pmatrix} 1 & 10 \\ 15 & 22 \end{pmatrix}$$

$A_{1,0} = 3 \quad (3\;\; 4)$

$i =$

0
1
2
3
4

$f(i) =$

0
1
4
9
16

16.2. Создание графиков

В Mathcad встроено несколько различных типов графиков *(см. рисунки этой главы),* которые можно разбить на две большие группы.

- Двумерные графики:
  - XY (декартовый) график (XY Plot);
  - полярный график (Polar Plot).
- Трехмерные графики:
  - график трехмерной поверхности (Surface Plot);
  - график линий уровня (Contour Plot);
  - трехмерная гистограмма (3D Bar Plot);
  - трехмерное множество точек (3D Scatter Plot);
  - векторное поле (Vector Field Plot).

Деление графиков на типы несколько условно, т. к., управляя установками многочисленных параметров, можно создавать комбинации типов графиков, а также новые типы (например двумерная гистограмма распределения является разновидностью простого XY-графика).

Все графики создаются совершенно одинаково, с помощью панели инструментов Graph (График), различия обусловлены отображаемыми данными.

Внимание!

Некорректное определение данных приводит, вместо построения графика, к выдаче сообщения об ошибке.

Чтобы создать график, например двумерный Декартов:

1. Поместите курсор ввода в то место документа, куда требуется вставить график.
2. Если на экране нет панели Graph (График), вызовите ее нажатием кнопки с изображением графиков на панели Math (Математика).
3. Нажмите на панели Graph (График) кнопку X-Y Plot для создания Декартового графика (рис. 16.1) или другую кнопку для иного желаемого типа графика.
4. В результате в обозначенном месте документа появится пустая область графика с одним или несколькими местозаполнителями (рис. 16.1, слева). Введите в местозаполнители имена переменных или функций, которые должны быть изображены на графике. В случае Декартова графика это два местозаполнителя данных, откладываемых по осям x и Y.

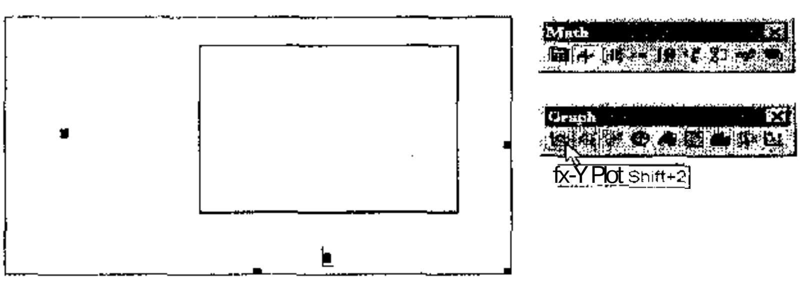


Рис. 16.1. Создание Декартового графика при помощи панели Graph

Если имена данных введены правильно, нужный график появится на экране. Созданный график можно изменить, в том числе меняя сами данные, форматируя его внешний вид или добавляя дополнительные элементы оформления.

Примечание

Правильному заданию данных и форматированию графиков посвящены соответствующие разделы этой главы.

Самый наглядный способ создания графика — с помошью панели инструментов **Graph** (График). Однако точно так же создаются графики путем выбора соответствующего элемента подменю **Insert / Graph** (Вставка / График), показанного на рис. 16.2, либо нажатием соответствующей типу графика горячей клавиши.

Чтобы удалить график, щелкните в его пределах и выберите в верхнем меню **Edit** (Правка) пункт **Cut** (Вырезать) или **Delete** (Удалить).

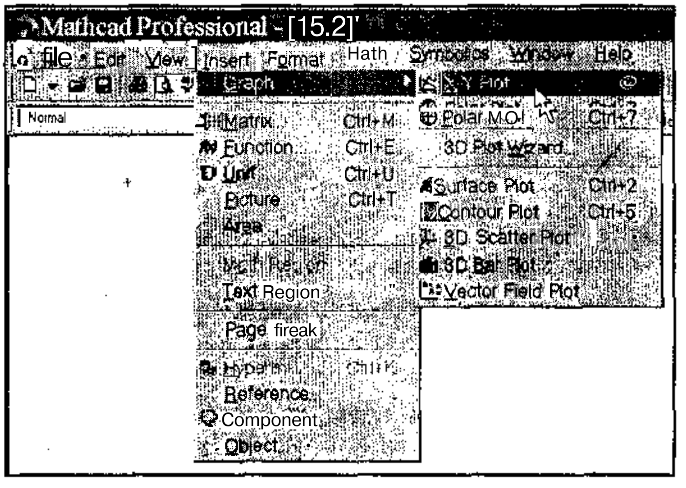


Рис. 16.2. Создание графика посредством меню

16.3. Двумерные графики

К двумерным графикам относят графики в Декартовой и полярной системах координат. Созданный однажды график одного типа нельзя переделать в график другого типа (в отличие от трехмерных фафиков). Для построения XY-графика необходимы два ряда данных, откладываемых по осям x и Y.

16.3.1. XY-график двух векторов

Самый простой и наглядный способ получить Декартов график — это сформировать два вектора данных, которые будут отложены вдоль осей x и Y. Последовательность построения графика двух векторов x и y показана на рис. 16.3. В этом случае в местозаполнители возле осей вводятся просто имена векторов. Также допускается откладывать по осям элементы векторов, т. е. вводить в местозаполнители возле осей имена x_i и y_i, соответственно (рис. 16.4). В результате получается график, на котором отложены точки, соответствующие парам элементов векторов, соединенные отрезками прямых линий. Образованная ими ломаная называется *рядом данных* или *кривой* (trace).

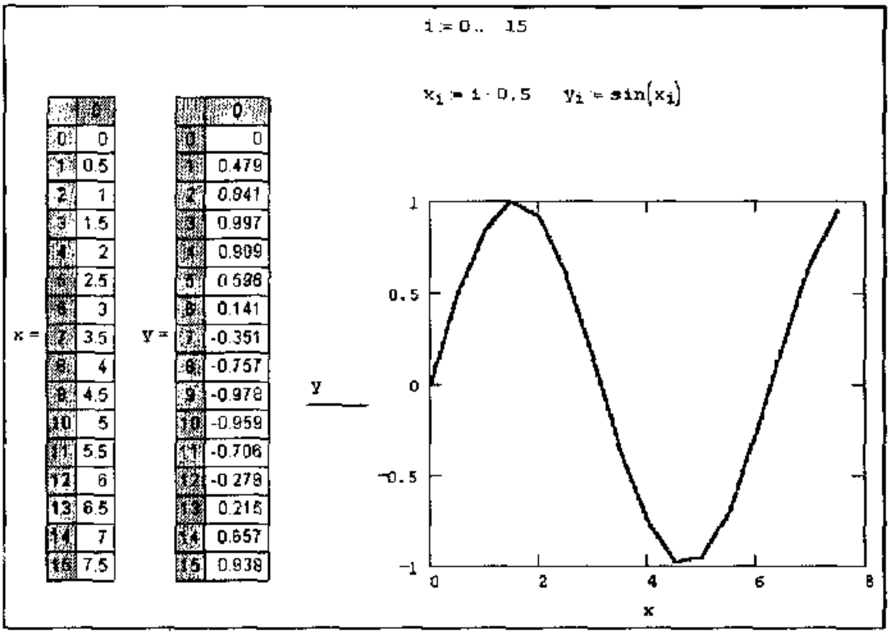


Рис. 16.3. XY-график двух векторов

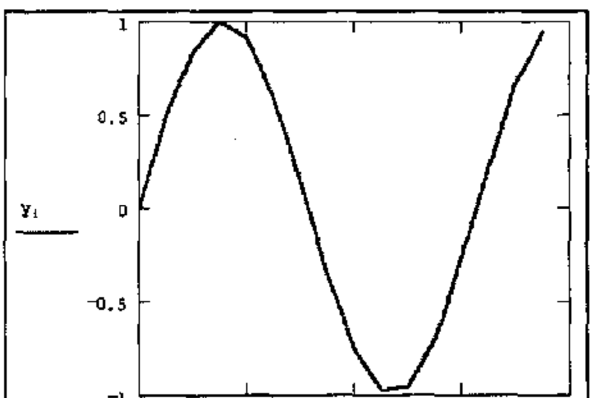


Рис. **16.4.** XY-график двух векторов, заданных элементами

Примечание

Обратите внимание, что Mathcad автоматически определяет границы графика, исходя из диапазона значений элементов векторов.

Стоит отметить, что подобным образом легко создать и XY-график столбцов или строк матрицы, применяя оператор выделения столбца и откладывая соответствующие выражения по осям графика *(множество подобных примеров Вы найдете на рисунках гл. 11 и 12).*

16.3.2. XY-график вектора и ранжированной переменной

В качестве переменных, откладываемых по любой из осей, можно использовать саму ранжированную переменную (рис. 16.5). При этом по другой оси должно быть отложено либо выражение, явно содержащее саму ранжированную переменную, либо элемент вектора с индексом по этой ранжированной переменной, но никак не сам вектор.

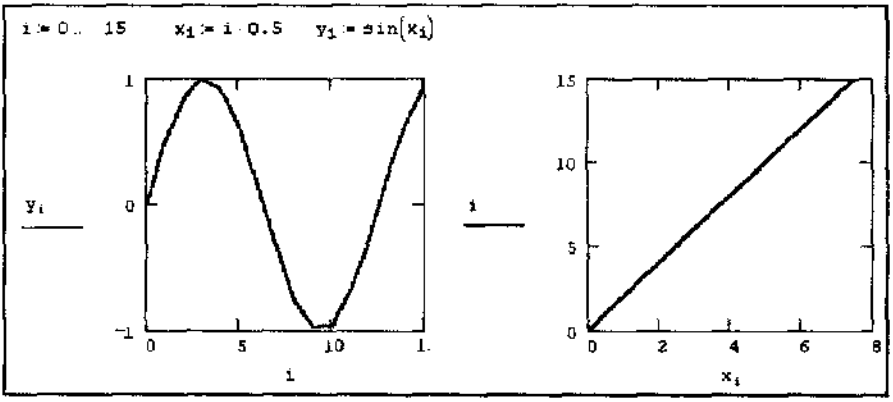


Рис. 16.5. Графики вектора и ранжированной переменной

16.3.3. XY-график функции

Нарисовать график любой скалярной функции f (x) можно двумя способами. Первый заключается в дискретизации значений функции, присвоении этих значений вектору и прорисовке графика вектора. Собственно, так и были получены графики синуса на рис. 16.3—16.5. Второй, более простой способ, называемый *быстрым построением графика,* заключается во введении функции в один из местозаполнителей (например у оси у), а имени аргумента — в местозаполнитель у другой оси (рис. 16.6). В результате Mathcad сам создает график функции в пределах значений аргумента, по умолчанию принятых равными от —10 до 10. Разумеется, впоследствии можно поменять диапазон значений аргумента, и график автоматически подстроится под него.

Необходимо заметить, что если переменной аргумента функции было присвоено некоторое значение до построения в документе графика, то вместо быстрого построения графика будет нарисована зависимость функции с учетом этого значения. Примеры двух таких графиков приведены на рис. 16.7.

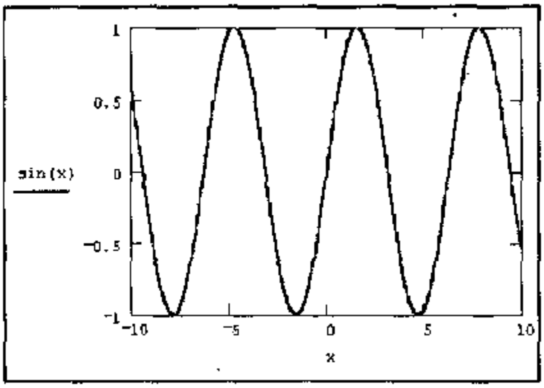


Рис. 16.6. Быстрое построение графика функции

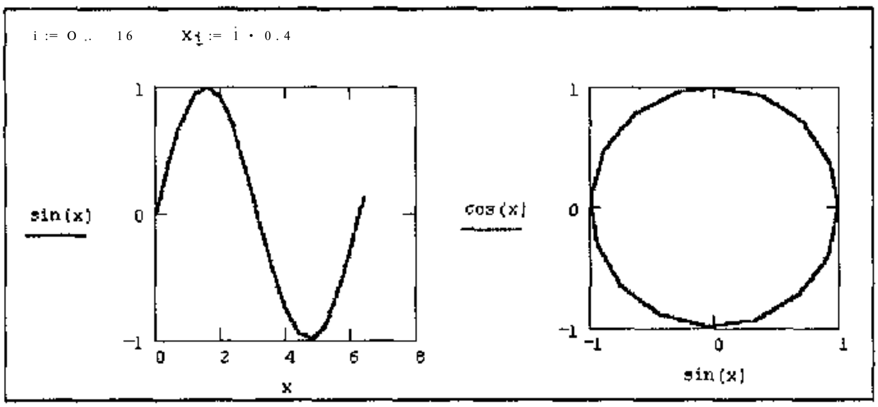


Рис. 16.7. Графики функций от векторного аргумента

16.3.4. Полярный график

Для создания полярного графика необходимо нажать кнопку Polar Plot на панели Graph (График) (рис. 16.8) и вставить в местозаполнители имена переменных и функций, которые будут нарисованы в полярной системе координат: угол (нижний местозаполнитель) и радиус-вектор (левый местозаполнитель). Точно так же, как при создании Декартова графика *(см. разд. 16.3.1—16.3.3)*, по осям могут быть отложены два вектора (рис. 16.8, слева), элементы векторов и ранжированные переменные в различных сочетаниях, а также может быть осуществлено быстрое построение графика функции (рис. 16.8, справа).

Форматирование полярных графиков практически идентично форматированию Декартовых, поэтому все, сказанное ниже об оформлении двумерных графиков на примере XY-графиков, в полной мере относится и к полярным.

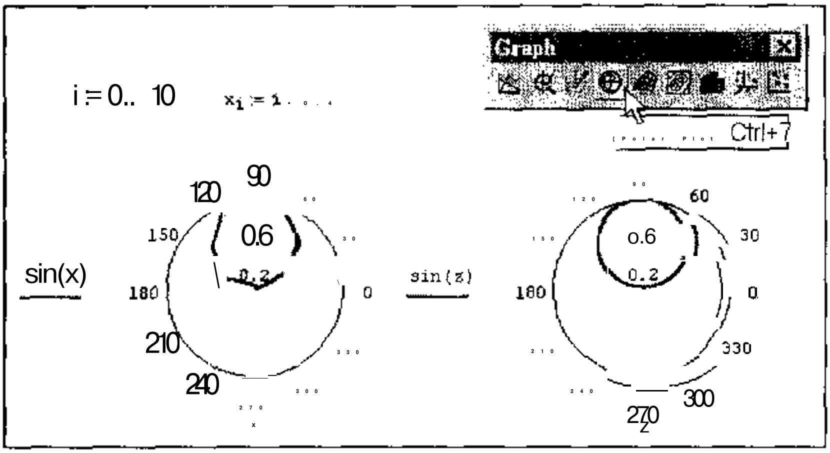


Рис. 16.8. Полярные графики

16.3.5. Построение нескольких рядов данных

На одном графике может быть отложено до 16 различных зависимостей. Чтобы построить на графике еще одну кривую, необходимо выполнить следующие действия:

1. Поместите линии ввода таким образом, чтобы они целиком захватывали выражение, стоящее в надписи координатной оси у (рис. 16.9).
2. Нажмите клавишу <,>.
3. В результате появится местозаполнитель, в который нужно ввести выражение для второй кривой.
4. Щелкните в любом месте вне этого выражения (на графике или вне его).

После этого вторая кривая будет отображена на графике. На рис. 16.9 уже нарисованы два ряда данных, а нажатие клавиши с запятой <,> приведет к появлению третьего местозаполнителя, с помощью которого можно определить третий ряд данных.

Примечание

Чтобы убрать один или несколько рядов данных с графика, удалите клавишами <BackSpace> или <Del> соответствующие им надписи у координатных осей.

Описанным способом будет создано несколько зависимостей, относящихся к одному аргументу. На рис. 16.9 построены графики пар точек $y(x_i)$ и $\cos(x_i)$ одного и того же аргумента — элементов вектора x_i. Об этом говорит единственная метка x у оси абсцисс. Вместе с тем, имеется возможность отображения на одном и том же графике зависимостей разных аргументов.

Для этого достаточно расставить по очереди метки всех зависимостей у обеих осей.

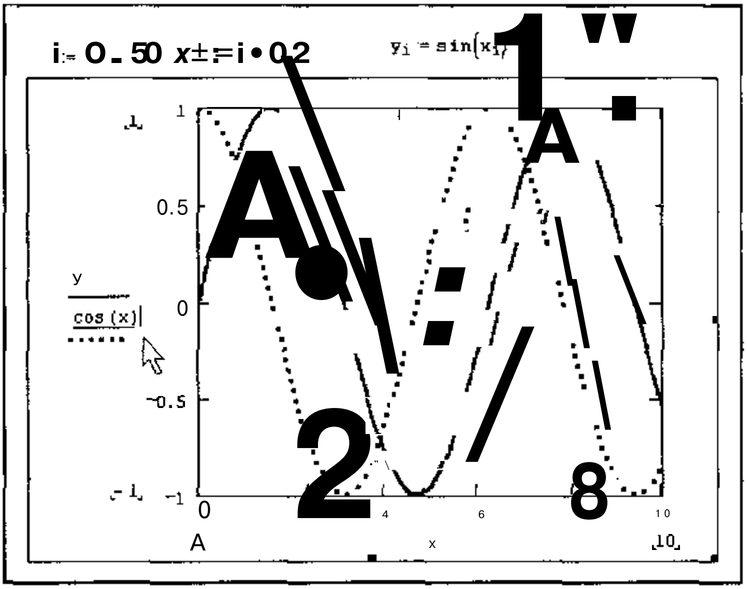


Рис. 16.9. Построение нескольких зависимостей на одном графике

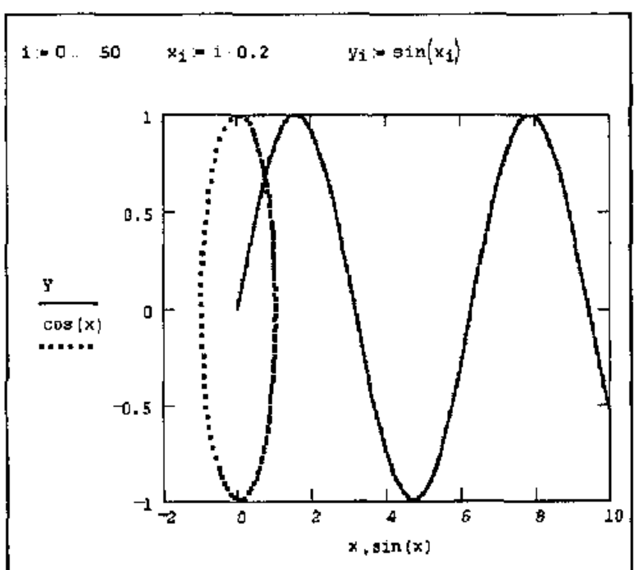


Рис. 16.10. Построение нескольких зависимостей от разного аргумента

Например, чтобы вместо второго (пунктирного) графика на рис. 16.9 построить график не cos(x$_i$), а график параметрической зависимости cos(sin(x$_i$)), достаточно добавить нажатием клавиши с запятой еще одну метку, на этот раз оси x, и ввести в нее выражение sin(x). Результат этих действий показан на рис. 16.10.

При построении на одном и том же графике нескольких зависимостей разного аргумента достаточно позаботиться только о соответствии типа данных для каждой пары точек в отдельности. Например, вполне можно совместно отобразить график функции от ранжированной переменной и график функции, созданный в режиме быстрого построения (рис. 16.11).

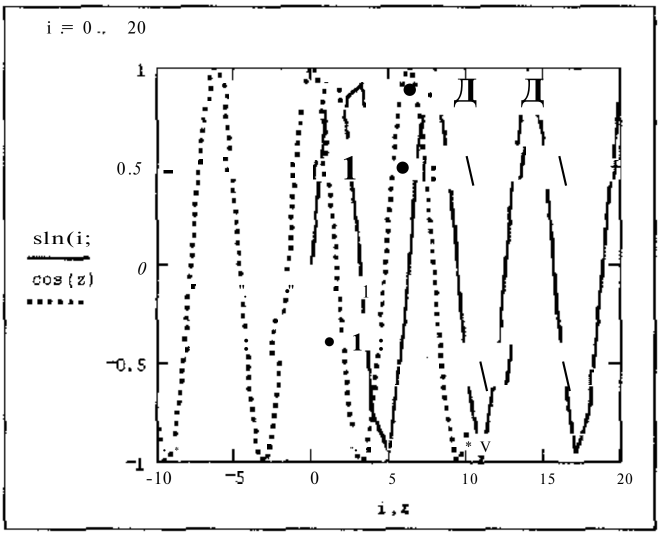


Рис. 16.11. Построение зависимостей от разного аргумента разного типа данных

16.3.6. Форматирование осей

Возможности форматирования координатных осей фафиков включают в себя управление их внешним видом, диапазоном, шкалой, нумерацией и отображением некоторых значений на осях при помощи маркеров.

Изменение диапазона осей

Когда график создается впервые, Mathcad выбирает представленный диапазон для обеих координатных осей автоматически. Чтобы изменить этот диапазон:

1. Перейдите к редактированию графика, щелкнув в его пределах мышью.
2. График будет выделен, а вблизи каждой из осей появятся два поля с числами, обозначающими границы диапазона. Щелкните мышью в об-

ласти одного из полей, чтобы редактировать соответствующую границу оси (например, верхнего предела оси x, как показано на рис. 16.12).

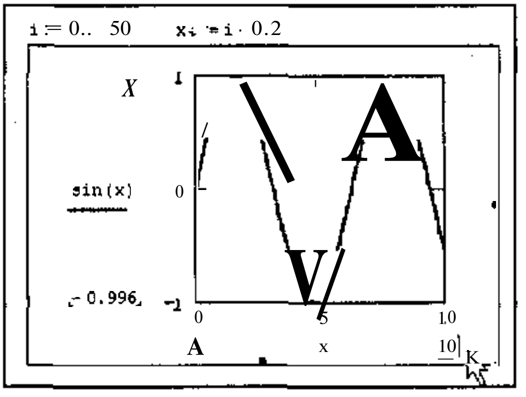


Рис. 16.12. Изменение диапазона оси x

3. Пользуясь клавишами управления курсором и клавишами <BackSpace> и <Del>, удалите содержимое поля.
4. Введите новое значение диапазона (например 20).
5. Щелкните за пределами поля, и график будет автоматически перерисован в новых пределах.

На рис. 16.13 показаны результаты изменения диапазона оси x на (0,20) и оси У — на (-2,2).

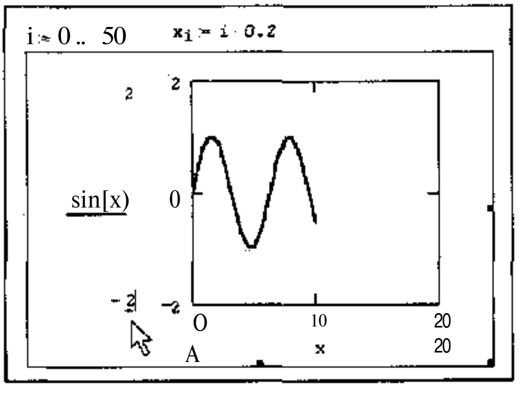


Рис. 16.13. Результат изменения диапазона осей

Чтобы вернуть автоматический выбор какого-либо диапазона, удалите число из соответствующего поля и щелкните вне его. Граница шкалы будет выбрана Mathcad, исходя из значений данных, представляемых на графике.

Форматирование шкалы

Изменение внешнего вида шкалы, нанесенной на координатную ось, производится с помощью диалогового окна **Formatting Currently Selected X-Y Plot** (Форматирование выбранного графика), в котором следует перейти на вкладку **X-Y Axes** (Оси X-Y) (рис. 16.14). Вызвать диалог можно двойным щелчком мыши в области графика или выполнением команды **Format / Graph / X-Y Plot** (Формат / График / X-Y График), или выбором в контекстном меню команды **Format** (Формат).

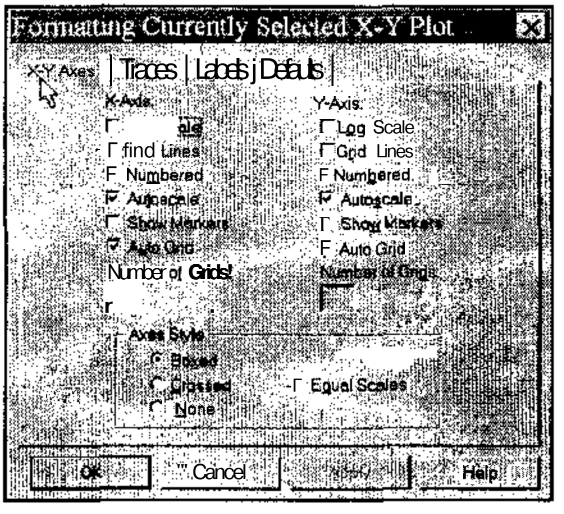


Рис. 16.14. Диалоговое окно **Formatting Currently Selected X-Y Plot**

С помощью флажков и переключателей легко поменять внешний вид каждой из осей. Перечислим доступные опции и поясним их действие:

- **Log Scale** (Логарифмический масштаб) — график по данной оси будет нарисован в логарифмическом масштабе. Это полезно, если данные разнятся на несколько порядков;
- **Grid Lines** (Линии сетки) — показать линии сетки (пример на рис. 16.15);
- **Numbered** (Нумерация) — показать нумерацию шкалы. Если убрать этот флажок, то числа, размечающие шкалу, пропадут (пример см. ниже на рис. 16.16);
- **Autoscale** (Автоматический масштаб) — выбор диапазона оси производится автоматически процессором Mathcad;
- **Show Markers** (Показать маркеры) — выделение значений на осях *(см. далее разд. "Маркеры" этой главы);*

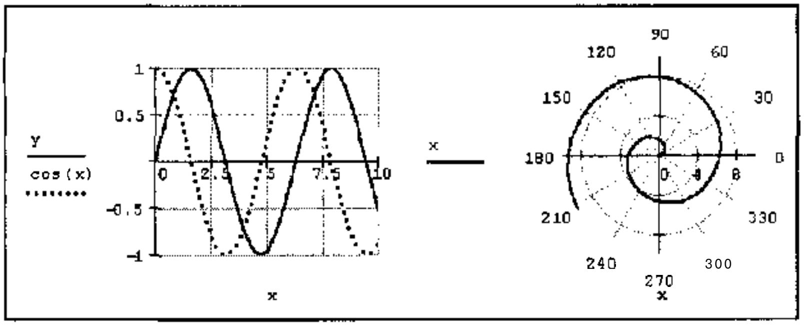


Рис. 16.15. Линии сетки на Декартовом и полярном графиках, вид осей — Crossed

- **AutoGrid** (Автоматическая шкала) — разбиение шкалы производится автоматически процессором Mathcad. Если этот флажок снят, в поле ввода рядом с ним следует указать желаемое количество меток шкалы;
- **Equal Scales** (Одинаковый масштаб) — оси х и Y принудительно рисуются в одинаковом масштабе;
- **Axes Style** (Вид оси) — можно выбрать один из трех видов системы координат:
  - **Boxed** (Прямоугольник) — как показано на рис. 16.10—16.13;
  - **Crossed** (Пересечение) — координатные оси в виде двух пересекающихся прямых (рис. 16.15);
  - **None** (Нет) — координатные оси не показываются на графике.

Примечание

Для полярного графика предусмотрены другие виды осей: Perimeter (Периметр), Crossed (Пересечение) и None (Нет). Первый тип оси показан выше (см. рис. 16.8), а второй Вы видите на рис. 16.15.

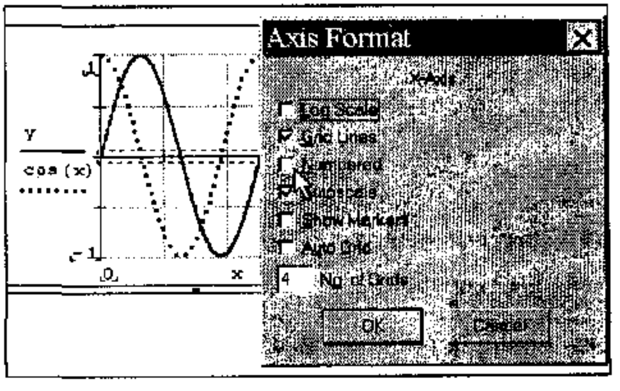


Рис. 16.16. Диалоговое окно Axis Format

Изменить описанные параметры можно и в диалоговом окне Axis Format (Формат оси), которое появляется, если щелкнуть дважды на самой оси (рис. 16.16).

Маркеры

Маркером на координатных осях отмечаются метки некоторых значений. Маркер представляет собой линию, перпендикулярную оси, снабженную числом или переменной. Чтобы создать маркер:

1. Дважды щелкните на графике.
2. На вкладке X-У Axes (Оси X-Y) диалога Formatting Currently Selected X-Y **Plot** (Форматирование выбранного графика), показанной на рис. 16.14, установите флажок Show Markers (Показать маркеры).
3. Нажмите кнопку ОК.
4. В появившийся местозаполнитель введите число или имя переменной, значение которой Вы хотите отобразить на оси маркером (рис. 16.17, слева).
5. Щелкните вне маркера.

Готовые маркеры показаны на рис. 16.17, справа. На каждой из осей допускается установить по два маркера. Если определен лишь один из них, то второй виден не будет.

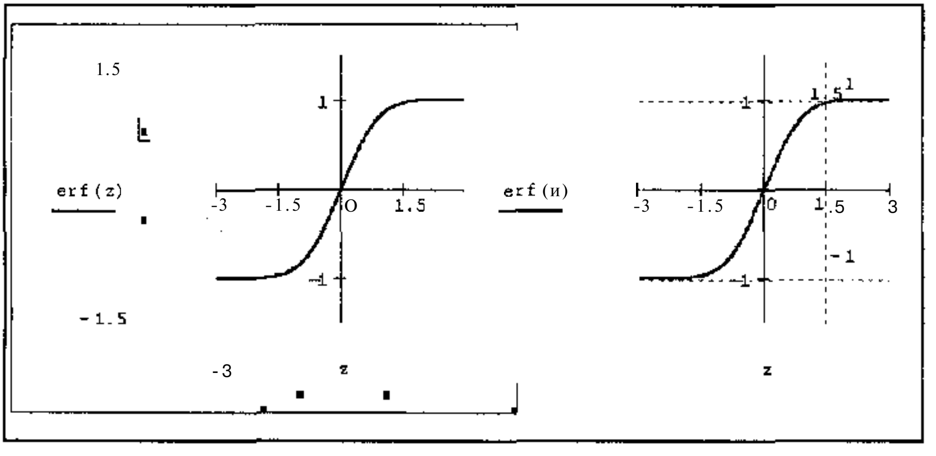


Рис. 16.17. Создание маркеров (слева) и готовые маркеры (справа)

Примечание

При создании маркеров очень полезной бывает трассировка графиков, позволяющая точно определить значение маркера *(см. разд. 16.3.10).*

16.3.7. Форматирование рядов данных

С помощью вкладки **Traces** (Ряды данных) диалогового окна **Formatting Currently Selected X-Y Plot** (Форматирование выбранного графика) легко установить комбинацию параметров линии и точек для каждого из рядов данных, представленных на графике. Пользователю требуется выделить в списке нужный ряд данных (его положение в списке соответствует положению метки зависимости у оси Y) И изменить в списках в середине диалогового окна желаемые установки (рис. 16.18).

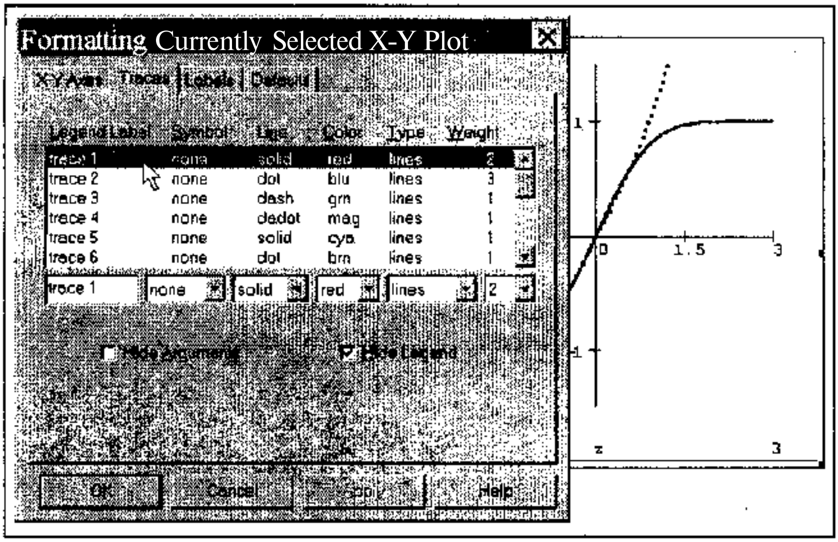


Рис. 16.18. Вкладка Traces диалога Formatting Currently Selected X-Y Plot

На вкладке **Traces** (Ряды данных) регулируются следующие параметры:

- **Legend Label** (Легенда) — текст легенды, описывающий ряд данных (на рис. 16.19—16.22 легенда объясняет смысл различных параметров);
- ❒ **Symbol** (Символ) — символ, которым обозначаются отдельные точки данных (рис. 16.21);
- ❒ **Line** (Линия) — стиль линии (рис. 16.19):
  - **solid** (сплошная);
  - **dot** (пунктир);
  - **dash** (штрих);
  - **dadot** (штрихпунктир).
- G **Color** (Цвет) — цвет линии и точек данных;
- ❒ **Weight** (Толщина) — толщина линии (рис. 16.20) и точек данных;

- **Type** (Тип) — тип представления ряда данных:
  - **lines** (линии);
  - **points** (точки);
  - **error** (ошибки);
  - **bar** (столбцы);
  - **step** (шаг);
  - **draw** (рисунок);
  - **stem** (стержень);
  - **solid bar** (гистограмма).

Примечание

Для некоторых типов графиков те или иные параметры недоступны (например нельзя задать символ для шаговой кривой).

Стиль, толщина и цвет линии

Изменяя параметры линии, можно добиться наилучшего восприятия разных зависимостей на одном графике (рис. 16.19—16.20).

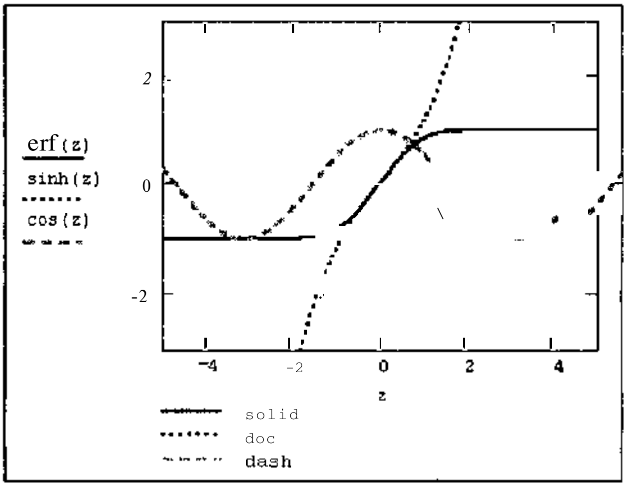


Рис. 16.19. Линии различного стиля

Форматирование точек данных

Чтобы построить график в виде только точек данных, перейдите в диалоге форматирования выбранного графика к списку **Type** (Тип) и выберите в нем пункт **points** (точки). Чтобы вместе с точками была показана и кривая, выберите другой тип ряда данных (например линии — **lines,** см. рис. 16.18).

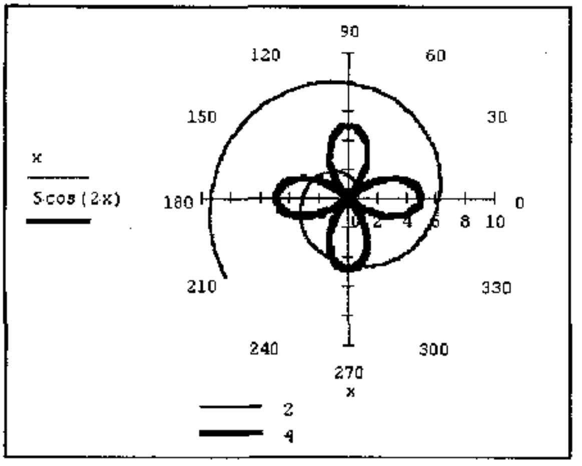


Рис. 16.20. Линии разной толщины на полярном графике

Внешний вид точки задает список Symbol (Символ), а их размер — Weight (Толщина). Примеры показаны на рис. 16.21.

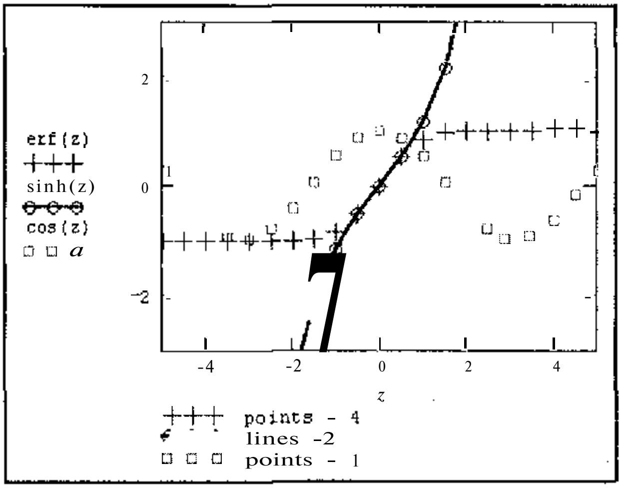


Рис. 16.21. Различный стиль и размер точек данных

Типы рядов данных

Несколько различных типов рядов данных представлено на рис. 16.22.

Примечание

Для полярных графиков также допускается устанавливать любые из перечисленных типов (пример см. ниже на рис. 16.25).

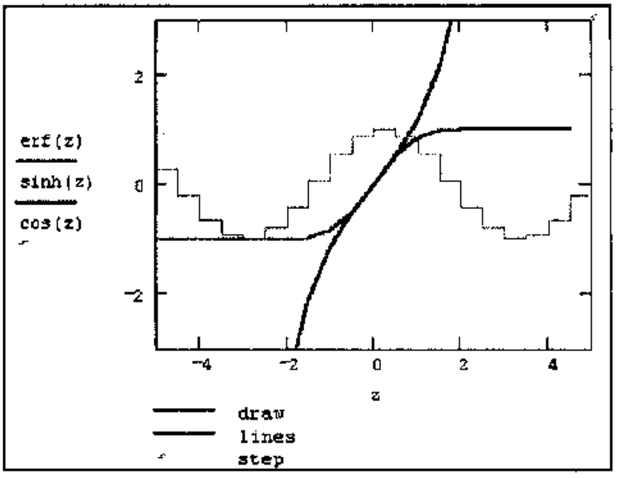


Рис. 16.22. Различные типы рядов данных

Столбчатые графики (гистограммы)

В Mathcad есть несколько столбчатых типов графиков, подходящих для построения гистограмм *(об их практическом применении читайте в гл. 13)*. Три различных типа иллюстрируются рис. 16.23.

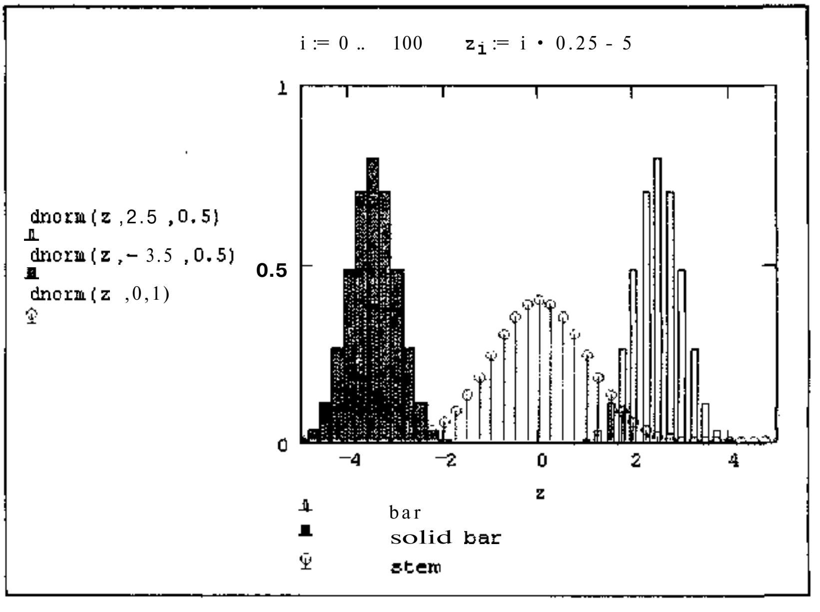


Рис. 16.23. Столбчатые типы графиков

Графики с отложенными ошибками

Тип графика с отложенными ошибками довольно сильно отличается от остальных типов, поскольку требует не двух, а трех серий данных. Помимо пар Декартовых (XY) ИЛИ полярных координат точек необходимо задать еще две последовательности данных, представляющих соответствующие значения ошибок для каждой пары точек (рис. 16.24).

Внимание!

График представления данных с погрешностями требует, чтобы два последовательных ряда данных имели тип графика с ошибками (error).

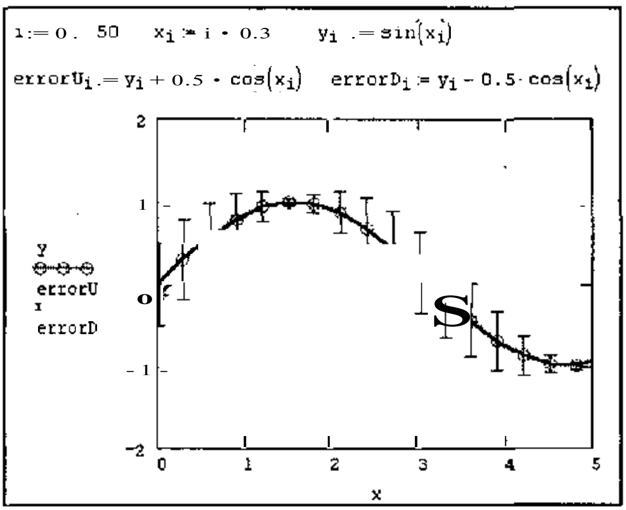


Рис. 16.24. Создание графика с отложенными ошибками

На рис. 16.24 отложено три ряда данных: y (сами данные), errorU (верхняя метка ошибок), errorD (нижняя метка ошибок). Для двух последних рядов выставлен тип error (ошибки).

Сохранение установок по умолчанию

На вкладке **Defaults** (По умолчанию) диалога **Formatting Currently Selected X-Y Plot** (Форматирование выбранного графика) находятся два элемента управления:

- кнопка **Change to Defaults** (Вернуть установки по умолчанию) — изменяет все установки выделенного графика на установки по умолчанию, принятые для текущего документа;
- флажок проверки Use **for Defaults** (Использовать для установок по умолчанию) — делает установками по умолчанию для данного документа установки выбранного графика.

16.3.8. Создание заголовка графика

Чтобы создать заголовок графика:

1. Дважды щелкните на графике.
2. В диалоге Formatting Currently Selected X-Y Plot (Форматирование выбранного графика) перейдите на вкладку Labels (Метки).
3. В поле Title (Заголовок) введите текст заголовка.
4. Установите флажок проверки Show Title (Показать заголовок).
5. Выберите переключатель Above (Сверху) или Below (Снизу), чтобы заголовок появился сверху или снизу графика, как показано на рис. 16.25.
6. Нажмите кнопку OK.

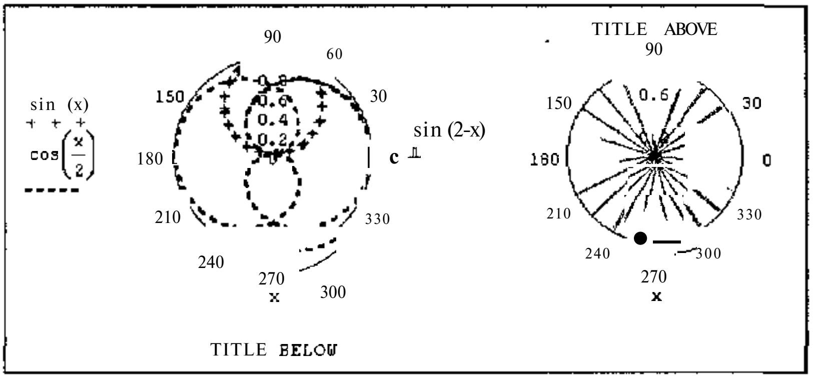


Рис. 16.25. Заголовок сверху и снизу графика

16.3.9. Изменение размера и положения графиков

Прежде чем переместить или изменить размер графика, выделите его щелчком мыши. Изменить положение графика в документе можно перетаскиванием, т. е. перемещением указателя при нажатой кнопке мыши. Чтобы изменить размер графика, растягивайте или сжимайте его, перемещая указателем мыши черные прямоугольные маркеры, расположенные на его сторонах.

16.3.10. Трассировка и увеличение графиков

Трассировка позволяет очень точно изучить строение графика. Для того чтобы включить режим трассировки, щелкните в области графика правой кнопкой мыши и выберите в контекстном меню пункт Trace (Трассировка). В результате появится окно трассировки (рис. 16.26), а в поле графика Вы увидите две пересекающиеся пунктирные линии.

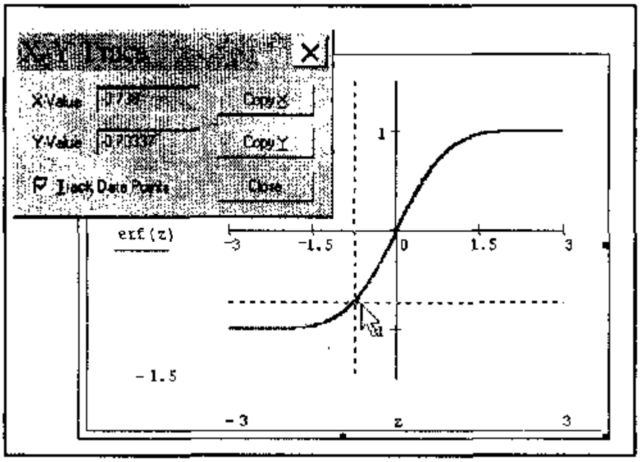


Рис. 16.26. Трассировка графика

Перемещая указатель мыши по графику, Вы тем самым передвигаете точку пересечения линий трассировки. При этом координаты точки указываются с высокой точностью в окне трассировки в полях **X-Value** (Значение X) и **Y-Value** (Значение Y). Нажатие кнопки **Copy X** (Копировать X) или **Copy Y** (Копировать Y) копирует соответствующее число в буфер обмена. В дальнейшем его можно вставить в любое место документа или в маркер, нажав клавиши <Ctrl>+<V>.

Если установлен флажок **Track Data Points** (Следовать за рядом данных), как это показано на рис. 16.26, то линии трассировки следуют точно вдоль графика. Если нет, то они могут перемещаться по всей области графика.

Помимо трассировки, в Mathcad предусмотрена еще одна удобная возможность просмотра графика в увеличенном масштабе. Для вызова диалогового окна **Zoom** (Масштаб графика) выберите в контекстном меню, либо в меню **Format (Формат) пункты Graph** (График) и **Zoom** (Масштаб). После этого указателем мыши выберите прямоугольную область на графике, которую Вы планируете просмотреть в увеличенном масштабе (рис. 16.27), и нажмите кнопку **Zoom** (Увеличить). В результате часть графика будет прорисована более крупно (рис. 16.28). Далее можно либо продолжать изменять масштаб, либо вернуться к прежнему виду графика кнопкой **Full View** (Показать целиком), либо закрыть диалог **Zoom** для окончательной перерисовки графика в крупном масштабе (нажав кнопку OK).

Примечание

Возможно, Вам покажется более удобным вызов окон трассировки и масштабирования графиков с помощью панели инструментов **Graph** (График) (рис. 16.29). Эти кнопки доступны только при выделенном двумерном графике.

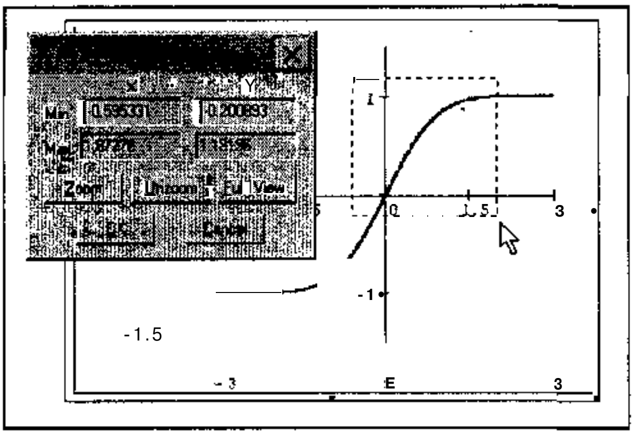


Рис. 16.27. Управление масштабом графика

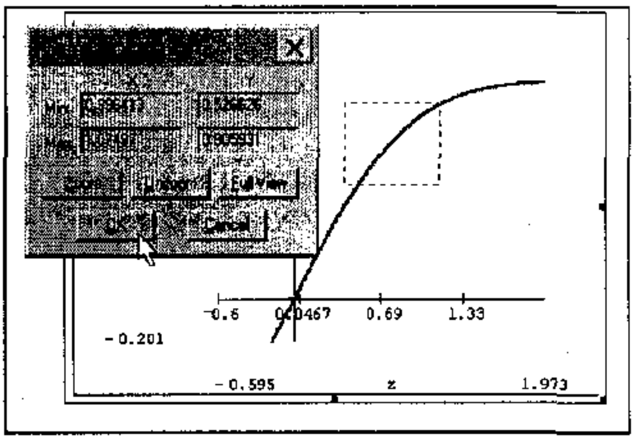


Рис. 16.28. Просмотр графика в увеличенном масштабе

Рис. 16.29. Кнопка трассировки на панели Graph, кнопка масштабирования —- слева от нее

16.4. Трехмерные графики

Коллекция трехмерных графиков — настоящее чудо, которое Mathcad дарит пользователю. За считанные секунды Вы можете создать великолепную презентацию результатов своих расчетов.

16.4.1. Создание трехмерных графиков

Чтобы создать трехмерный график, требуется нажать кнопку с изображением любого из типов трехмерных графиков на панели инструментов Graph (График) (если возникнут сложности, обратитесь к *разд. 16.2).* В результате появится пустая область графика с тремя осями (рис. 16.30) и единственным местозаполнителем в нижнем левом углу. В этот местозаполнитель следует ввести либо имя z функции z(x,y) двух переменных для *быстрого построения трехмерного графика,* либо имя матричной переменной z, которая задаст распределение данных $z_{x,y}$ на плоскости XY.

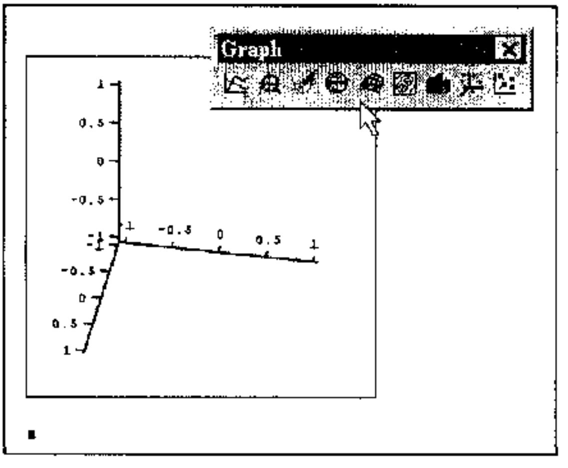


Рис. 16.30. Создание трехмерного графика

Рассмотрим на простом примере функции z(x,y) и матрицы z (они заданы в листингах 16.3 и 16.4, соответственно) примеры построения трехмерных графиков различных типов, создаваемых нажатием той или иной кнопки на панели Graph (График). Еще раз отметим, что для получения фафиков не требуется никакого текста, кроме введения имени соответствующей функции или матрицы в местозаполнитель.

Внимание!

Для графиков, задаваемых матрицами, шкалу плоскости XY приходится задавать вручную. Mathcad просто рисует поверхность, точки в пространстве или линии уровня, основываясь на двумерной структуре этой матрицы. При быстром же построении графиков имеется возможность строить их в различном диапазоне аргументов, подобно двумерным графикам.

Листинг 16.3. Функция для быстрого построения трехмерных графиков

$z(x, y) := x^2 + y^2$

Листинг 16.4. Матрица для отображения на трехмерных графиках

$$Z := \begin{pmatrix} 1 & 1 & 0 & 1.1 & 1.2 \\ 1 & 2 & 3 & 2.1 & 1.5 \\ 1.3 & 3.3 & 5 & 3.7 & 2 \\ 1.3 & 3 & 5.7 & 4.1 & 2.9 \\ 1.5 & 2 & 6.5 & 4.8 & 4 \end{pmatrix}$$

Surface Plot - график поверхности (рис. 16.31 и 16.32)

Для построения графика поверхности необходимо воспользоваться клавишей Surface Plot панели инструментов Graph и следовать указаниям, описанным выше.

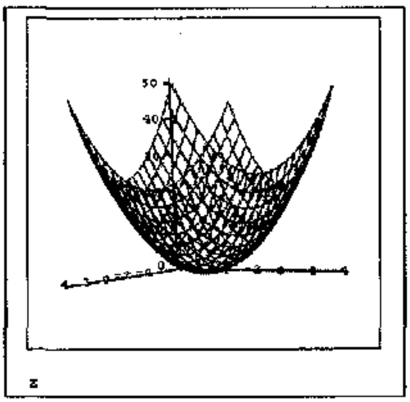


Рис. 16.31. Быстрое построение графика поверхности функции (листинг 16.3)

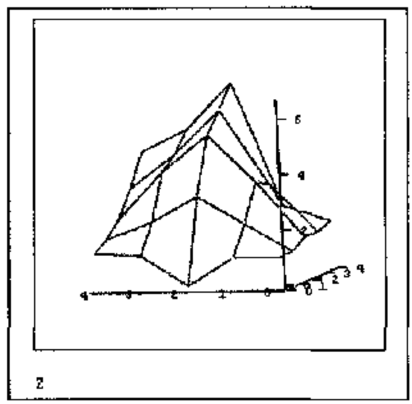


Рис. 16.32. График поверхности, заданный матрицей (листинг 16.4)

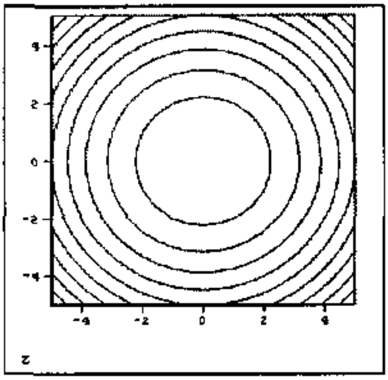


Рис. 16.33. Быстрое построение графика линий уровня функции (листинг 16.3)

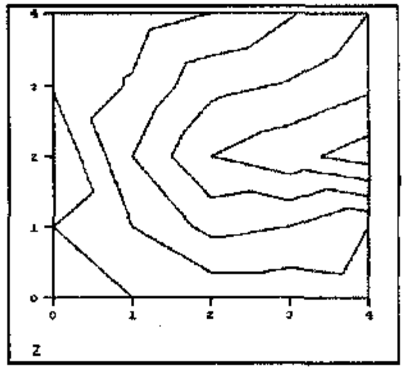


Рис. 16.34. График линий уровня, заданный матрицей (листинг 16.4)

Contour Plot - график линий уровня (рис. 16.33 и 16.34)

Для построения графика поверхности необходимо воспользоваться клавишей Contour Plot панели инструментов Graph и следовать указаниям, описанным выше.

3D Bar Plot - график трехмерной гистограммы (рис. 16.35 и 16.36)

Для построения графика поверхности необходимо воспользоваться клавишей 3D Bar Plot панели инструментов Graph и следовать указаниям, описанным выше.

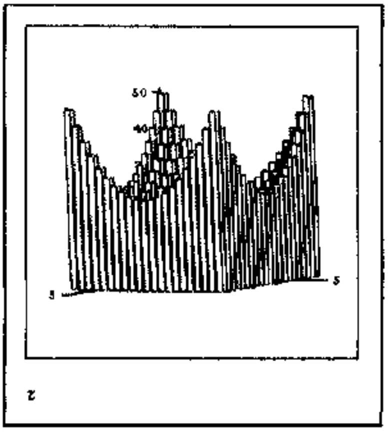


Рис. 16.35. Быстрое построение графика трехмерной гистограммы функции (листинг 16.3)

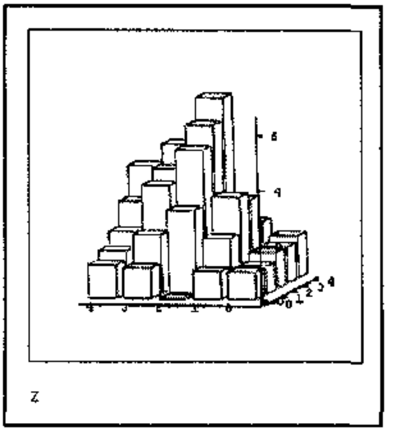


Рис. 16.36. График трехмерной гистограммы, заданный матрицей (листинг 16.4)

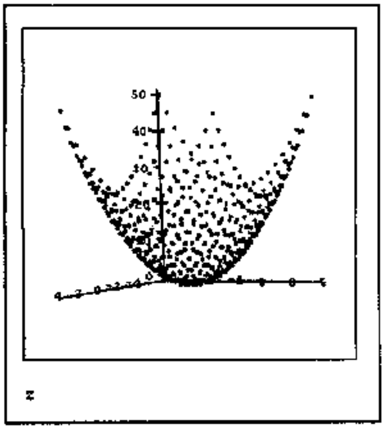


Рис. 16.37. Быстрое построение графика множества точек функции (листинг 16.3)

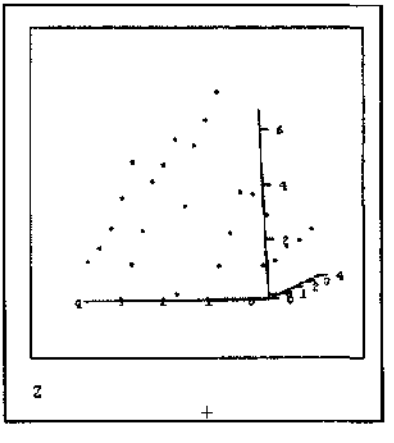


Рис. 16.38. График множества точек, заданный матрицей (листинг 16.4)

3D Scatter Plot - график множества точек (рис. 16.37 и 16.38)

Для построения графика поверхности необходимо воспользоваться клавишей **3D Scatter Plot** панели инструментов **Graph** и следовать указаниям, описанным выше.

Vector Field Plot - график векторного поля (рис. 16.39)

График векторного поля несколько отличается от остальных типов двумерных графиков. Его смысл заключается в построении некоторого вектора в каждой точке плоскости XY. Чтобы задать вектор на плоскости, требуются два скалярных числа. Поэтому в Mathcad принято, что векторное поле задает комплексная матрица. Действительные части каждого ее элемента задают проекцию вектора на ось x, а мнимые — на ось Y.

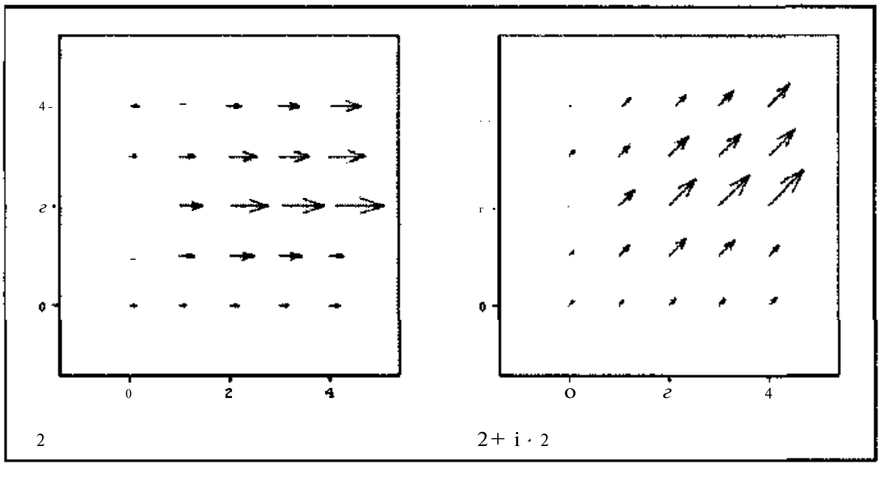


Рис. 16.39. Графики векторного поля, заданные матрицами (листинг 16.4)

Приведенные рисунки являются лишь первым шагом в создании красочных графиков. О том, как правильно отформатировать вновь созданные графики, чтобы они приобрели оптимальный с математической точки зрения и эффектный вид, рассказано в следующих разделах.

Совет

Улучшить трехмерное представление графика часто позволяет применение к исходным данным интерполяции *(см. разд. "Многомерная интерполяция" гл. 15).*

16.4.2. Форматирование трехмерных графиков

Форматирование трехмерных графиков выполняется с помощью диалогового окна **3-D Plot Format** (Форматирование 3-D графика), которое вызывается двойным щелчком мыши в области графика (рис. 16.40). Параметры трехмерных графиков всех типов устанавливаются посредством этого диалогового окна.

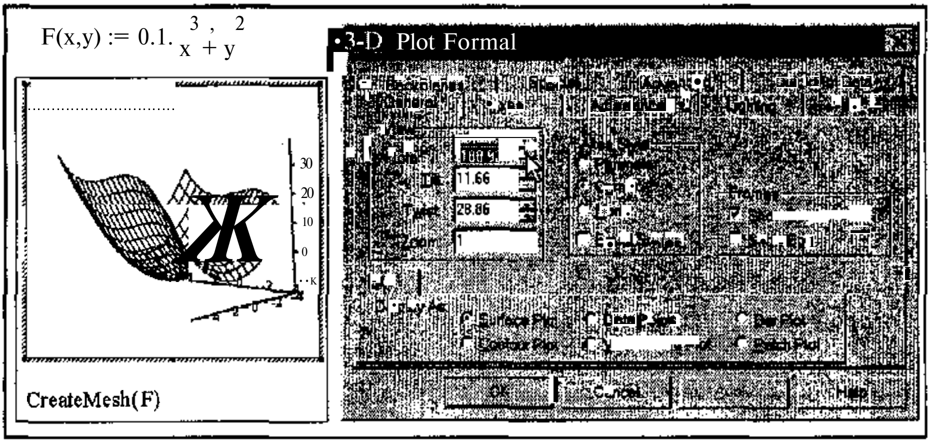


Рис. 16.40. Диалоговое окно **3-D Plot Format**

В диалоге 3-D Plot Format (Форматирование 3-D графика) доступно большое количество параметров, изменение которых способно очень сильно повлиять на внешний вид графика. Они сгруппированы по принципу действия на нескольких вкладках. Остановимся коротко на возможностях оформления трехмерных графиков, поясняя их, главным образом, примерами.

Изменение типа графика

Чтобы поменять тип уже имеющегося графика (например построить вместо поверхности график линий уровня и т. д.), просто установите соответствующий переключатель в нижней части вкладки General (Общие) и нажмите кнопку ОК. График будет перерисован.

Вращение графика

Самый простой способ ориентации системы координат с графиком в трехмерном пространстве — это перетаскивание ее указателем мыши. Попробуйте перемещать при нажатой левой кнопке мыши указатель в пределах графика, и Вы увидите, как поворачивается график.

Примечание

Разумеется, поворачивать можно лишь графики в трехмерном пространстве; векторное поле и линии уровня строятся, по определению, на прямоугольном участке плоскости.

Другой способ изменения ориентации графика — с помощью полей Rotation (Вращение), Tilt (Наклон) и Twist (Поворот) на вкладке General (Общие) (см. рис. 16.40), которые в совокупности определяют соответствующие углы (в градусах) и тем самым задают направление всех трех осей координат в пространстве. Сравнивая рис. 16.40—16.42, Вы разберетесь, как эти углы влияют на ориентацию графика.

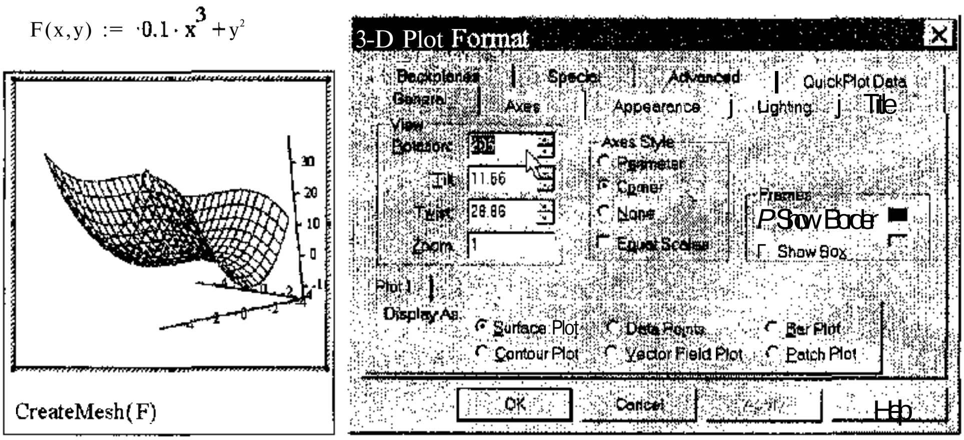


Рис. 16.41. Изменение параметра **Rotation** (сравните с рис. 16.40)

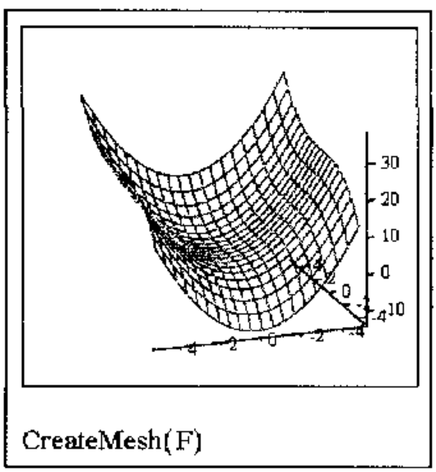


Рис. 16.42. Вид того же графика (рис. 16.41) при углах Rotation=0, **Tilt**=20 и **Twist**=200

Стиль осей

С помощью группы переключателей Axes Style (Стиль осей) можно задать один из следующих стилей осей координат:

- Perimeter (Периметр) — как на рис. 16.43;

- Corner (Углом) — как на рис. 16.41—16.42;
- None (Нет) — оси отсутствуют.

Если установить флажок Show Box (Показать куб), то координатное пространство будет изображено в виде куба (рис. 16.44).

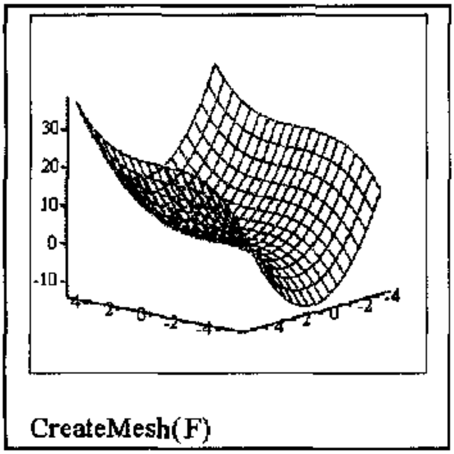


Рис. 16.43. Расположение координатных осей по периметру

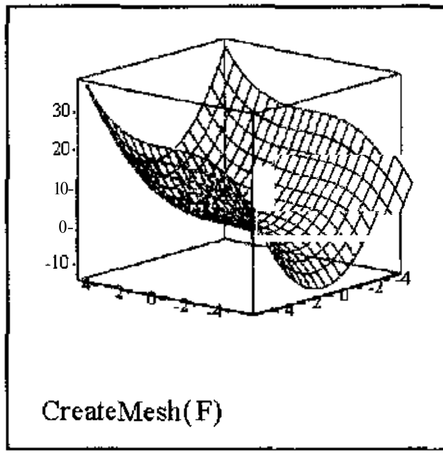


Рис. 16.44. Установлен флажок **Show Box**

Масштабирование графика

В поле **Zoom** (Масштаб) вкладки **General** (Общие) (см. рис. 16.40) можно задать числовое значение масштаба (рис. 16.45).

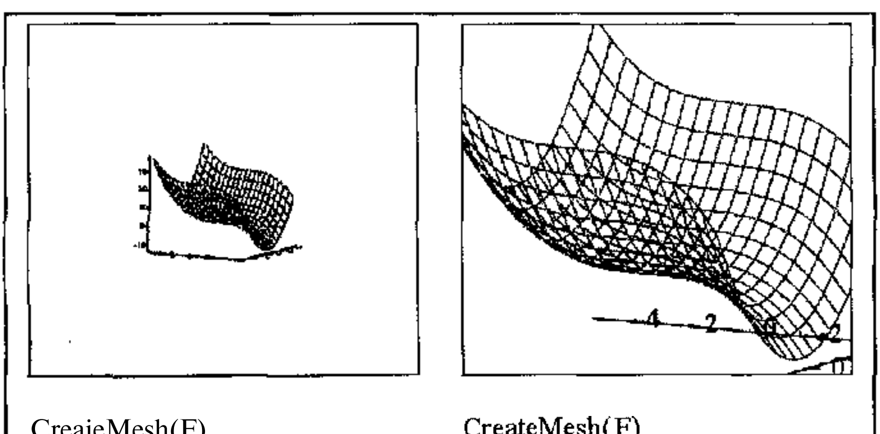


Рис. 16.45. Вид графика поверхности при масштабе 0,5 (слева) и 1,5 (справа)

Форматирование осей

Вкладка **Axes** (Оси) содержит три вложенных вкладки, в которых задаются параметры для каждой из трех координатных осей. В частности, можно включить или отключить показ линий сетки, нумерации и задать диапазон по каждой из осей (рис. 16.46). Смысл этих операций сходен с аналогичными операциями для двумерных графиков. При помощи еще одной вкладки — **Backplanes** (Плоскости заднего плана) задается показ проекций координатной сетки на три скрытые плоскости трехмерного графика (пример форматирования плоскости XY показан на рис. 16.47).

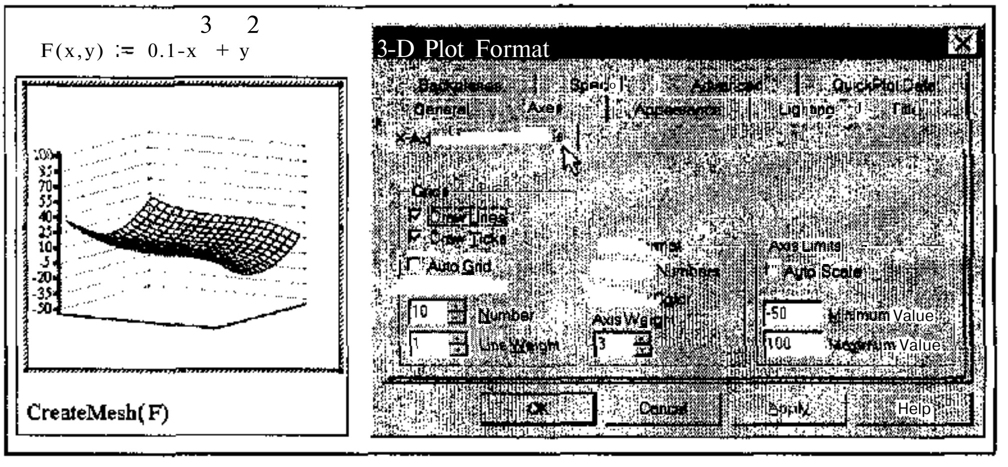


Рис. 16.46. Форматирование осей координат

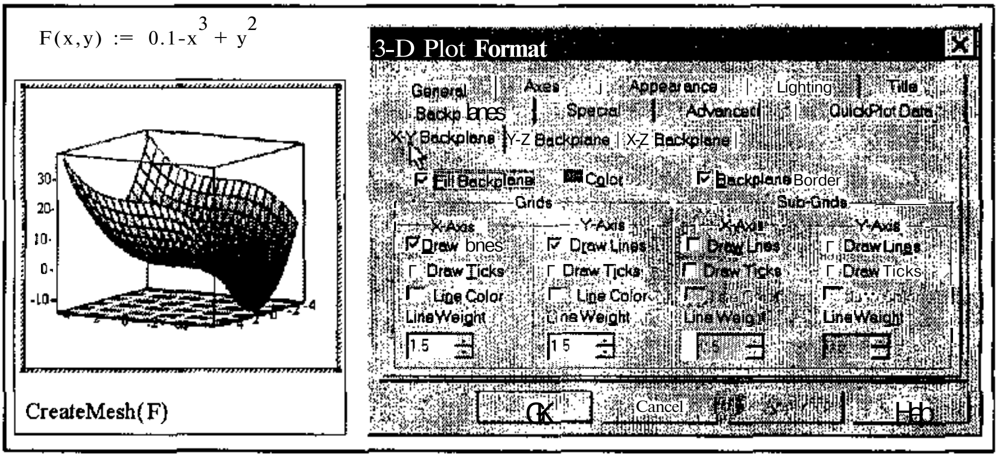


Рис. 16.47. Форматирование скрытых плоскостей графика

Стиль заливки и линий

На рис. 16.48 и 16.49 показано влияние различного стиля задания заливки и линий с помощью вкладки Appearance (Появление) для контурного и поверхностного графиков. При выборе переключателя Fill Surface (Заливка поверхности) из группы Fill Options (Опции заливки) (рис. 16.48) Вы получаете доступ к опциям цвета (в группе Color Options). Если выбрать переключатель Solid Color (Один цвет), то получится однотонная заливка поверхности, показанная на рис. 16.49. Если установить переключатель Colormap (Цветовая схема), то поверхность или контурный график будут залиты разными цветами и оттенками (рис. 16.48), причем выбрать цветовую схему можно на вкладке Advanced (Дополнительно) (рис. 16.50).

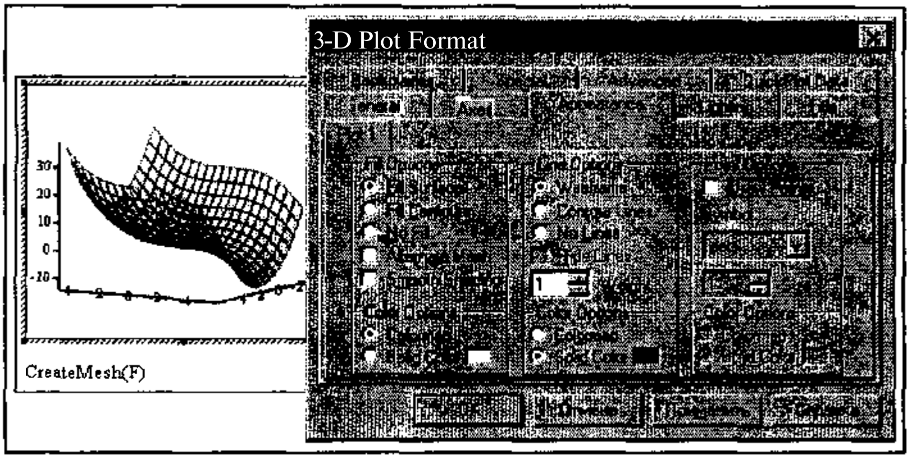


Рис. 16.48, Настройка заливки графика поверхности

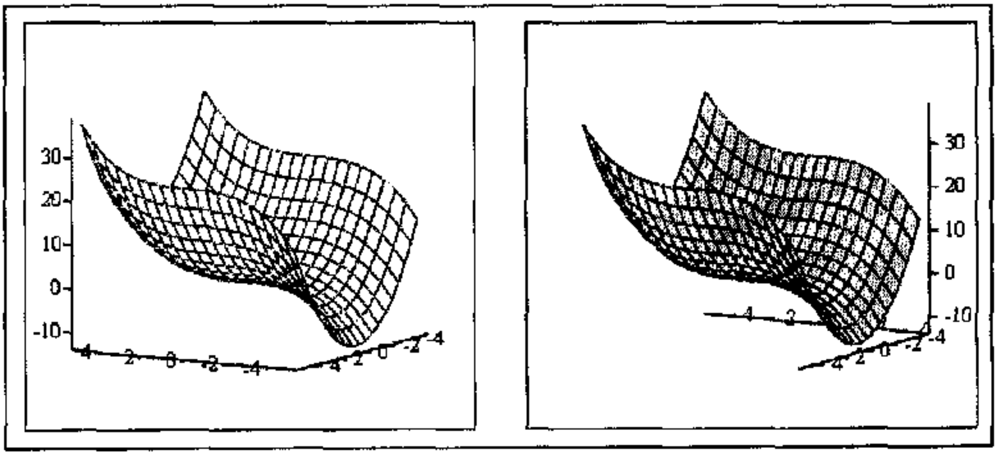


Рис. 16.49. Заливка поверхности белым цветом (слева) и серым цветом (справа)

Поэкспериментируйте с разными цветовыми схемами и представлениями заливки и линий, задаваемых полем **Line Options** (Опции линии) (рис. 16.48), чтобы представлять себе богатство возможностей Mathcad. Некоторые параметры, влияющие на представление контуров графиков, доступны на вкладке **Special** (Специальные) (рис. 16.51). Сочетаний различных цветовых схем, заливок и других параметров настолько много, что лучше предоставить читателю самому попробовать применить их различные комбинации и выбрать из них наиболее понравившиеся.

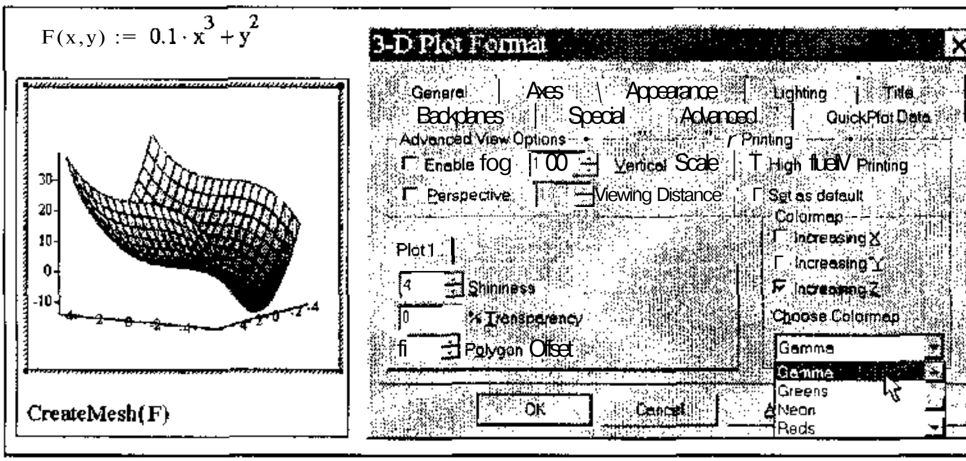


Рис. 16.50. Выбор цветовой схемы

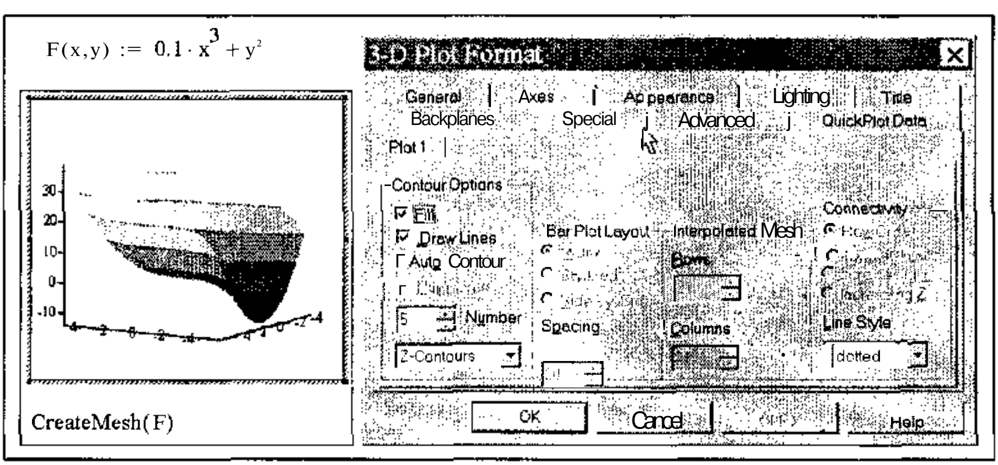


Рис. 16.51. Вкладка Special

Спецэффекты

В той же вкладке Advanced (Дополнительно) (рис. 16.50) имеется доступ к управлению несколькими специальными эффектами оформления графиков, благодаря которым они смотрятся более красиво.

Перечислим эти эффекты:

- Shininess (Сияние) — имеется возможность регулировать сияние в пределах от 0 до 128;
- Fog (Туман) — эффект тумана (рис. 16.52);

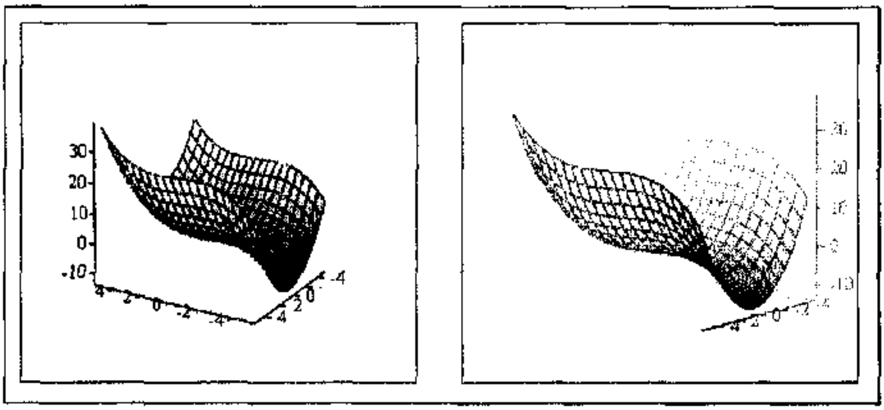


Рис. 16.52. Эффекты перспективы (слева) и тумана (справа)

- Transparency (Прозрачность) — задается процент прозрачности графика (рис. 16.53);
- Perspective (Перспектива) — показ перспективы с определением видимости расстояния (рис. 16.52).

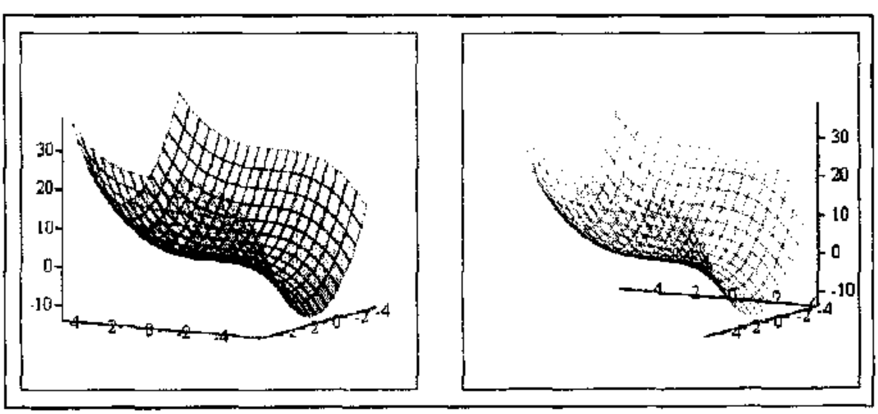


Рис. 16.53. Эффект прозрачности в 40% (слева) и 80% (справа)

Еще один спецэффект подсветки графика задается на вкладке **Lighting** (Подсветка) (рис. 16.54), причем имеются как встроенные схемы подсветки (на рис. 16.54 на них наведен указатель мыши), так и возможность задавать ее цвет и направление самому пользователю.

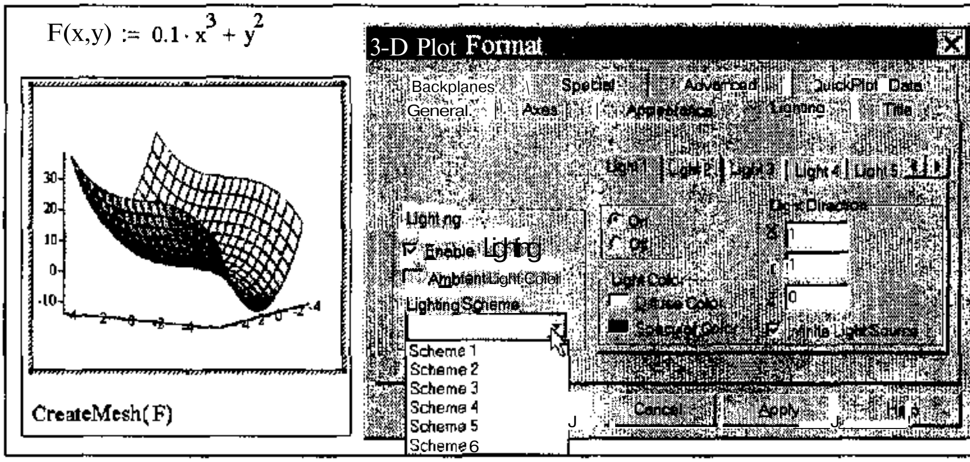


Рис. 16.54. Эффект подсветки

Заголовок графика

Заголовок графика определяется на вкладке **Title** (Заголовок) и может быть расположен как сверху, так и снизу графика (рис. 16.55).

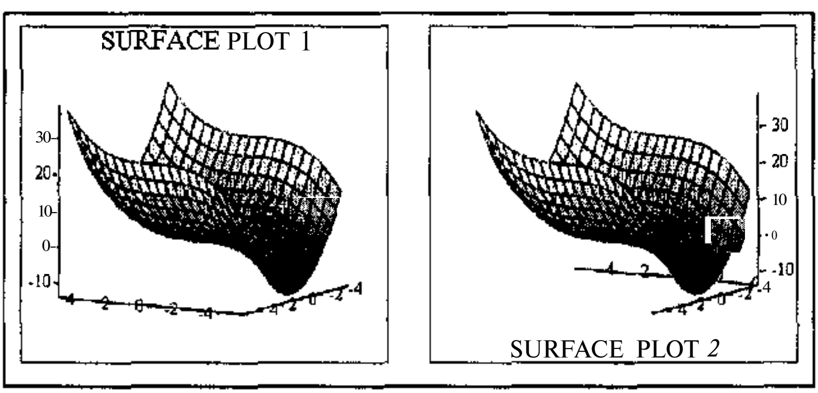


Рис. 16.55. Графики с заголовками

Редактирование точек данных

На многих типах графиков допускается показ точек данных. Формат точек, включая тип символа, размер, соединение их линией задается на вкладке

Appearance (Оформление) (рис. 16.56). Опции форматирования точек те же самые, что и для двумерных графиков *(см. разд. 16.3.7).*

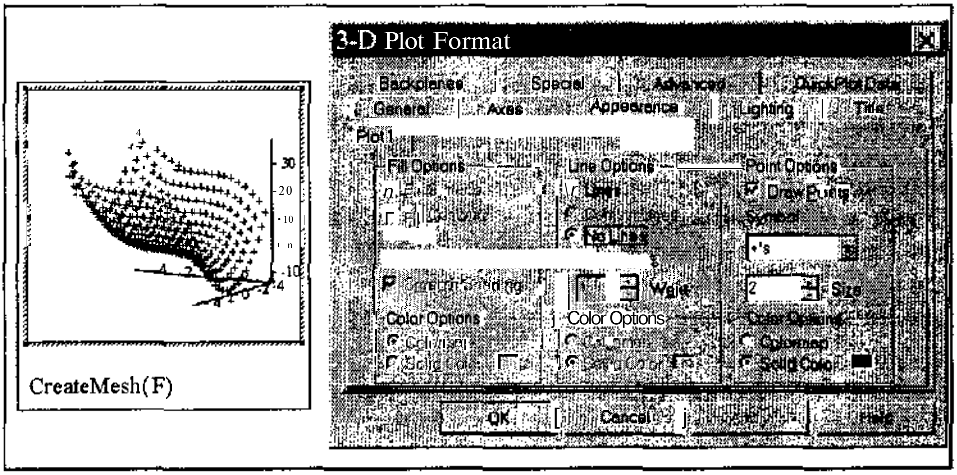


Рис. 16.56. Форматирование точек данных

Быстрое изменение формата графика

Удобный способ применения некоторых видов форматирования трехмерных графиков заключается в использовании контекстного меню (рис. 16.57). Достаточно нажать на графике правую кнопку мыши и выбрать в контекстном меню одну из опций форматирования. Внешний вид графика сразу изменится.

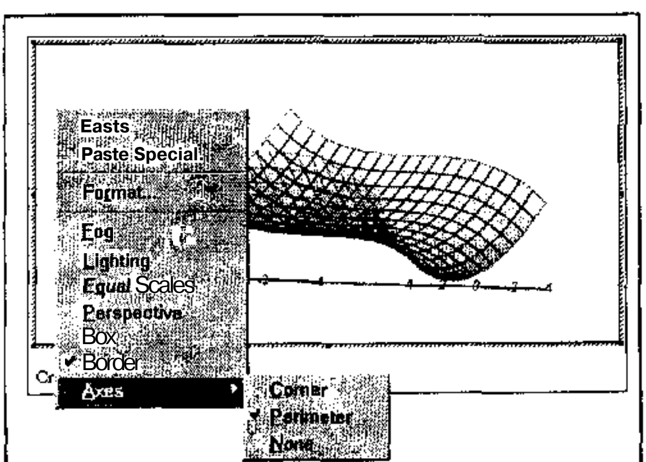


Рис. 16.57. Форматирование графика при помощи контекстного меню

16.5. Создание анимации

Во многих случаях самый зрелищный способ представления результатов математических расчетов — это анимация. Mathcad позволяет создавать анимационные ролики и сохранять их в видеофайлах.

Основной принцип анимации в Mathcad — покадровая анимация. Ролик анимации — это просто последовательность кадров, представляющих собой некоторый участок документа, который выделяется пользователем. Расчеты производятся обособленно для каждого кадра, причем формулы и графики, которые в нем содержатся, должны быть функцией от номера кадра. Номер кадра задается системной переменной FRAME, которая может принимать лишь натуральные значения. По умолчанию, если не включен режим подготовки анимации, FRAME=0.

Рассмотрим последовательность действий для создания ролика анимации, например демонстрирующего перемещение гармонической бегущей волны. При этом каждый момент времени будет задаваться переменной FRAME.

1. Введите в документ необходимые выражения и графики, в которых участвует переменная номера кадра FRAME. Подготовьте часть документа, которую Вы желаете сделать анимацией, таким образом, чтобы она находилась в поле Вашего зрения на экране. В нашем примере подготовка сводится к определению функции f(x,t) :=sin(x-t) и создании ее Декартова графика y(x,FRAME).
2. Выполните команду Tools / Animation / Record (Сервис / Анимация / Запись).
3. В диалоговом окне Animate (Анимация) задайте номер первого кадра в поле **From** (От), номер последнего кадра в поле То (До) и скорость анимации в поле At (Скорость) в кадрах в секунду (рис. 16.58).
4. Выделите протаскиванием указателя мыши при нажатой левой кнопке мыши область в документе, которая станет роликом анимации.
5. В диалоговом окне Animate (Анимация) нажмите кнопку Animate (Анимация). После этого в окошке диалогового окна Animate (Анимация) будут появляться результаты расчетов выделенной области, сопровождающиеся выводом текущего значения переменной FRAME. По окончании этого процесса на экране появится окно проигрывателя анимации (рис. 16.59).
6. Запустите просмотр анимации в проигрывателе нажатием кнопки воспроизведения в левом нижнем углу окна проигрывателя.
7. В случае если вид анимации Вас устраивает, сохраните ее в виде видеофайла, нажав кнопку Save As (Сохранить как) в диалоговом окне Animate (Анимация). В появившемся диалоговом окне Save Animation (Сохранить

анимацию) обычным для Windows способом укажите имя файла и его расположение на диске.

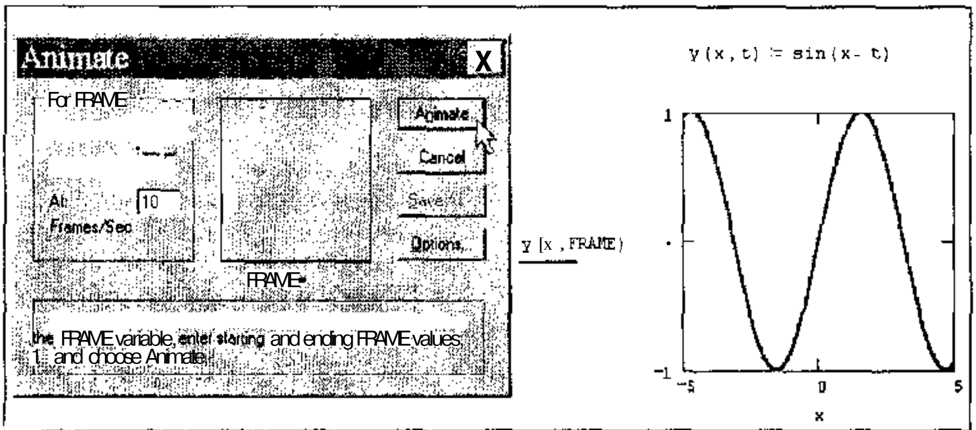


Рис. 16.58. Начало создания анимации

Закройте диалог **Animate** (Анимация) нажатием кнопки **Cancel** (Отмена) или кнопки управления его окном. Сохраненный видеофайл можно использовать за пределами Mathcad. Скорее всего, если в проводнике Windows дважды щелкнуть на имени этого файла, он будет загружен в проигрыватель видеофайлов Windows, и Вы увидите его на экране компьютера. Таким образом, запуская видеофайлы обычным образом, можно устроить красочную презентацию результатов работы как на своем, так и на другом компьютере.

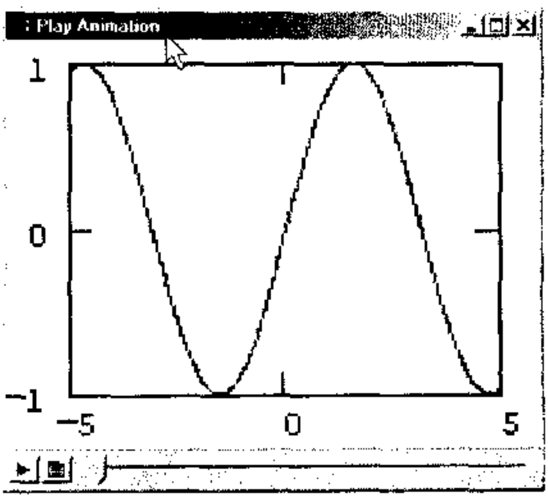


Рис. 16.59. Просмотр созданного ролика анимации

Примечание

При создании файлов анимации допускается выбирать программу видеосжатия (кодек) и качество компрессии. Делается это с помощью кнопки **Options** (Опции) в диалоговом окне **Animate** (Анимация).

16.6. Ввод-вывод во внешние файлы

Важный компонент ввода-вывода — это ввод-вывод во внешние файлы. Ввод внешних данных в документы Mathcad применяется чаще вывода, поскольку Mathcad имеет гораздо лучшие возможности представления результатов расчетов, чем многие пользовательские программы. Для общения с внешними файлами данных в Mathcad имеется несколько разных способов. Самый простой из них — использовать имеющееся семейство встроенных функций.

16.6.1. Текстовые файлы

Перечислим встроенные функции для работы с текстовыми файлами:

- `READPRN("file")` — чтение данных в матрицу из текстового файла;
- `WRITEPRN("file")` — запись данных в текстовый файл;
- `APPENDPRN("file")` — дозапись данных в существующий текстовый файл;
    - `file` — путь к файлу.

Примечание

Можно задавать как полный путь к файлу, например, C:\Мои документы, так и относительный, имея в виду, что он будет отсчитываться от папки, в которой находится файл с документом Mathcad.

Примеры использования встроенных функций иллюстрируются листингами 16.5—16.7. Результат действия листингов 16.5 и 16.7 можно понять, просмотрев получающиеся текстовые файлы, например с помощью Блокнота Windows (рис. 16.60 и 16.61, соответственно).

Листинг 16.5. Запись матрицы I в текстовый файл

$$\mathbf{z} := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 99 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

WRITEPRN ("datafile.prn" } $:= Z$

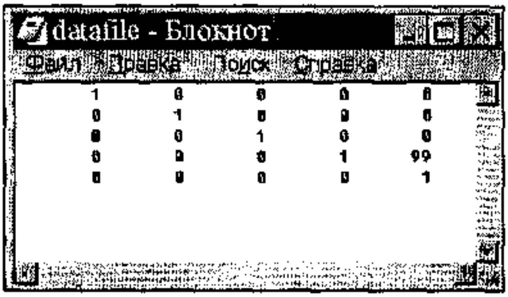


Рис. 16.60. Файл, созданный листингом 16.5

Листинг 16.6. Чтение данных из текстового файла в матрицу c

c := READPRN("datafile.prn")

$$c = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 99 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Листинг 16.7. Дозапись вектора k в существующий текстовый файл

$k := (1\ \ 2\ \ 3\ \ 4\ \ 5)$

APPENDPRN("datafile.prn") := k

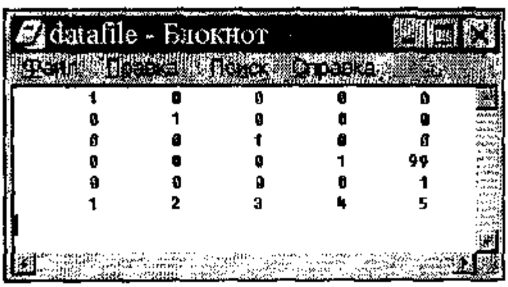


Рис. 16.61. Файл, созданный листингами 16.5 и 16.7

Обратите внимание, что если Вы выводите данные в файл, пользуясь встроенной функцией WRITEPRN, то в любом случае создается новый текстовый файл. Если даже до записи данных файл с таким именем существовал, то его содержимое будет уничтожено, заменившись новыми данными. Если Вы хотите сохранить прежнее содержимое текстового файла с данными, пользуйтесь функцией APPENDPRN. Эта встроенная функция может применяться и для создания нового файла. Иными словами, если файла с заданным именем не существовало, то он, после применения, будет создан и наполнен теми данными, которые Вами определены в документе.

16.6.2. Графические файлы

Подобно вводу-выводу в текстовые файлы, можно организовать чтение и запись данных в графические файлы различных форматов.

При этом данные отождествляются с интенсивностью того или иного цвета пиксела изображения, находящегося в файле. Перечислим основные функции:

- `READRGB("file")` — чтение цветного изображения;
- `READBMP("file")` — чтение изображения в оттенках серого;
- `WRITERGB("file")` — запись цветного изображения;
- `WRITEBMP("file")` — запись изображения в оттенках серого;
  - `file` — путь к файлу.

Примечание

Имеется также большое количество функций специального доступа к графическим файлам, например, чтение интенсивности цветов в других цветовых моделях (яркость-насыщенность-оттенок), а также чтение только одного из основных цветов и т. п. Вы без труда найдете информацию об этих функциях в справочной системе **Mathcad**.

- Действие функций доступа к графическим файлам иллюстрируется листингами 16.8—16.10. Заметим, что для создания изображения используется встроенная функция `identity`, создающая единичную матрицу. Изображение, созданное листингом 16.8, приведено на рис. 16.62.

Листинг 16.8. Запись матрицы I в графический файл

$I := identity(100) \cdot 100$

$I_{3,9} := 500$

$WRITEBMP("data.bmp") := I$

Листинг 16.9. Чтение из графического файла

$C := READBMP("data.bmp")$

Листинг 16.10. Запись в цветной графический файл

$R := identity(100) \cdot 100$

$G := identity(100)$

B := identity (100)

WRITERGB ("color.bmp") := augment (R , G , B)

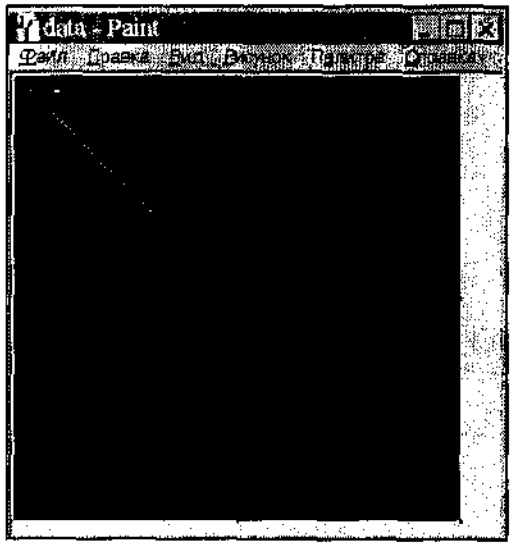


Рис. 16.62. Файл, созданный листингом 16.8

16.6.3. Звуковые файлы

В Mathcad версии 2001 появилась возможность записывать и считывать амплитуду акустических сигналов, записанных в звуковые файлы с расширением .wav:

- READWAV("file") — чтение звукового файла в матрицу;
- WRITEWAV("file" ,s,b) — запись данных в звуковой файл;
- GETWAVINFO("file") — создает вектор из четырех элементов с информацией о звуковом файле;
  - file — путь кфайлу;
  - s — скорость следования сэмплов, задаваемых матрицей;
  - b — разрешение звука в битах.

Использование этих встроенных функций позволяет организовать обработку звука.

ГЛАВА 17

Оформление документов

В этой главе рассматриваются приемы оформления результатов работы в Mathcad. Помимо того, что Mathcad является мощным математическим редактором, позволяющим проводить самые различные численные и символьные расчеты по формулам, в нем еще предусмотрены богатые возможности форматирования представления внешнего вида расчетов. Если уметь правильно пользоваться инструментами, имеющимися в Mathcad для оформления документов, то результаты работы можно подать в очень эффектной и математически понятной форме.

Основные возможности редактора Mathcad были рассмотрены в первой части книги *(см. гл. 2)*. В начале данной главы перечисляются различные элементы оформления, как встроенные, так и внешние, которые допускается применять в документах Mathcad для выделения областей *(см. разд. 17.1)*, шрифтового оформления текста и формул *(см. разд. 17.2)*, разметки страниц и установки колонтитулов *(см. разд. 17.3)*. Вспомогательными, но очень важными элементами оформления являются гиперссылки, позволяющие организовать оперативный обзор документов Mathcad *(см. разд. 17.4)*, а также рисунки, которые можно импортировать из внешних графических файлов *(см. разд. 17.5)*.

17.1. Элементы оформления документов

Расчеты в Mathcad могут быть оформлены по-разному:

- печатные материалы — документы, распечатанные на принтере;
- Web-страницы — документы, просматриваемые с помощью браузеров, которые могут быть размещены в Интернете;
- документы Mathcad — для представления аудитории с помощью самого приложения Mathcad;
- электронные книги — оформленные специальным образом интерактивные документы Mathcad, построенные по принципу, который можно наблюдать на примере различных Ресурсов Mathcad;

- фрагменты документов, экспортированные и оформленные в других приложениях (например в документах Microsoft Word или в презентациях Microsoft PowerPoint).

17.1.1. Элементы оформления

Перечислим элементы оформления документов, которые допускается применять в Mathcad как, собственно, для проведения математических расчетов, так и в чисто декоративных целях (рис. 17.1, сверху вниз):

- текстовые области (text region);
- математические области, или формулы (math region);
- графики или графические области (graphics region);
- компоненты других приложений (component);
- внедренные объекты (object).

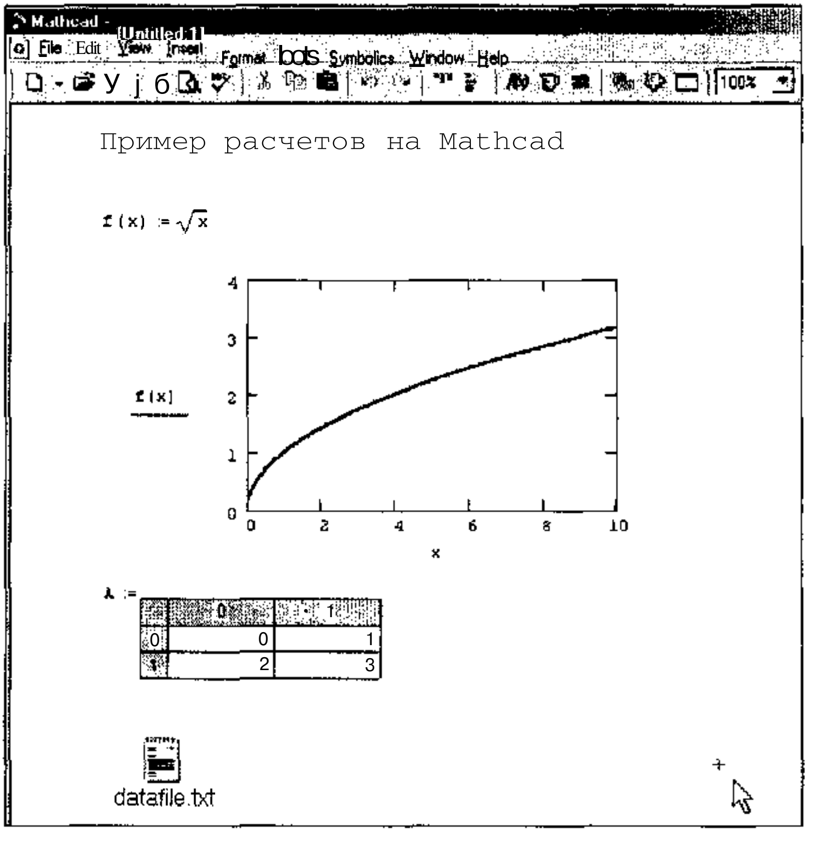


Рис. 17.1. Основные элементы оформления документов Mathcad

За пределами границ областей находится пустая часть документа. Кроме перечисленных, часто бывает полезным применение следующих дополнительных элементов оформления:

- закрытые и выделенные области (locked and highlighted area);
- колонтитулы (header, footer);
- разметка документов — разрывы страниц (page break), стили (styles) и поля (margins);
- ссылки (references);
- гиперссылки (hyperlinks);
- рисунки (pictures).

17.1.2. Размещение элементов оформления в документах

Важной составляющей оформления расчетов является правильное и понятное размещение объектов по документу Mathcad.

Вставка новой области

Для вставки того или иного элемента нужно предварительно выбрать место в документе, куда он будет вставлен. Это осуществляется с помощью курсора ввода (крестика, показанного на рис. 17.1 в правом нижнем углу, на который наведен указатель мыши). Затем следует воспользоваться соответствующим пунктом меню **Insert** (Вставка), либо одной из панелей инструментов, либо, как для ввода формулы, просто начать вводить символы с клавиатуры.

Примечание

Приемы вставки различных областей были рассмотрены выше в соответствующих разделах книги *(например, формулы* — в *гл. 2, графики* — в *гл. 15 и т. д.).*

Помните, что компоненты вставляются при помощи пункта меню **Insert / Component** (Вставка / Компонент), а внедренный объект можно вставить, поместив его в буфер обмена из области другого приложения и, после переключения в Mathcad, нажав сочетание клавиш <Ctrl>+<V>.

Перемещение областей по документу

Чтобы изменить место расположения любой области в документе Mathcad:

1. Щелкните в ее пределах мышью. После этого область будет выделена, а курсор, оказавшись внутри нее, приобретет форму линий ввода. Выделение различных элементов показано в виде коллажа на рис. 17.2.

2. Не нажимая кнопок, поместите указатель мыши на границу области, чтобы он сменил вид стрелки на форму руки.
3. Нажмите левую кнопку мыши и, удерживая ее, перетащите объект на новое место.

Внимание!

Помните о том, что порядок следования формул и графиков в документе влияет на расчеты. Вычисление формул производится в порядке слева направо и сверху вниз.

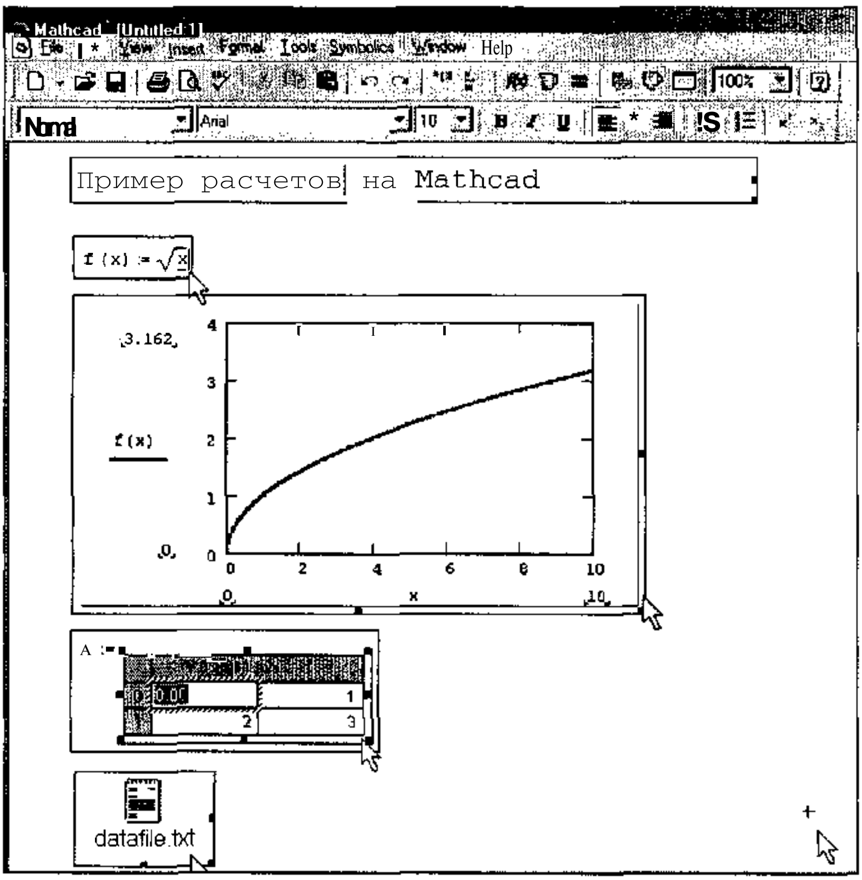


Рис. 17.2. Выделение основных элементов оформления (коллаж)

Чтобы создать копию области в другом месте документа, начните перетаскивать ее обычным образом, а затем нажмите и удерживайте, вплоть до отпускания клавиши мыши там, где хотите поместить область, клавишу <Ctrl>.

Изменение размера областей

Рассматривая рис. 17.2, можно заметить, что по сторонам всех областей, за исключением математических, при выделении располагаются черные прямоугольники, называемые маркерами или ручками (handle). С их помощью осуществляется растягивание или сжатие областей в соответствующих направлениях. Если поместить указатель мыши на ручку, то он приобретет форму двойной стрелки, перетаскивая которую можно изменить размер области.

Размер формул изменить таким образом нельзя. Для форматирования (в том числе, управления размером и типом шрифта) как формул, так и текста в текстовых областях, следует пользоваться панелью **Formatting** (Форматирование) *(см. разд. 17.2)*.

Примечание

Помните о том, что изменение масштаба представления документа в меню **View / Zoom** (Вид / Масштаб) влияет только на его экранное представление внутри Mathcad, но не сохраняется в файле и не влияет на распечатку на принтере.

Разделение областей

Часто при работе с Mathcad, по мере перетаскивания областей с места на место, оказывается, что одни из них перекрываются другими и перестают быть видны на экране. Чтобы разделить области, примените удобную возможность, предусмотренную разработчиками Mathcad.

1. Выделите группу областей протаскиванием через них указателя при нажатой кнопке мыши (рис. 17.3).
2. Выберите команду **Format / Separate Regions** (Формат / Разделить регионы).

В результате области в документе будут разделены как по вертикали, так и по горизонтали.

Примечание

Если Вы по какой-либо причине считаете, что некоторые регионы должны в документе перекрываться, то порядком их наложения друг на друга можно управлять при помощи команд контекстного меню **Bring to Front** (На передний план) или **Send to Back** (На задний план).

Удаление области

Универсальный метод удаления области целиком — выбор в главном меню пункта **Edit / Delete** (Правка / Удалить). Различные области можно также удалить многократным нажатием клавиш <Del> или <BackSpace>, однако эти клавиши используются, в основном, для удаления содержимого внутри областей.

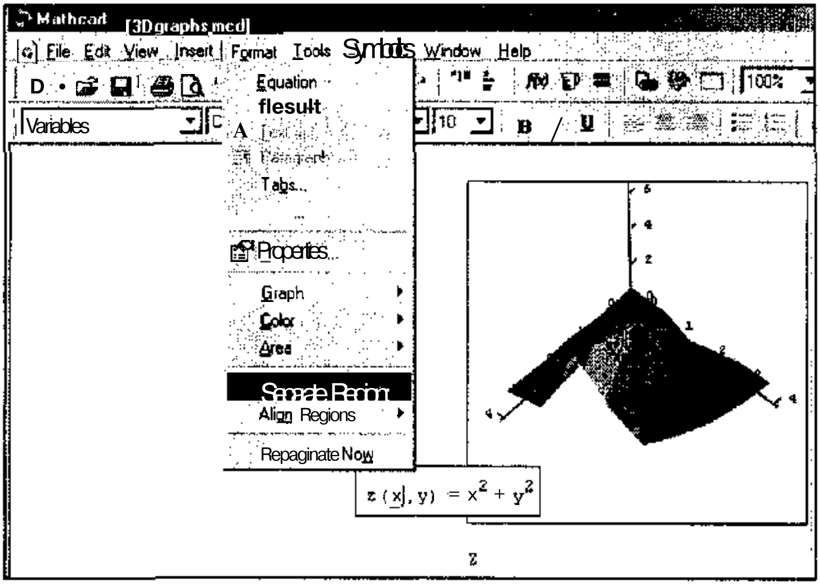


Рис. 17.3. Разделение областей

> **Примечание**
>
> Редактор Mathcad, в силу математической специфики, **отличается** от большинства текстовых процессоров, с которыми, как правило, пользователи не испытывают трудностей. Встретив сложности при работе **с** редактором, что не является редкостью, обратитесь к первой части книги *(см. гл. 2).*

17.1,3. Выделение областей

В документах Mathcad можно выделять некоторые области цветом или обрамлением.

Выделение области цветом

Чтобы выделить область цветом, вызовите нажатием на ней правой кнопкой мыши контекстное меню и выберите в нем пункт **Properties** (Свойства), либо выберите такой же пункт в меню **Format** (Формат). Установите в диалоговом окне **Properties** (Свойства) флажок **Highlight Region** (Выделить цветом) и нажмите кнопку **OK** (рис. 17.4). Область будет выделена цветом, по умолчанию желтым (текст, показанный в левой верхней части рис. 17.4, выделен серым).

После установки флажка **Highlight Region** (Выделить цветом) становится доступной кнопка **Choose Color** (Выбрать цвет), с помощью которой можно выбрать любой другой цвет выделения из палитры. Именно так была выделена формула на рис. 17.4.

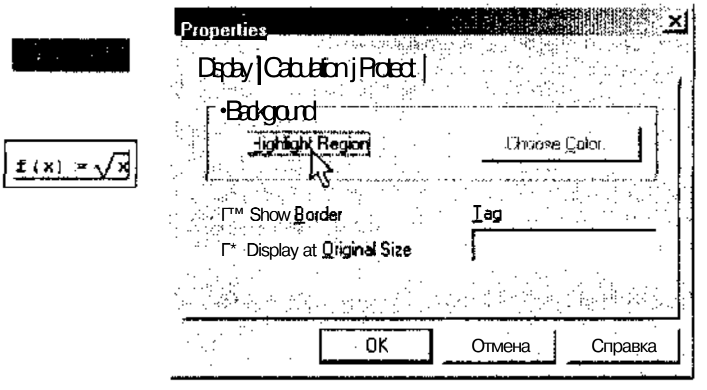


Рис. 17.4. Выделение области цветом

Примечание

Многие формулы из **Mathcad Resources** выделены цветом. После того как Вы скопировали какие-либо формулы в документ, выделение можно убрать снятием флажка проверки в диалоговом окне **Properties** (Свойства).

Для того чтобы задать цвет фона всего документа, выберите команду Format / Color / Background (Формат / Цвет / Фон) и определите в палитре понравившийся Вам цвет.

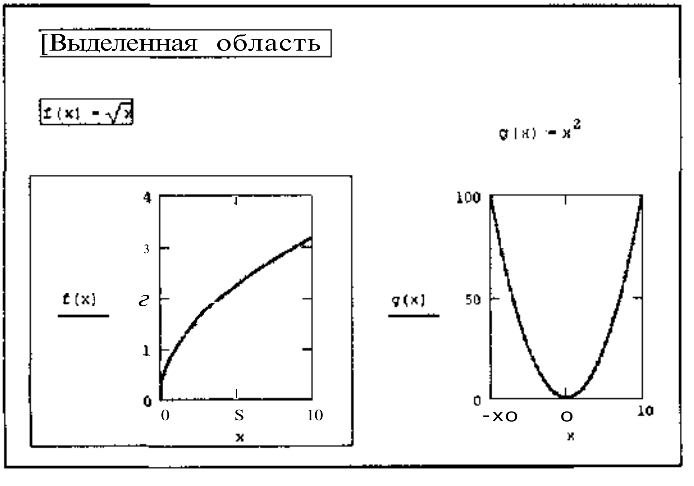


Рис. 17.5. Области с обрамлением (слева) и без обрамления (справа)

Выделение области обрамлением

Выделить область можно не только цветом, но и обрамлением (рис. 17.5). Для включения обрамления установите флажок Show Border (Показать рамку) в том же самом диалоговом окне **Properties** (Свойства). Обрамление может применяться вместе с выделением цветом (так выделена текстовая область на рис. 17.5).

Примечание

Напомним, что при помощи закладки **Calculations** (Вычисления) того же самого диалога **Properties** (Свойства) можно выключить отдельные формулы из процесса вычислений *(см. гл. 3).* Такие формулы отображаются в документе с прямоугольной точкой в углу (рис. 17.6).

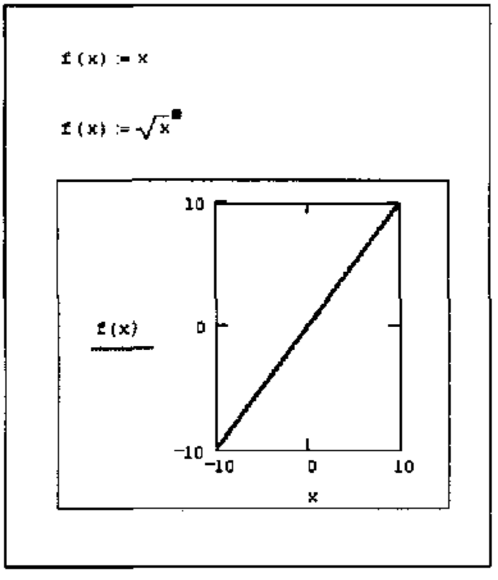


Рис. 17.6. Вычисление формулы в центре выключено

17.1.4. Работа с зонами

Участки документа Mathcad можно объединять в зоны (area). Часто их в русском переводе называют областями, что создает путаницу с термином *область,* который в данной книге эквивалентен понятию *регион* (region). Например, в этой и других главах мы говорим о текстовой или математической области. Собственно говоря, зоны включают в себя различные области.

Зоны могут понадобиться для следующих целей:

- разграничение участков документа по смыслу;
- временное скрытие участков документов;
- запирание участков документов таким образом, чтобы несанкционированный доступ к ним был запрещен другим пользователям, за исключением одобренных разработчиком документа.

> **Совет**
>
> Зоны могут использоваться при создании обучающих систем на основе Mathcad для организации областей, которые пользователям не рекомендуется или вовсе запрещается модифицировать.

Создание зоны

Чтобы создать новую зону в документе, достаточно поместить курсор ввода в желаемое место и выбрать в верхнем меню пункт Insert / **Area** (Вставить / Область). В результате, в выбранном месте документа появится пара горизонтальных линий, отмеченных у левого края значком в виде черного треугольника (рис. 17.7). Часть документа, оказавшаяся между этими линиями, и образует зону.

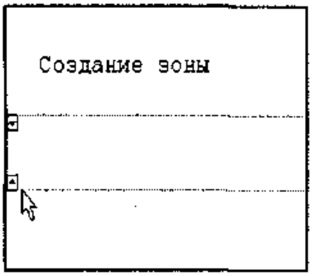


Рис. 17.7. Создание зоны

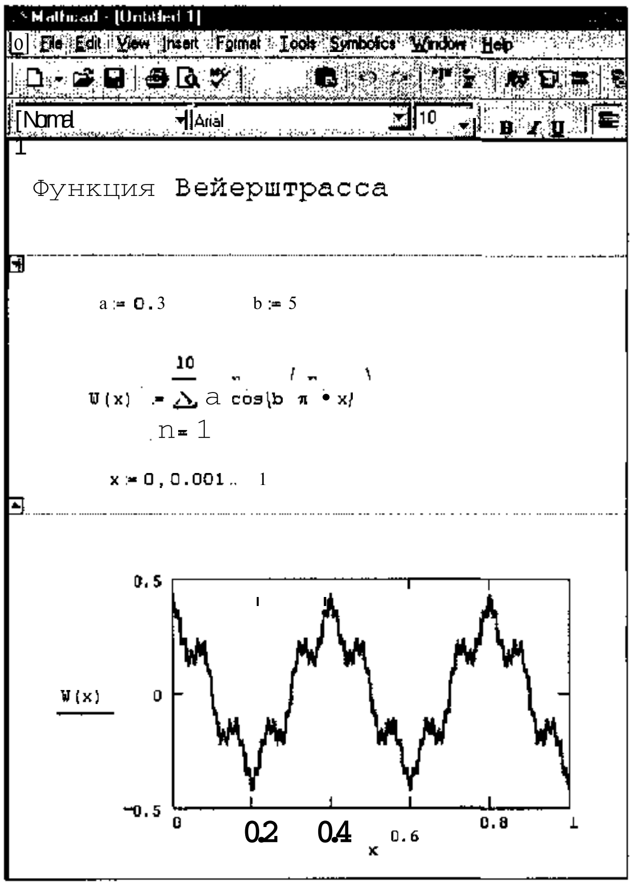


Рис. 17.8. Помещение формул внутрь зоны

Изменить размеры зоны можно в любой момент, щелкнув на любой из горизонтальных линий (в результате чего она будет выделена) и передвинув ее в новое положение. Для удаления зоны из документа выделите щелчком мыши любую из горизонтальных линий и нажмите клавишу <Del>.

Чтобы поместить формулу внутрь зоны, просто перетащите ее туда указателем мыши (рис. 17.8).

Скрытие зоны

Чтобы скрыть зону, дважды щелкните мышью на любой из линий, ее выделяющих. Альтернативный способ заключается в помещении курсора внутрь зоны и выполнении команды **Format** / **Area** / **Collapse** (Формат / Область / Скрыть). Сразу после этого содержимое зоны будет убрано с экрана, но по-прежнему будет участвовать в расчетах (рис. 17.9).

Вернуть зону на экран можно точно так же: двойным щелчком на линии, показывающей наличие скрытой зоны (рис. 17.9), либо выбором команды **Format / Area / Expand** (Формат / Область / Раскрыть).

Совет

Применение скрытых зон эффективно в документах большого размера.

Рис. **17.9.** Скрытие зоны

Запирание зоны

Зоны можно не только скрывать и раскрывать, но и запирать или *блокировать*, т. е. закрывать для любого редактирования. Для запирания зоны выберите команду **Format** / **Area** / **Lock** (Формат / Область / Запереть). В появившемся диалоговом окне (рис. 17.10) следует нажать кнопку OK. Для того чтобы запретить несанкционированный доступ к редактированию зоны другим пользователям, предварительно введите пароль **(Password)** в верхнем текстовом поле, повторите ввод пароля в нижнем текстовом поле, дабы избежать возможных ошибок, и затем нажмите кнопку OK.

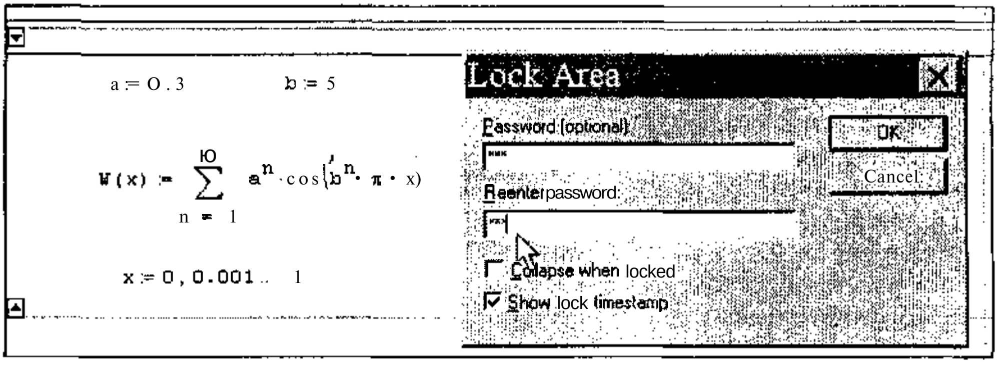


Рис. 17.10. Блокирование зоны

Чтобы открыть или *разблокировать* зону, выберите в главном меню пункт **Format / Area / Unlock** (Формат / Область / Открыть). Если зона была заперта с защитой паролем, то этот пароль будет запрошен при разблокировании зоны (рис. 17.11). Запертая зона отображается со значками в форме замков в левом углу горизонтальных линий, ее обрамляющих.

На рис. 17.11 вы видите запертую зону с информацией о времени ее блокирования, поскольку в диалоге **Lock Area** (Запереть область) был установлен флажок **Show lock timestamp** (Показать время блокировки). Если в диалоге Lock Area установить другой флажок — **Collapse when locked** (Скрыть при блокировке), то запертая зона будет скрыта с экрана.

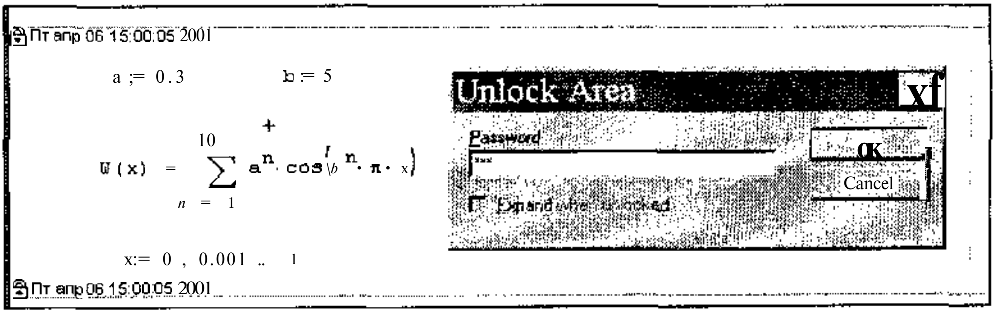


Рис. 17.11. Разблокирование зоны

17.2. Форматирование текста и формул

Для форматирования текста и формул служит панель инструментов Formatting (Форматирование). С ее помощью текстовые области можно форматировать двумя способами.

- Применять к ним текстовые стили, что сказывается на изменении формата текстовой области целиком *(см. разд. 17.2.1).*
- Форматировать отдельные элементы текста.

Для применения стиля к текстовому региону или формуле используется поле с раскрывающимся списком стилей на панели Formatting (Форматирование) (на него наведен указатель мыши на рис. 17.12). Все элементы управления, расположенные на этой панели правее списка со стилями, служат для форматирования отдельных частей текста. Особенность форматирования ими формул заключается в том, что при изменении шрифта отдельной формулы изменяется соответствующий параметр математического стиля во всем документе.

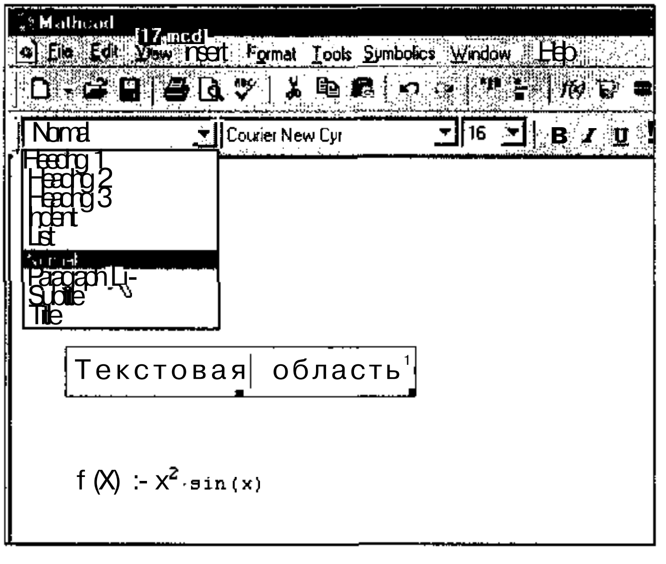


Рис. 17.12. Список стилей форматирования текста

Форматирование текста (и, отчасти, формул) в Mathcad во многом похоже на форматирование в большинстве текстовых редакторов, поэтому остановимся коротко на возможностях шрифтового оформления и редактировании параметров абзацев и страниц. Разберем сначала второй способ, относящийся к форматированию текста, чтобы ясно представлять соответствующие возможности Mathcad.

17.2.1. Форматирование текста

Форматирование текста заключается в управлении его двумя основными составляющими:

- форматом шрифта;
- форматом абзаца.

Шрифт

Шрифт выщеленного текста можно поменять при помощи панели Formatting (Форматирование) (рис. 17.13), либо при помощи диалогового окна Text Format (Формат текста) (рис. 17.14). Чтобы вызвать это диалоговое окно, следует выбрать пункт Text (Текст) в верхнем меню Format (Формат) или пункт Font (Шрифт) в контекстном меню. Перечислим параметры шрифта и соответствующие элементы этой панели, которыми допускается управлять:

- Font (Шрифт) — поле в окне Text Format (Формат текста);
- Size (Размер) — поле в окне Text Format (Формат текста) (на аналогичное поле панели форматирования (рис. 17.13) наведен курсор);

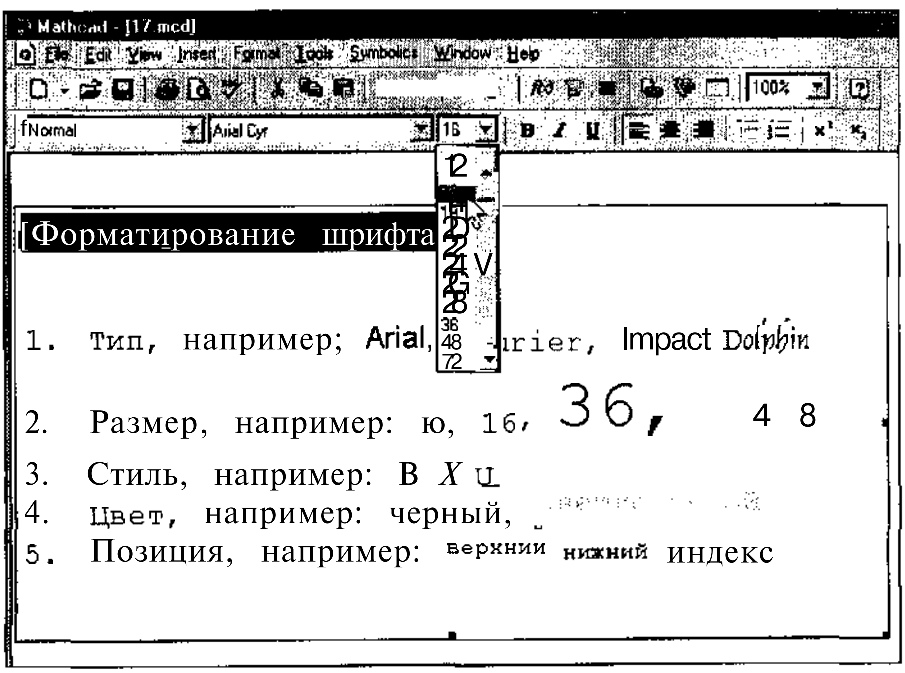


Рис. 17.13. Примеры шрифтового оформления текста

- Font Style (Стиль шрифта) — поле в окне Text Format (Формат текста), ему соответствуют кнопки на панели форматирования:
  - Bold (Полужирный);
  - Italic (Наклонный);
  - Underlined (Подчеркнутый).
- Strikeout (Зачеркнутый) — флажок в диалоге Text Format;
- Color (Цвет) — поле со списком в диалоге Text Format;
- Position (Позиция) — флажки в диалоге Text Format:
  - **Superscript** (Верхний индекс);
  - Subscript (Нижний индекс).

Примеры форматирования шрифта показаны на рис. 17.13.

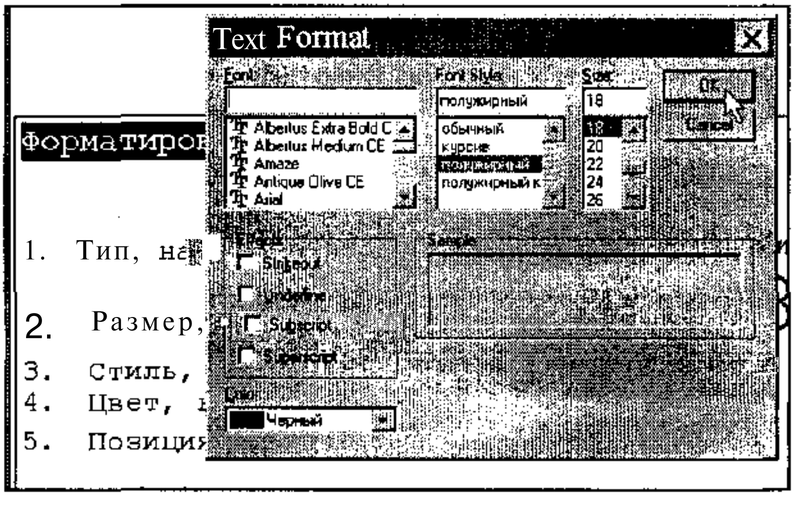


Рис. 17.14. Диалоговое окно Text Format

Абзац

Для установки параметров абзаца применяются:

- абзацный отступ — три маркера на линейке в верхней части экрана задают левую границу первой строки абзаца (верхний левый маркер) и его остальных строк (нижний левый маркер), а также правую границу абзаца (правый маркер);
- нумерованный и маркированный списки — две крайние правые кнопки на панели Formatting (Форматирование);

- выравнивание — задается кнопками на панели форматирования (рис. 17.15):
  - по левому краю;
  - по центру;
  - по правому краю.

Все параметры абзаца можно изменить и в диалоговом окне Paragraph Format (Формат абзаца), которое вызывается выбором пункта Paragraph (Абзац) меню Format (Формат) или одноименного пункта контекстного меню (рис. 17.16).

Примечание

Дополнительная и реже применяемая возможность форматирования текста — установка. Инструкции по установке знаков табуляции Вы найдете в справочной системе Mathcad.

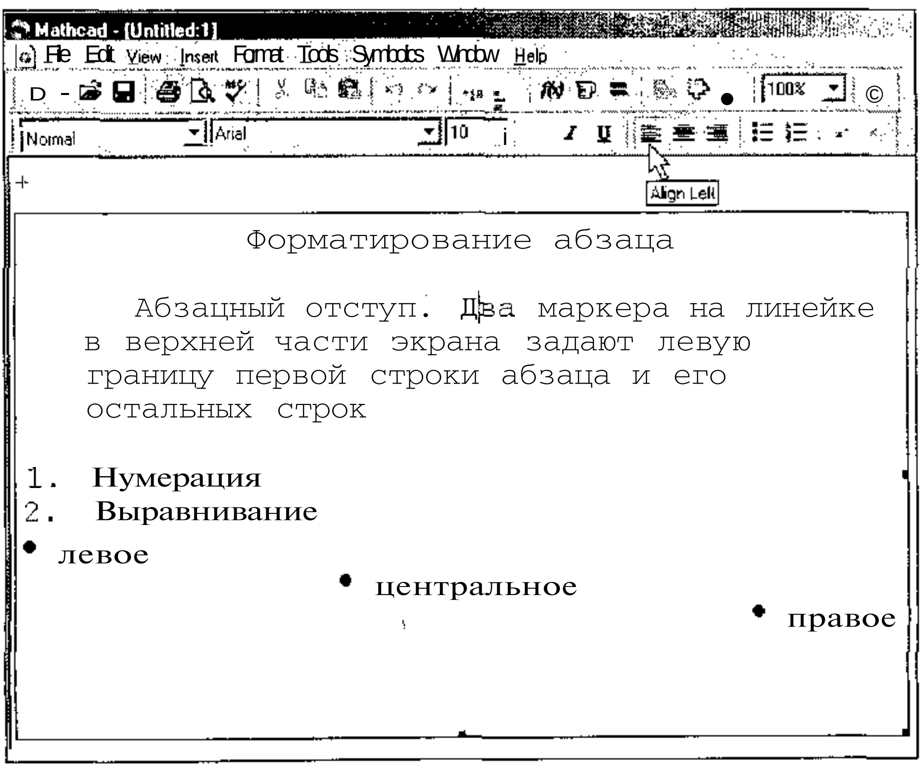


Рис. 17.15. Примеры оформления абзаца

16 Зак 984

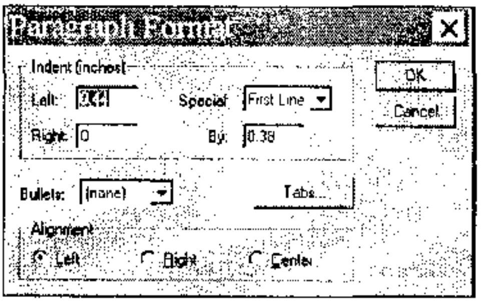


Рис. 17.16. Диалоговое окно Paragraph Format

17.2.2. Стили текста и формул

Когда Вы начинаете вводить текст в текстовый регион или формулы в математический регион, формат шрифта и абзаца выбирается в соответствии со *стилями,* выбранными по умолчанию и сохраненными в шаблоне документа.

Текстовый стиль содержит информацию обо всех установках шрифта и абзаца текстового региона, а математический стиль — об установках только шрифта различных элементов формул (но не абзаца, т. к. каждая формула по умолчанию не может занимать более одной строки).

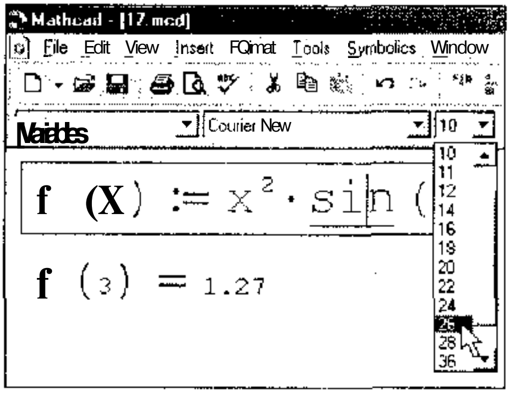


Рис. 17.17. Изменение шрифта элементов формулы, выполненных в стиле переменных

Форматирование формул

Для математических регионов можно применять все рассмотренные способы форматирования шрифта при помощи панели Formatting (Форматирование). Особенность форматирования формул заключается в том, что изменения шрифта, примененные к отдельному параметру в одной формуле, немедленно приводят к его изменению во всех формулах в документе (там, где этот параметр присутствует) (рис. 17.17, 17.18). При этом следует знать, что формулы содержат элементы, выполненные в нескольких математических стилях. К примеру, на рис. 17.17 изменяется шрифт составляющих всех формул документа, выполненных в стиле Variables (Переменные), а па рис. 17.18— шрифт элементов формул и стиле Constants (Константы). О принадлежности редактируемой части формулы к тому или иному стилю можно судить по имени стиля, отображаемому в левом углу панели Formatting (Форматирование).

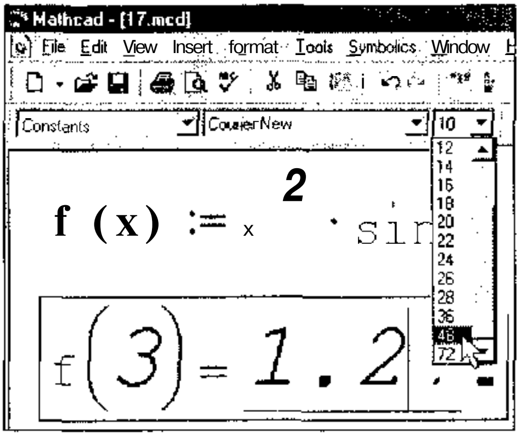


Рис. 17.18. Изменение шрифта элементов формулы, выполненных в стиле констант

Применение стиля к формуле или тексту

Изменить форматирование формулы или абзаца текстового региона целиком можно с помощью применения к нему стиля. Для этого выделите абзац или формулу, а затем выберите из списка стилей стиль, который Вы желаете применить к формуле или абзацу текста (см. рис. 17.12).

Изменение стиля

Для того чтобы изменить установки текстового или математического стиля или создать новый стиль пользователя, выберите в меню Format пункт Style (Стиль) или Equation (Формула). Рассмотрим изменение текстового стиля (рис. 17.19). В диалоговом окне Text Style (Стиль текста) нажмите кнопку Modify (Изменить) и в появившемся новом диалоговом окне Define Style (Определение стиля) отредактируйте параметры шрифта и абзаца, которые будут присущи данному стилю. Текущее описание стиля можно наблюдать в нижней части диалога Define Style (Определение стиля) в рамке Description (Описание).

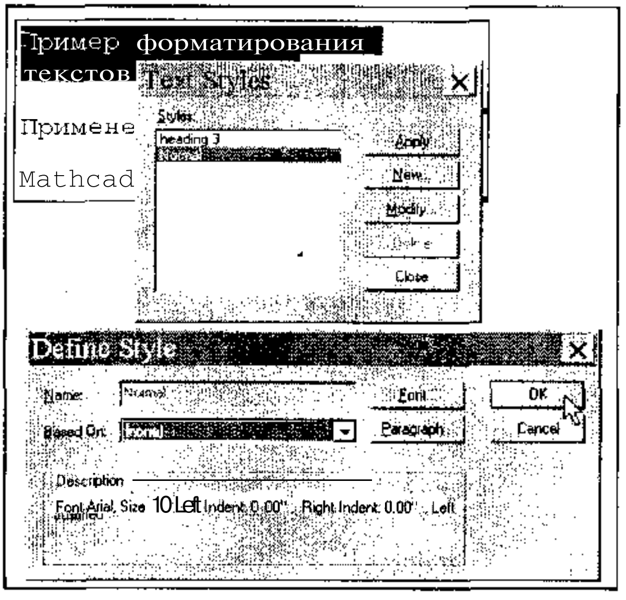


Рис. 17.19. Изменение текстового стиля

Применение стиля к формуле

Иногда требуется применить стиль шрифта к одной из переменных или чисел, чтобы она отличалась от остальных. Для этого измените математический стиль переменной. Выполните следующие действия (рис. 17.20):

1. Щелкните на имени переменной или числе.
2. Выберите команду Format / Equation (Формат / Формула).
3. В диалоге Equation Format (Формат формулы) выберите стиль формулы в списке стилей Style Name (на него наведен указатель мыши на рис. 17.20).
4. Если требуется поменять какие-либо установки шрифта, задающие стиль, отредактируйте их, нажав кнопку Modify (Изменить).
5. Нажмите кнопку OK.

В результате шрифт переменной будет отформатирован в соответствии с выбранным стилем.

Внимание!

Одноименные переменные, записанные в разном стиле, являются разными переменными! Если Вы желаете поменять математический стиль переменной, меняйте его везде, где переменная встречается в документе. Соответствующий пример приведен на рис. 17.21, на котором переменная x в разных стилях по-разному идентифицируется Mathcad.

Если стиль не нужно изменять, применить его к формуле можно более просто, выбрав из списка стилей на панели форматирования.

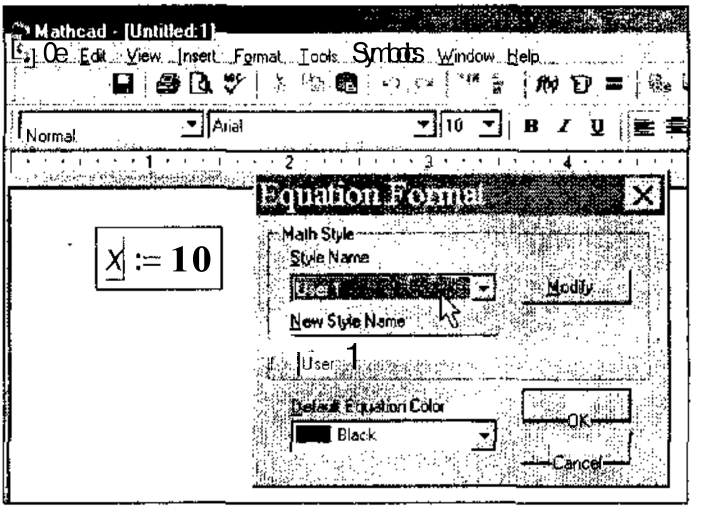


Рис. 17.20. Изменение стиля переменной

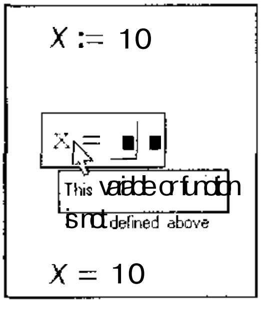


Рис. 17.21. Стиль влияет на идентификацию переменной Mathcad

17.3. Оформление страниц

Mathcad имеет некоторый набор средств для оформления страниц в целом, которые можно разделить на управление параметрами разметки страницы и на создание верхних и нижних колонтитулов.

17.3.1. Параметры страницы

Расположение документа на странице при распечатке его на принтере задается в диалоговом окне Page Setup (Параметры страницы) (рис. 17.22), которое вызывается одноименной командой меню **File** (Файл). В этом диалоговом окне определяются следующие параметры:

- **Size** (Размер страницы)
- Source (Тип подачи бумаги)
- **Orientation** (Ориентация)
  - **Portrait** (Вертикальная)
  - **Landscape** (Горизонтальная)
- **Margins (Поля)**
  - **Left** (Левое)
  - **Right** (Правое)
  - **Top** (Отступ сверху)
  - **Bottom** (Отступ снизу)

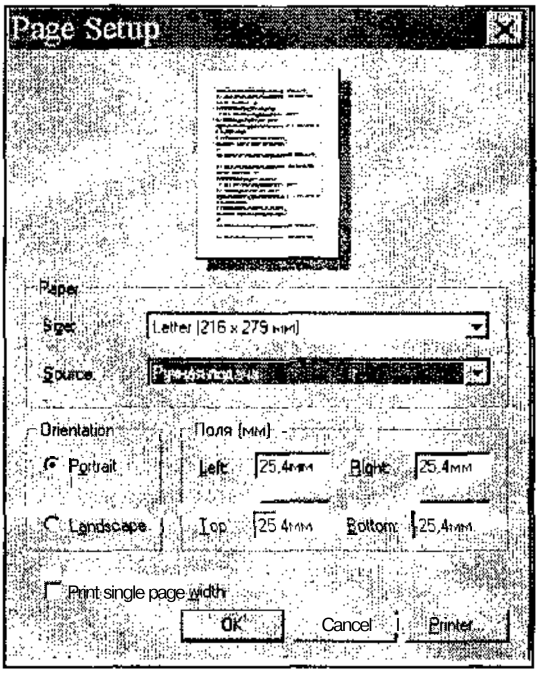


Рис. 17.22. Параметры страницы

При изменении какого-либо параметра его влияние можно оценить в области предварительного просмотра и верхней масти диалогового окна, в которой изображается макет печатной копии страницы. Кроме того, установка полей и диалоговом окне **Page Setup** (Параметры страницы) влияет на положение линий раздела границ, которые Вы видите на рабочей области документа Mathcad.

17.3.2. Колонтулы

Колонтитулами называют элементы оформления документа, которые появляются в унифицированном виде на каждой странице печатной копии документа. Чтобы вставить колонтитулы в документ:

1. Выберите пункт Headers/Footers (Колонтитулы) меню View (Вид).
2. В зависимости от типа колонтитула перейдите в диалоге Headers/Footers (Колонтитулы), изображенном на рис. 17.23, на одну из вкладок: Header (Верхний колонтитул) или Footer (Нижний колонтитул).

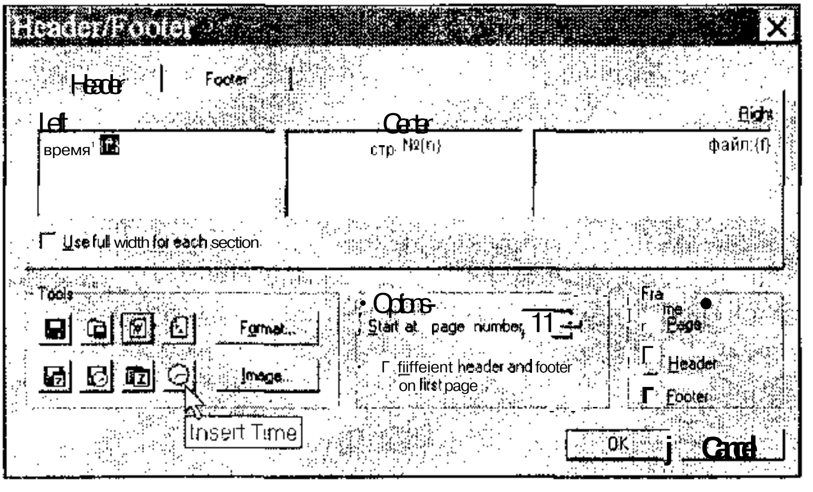


Рис. 17.23. Установка колонтитулов

3. Щелкните в одном из трех текстовых полей, в зависимости от того, куда Вы желаете поместить колонтитул: слева, в центр или справа.
4. Вставьте текст колонтитула, комбинируя его с информацией, которая может быть вставлена автоматически (дата, номер страницы, имя файла или т. п.). Для ввода этой информации просто нажмите одну из кнопок с соответствующим значком, которые находятся в нижней левой части диалогового окна.
5. При необходимости повторите пп. 3—4 для разных колонтитулов.

Примечание

Колонтитулы влияют только на вид печатных копий документа. Просмотреть их можно, к примеру, в режиме предварительного просмотра, выбрав команду **File/ Print Preview** (Файл/ Предварительный просмотр), как показано на рис. 17.24.

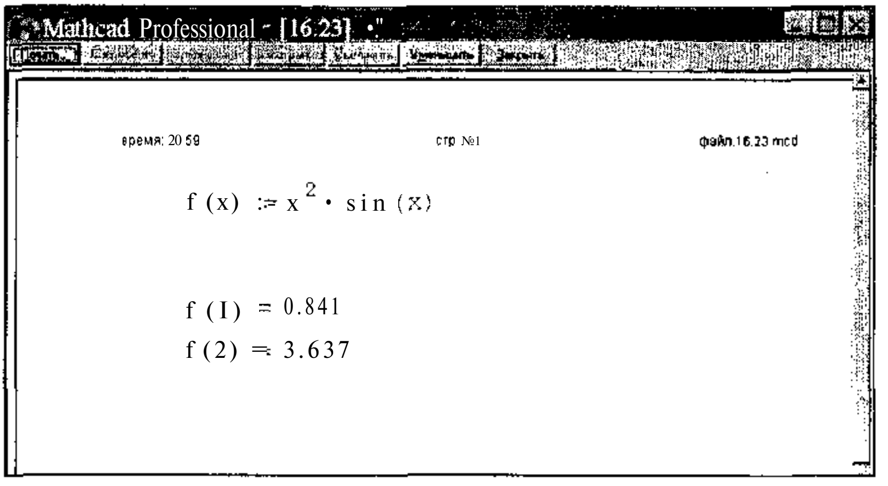


Рис. 17.24. Внешний вид колонтитулов в режиме предварительного просмотра страницы

Перечислим кнопки, которые осуществляют автоматическую вставку информации в колонтитул. В фигурных скобках отображен символ, который определяет соответствующую информацию в поле колонтитула.

- **File Name** {f} {Имя файла);
- **File Path** {p} (Путь к файлу);
- **Page Number** {n} (Номер страницы);
- **Number of Pages** {nn} (Число страниц);
- **Date Last Saved** {fd} (Дата последнего сохранения);
- **Time Last Saved** {ft} (Время последнего сохранения);
- **Date** {d} (Текущая дата);
- **Time** {t} (Текущее время).

Примечание

Задать начало нумерации страниц с любой цифры можно в поле Start at page number (Начать нумерацию) в диалоге Headers/Footers (Колонтитулы).

17.3.3. Установки документа

Основные элементы оформления документа сохраняются в его *установках* (settings). Они автоматически сохраняются вместе с содержимым в Mathcad-файле и могут быть использованы в качестве установок по умолчанию при создании нового документа на основе шаблона *(см. разд. "Создание документа на основе шаблона" гл. 2).*

Перечислим установки документа:

- свойства текста по умолчанию;
- определение всех текстовых и математических стилей;
- колонтитулы;
- установка полей для печати документов;
- численный формат результатов вычислений;
- значения встроенных переменных;
- основные размерности переменных;
- система исчисления по умолчанию;
- режим вычислений по умолчанию.

17.4. Ссылки и гиперссылки

Гиперссылки — это активные области в документах Mathcad, которые выводят на экран какое-либо другое место в активном документе, другой документ Mathcad или другого приложения, либо на сайт в Интернете. Гиперссылки эффективны в больших документах, а также в обучающих и презентационных системах, выполненных в Mathcad. Особенно важно уметь пользоваться гиперссылками, если Вы разрабатываете электронные книги.

17.4.1. Установка тега

Прежде чем определить гиперссылку, можно сначала точно определить место в документе, на которое эта гиперссылка будет переводить курсор и которое в Mathcad называется *тегом* (tag). Для установки тега:

1. Щелкните на том месте, где Вы хотите расположить тег, правой кнопкой мыши.
2. В контекстном меню выберите пункт **Properties** (Свойства).
3. В диалоге **Properties** (Свойства) перейдите на вкладку **Display** (Отображение).
4. В поле **Tag** (Тег) введите имя тега, которое будет идентифицировать данное место в документе (рис. 17.25).
5. Нажмите кнопку OK.

17.4.2. Вставка гиперссылки

Создавать гиперссылку можно в любом месте любого документа. Щелчок на гиперссылке будет переводить курсор на место тега, на который установлена гиперссылка, либо (при отсутствии тега) в начало документа, на который произведена гиперссылка. Для вставки гиперссылки;

1. Щелкните на текстовой или формульной области документа, которую Вы хотите сделать гиперссылкой.

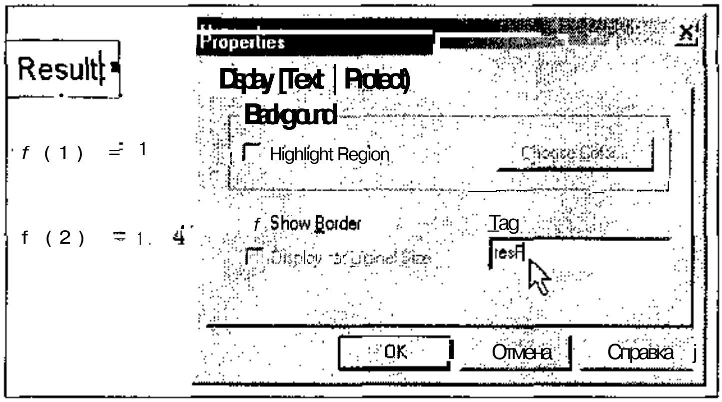


Рис. 17.25. Установка тега

2. Выберите в меню **Insert** (Вставка) пункт **Hyperlink** (Гиперссылка).
3. В диалоговом окне **Edit Hyperlink** (Правка гиперссылки), в текстовом поле **Link to file or URL** (Связать с файлом или URL (Universal Resource Locator, универсальный адрес ресурсов)) определите путь к документу, на который Вы собираетесь ссылаться, а также (необязательно) имя тега (`tagename`) С указанием имени файла (filename) В формате `filename#tagname` (рис. J7.26).
4. По желанию, в нижнем текстовом поле задайте текст, который будет появляться на строке состояния при наведении указателя мыши на гиперссылку.
5. Нажмите кнопку OK.

Примечание

Имя файла необходимо указывать, даже если тег расположен в том же документе, что и гиперссылка.

Если Вы все сделали правильно, то при двойном щелчке на гиперссылке будет осуществлен переход на место, где расположен тег, т. е. в нашем примере на документ 17.25.mcd. Чтобы отредактировать гиперссылку, достаточно, находясь на ее области, выбрать тот же пункт меню **Insert/ Hyperlink** (Вставка/ Гиперссылка). Появится диалог **Edit Hyperlink** (Правка гиперссылки), в котором можно исправить ее параметры (см. рис. 17.26). Удалить гиперссылку можно нажатием кнопки **Remove Link** (Удалить гиперссылку).

Помимо гиперссылок на документы Mathcad, допускается создавать гиперссылки на другие файлы (например видеофайлы или HTML-файлы), в том числе, находящиеся в Интернете. Для этого достаточно указать соответствующий адрес URL в верхнем текстовом поле диалога **Edit Hyperlink** (Вставка гиперссылки).

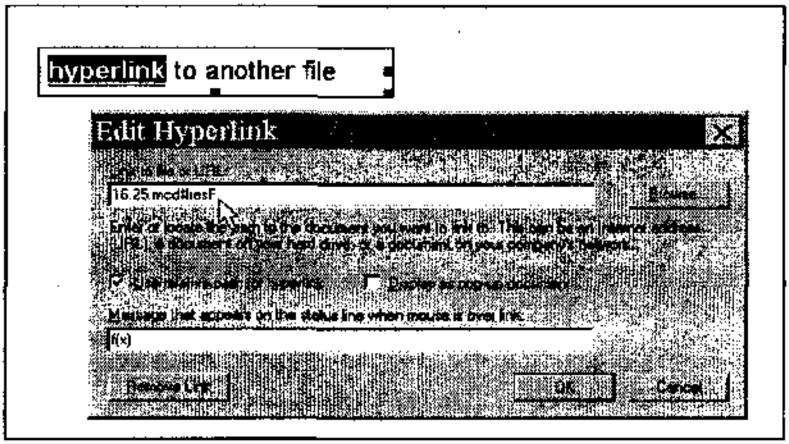


Рис. 17.26. Правка гиперссылки

17.4.3. Ссылки

Помимо гиперссылок, иногда стоит применять другие схожие с ними объекты, называемые *ссылками* (reference). Ссылка на документ А, вставленная в некоторое место документа В, приводит к расчету всего документа А внутри документа В. Таким образом, ссылки позволяют хранить вложенные друг в друга расчеты в разных файлах.

Совет

Ссылки могут понадобиться, если группа разработчиков решает одну большую задачу. В этом случае после распределения задач и соглашения об именах глобальных и локальных переменных каждый разработчик создает свой файл с расчетами.

Для установки ссылки достаточно выбрать команду **Insert / Reference** (Вставка / Ссылка) и затем в диалоговом окне **Insert Reference** (Вставка ссылки) определить путь к имени файла-ссылки. В примере, показанном на рис. 17.27, в файле отсутствует описание функции f(x), зато оно есть в файле, на который оформляется ссылка. Поэтому после нажатия кнопки ОК на месте курсора ввода появится информация о файле-ссылке, а результат функции `f(1)` будет рассчитан в соответствии с формулами этого файла.

17.5. Рисунки

Mathcad 11 имеет средства оформления документов, позволяющие вставлять и редактировать рисунки, сохраненные в файлах самых разных графических форматов. Эти средства придают Mathcad 11 основные функции графического редактора.

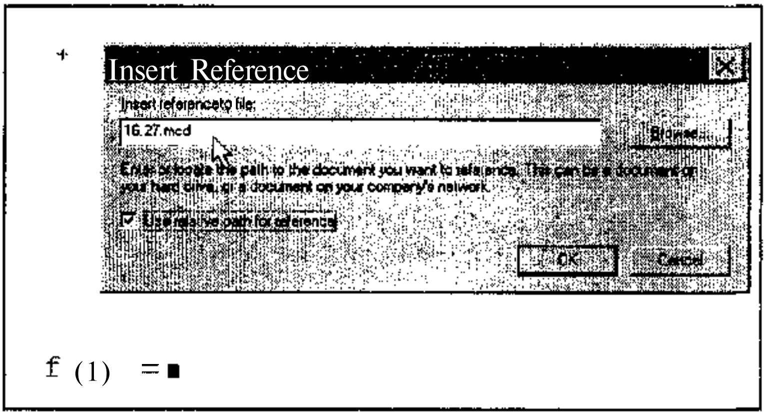


Рис. 17.27. Создание ссылки

Для вставки рисунка в документ:

1. Сохраните его в файле и поместите этот файл в ту же папку на диске, что и документ Mathcad.
2. Если панели **Matrix** (Матрица) нет на экране, вызовите ее.
3. Нажмите кнопку **Picture** (Рисунок) на панели **Matrix** (Матрица) (рис. 17.28).

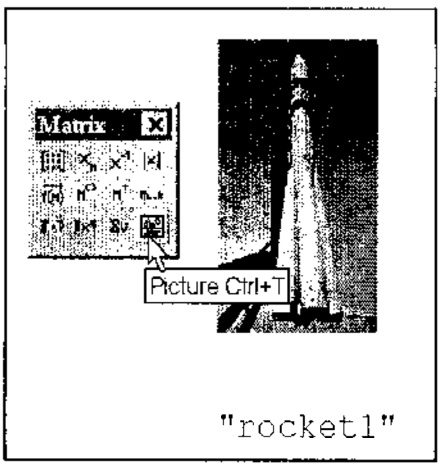


Рис. 17.28. Вставка области с рисунком

4. В местозаполнитель появившейся области введите в кавычках имя файла.

В примере, показанном на рис. 17.28, рисунок был сохранен в файле rocket1.gif. В результате содержимое графического файла появится в области рисунка.

Примечание

Если ввести в местозаполнитель имя какой-либо определенной ранее в документе матрицы, то созданный рисунок отразит строение этой матрицы. Данная опция очень эффективна для визуализации матриц большого размера, особенно разреженных.

Как только пользователь выделит рисунок, щелкнув на нем мышью, на экран автоматически будет вызвана панель инструментов **Picture** (Рисунок) (рис. 17.29). Она позволяет редактировать рисунок, применяя довольно развитые графические средства, например, отражение рисунка, увеличение его фрагмента и т. п. Назначение большинства кнопок на панели **Picture** (Рисунок) совпадает с наиболее известными графическими редакторами. Вставка областей с рисунками позволяет оформить документы более эффектно.

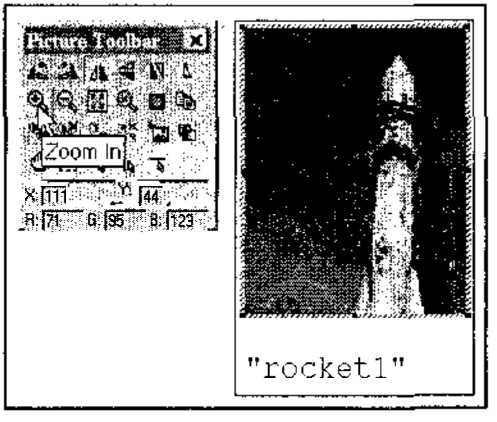


Рис. 17.29. Редактирование рисунка при помощи панели **Picture**

ПРИЛОЖЕНИЕ 1

Новые возможности Mathcad 2001 и 2001 i

- Улучшенный процессор, позволяющий проводить высокоскоростные расчеты, т. н. *режим ускоренных вычислений* (higher speed calculation). Оптимизация вычислений, улучшенное вычисление неопределенных выражений типа `0·ln(0)` и предварительная проверка матриц на сингулярность *(см. разд. 3.3.6)*.
- Повышенная Web-интеграция, совместимость с математическим браузером Techexplorer Hypermedia Browser компании IBM *(см. разд. 2.1.3)*.
- Возможности организации гиперссылок, в том числе из одного региона на другой, что осуществляется расстановкой тегов *(см. разд. 17.4)*.
- Новые встроенные функции и возможности:
  - Функции преобразования Декартовых, сферических, цилиндрических координат *(см. разд. 10.10)*.
  - Новые функции регрессии, например, логарифмической и линейной специального вида *(см. разд. 15.2.3—15.2.4)*.
  - Семейство `lookup`-функций для выборки значений из матриц.
  - Функции оцифровки звука из звуковых файлов *(см. разд. 16.6.3)*.
  - Новый тип графика для подготовки классических гистограмм *(см. разд. 14.2.1)*.
  - Возможность представления числа в виде простой дроби *(см. разд. 4.3.1)*.
- Улучшенная связь с другими приложениями, благодаря вставке компонентов *(см. разд. 17.1)*.
- Новые мощные средства для вставки и редактирования рисунков, в том числе панель **Picture** (Рисунок) *(см. разд. 17.5)*.

- Новые главы справочной системы, адресованные разработчикам компонентов и электронных книг *(см. разд. 1.4 и 17.6)*.
- В Mathcad 2001i обеспечена полная поддержка работы в ОС Windows XP.
- В Mathcad 2001i введена новая встроенная функция для решения жестких систем обыкновенных дифференциальных уравнений Radau, не требующая явного ввода в качестве аргумента якобиана системы *(см. разд. 11.5.2)*.

ПРИЛОЖЕНИЕ 2

Команды меню и панели инструментов

Таблица ***П2.1.*** *Команды меню*

Меню	Команда	Перевод	Сочетание клавиш	Описание
File (Файл)	**New**	Создать	<Ctrl>+<N>	Создать новый документ
	Open	Открыть	<Ctrl>+<0>	Открыть существующий документ
	Close	Закрыть	<Ctrl>+<W>	Закрыть активный документ
	Save	Сохранить	<Ctrl>+<S>	Сохранить активный документ
	Save As	Сохранить как		Сохранить активный документ в другом файле
	Save As Web Page	Сохранить как Web-страницу		Сохранить копию активного документа в файле формата HTML
	Page Setup	Параметры страницы		Опции вывода активного документа на печать
	Print Preview	Просмотр		Предварительный просмотр на экране вывода на печать активного документа

17 Зак. 984

Таблица П2.1 (продолжение)

Меню	Команда	Перевод	Сочетание клавиш	Описание
	Print	Печать	<Ctrl>+<P>	Распечатать активный документ
	Send	Отправить		Отправить активный документ по электронной почте
	Exit	Выход		Завершение работы с Mathcad
Edit (Правка)	**Undo**	Отменить	<Ctrl>+<Z>	Отменить последнее действие
	Redo	Повторить	<Ctrl>+<Y>	Повторить последнее отмененное действие
	Cut	Вырезать	<Ctrl>+<X>	Вырезать выбранное выражение в буфер
	Copy	Копировать	<Ctrl>+<C>	Копировать выбранное выражение в буфер
	Paste	Вставить	<Ctrl>+<V>	Вставить выражение из буфера
	Paste Special	Специальная вставка		Вставить объект специального формата, находящийся в буфере
	Delete	Удалить	<Ctrl>+<D>	Удалить выбранный регион
	Select All	Выделить все	<Ctrl>+<A>	Выделить всю рабочую область
	Find	Найти	<Ctrl>+<F>	Поиск текста
	Replace	Заменить	<Ctrl>+<H>	Замена искомого текста другим
	Go to Page	Перейти к странице		Переход к другой странице
	Links	Ссылки		Управление связями OLE с другими приложениями

Таблица П2.1. (продолжение)

Меню	Команда	Перевод	Сочетание клавиш	Описание
	Object	Объект		Активизировать вставленный объект OLE
View (Вид)	**Toolbars**	Панели инструментов		Показать или скрыть панели инструментов
	Ruler	Линейка		Показать или скрыть линейку
	Status Bar	Строка состояния		Показать или скрыть строку состояния
	Headers/ Footers	Верхние/Нижние колонтитулы		Изменить колонтитулы для распечатываемых страниц
	Regions	Регионы		Показать или скрыть границы регионов
	Refresh	Обновить	<Ctrl>+<R>	Обновить документ
	Zoom	Масштаб		Изменить масштаб отображения документа
Insert (Вставка)	**Graph**	График		Вставить график (с выбором типа графика из подменю)
	Matrix	Матрица	<Ctrl>+<M>	Вставить матрицу или вектор
	Function	Функция	<Ctrl>+<E>	Вставить встроенную функцию
	Unit	Единицы	<Ctrl>+<U>	Вставить единицы измерения размерной величины
	Picture	Рисунок	<Ctrl>+<T>	Создать рисунок для отображения матрицы
	Area	Область		Создать зону
	Page Break	Разрыв страницы		Начать новую страницу

Таблица П2.1. (продолжение)

Меню	Команда	Перевод	Сочетание клавиш	Описание
	Math Region	Математическая область	<Ctrl>+ +<Shift>+<A>	Создать математическую область в тексте
	Text Region	Текстовая область	<">	Создать текстовую область е документе
	Component	Компонент		Вставить компонент другого приложения
	Data	Данные		Вставить данные в различных форматах
	Control	Элемент управления		Вставить элемент управления
	Object	Объект		Внедрение объекта
	Reference	Ссылка		Вставить ссылку на другой документ
	Hyperlink	Гиперссылка	<Ctrl>+<K>	Вставить гиперссылку
Format (Формат)	**Equation**	Формула		Форматирование формул
	Result	Результат		Форматирование вывода результатов вычислений
	Text	Текст		Форматирование текста
	Paragraph	Абзац		Изменение разметки абзаца
	Tabs	Табуляции		Установить табуляцию для документа или выделенного участка текста
	Style	Стиль текста		Определить или применить стиль — комбинацию настроек текстового формата

Таблица П2.1. (продолжение)

Меню	Команда	Перевод	Сочетание клавиш	Описание
	Properties	Свойства		Изменение свойств области
	Graph	График		Изменения в графиках
	Color	Цвет		Настройка цвета
	Area	Область		Работа с зоной
	Separate Regions	Разделить регионы		Разделить перекрывающиеся регионы в документе
	Align Regions	Выровнять регионы		Выравнивание региона по горизонтали или вертикали
	Repaginate Now	Разбить на страницы		Разбиение документа на страницы
Tools (Сервис)	**Spelling**	Проверка орфографии		Проверка орфографии текстовых регионов
	Animation	Анимация		Создать или воспроизвести анимацию
	Protect Worksheet	Запереть документ		Защита документа от редактирования
	Calculate	Пересчитать	<F9>	Управление вычислением формул
	Optimize	Оптимизировать		Управление режимом оптимизации расчетов
	Disable Evabion	Отключить вычисления		Вкл. / еыкл. вычисления формулы
	Trace Error	Трассировка ошибки		Трассировка источника сообщения об ошибке
	Worksheet Options	Опции документа		Установка опций математики
	Preferences	Настройки		Изменить основные настройки

Таблица П2.1. (продолжение)

Меню	Команда	Перевод	Сочетание клавиш	Описание
Symbolics (Символика)	**Evaluate**	Вычислить		Вычислить выражение в виде числа, если это возможно
	Simplify	Упростить		Упростить выражение
	Expand	Разложить		Представить выражение в более развернутом виде
	Factor	Разложить на множители		Разложить полином или целое число на простые множители
	Collect	Привести подобные		Привести подобные слагаемые
	Polynomial Coefficients	Коэффициенты полинома		Вычислить полиномиальные коэффициенты
	Variable	Переменная		Символьные действия с выделенной переменной
	Matrix	Матрица		Символьные действия с матрицей
	Transform	Преобразование		Символьные интегральные преобразования
	Evaluation Style	Стиль вычислений		Изменить показ символьных ответов
Window (Окно)	**Cascade**	Каскад		Расположить окна документов каскадом
	Tile Horizontal	По горизонтали		Расположить окна документов по горизонтали
	Tile Vertical	По вертикали		Расположить окна документов по вертикали
	*	(Имя документа)		Активизировать окно *

Таблица П2.1. *(окончание)*

Меню	Команда	Перевод	Сочетание клавиш	Описание
Help (Справка)	**Mathcad Help**	Справка	<F1>	Получение справочной информации
	What's This?	Что это такое?		Быстрая интерактивная справка об элементах интерфейса
	Developer's Reference	Справка для разработчиков		Дополнительная справка для разработчиков
	Author's Reference	Справка для авторов		Дополнительная справка для авторов электронных книг
	Tutorials	Учебники		Доступ к электронным книгам учебников
	QuickSheets	Быстрые шпаргалки		Доступ к электронным книгам Быстрых шпаргалок
	Reference Tables	Справочный стол		Доступ к электронной книге со справочными таблицами
	E-Books	Электронные книги		Открыть существующую в виде файла электронную книгу или пакет расширения
	User's Forum	Форум		Перейти на форум пользователей Mathcad
	Mathcad.com	Mathcad.com		Перейти на сайт Mathcad
	Mathcad Update	Обновление Mathcad		Проверить сайт компании MathSoft на наличие обновлений версии Mathcad 11
	About Mathcad	О программе		Информация о текущей версии Mathcad

***Таблица** П2.2. Панель **Math** (Математика)*

Кнопка	Перевод
Calculator Toolbar	Калькулятор
Graph Toolbar	График
Matrix Toolbar	Матрица
Evaluation	Выражения
Calculus	Вычисления
Boolean	Булевы операторы
Programming	Программирование
Greek Symbols Toolbar	Греческие символы
Symbolic Keyword Toolbar	Символика

***Таблица П2.3.** Панель **Calculator** (Калькулятор)*

Кнопка	Перевод	Сочетание клавиш
Sin	Синус	
Cos	Косинус	
Tan	Тангенс	
Ln	Натуральный логарифм	
Log	Десятичный логарифм	
n!	Факториал	<!>
I	Ввод мнимой единицы	<1>, <i>
|x|	Модуль	<Shift>+<\>
$\sqrt{\ }$	Квадратный корень	<\>
$\sqrt[n]{\ }$	Корень n-й степени	<Ctrl>+<\>
ex	Экспонента в n-й степени	
1/x	Обратная величина	

Таблица ***П2.3.*** *(окончание)*

Кнопка	Перевод	Сочетание клавиш
0	Скобки	<'>
x2	Возведение в квадрат	
xy	Возведение в степень у	<^>
π	Ввод числа π	<Ctrl>+<Shift>+<P>
/	Деление	</>
·/	Умножение и деление	<Ctrl>+<Shift>+<+>
x	Умножение	<*>
+	Деление в одну строку	<Ctrl>+</>
+	Сложение	<+>
:=	Присваивание	<:>
.	Десятичная точка	<.>
0, 1, 2, ..., 9	Числа 0...9	<0>,<1>, <2>, ..., <9>
-	Вычитание ("минус")	<->
=	Вычислить численно ("равно")	<=>

Таблица ***П2.4.*** *Панель Graph (График)*

Кнопка	Перевод	Сочетание клавиш
XY Plot	XY (Декартовый) график	<Shift>+<2>
Zoom	Масштаб графика	
Trace	Трассировка графика	
Polar Plot	Полярный график	<Ctrl>+<7>
Surface Plot	График трехмерной поверхности	<Ctrl>+<2>
Contour Plot	График линий уровня	<Ctrl>+<5>
3D Bar Plot	Трехмерная гистограмма	
Vector Field Plot	Векторное поле	
3D Scatter Plot	Трехмерное множество точек	

Таблица П2.5. Панель **Matrix** (Матрица)

Кнопка	Перевод	Сочетание клавиш
Matrix or Vector	Матрица или вектор	<Ctrl>+<M>
Subscript	Нижний индекс	<[>
Inverse	Обратная матрица	
Determinant	Определитель	<|><Shift>+<\>
Vectorize	Векторизовать	<Ctrl>+<->
Matrix Column	Выделение столбца	<Ctrl>+<6>
Matrix Transpose	Транспонирование	<Ctrl>+<1>
Range Variable	Ранжированная переменная	<;>
Cross Product	Векторное произведение	<Ctrl>+<8>
Dot Product	Умножение	<*>
Vector Sum	Сумма вектора	<Ctrl>+<4>
Picture	Рисунок	<Ctrl>+<T>

Таблица П2.6. Панель Evaluation (Выражения)

Кнопка	Перевод	Сочетание клавиш
Evaluate Numerically =	Вычислить численно ("равно")	<=>
Definition :=	Присваивание	<:>
Global Definition ≡	Глобальное присваивание	<~>
Evaluate Symbolically →	Вычислить символьно	<Ctrl>+<.>
Symbolic Keyword Evaluation • →	Символьное вычисление с ключевым словом	<Ctrl>+<Shift>+<.>
Prefix Operator fx	Оператор "перед"	
Postfix Operator xf	Оператор "после"	
Infix Operator xfy	Оператор "внутри"	
Tree Operator xfy	Оператор "дерево"	

Таблица П2.7. Панель **Calculus** (Вычисления)

Кнопка	Перевод	Сочетание клавиш
Derivative	Производная	<?>
Nth Derivative	n-я производная	<Ctrl>+<?>
Infinity ∞	Символ бесконечности	<Ctrl>+<Shift>+<Z>
Definite Integral	Определенный интеграл	<&>
Summation	Сумма	<Ctrl>+<Shift>+<4>
Iterated product	Произведение	<Ctrl>+<Shift>+<3>
Indefinite Integral	Неопределенный интеграл	<Ctrl>+<I>
Summation with range variables	Сумма ранжированной переменной	<Ctrl>+<4>
Iterated product with range variables	Произведение ранжированной переменной	<Ctrl>+<3>
Two-sided limit	Предел	<Ctrl>+<L>
Left-sided limit	Левый предел	<Ctrl>+<A>
Right-sided limit	Правый предел	<Ctrl>+<B>

Таблица П2.8. Панель **Boolean** (Булевы операторы)

Кнопка	Перевод	Сочетание клавиш
= equal	Равно	<Ctrl>+<=>
< less than	Меньше	<<>
> greater than	Больше	<>>
≤ less than or equal	Меньше или равно	<Ctrl>+<9>
≥ greater than or equal	Больше или равно	<Ctrl>+<0>
≠ not equal to	Не равно	<Ctrl>+<3>
-,Not	Не	<Ctrl>+<Shift>+<1>
л And	И	<Ctrl>+<Shift>+<7>
v Or	Или	<Ctrl>+<Shift>+<6>
©Exclusive or	Исключающее или	<Ctrl>+<Shift>+<5>

Таблица П2.9. *Панель* ***Controls*** *(Элементы управления)*

Кнопка	Перевод
Check Box	флажок проверки
Radio Button	переключатель
Push Button	кнопка
Slider	ползунковый регулятор
Text Box	поле текстового ввода
List Box	список

ПРИЛОЖЕНИЕ 3

Встроенные операторы и функции

Таблица П3.1. *Арифметические операторы*

Оператор	Клавиши	Скаляр	Вектор	Матрица
:-	<:>	Присваивание		
≡	<~>	Глобальное присваивание		
=	<=>	Численный вывод		
→	<Ctrl!>+<=>	Символьный вывод		
+	<+>	Сложение		
-	<->	Вычитание или отрицание (унарная операция)		
·	<*>	Умножение	Матричное умножение, умножение на скаляр	
			Скалярное произведение	
x	<Ctrl>+<8>		Деление	
/ либо ÷	</> либо <Ctrl>+</>		Факториал	
!	<!>			
–	<">	Комплексное сопряжение		

Таблица П3.1. *(окончание)*

Оператор	**Клавиши**	**Скаляр**	**Вектор**	**Матрица**
$\sqrt{\ }$	<\>	Квадратный корень		
$\sqrt[\blacksquare]{\ }$	<Ctrl>+<\>	Корень n-й степени		
$(\bullet)$	O	Скобки (изменение приоритета)		
$\blacksquare_{\blacksquare}$	<[>		Нижний индекс	
$\bullet^{T}$	<Ctrl>+<1>		Транспонирование	
$\lvert\blacksquare\rvert$	<Shift>+<\>	Модуль	Модуль вектора	Определитель
$\sum\blacksquare$	<Ctrl>+<4>		Сумма элементов	
$\bullet^{-1}$		Обратная величина		Обратная матрица
$\bullet^{n}$	<^>+n	Возведение в степень п		Возведение матрицы в степень п
$\to$	<Ctrl>+<->		Векторизация	
$\blacksquare^{<\blacksquare>}$	<Ctrl>+<6>		Выделение столбца	

Примечание

Скалярные операции над векторами и матрицами, если это не оговорено особо, производятся независимо над их каждым элементом, как над скаляром.

Таблица П3.2. *Вычислительные операторы*

Оператор	**Клавиши**	**Описание**	**Ссылка**
$\int_{\blacksquare}^{\blacksquare} \bullet\ da$	<Shift>+<7>	Определенный интеграл	7.1
$\int \blacksquare\ d\blacksquare$	<Ctrl>+<I>	Неопределенный интеграл	7.1.3

Таблица П3.2. *(окончание)*

Оператор	Клавиши	Описание	Ссылка
$\frac{d}{d\blacksquare}\blacksquare$	<?>	Дифференцирование	7.2
$\frac{d^n}{d\blacksquare^n}\blacksquare$	<Ctrl>+<?>	Вычисление n-й производной	7.2
$\sum_{\blacksquare=\blacksquare}^{\blacksquare}\blacksquare$	<Ctrl>+<Shift>+<4>	Сумма	3.2.2
$\sum_{\blacksquare}\blacksquare$	<Ctrl>+<4>	Сумма ранжированной переменной	3.2.2
$\prod_{\blacksquare=\blacksquare}^{\blacksquare}\blacksquare$	<Ctrl>+<Shift>+<3>	Произведение	3.2.2
$\prod_{\blacksquare}\blacksquare$	<Ctrl>+<3>	Произведение ранжированной переменной	3.2.2
$\lim_{\blacksquare\to\blacksquare}\blacksquare$	<Ctrl>+<L>	Предел	3.2.2
$\lim_{\blacksquare\to\blacksquare^-}\blacksquare$	<Ctrl>+<A>	Левый предел	3.2.2
$\lim_{\blacksquare\to\blacksquare^+}\blacksquare$	<Ctrl>+<B>	Правый предел	3.2.2

Таблица П3.3. *Встроенные функции по алфавиту*

Оператор	Клавиши	Описание	Ссылка
a*(z)	*z* — аргумент	Обратная тригонометрическая или гиперболическая функция *	10.4—5
Ai(x)	x — аргумент	Функция Эйри первого рода	15.1.3
angle (x,y)	x,y — координаты точки	Угол между точкой и осью OX	10.4

ТаблицаП3.3. *(продолжение)*

Оператор	Клавиши	Описание	Ссылка
APPENDPRN(file)	file — строковое представление пути к файлу	Дозапись данных в существующий текстовый файл	16.6.1
arg(z>	z — аргумент функции	Аргумент комплексного числа	10.2
atan2(x,y)	x,y — координаты точки	Угол, отсчитываемый от оси ОХ до точки (x,y)	10.4
Augment (A,B,C,...)	А,в,с,... — векторы или матрицы	Слияние матриц слева направо	9.2.2
bei(n,x) ber(n,x)	п — порядок x — аргумент	Мнимая и действительная части функции Бесселя —Кельвина	15.1.4
Bi(x)	x — аргумент	Функция Эйри второго рода	15.1.3
bspline(x,y,u,n)	x,y — векторы данных и — вектор значений сшивок В-сплайнов n — порядок полиномов	Вектор коэффициентов В-сплайна	15.1.3
Bulstoer (y0,t0,t1,M,D)	См. rkfixed	Возвращает матрицу с решением задачи Коши для системы ОДУ методом Булирша-Штера	11.3
bulstoer (y0,t0,t1, acc,D,k,s)	См. rkadapt	Возвращает матрицу с решением задачи Коши для системы ОДУ методом Булирша-Штера (для определения только последней точки интервала)	11.3
Bvalfit (z1,z2,x0,x1, xf,D,load1, load2,score)	zl,z2 — вектор начальных значений для недостающих левых и правых граничных условий xO — левая граница x1— правая граница xf — внутренняя точка D(x,y) — векторная функция, задающая систему ОДУ	Возвращает вектор недостающих граничных условий у краевой задачи для системы N ОДУ с дополнительным условием в промежуточной точке	12.1.4

Таблица П3.3. (продолжение)

Оператор	Клавиши	Описание	Ссылка
	load1(x0,z), load2(x1,z) — векторные функции, задающие левые и правые граничные условия score(xf,y) — векторная функция, задающая сшивку решений в xf		
ceil(x)	x — аргумент	Наименьшее целое, не меньшее x	10.8
cfft(y) CFFT(y)	y — вектор данных	Вектор прямого комплексного преобразования Фурье (в разных нормировках)	15.4.1
cholesky(A)	A — квадратная, определенная матрица	Разложение Холецкого	9.5.1
cols(A)	A — матрица или вектор	Число столбцов	9.2.3
concat(S1,S2, ...)	S1,S2,... — строки	Объединение строковых переменных	10.7
cond1(A) cond2(A) conde(A) condi(A)	A — квадратная матрица	Числа обусловленности в разных нормах (ы, L2, Евклидова, ∞)	9.2.6
cos(z)	z — аргумент	Косинус	10.4
cosh(z)	z — аргумент	Гиперболический косинус	10.5
cot(z)	z — аргумент	Котангенс	10.4
coth(z)	z — аргумент	Гиперболический котангенс	10.5
csort(A,i)	A — матрица i — индекс столбца	Сортировка строк матрицы по элементам i-го столбца	9.2.4
CreateMesh (F,s0,s1, t0,t1,sgr, tgr,fmap)	F(s,t) — векторная функция из трех элементов t0,t1 — пределы t s0,s1 — пределы s tgr,sgr — число точек сетки по t и s fmap — функция преобразования координат	Создание вложенного массива, представляющего x-, y- и z-координаты параметрической поверхности, заданной функцией F	9.2.1

Таблица П3.3. (продолжение)

Оператор	Клавиши	Описание	Ссылка
CreateSpace(F[,t0,t1, tgr,fmap])	F(t) — векторная функция из трех элементов t0,t1 — пределы t tgr — число точек сетки по t fmap — функция преобразования координат	Создание вложенного массива, представляющего x-, y- и z-координаты параметрической пространственной кривой, заданной функцией F	9.2.1
csc(z)	z — аргумент	Косеканс	10.4
csch(z)	z — аргумент	Гиперболический косеканс	10.5
csgn(z)	z — аргумент	Комплексный знак числа	10.2
cspline(x,y)	x,y — векторы данных	Вектор коэффициентов кубического сплайна	15.1.2
cyl2xyz(r,θ,z)	r,θ,z — цилиндрические координаты	Преобразование цилиндрических координат в прямоугольные	10.10
D* (x,par)	x — значение случайной величины par — список параметров распределения *	Плотность вероятности со статистикой распределения *	14.1.4
diag(v)	v — вектор	Диагональная матрица, на диагонали которой находятся элементы вектора	9.2.1
eigenvals(A)	A — квадратная матрица	Собственные значения матрицы	9.4
eigenvec(A,λ)	A — квадратная матрица λ — собственное значение	Собственный вектор матрицы, соответствующий заданному собственному значению	9.4
eigenvecs(A)	A — квадратная матрица	Собственные векторы матрицы	9.4
erf(x)	x — аргумент	Функция ошибок	14.1.1
erfc(x)	x — аргумент	Обратная функция ошибок	14.1.1
error(S)	S — строка	Возвращает строку s как сообщение об ошибке	10.7

Таблица П3.3. (продолжение)

Оператор	Клавиши	Описание	Ссылка
`exp(z)`	z — аргумент	Экспонента в степени z	10.3
`expfit(x,y,g)`	x,y — векторы данных g — вектор начальных значений **a,b,c**	Регрессия экспонентой a-ebx+c	15.2.3
`fft(y)` `FFT(y)`	y — вектор данных	Вектор прямого преобразования Фурье (в разных нормировках)	15.4.1
`fhyper(a,b,c,x)`	**a,b,c** — параметры x — **аргумент**, -1<x<1	Гауссова гипергеометрическая функция	10.6
`Find (x1,x2,...)`	xl,x2,... — переменные	Возвращает корень алгебраического уравнения (скаляр) или системы (вектор), определенных в блоке с Given	8.3—8.4
`floor(x)`	x — аргумент	Наибольшее целое число, меньшее или равное x	10.8
`Gamma(x)` `Gamma(a,x)`	x — аргумент	Гамма-функция Эйлера или неполная гамма-функция порядка a	10.6
`genfit(x,y,g,G)`	x,y — векторы данных g — вектор начальных значений параметров регрессии G (x, C) — векторная функция, составленная из функции пользователя и ее частных производных по каждому параметру	Вектор коэффициентов регрессии функциями пользователя общего вида	15.2.4
`geninv(A)`	A — матрица	Создание обратной матрицы	9.2.1
`genvals(A,B)`	A,B — квадратные матрицы	Расчет обобщенных собственных значений	9.4
`genvecs(A,B)`	A,B — квадратные матрицы	Расчет обобщенных собственных векторов	9.4
`Given`		Ключевое слово для ввода систем уравнений, неравенств и т. п.	8.3

Таблица П3.3. (продолжение)

Оператор	Клавиши	Описание	Ссылка
heaviside step(x)	x — аргумент	Функция Хевисайда	10.9
Her(n,x)	x — аргумент n — порядок	Полином Эрмита	10.6
I0(x) I1(x) In(m,x)	x — аргумент	Модифицированная функция Бесселя первого рода нулевого, первого и m-го порядка	10.1.2
ibeta(a,x,y)	x,y — аргументы a — параметр	Неполная бета-функция	10.6
identity(N)	N— размер матрицы	Создание единичной матрицы	9.2.1
icfft(v) ICFFT(v)	v — вектор частотных данных Фурье-спектра	Вектор комплексного обратного преобразования Фурье (в разных нормировках)	15.4.1
if (cond,x,y)	cond — логическое условие x,y— значения, возвращаемые, если условие верно (ложно)	Функция условия	10.9
ifft(v) IFFT(v)	v — вектор частотных данных Фурье-спектра	Вектор обратного преобразования Фурье (в разных нормировках)	15.4.1
IsString(x)	x — аргумент	Возвращает 1, если x — строка, и 0 в остальных случаях	10.7
iwave(v)	V— вектор частотных данных вейвлет-спектра	Вектор обратного вейвлет-преобразования	15.4.2
Im(z)	z — аргумент	Мнимая часть комплексного числа	10.2
interp(s,x,y,t)	s — вектор вторых производных x,y — векторы данных t — аргумент	Сплайн-интерполяция	15.1.2
intercept(x,y)	x,y — векторы данных	Коэффициент b линейной регрессии b+a·x	15.2.1

Таблица П3.3. *(продолжение)*

Оператор	Клавиши	Описание	Ссылка

`J0(x)`
`J1(x)`
`Jn(m,x)`
	x — аргумент	Функция Бесселя первого рода нулевого, первого и m-го порядка	10.1.1

`Jac(n,a,b,x)`	x — аргумент a,b — параметры n — порядок	Полином Якоби	10.6
`js(n,x)`	n — порядок x — аргумент	Сферическая функция Бесселя первого рода	10.1.5

`K0(x)`
`K1(x)`
`Kn(m,x)`
	x — аргумент	Модифицированная функция Бесселя второго рода нулевого, первого и т-го порядка	10.1.2

`Kronecker delta(x,y)`	x,y — аргументы	Дельта-символ Кронекера	10.9
`ksmooth(x,y,b)`	x,y — векторы данных b — ширина окна сглаживания	Сглаживание с помощью функции Гаусса	15.3.1
`Lag(n,x)`	x — аргумент п — порядок	Полином Лагерра	10.6
`last(v)`	v — вектор	Индекс последнего элемента вектора	9.2.3
`Leg(n,x)`	x — аргумент п — порядок	Полином Лежандра	10.6
`length(v)`	v — вектор	Число элементов вектора	9.2.3
`line(x,y)`	x,y — векторы данных	Вектор из коэффициентов линейной регрессии b+ax	15.2.1
`linfit(x,y,F)`	x,y — векторы данных F(x) — векторная функция пользователя	Вектор коэффициентов регрессии функцией пользователя	15.2.4
`linterp(x,y,t)`	x,y — векторы данных t — аргумент	Кусочно-линейная интерполяция	15.1.1
`lgsfit(x,y,g)`	x,y — векторы данных g — вектор начальных значений a,b,c	Регрессия логистической функцией a/(1+b·e-cx)	15.2.3

Таблица П3.3. (продолжение)

Оператор	Клавиши	Описание	Ссылка
ln(z)	z — аргумент	Натуральный логарифм	10.3
lnfit(x,y)	x,y — векторы данных	Регрессия логарифмической функцией a·ln (x) +b	15.2.3
loess(x,y,span)	x,y — векторы данных span — параметр размера полиномов	Вектор коэффициентов для регрессии отрезками полиномов (применяется вместе с interp)	15.2.2
log(z)	z — аргумент	Десятичный логарифм	10.3
log(z, b)	z ~ аргумент	Логарифм z по основанию b	10.3
logfit(x,y,g)	x,y — векторы данных g — вектор начальных значений a,b,c	Регрессия логарифмической функцией a·ln(x+b) +c	15.2.3
lsolve(A,b)	A — матрица СЛАУ Ь — вектор правых частей	Решение системы линейных уравнений (СЛАУ)	9.3
lspline(x,y)	x,y — векторы данных	Вектор коэффициентов линейного сплайна	15.1.2
lu(A)	A — квадратная матрица	LU-разложение	9.5.3
matrix(M,N,f)	M — количество строк N — количество столбцов f(i,j) — функция	Создание матрицы с элементами f(i,j)	9.2.1
Maximize (f,x1,...)	f(x1,...) — функция x1,... — аргументы, по которым производится максимизация	Вектор значений аргументов, при которых функция f достигает максимума (возможно задание дополнительных условий в блоке с Given)	8.6
mhyper(a,b,x)	x — аргумент a,b — параметры	Конфлюэнтная гипергеометрическая функция	10.6

Таблица П3.3. (продолжение)

Оператор	Клавиши	Описание	Ссылка
Minerr (x1,x2,...)	x1,x2,... — переменные	Возвращает вектор приближенного решения системы уравнений и неравенств, определенных в блоке с Given	8.5
Minimize (f,x1,...)	f(x1,...) — функция x1,... — аргументы, по которым производится минимизация	Вектор значений аргументов, при которых функция f достигает минимума (возможно задание дополнительных условий в блоке с Given)	8.6
medsmooth(y,b)	y — вектор данных b — ширина окна сглаживания	Сглаживание методом "бегущих медиан"	15.3.1
Multigrid (F,ncycle)	F — матрица правой части уравнения Пуассона ncycle — параметр алгоритма (2)	Матрица решения уравнения Пуассона на квадратной области с нулевыми граничными условиями	12.4.1
n* (M,par)	M — размерность вектора x — значение случайной величины par — список параметров распределения *	Вектор случайных чисел со статистикой *	14.1.4
norm1(A) norm2(A) norme(A) normi(A)	A — квадратная матрица	Нормы матриц (L1, L2, Евклидова, ∞)	9.2.5
num2str(z)	z — число	Возвращает строку, чьи знаки соответствуют десятичному значению числа z	10.7
Odesolve (t,t1[,step])	t — переменная интегрирования ОДУ t1 — конечная точка интервала интегрирования step — число шагов интегрирования ОДУ	Возвращает матрицу с решением задачи Коши для одного ОДУ, определенного в блоке с Given и начальными условиями в точке t0	11.1.1, 11.2

ТаблицаП3.3.(продолжение)

Оператор	Клавиши	Описание	Ссылка
p* (x,par)	x — значение случайной величины par — список параметров распределения *	Функция распределения со статистикой *	14.1.4
pdesolve(u, x, xrange, t, trange, [xpts], [tpts]))	и — вектор имен функций x —пространственная переменная xrange — интервал интегрирования по пространству t — временная переменная trange — интервал интегрирования по времени xpts — число пространственных узлов сетки tpts — число временных шагов сетки	Возвращает скалярную функцию двух аргументов (x,t), являющуюся решением дифференциального уравнения (или системы уравнений) в частных производных	13.3.1
pol2xy(r,θ)	r,θ — полярные координаты	Преобразование полярных координат в прямоугольные	10.10
polyroots (v)	v — вектор, составленный из коэффициентов полинома	Возвращает вектор всех корней полинома	8.2
predict(y,m,n)	y — исходный вектор m — число элементов y, по которым строится экстраполяция п — количество предсказываемых элементов	Функция предсказания, экстраполирующая вектор	15.1.4
pspline(x,y)	x,y — векторы данных	Вектор коэффициентов квадратичного сплайна	15.1.2
pw fit fx,y,g)	x,y — векторы данных g — вектор начальных значений a,b,c	Регрессия степенной функцией a·xb+c	15.2.3
q* (p,par)	p — значение вероятности par— список параметров распределения *	Квантиль (функция, обратная функции распределения) со статистикой *	14.1.4

Таблица П3.3. *(продолжение)*

Оператор	Клавиши	Описание	Ссылка
qr(A)	A — вектор или матрица	QR-разложение	9.5.2
Radau (y0,t0,t1,M,D)	См. rkfixed	Возвращает матрицу с решением задачи Коши для жесткой системы ОДУ методом RADAUS	11.5.2
Radau (y0,t0,t1,M,D)	См. rkfixed	Возвращает матрицу с решением задачи Коши для жесткой системы ОДУ методом RADAUS (для определения только последней точки интервала)	11.5.2
rank(A)	A — матрица	Ранг матрицы	9.2.7
Re(z)	z — аргумент	Действительная часть комплексного числа	10.2
READ*(file)	file — строковое представление пути к файлу	Запись данных в файл типа *	15.6
regress(x,y,k)	x,y — векторы данных к — степень полинома	Вектор коэффициентов для полиномиальной регрессии (применяется вместе с interp)	15.2.2
Relax (a,b,c,d,e,P, v, rjac)	a,b,c,d,e — матрицы коэффициентов разностной схемы F — матрица правой части уравнения v — матрица граничных условий rjac — параметр алгоритма (0...1)	Матрица решения методом сеток дифференциального уравнения в частных производных на квадратной области	12.4.1, 12.4.3
reverse(v)	v — вектор	Перестановка элементов вектора в обратном порядке	9.2.4
Rkadapt (y0,t0,t1, acc,D,k,s)	yO — вектор начальных условий (t0,t1) — интервал интегрирования acc — погрешность вычисления D(t,y) — векторная функция, задающая систему ОДУ	Возвращает матрицу с решением задачи Коши для системы ОДУ методом Рунге-Кутты с переменным шагом и заданной точностью (для определения только последней точки интервала)	11.3

Таблица П3.3. *(продолжение)*

Оператор	Клавиши	Описание	Ссылка
	k— максимальное число шагов интегрирования s — минимальный шаг интегрирования		
Rkadapt (y0,t0,t1,M,D)	См. rkfixed	Возвращает матрицу с решением задачи Коши для системы ОДУ методом Рунге-Кутты с переменным шагом	11.3
rkfixed (y0,t0,t1,M,D)	y0 — вектор начальных условий (t0,t1)— интервал интегрирования M— число шагов интегрирования D(t,y) — векторная функция, задающая систему ОДУ	Возвращает матрицу с решением задачи Коши для системы ОДУ методом Рунге-Кутты с фиксированным шагом	11.1.2, 11.3
root (f(x, ...),x[a,b])	f(x, . . .} — функция x— переменная (a,b) — интервал поиска корня	Возвращает корень функции	8.1
round(x, n)	x — аргумент n — число знаков округления после десятичной **точки**	Округление	10.8
rows(A)	A — матрица или вектор	Число строк	9.2.3
rref(A)	A — матрица или вектор	Преобразование матрицы в ступенчатый **вид**	9.2.1
rsort(A,i)	A — матрица i — индекс строки	Сортировка матрицы по элементам i -й строки	9.2.4
sbval (z,x0,x1,D, load,score)	z — вектор начальных приближений для недостающих начальных условий xO — левая граница x1— правая граница	Возвращает вектор недостающих начальных условий для двухточечной краевой задачи для системы ОДУ	12.1.3

Таблица П3.3. *(продолжение)*

Оператор	Клавиши	Описание	Ссылка
	D(X,y) — векторная функция, задающая систему ОДУ load(x0,z) — векторная функция с начальными условиями score(xl,y) — векторная функция, задающая правые граничные условия		
`search(S,Subs,m)`	S — строка Sub — подстрока m — стартовая позиция поиска	Стартовая позиция подстроки в строке	10.7
`sec(z)`	z — аргумент	Секанс	10.4
`sech(z)`	z — аргумент	Гиперболический секанс	10.5
`sign(x)`	x — аргумент	Знак числа	10.9
`signum(z)`	z — аргумент	Комплексный знак числа $z/\|z\|$	10.2
`sin(z)`	z — аргумент	Синус	10.4
`sinh(z)`	z — аргумент	Гиперболический синус	10.5
`sinfit(x,y,g)`	x,y — векторы данных g — вектор начальных значений a,b,c	Регрессия синусоидой $f(x)=a\cdot\sin(x+b)+c$	15.2.3
`sinc(z)`	z — аргумент	Sinc-функция	10.11
`slope(x,y)`	x,y — векторы данных	Коэффициент a линейной регрессии b+ax	15.2.1
`sort(v)`	v — вектор	Сортировка элементов вектора	9.2.4
`sph2xyz(r,θ,φ)`	r,θ,φ — сферические координаты	Преобразование сферических координат в прямоугольные	10.10
`stack(A,B,C,...)`	A,B,C,... — векторы или матрицы	Слияние матриц сверху вниз	9.2.2

Таблица П3.3. *(продолжение)*

Оператор	Клавиши	Описание	Ссылка
Stiffb (y0,t0,t1, M,D,J)	См. rkfixed J(t,y) — матричная функция Якоби для D(t,y)	Возвращает матрицу с решением задачи Коши для жесткой системы ОДУ методом Булирша-Штера	11.5.2
stiffb (y0,t0,t1, acc,D,J,k,s)	См. rkadapt J(t,y) — матричная функция Якоби для D(t,y)	Возвращает матрицу с решением задачи Коши для жесткой системы ОДУ методом Булирша-Штера (для определения только последней точки интервала)	11.5.2
Stiffr (y0,t0,t1, M,D,J)	См. Stiffb	Возвращает матрицу с решением задачи Коши для жесткой системы ОДУ методом Розенброка	11.5.2
stiffr (y0,t0,t1,acc, D,J,k,s)	См. stiffb	Возвращает матрицу с решением задачи Коши для жесткой системы ОДУ методом Розенброка (для определения только последней точки интервала)	11.5.2
str2num(S)	S — строка	Преобразование строкового представления в действительное число	10.7
str2vec(S)	S — строка	Преобразование строкового представления в вектор ASCII-кодов	10.7
strlen(S)	S — строка	Количество знаков в строке	10.7
submatrix(A,ir,jr, ic,jc)	A — матрица ir, jr — строки ic, jc — столбцы	Возвращает часть матрицы, находящуюся между ir,jr-строками и ic,jc-толбцами	9.2.2
substr(S,m,n)	S — строка	Подстрока, полученная из строки S выделением п знаков, начиная с позиции m в строке S	10.7
supsmooth(x,y)	x,y — векторы данных	Сглаживание с помощью адаптивного алгоритма	15.3.1

Таблица П3.3. (окончание)

Оператор	Клавиши	Описание	Ссылка
`svd(A)`	A — действительная матрица	Сингулярное разложение	9.5.4
`svds(A)`	A— действительная матрица	Вектор, состоящий из сингулярных чисел	9.5.4
`tan(z)`	z — аргумент	Тангенс	10.4
`tanh(z)`	z — аргумент	Гиперболический тангенс	10.5
`Tcheb(n,x)`	x — аргумент n — порядок	Полином Чебышева первого рода	10.6
`tr(A)`	A— квадратная матрица	След матрицы	9.1.8
`trunc(x)`	x — аргумент	Целая часть числа	10.8
`Ucheb(n,x)`	x — аргумент п — порядок	Полином Чебышева второго рода	10.6
`vec2str(v)`	v — вектор ASCII-кодов	Строковое представление элементов вектора v	10.7
`wave(y)`	y— вектор данных	Вектор прямого вейвлет-преобразования	15.4.2
`WRITE*(file)`	`file` — строковое представление пути к файлу	Запись данных в файл типа *	16.6
`xy2pol(x,y)`	x,y — прямоугольные координаты на плоскости	Преобразование прямоугольных координат в полярные	10.10
`xyz2cyl(x,y,z)`	x,y,z — прямоугольные координаты	Преобразование прямоугольных координат в цилиндрические	10.10
`xyz2sph(x,y,z)`	x,y,z — прямоугольные координаты	Преобразование прямоугольных координат в сферические	10.10

`Y0(x)`
`YI(x)`
`Yn(m, x)`
	x — аргумент, x>0	Функция Бесселя второго рода нулевого, первого и m-го порядка	10.1.1

`ys(n,x)`	п — порядок x — аргумент	Сферическая функция Бесселя второго рода	10.1.5

Примечание

Некоторые функции, составляющие семейства типовых функций, приведены в сокращенном виде с недостающей частью имени в виде звездочки *. Например, различные статистические функции, описывающие различные распределения, или функции вывода в файлы. Подробные сведения содержатся в разделе, на который указывает соответствующая ссылка.

ПРИЛОЖЕНИЕ 4

Сообщения об ошибках

Таблица П4.1. Сообщения об ошибках

Ошибка	Перевод	Вероятная причина	Возможные пути устранения
	Сообщения об ошибках в численных вычислениях		
A "Find" or "Minerr" must be preceded by a matching "Given"	Find или Minerr должны предваряться ключевым словом Given	Эта ошибка выделяет функцию Find или Minerr при их несогласованности с Given	Каждый вычислительный блок, который заканчивается функцией Find или Minerr, должен начинаться с ключевого слова Given
All evaluations resulted in either an error or a complex result	Вычисления приводят к ошибке или комплексному результату	Mathcad не может начертить некоторые точки, потому что не существует действительных значений для их нанесения на график	Это сообщение может появиться, если имеется ошибка или все значения комплексные
Arguments in function definitions must be names	Аргументы в определениях функции должны быть именами	Выделенное определение функции содержит неправильный перечень аргументов	В списке аргументов должны быть правильно поименованы переменные, или список имен необходимо отделить запятыми
At least one limit must be infinity	По крайней мере один предел должен быть бесконечным	Когда для интегрирования выбран алгоритм бесконечного предела, то по крайней мере один из пределов интеграла должен быть бесконечным	Тип бесконечности вводится нажатием сочетания клавиш <Ctrl>+<Shift>+<Z>. Для изменения алгоритма, использующего бесконечный предел, или для вычисления какого-либо другого интеграла щелкните на интеграле правой кнопкой мыши и измените алгоритм с помощью контекстного меню

Таблица П4.1. *(продолжение)*

Ошибка	Перевод	Вероятная причина	Возможные пути устранения
Can only evaluate an n-th order derivative when n=0,1..5.	Можно вычислить n-й порядок производной, только когда n=0,1..5	Порядок производной должен быть одним из следующих чисел: 0, 1, 2,..5.	Если Вы хотите посчитать производную более высокого порядка, то делайте это с помощью символьного дифференцирования
Can't evaluate this function when its argument less than or equal to zero	Невозможно вычислить эту функцию, когда ее аргумент меньше или равен нулю	Такое сообщение может касаться XY- или полярных графиков, имеющих логарифмические оси, на которых или пределы, или некоторые из значений, не положительны	Отрицательные числа и ноль не могут быть расположены нигде на логарифмических осях. Смените тип осей графика или постройте его для других значений
Can't converge to a solution	Не сходится к решению	Численный метод расходится (не может найти решения)	Убедитесь, что операция не применяется к функции в области непосредственной близости точки ее сингулярности (деления на ноль). Попробуйте поменять параметры численного метода (например начальное приближение). Попробуйте увеличить константу TOL, т. е. осуществить поиск решения с худшей погрешностью. Попробуйте поменять численный алгоритм, если это возможно (вызвав контекстное меню нажатием на месте ошибки правой кнопки мыши)
Can't define the same variable more than once in the same expression	Невозможно определить ту же самую переменную более одного раза в одном и том же выражении	Вы пытаетесь вычислить одну и ту же переменную дважды в одном выражении	Пример подобной ошибки: если Вы создаете вектор с левой стороной a := и используете это же имя справа, то получите ошибку
Can't determine what units the result of this operation should have	Невозможно определить, в каких единицах следует быть результату этой операции	Вы возвели выражение, содержащее единицы измерения, в степень, являющуюся переменной в неких пределах или вектором. В результате невозможно определить размерность результата	Если выражение включает в себя единицы измерений, то можно возводить его только в действительную фиксированную степень

Таблица П4.1. *(продолжение)*

Ошибка	Перевод	Вероятная причина	возможные пути устранения
Can't divide by zero	Деление на ноль невозможно	Где-то в программе или внутри численного метода возникло деление на ноль	Найдите место деления на ноль и устраните его Попробуйте поменять параметры численного метода, константы точности или сам численный алгоритм
Could not find a solution	Невозможно найти решение	Численный метод расходится {не может найти решения)	См. "Can't converge to a solution"
Can't find the data file you're trying to use	Невозможно найти файл, который Вы пытаетесь использовать	Невозможно найти файл данных или другой тип файла, к которому Вы обращаетесь	Удостоверьтесь, что такой файл существует в указанном месте
Can't have anything with units or dimensions here	Здесь нет ничего в единицах измерений или в размерностях	Это выражение использует единицы измерений где-то, где они не разрешены	Единицы измерений не разрешены: - в аргументах большинства функций; - в экспонентах; - в верхних и нижних индексах. Для того чтобы использовать выражения с единицами измерений, вначале переведите это выражение в UnitsOf (выражение)
Can't have more than one array in a contour plot	Нельзя иметь более одного массива в контурном графике	Вы вводите более одного массива в местозаполнитель контурного или поверхностного графика	Можно иметь только один массив в данном местозаполнителе, т. к. графики могут выдавать лишь одну поверхность в один момент времени
Can't perform this operation on the entire array at once. Try using "vectorize" to perform it element by element	Невозможно представить эту операцию в целом массиве сразу. Попытайтесь использовать векторизацию, чтобы представить элемент за элементом	Например, можно увидеть это сообщение при попытке разделить один вектор на другой	Для того чтобы применять функцию или оператор к каждому элементу вектора или матрицы, используйте оператор векторизации
Can't plot this many points	Невозможно начертить график с таким большим количеством точек	Попытка построения графика с числом точек, превосходящим возможное	Попробуйте сделать число точек меньше, чем 150 000

Таблица П4.1. (продолжение)

Ошибка	Перевод	Вероятная причина	Возможные пути устранения
Can't put a ":=" inside a solve block	Нельзя помещать ":=" внутрь вычислительного блока	Внутри вычислительного блока не должно быть формулировки присваивания. Он должен содержать только булевы выражения	Используйте панель с булевыми операторами
Can't raise an expression having units to a complex power	Нельзя возводить в комплексную степень выражение, имеющее единицы измерений	Это выражение содержит единицы измерений, а Вы возводите его в комплексную степень	Выражение с единицами измерений можно возводить только в действительную степень. Для того чтобы возводить в комплексную степень выражение с единицами измерений, вначале переведите это выражение в UnitsOf (выражение) — единицы измерений будут отменены
Can't solve a system having this many equations	Невозможно решить систему, имеющую так много уравнений	Mathcad не способен решить систему	См. определение термина "вычислительный блок " (гл. 8)
Can't understand something in this data file	Невозможно что-то понять в файле данных	Файл, к которому Вы пытаетесь получить доступ при помощи READ или READ*, имеет дефект	Файл должен быть ASCII-текстом Все строки файла должны иметь тот же номер значений, что используется в READ* Если файл имеет требуемый формат, а это сообщение продолжает появляться, попробуйте удалить любую часть текста из файла
Can't understand the name of this function	Невозможно понять имя этой функции	Такое сообщение может появиться, если в качестве имени функции используется, например, число 6(x)	Выражение должно соответствовать требованиям, предъявляемым в Mathcad к написанию имен функций
Can't understand the way this range variable is defined	Невозможно понять определение ранжированной переменной	Определение ранжированной переменной неверно	Вводя область определения ранжированной переменной, необходимо использовать один из следующих видов: Rvar := n1 .. n2 Rvar := n1,n2 ..n3

Таблица П4.1. (продолжение)

Ошибка	Перевод	Вероятная причина	Возможные пути устранения
Can't understand this number	Невозможно понять это число	Это выражение содержит символ или десятичную точку там, где это непозволительно	Вы увидите эту ошибку, например, если случайно запишете число так: .452.
Can't use a range variable in a solve block	Невозможно использовать ранжированную переменную в вычислительном блоке	Эта ошибка появится, если использовать область определения переменной в неподходящем месте	Придумайте алгоритм, не допускающий применения ранжированной переменной в вычислительном блоке
Cannot evaluate this accurately at one or more of the values you specified	Невозможно точно вычислить одно или более значений	Эта ошибка появляется, если попытаться вычислить функцию для аргумента, находящегося за пределами точной области определения функции	Проверьте область определения функции
Cross product is defined only for vectors having exactly three elements	Векторное произведение определяется только для векторов, имеющих точно три элемента	Векторное произведение определяется только для векторов, имеющих точно три элемента	См. определение векторного произведения (гл. 9)
Can't evaluate this expression. It may have resulted in an overflow or an infinite loop	Невозможно вычислить это выражение. Это может быть результатом переполнения или бесконечных циклов	Это функциональное определение может содержать слишком много вложенных функций. Или функция может быть константой в бесконечных циклах	Проверьте несколько итераций цикла
Degree of the polynomial must be between 1 and 99.	Степень полинома должна находиться в пределах между 1 и 99	Вектор, пропущенный через функцию поиска корней полинома, должен содержать по крайней мере 2 и не более 99 элементов	
Dimensions must be >4	Размерность должна быть >4	Эта матрица должна иметь по крайней мере 4 ряда и 4 столбца	

Таблица П4.1. *(продолжение)*

Ошибка	Перевод	Вероятная причина	Возможные пути устранения
End of file	Конец файла	Вы пытаетесь прочитать больше значений в файле данных, чем там имеется	Например, если файл данных имеет 10 значений, а записано выражение `i:=1…100` `xi:=READ*(file)`, то появится это сообщение
End points cannot be the same	Конечные точки не могут быть одинаковы	Это сообщение появляется при некорректном решении дифференциальных уравнений	Конечные точки интервала, на котором будет вычисляться решение, должны быть различны
Equation too large	Уравнение слишком большое	Это выражение слишком сложное для вычисления	Разбейте выражение на два или более простых
Floating point error	Ошибка вычислений с плавающей точкой	Функция вычисляется в точке, в которой это не разрешено	
Found a singularity while evaluating this expression. You may be dividing by zero	Найдена сингулярность при вычислении этого выражения. Возможно, Вы делите на ноль	Вычисляется функция или выполняется операция с недопустимым значением	Например, это сообщение возникнет при попытке деления на ноль или обращения сингулярной матрицы; разберитесь, где это происходит
Found a number with a magnitude greater than 10^307	Найдено число, превышающее значение 10^{307}		Попробуйте поменять параметры численного алгоритма или сам алгоритм
Illegal context. Press <F1> for help	Недопустимый контекст. Нажмите <F1>, чтобы получить помощь	Часто встречается при синтаксических ошибках	Проверьте синтаксис и порядок расположения формул в документе
Illegal dimensions	Недопустимые размерности	Матрица, на которую Вы ссылаетесь, не имеет достаточно строк или столбцов	Введите имя матрицы с клавиатуры и нажмите знак "=", чтобы проверить число ее строк и столбцов
Integer too large/ Integer too small	Целое число слишком большое/ слишком маленькое	Это число слишком велико/мало для работы с ним	Если Вы работаете со встроенными функциями, то щелкните на имени функции и вызовите подсказку с помощью <F1>

Таблица П4.1. *(продолжение)*

Ошибка	Перевод	Вероятная причина	Возможные пути устранения
Invalid format	Недопустимый формат	Аргументы этой функции могут быть некорректны	Если Вы работаете со встроенными функциями, то щелкните на имени функции и вызовите подсказку с помощью <F1>
Live symbolics not available	Символьные вычисления неприменимы		*См. гл. 5*
Must be <= 10000	Это значение должно быть <=10000		
Must be >= 10^-16	Это значение должно быть $>=10^{-16}$		
Must be string	Функция или оператор должен быть строковым аргументом		*См. гл. 10*
Must be between two lock regions	Должно быть между двумя запертыми областями		*См. гл. 16.*
Must be function	Этот аргумент должен быть функцией		
Must be increasing	Значения вектора должны быть возрастающими		Введите с клавиатуры имя вектора и знак =, чтобы проверить его значения
Must be less than the number of data points	Должен быть меньше, чем число точек данных	Этот аргумент должен быть меньше, чем число точек имеющихся данных	
Must be positive	Должен быть положительным	Невозможно вычислить эту функцию, когда ее значения меньше или равны нулю	Это сообщение может касаться построения XY- или полярных графиков с логарифмическими осями. Отрицательные числа или ноль не могут располагаться на логарифмических осях
Must be real	Должно быть действительным	Это значение должно быть действительным. Его мнимая часть должна быть нулем	Примером таких выражений могут служить нижний и верхний индексы, решения дифференциальных уравнений, углы

Таблица П4.1. (продолжение)

Ошибка	Перевод	Вероятная причина	Возможные пути устранения
Must be real scalar	Должно быть действительным скаляром	Это значение не должно быть комплексным или мнимым	
Must be real vector	Должно быть действительным вектором	Этот вектор не может иметь комплексные или мнимые элементы. Он должен также быть вектором-столбцом, а не строкой	
Must be square	Должна быть квадратной	Эта ошибка выделяет неквадратную матрицу в той операции или функции, в которой ей следует быть квадратной	Например, матрица должна быть квадратной при обращении, возведении ее в степень, или в функциях eigenvals и eigenvec
No solution found	Не найдено решение		Если Вы используете встроенные функции, то щелкните мышью на имени функции и нажмите клавишу <F1>, для того чтобы быть уверенным в корректности использования функции. Однако решение может просто не существовать. См. также "Can't converge to a solution"
Not enough memory for this operation	Для этой операции недостаточно памяти	Не хватает памяти, чтобы завершить это вычисление	Попытайтесь освободить немного памяти путем уменьшения массива или матрицы (Mathcad тратит около 8 байт памяти на каждый элемент матрицы), или удаления каких-либо больших побитовых отображений, массивов, матриц
Singular matrix	Сингулярная матрица	Эта матрица не может быть ни сингулярной, ни близкой к сингулярности	Матрица называется сингулярной, если ее определитель равен нулю. Матрица близка к сингулярной, если она имеет высокое число обусловленности *(см. гл. 9)*
The expression to the left of the equal sign cannot be defined	Выражение слева от знака равенства не может быть определено	В левой части находится что-то, что не является допустимым определяемым выражением	В левой части можно разместить одно из следующих определений: - имя переменной; - имя переменной с верхним или нижним индексом; - явный вектор или матрицу; - имя функции с аргументами: f(x,y) Любые другие выражение не допустимы

Таблица П4.1. *(продолжение)*

Ошибка	Перевод	Вероятная причина	Возможные пути устранения
The number of rows and/or columns in these arrays do not match	Число рядов и/или столбцов в этих массивах не согласовано	Попытка произвести матричные или векторные операции над массивами, размеры которых не совпадают	Например, сложение двух матриц разного размера недопустимо. Матричное умножение требует, чтобы число столбцов первой матрицы совпадало с числом строк второй *(см. разд. 9.1)*
The units in this expression do not match	Размерности в этом выражении не согласованы	Это сообщение появится, если складываются два элемента разной размерности либо создана матрица, элементы которой имеют разную размерность, либо Вы пытаетесь решить систему уравнений для неизвестных переменных разной размерности	Проверьте использование размерных переменных
There is an extra comma in this expression	В выражении лишняя запятая		Запятые должны использоваться для того, чтобы отделять: аргументы в функции первые два элемента области в определении интервала выражения в графике элементы во входной таблице нижние индексы в матрице Любые другие применения запятой приводят к ошибке. Например, запись 4,000 неправильная, а запись 4000 — правильная
This expression is incomplete. You must fill in the placeholders	Это выражение неполное. Необходимо добавить содержимое в местозаполнители	Не заполнены указанные **местозаполнители**	Необходимо дописать числа или выражения в указанные **местозаполнители**
This expression is incomplete. You must provide an operator	Это выражение неполное. Необходимо вставить оператор	Не заполнены **местозаполнители** оператора или пустое пространство между двумя операндами	Это могло произойти при удалении оператора, проверьте правильность ввода выражения

Таблица П4.1. *(продолжение)*

Ошибка	Перевод	Вероятная причина	Возможные пути устранения
This function has too many arguments	Эта функция имеет слишком много аргументов	Выделенное выражение содержит функцию с числом аргументов большим, нежели требуется	Проверьте правильность применения функции
This function is undefined at one or more of the points you specified	Эта функция не определена для одной или более точек	Попытка вычисления оператора или функции с неподходящими значениями	Например, -3! — выдаст ошибку, in{0), т. к. факториал не определен для отрицательного числа, а логарифм для нуля
This function needs more arguments	Этой функции не хватает аргументов	Выделенное выражение содержит функцию с меньшим, нежели требуется, числом аргументов	Для встроенных функций щелкните мышью на имени функции и воспользуйтесь подсказкой <F1>, чтобы проверить правильность числа и типа аргументов; для функции пользователя проверьте ее определение
This operation can only be performed on a function	Эта операция может быть произведена только над функцией	Этот аргумент должен быть функцией	Для встроенных функций щелкните мышью на имени функции и воспользуйтесь подсказкой <F1>
This operation can only be performed on an array. It can't be performed on a number	Эта операция может быть произведена только над массивом. Она не может быть произведена над числом		Например, это сообщение появится, если переменная верхнего индекса определена как скаляр. Поскольку переменная верхнего индекса представляет собой столбец матрицы, то ее следует определять как вектор Для поверхностных или контурных графиков массив данных должен иметь, по крайней мере, два ряда и два столбца
This operation can only be performed on a number or an array	Эта операция может быть произведена только над числом или массивом	Используемая функция или оператор требуют представления в виде константы, матрицы или вектора	

Таблица П4.1. (продолжение)

Ошибка	Перевод	Вероятная причина	Возможные пути устранения
This operation can only be performed on a string	Эта операция может быть произведена только над строкой	Используемая функция или оператор требуют представления в виде строки. Например строковые функции обычно требуют по крайней мере одного строкового аргумента	
This subscript is too large	Этот нижний индекс слишком велик	Попытка использовать верхний или нижний индекс, который превышает ограничения	
This value must be a matrix	Это значение должно быть матрицей	Попытка произвести матричную операцию не над матрицей	
This value must be a vector. It can be neither a matrix nor a scalar	Это значение должно быть вектором. Оно не может быть ни матрицей, ни скаляром	Это сообщение маркирует матрицу или скаляр в операциях, которые требуют вектора (одностолбцового массива). Например суммирование элементов вектора	
This value must be an integer greater than 1	Это значение должно быть целым числом, превосходящим 1	Это значение должно быть ≥ 1	При использовании встроенных функций щелкните мышью на имени функции и нажмите клавишу <F1>
This variable or function is not defined above	Эта переменная или функция не определена выше	Имя неопределенной функции будет помечено красным цветом	Удостоверьтесь, что эта функция или переменнуя определена выше. Это сообщение появится, если переменная некорректно используется в глобальном определении. Эта ошибка часто свидетельствует о том, что другое уравнение выше в документе является ошибкой. В этом случае все выражения, использующие выражение с ошибкой, будут помечены красным цветом

ТаблицаП4.1.(*продолжение*)

Ошибка	Перевод	Вероятная причина	Возможные пути устранения
Underflow	Потеря значимости (исчезновение значащих разрядов)	Из-за ограничений, присущих представлению чисел на компьютере, числа, которые слишком малы, не могут быть представлены. Это сообщение появляется, когда выражение включает такое число. Иногда, особенно в сложных вычислениях, промежуточный результат будет слишком мал, и вся разрядная сетка заполнится нулями	
Value of subscript or superscript is too big (or too small) for this array	Значение нижнего или верхнего индекса слишком велико (или слишком мало) для этого массива	Это выражение использует нижний или верхний индекс, который относится к несуществующему элементу массива	
This is not a scalar. Press <F1> for help	Это не скаляр. Нажмите клавишу <F1>, чтобы получить помощь	Использован вектор или выражение с интервалами, или какой-то другой тип выражения, где требуется применение скаляра	
You have one solve block inside another. Every "Given" must have a matching "Find" or "Minerr".	Один вычислительный блок содержится внутри другого. Каждому ключевому слову Given должно сопоставляться Find или Minerr	Указаны два ключевых слова Given подряд без Find или Minerr посередине. Вычислительный блок не может иметь внутри себя другой вычислительный блок	В качестве альтернативы можно задать функцию в терминах одного вычислительного блока и использовать ее внутри другого вычислительного блока. Во многих случаях это дает тот же самый эффект
You interrupted calculation. To resume, click here and choose "Calculate" from the "Math" menu	Вычисления прерваны. Для того чтобы продолжить, щелкните здесь и выберите пункт **Calculate** меню **Math**		Вычисления прерваны нажатием клавиши <Esc>. Для того чтобы пересчитать выделенное уравнение, наведите на него курсор и воспользуйтесь меню **Math / Calculate** (Математика / Вычислить)

Таблица П4.1. (продолжение)

Ошибка	Перевод	Вероятная причина	Возможные пути устранения
Сообщение об ошибках в символьных вычислениях			
Argument too large (Integer too large in context, Object too large)	Аргумент слишком велик	Обычно это результат вычисления выражения с плавающей точкой со значением большим, чем около 10x10 миллиардов	
Discarding large result	Сброс большого результата	Ответ слишком велик для отображения его в отформатированной математической области	Можно разместить ответ в буфере обмена
Expecting array or list	Ожидается массив или список	Операторы в упрощаемом или вычисляемом выражении требуют векторных или матричных операндов	
Expression contains non-symbolic operators	Выражение содержит несимвольные операторы	Применена символьная операция к выражению, содержащему местозаполнители оператора или переменной	
Floats not handled	С плавающей запятой не поддерживается	Команда **Factor** была применена к выражению с десятичным числом	
Illegal function syntax	Недопустимый синтаксис функции	Символьный процессор не может интерпретировать выражение, подобное (f)(x)	
Invalid arguments	Недопустимые аргументы	Символьный процессор не может выполнить требуемую операцию для данных аргументов	Это сообщение появится, если, например, применить скалярную функцию к массиву без использования оператора векторизации и выбрать команду **Symbolics / Simplify** (Символика / Упростить)

Таблица П4.1. *(окончание)*

Ошибка	Перевод	Вероятная причина	Возможные пути устранения
Invalid range	Недопустимый интервал	Для поиска численного решения уравнения символьный процессор пытается вычислить одну из своих встроенных функций за пределами области ее определения	
No answer found; stack limit reached	Ответа не найдено	Символьный процессор достиг предела своих возможностей без вычисления или упрощения, которое затребовал пользователь	
No answer found	Ответа не найдено	Символьный процессор не смог найти точного решения уравнения	
No closed form found for	Не найдено замкнутой формы для	Символьный процессор не смог найти интеграл или сумму, или произведение в замкнутой форме	
Syntax error	Синтаксическая ошибка	Обычно результат применения символьной операции в неподходящих или некорректных выражениях. Символьные вычисления выражений с размерностями также приведут к появлению этого сообщения	

Предметный указатель

M

N

O

P

Q

R

S

T

U

V

W

X

Z

А

Б

В

Г

Д

Е

З

И

К

Л

М

Н

О

П

Р

С

Т

У

Ф

Книги издательства "БХВ-Петербург" в продаже:

Серия "В подлиннике"

Андреев А. и др. MS Windows XP: Home Edition и Professional	848 с.
Андреев А. и др. Windows 2000 Professional. Русская версия	700 с.
Андреев А. и др. Microsoft Windows 2000 Server. Русская версия	960 с,
Андреев А. и др. Новые технологии Windows 2000	576 с.
Андреев А. и др. Microsoft Windows 2000 Server и Professional. Русские версии	1056 с.
Ахаян Р. Macromedia ColdFusion	672 с.
Браун М. HTML 3.2 {с компакт-диском)	1040 с.
Вебер Дж. Технология Java {с компакт-диском)	1104 с.
Власенко С. Компакт-диск с примерами к книгам серии "В подлиннике": "MS Office XP в целом", "MS Access 2002", "MS Word 2002", "MS Excel 2002"	32 с.
Власенко С Microsoft Word 2002	992 с
Гофман В., Хомоненко А. Delphi 6	1152 с.
Долженков В. MS Excel 2002	1072 с
Закер К. Компьютерные сети. Модернизация и поиск неисправностей	1008 с.
Колесниченко О., Шишигин И. Аппаратные средства PC, 4-е издание	1024 с.
Мамаев Е. MS SQL Server 2000	1280 с.
Матросов А. и др. HTML 4.0	672 с.
Михеева В., Харитонова И. Microsoft Access 2000	1088 с.
Михеева В., Харитонова И. Microsoft Access 2002	1040 с.
Новиков Ф., Яценко А. Microsoft Office 2000 в целом	728 с.
Новиков Ф., Яценко А. Microsoft Office XP в целом	928 с.
Ноутон П., Шилдт Г. Java 2	1072 с.
Пауэлл Т. Web-дизайн	1024 с.

Персон Р. Word 97	1120 с.
Питц М., Кирк Ч. XML	736 с.
Пономаренко С. Adobe Illustrator 9.0	608 с.
Пономаренко С. Adobe Photoshop 6.0	832 с.
Пономаренко С. CorelDRAW 9	576 с.
Пономаренко С. Macromedia FreeHand 9	432 с.
Русеев С. WAP: технология и приложения	432 с.
Секунов Н. Обработка звука на PC (с дискетой)	1248 с.
Сузи Р. Python (с компакт-диском)	768 с.
Тайц А. М., Тайц А. А, Adobe PageMaker 7.0	784 с.
Тайц А. М., Тайц А. А. Adobe InDesign	704 с.
Тайц А. М., Тайц А. А. CorelDRAW 9: все программы пакета	1136 с.
Тайц А. М., Тайц А. А. CorelDRAW 10: все программы пакета	1136 с.
Тихомиров Ю. Microsoft SQL Server 7.0	720 с.
Уильямс Э. и др. Active Server Pages (с компакт-диском)	672 с.
Усаров Г. Microsoft Outlook 2002	656 с.
Ханкт Ш. Эффекты CorelDRAW (с компакт-диском)	704 с.

Серия "Мастер"

CD-ROM с примерами к книгам "Ресурсы MS Windows NT Server 4.0" и "Сетевые средства Windows NT Server 4"	
Microsoft Press. Электронная коммерция. B2B-программирование (с компакт-диском)	368 с.
Microsoft Press. Visual Basic 6.0	992 с.
Microsoft Press. Ресурсы MS Windows NT Server 4.0	752 с
Айзекс С. Dynamic HTML (с компакт-диском)	496 с.
Анин Б. Защита компьютерной информации	384 с.
Асбари С. Корпоративные решения на базе Linux	496 с.
Березин С. Факс-модемы: выбор, подключение, выход в Интернет	256 с.
Березин С. Факсимильная связь в Windows	250 с.
Борн Г. Реестр Windows 98 (с дискетой)	496 с.

Бухвалов А. и др. Финансовые вычисления для профессионалов	320 с.
Валиков А. Технология XSLT	432 с.
Габбасов Ю. Internet 2000	448 с.
Гарбар П, Novell GroupWise 5.5: система электронной почты и коллективной работы	480 с.
Гарнаев А. Microsoft Excel 2000: разработка приложений	576 с.
Гарнаев А. Excel, VBA, Internet в экономике и финансах	816 с.
Гарнаев А., Гарнаев С. Web-программирование на Java и JavaScript	1040 с.
Гордеев О. Программирование звука в Windows (с дискетой)	384 с.
Гофман В., Хомоненко А. Работа с базами данных в Delphi	656 с.
Дарахвелидзе П. и др. Программирование в Delphi 5 (с дискетой)	784 с.
Дронов В. JavaScript в Web-дизайне	880 с.
Дубина А. и др. MS Excel в электронике и электротехнике	304 с.
Дубина А. Машиностроительные расчеты в среде Excel 97/2000 (с дискетой)	416 с.
Дунаев С. Технологии Интернет-программирования	480 с.
Жарков С. Shareware: профессиональная разработка и продвижение программ	320 с.
Зима В. и др. Безопасность глобальных сетевых технологий	320 с.
Киммел П. Borland C++ 5	976 с.
Костарев А. PHP в Web-дизайне	592 с.
Краснов М. DirectX. Графика в проектах Delphi (с компакт-диском)	416 с.
Краснов М. Open GL в проектах Delphi (с дискетой)	352 с.
Кубенский А. Создание и обработка структур данных в примерах на Java	336 с.
Кулагин Б. 3ds max 4: от объекта до анимации	448 с.
Купенштейн В. MS Office и Project в управлении и делопроизводстве	400 с.
Куприянов М. и др. Коммуникационные контроллеры фирмы Motorola	560 с.
Лавров С. Программирование. Математические основы, средства, теория	304 с.
Лукацкий А. Обнаружение атак	624 с.
Матросов А. Maple 6. Решение задач высшей математики и механики	528 с.
Медведев Е., Трусова В. "Живая" музыка на PC (с дискетой)	720 с.

Мешков А., Тихомиров Ю. Visual C++ и MFC, 2-е издание (с дискетой)	1040 с.
Миронов Д. Создание Web-страниц в MS Office 2000	320 с.
Мещеряков Е., Хомоненко А. Публикация баз данных в Интернете	560 с.
Михеева В., Харитонова И. Microsoft Access 2000: разработка приложений	832 с.
Новиков Ф. и др. Microsoft Office 2000: разработка приложений	680 с.
Нортон П. Разработка приложений в Access 97 (с компакт-диском)	656 с.
Одинцов И. Профессиональное программирование. Системный подход	512 с.
Олифер В., Олифер Н. Новые технологии и оборудование IP-сетей	512 с.
Подольский С. и др. Разработка интернет-приложений в Delphi (с дискетой)	432 с.
Полещук Н. Visual LISP и секреты адаптации AutoCAD	576 с.
Понамарев В. COM и ActiveX в Delphi	320 с.
Пономаренко С. Adobe InDesign: дизайн и верстка	544 с.
Попов А. Командные файлы и сценарии Windows Scripting Host	320 с.
Приписное Д. Моделирование в 3D Studio MAX 3.0 (с компакт-диском)	352 с.
Роббинс Дж. Отладка приложений	512 с.
Рудометов В., Рудометов Е. PC: настройка, оптимизация и разгон, 2-е издание	336 с.
Русеев Д. Технологии беспроводного доступа. Справочник	352 с.
Соколенко П. Программирование SVGA-графики для IBM	432 с.
Тайц А. Каталог Photoshop Plug-Ins	464 с.
Тихомиров Ю. MS SQL Server 2000: разработка приложений	368 с.
Тихомиров Ю. SQL Server 7.0: разработка приложений	370 с.
Тихомиров Ю. Программирование трехмерной графики в Visual C++ (с дискетой)	256 с.
Трельсен Э. Модель COM и библиотека ATL 3.0 (с дискетой)	928 с.
Федоров А., Елманова Н. ADO в Delphi (с компакт-диском)	816 с.
Федорчук А. Офис, графика, Web в Linux	416 с.
Чекмарев А. Windows 2000 Active Directory	400 с.

Чекмарев А. Средства проектирования на Java (с компакт-диском)	400 с.
Шапошников И. Web-сайт своими руками	224 с.
Шапошников И. Интернет-программирование	224 с.
Шапошников И. Справочник Web-мастера. XML	304 с.
Шилдт Г. Теория и практика С++	416 с.
Яцюк О., Романычева Э. Компьютерные технологии в дизайне, Логотипы, упаковка, буклеты (с компакт-диском)	464 с.

Серия "Изучаем вместе с BHV"

Березин С. Internet у вас дома, 2-е издание	752 с.
Тайц А. Adobe Photoshop 5.0 (с дискетой)	448 с.

Серия "Самоучитель"

Ананьев А., Федоров А. Самоучитель Visual Basic 6.0	624 с.
Васильев В. Основы работы на ПК	448 с.
Гарнаев А. Самоучитель VBA	512 с.
Герасевич В. Самоучитель. Компьютер для врача	640 с.
Дмитриева М. Самоучитель JavaScript	512 с.
Долженков В. Самоучитель Excel 2000 (с дискетой)	368 с.
Исагулиев К. Macromedia Dreamweaver 4	560 с.
Исагулиев К. Macromedia Flash 5	368 с.
Кетков Ю., Кетков А. Практика программирования: Бейсик, Си, Паскаль (с дискетой)	480 с.
Кирьянов Д. Самоучитель Adobe Premiere 6.0	432 с.
Кирьянов Д. Самоучитель MathCAD 2001	544 с.
Коркин И. Самоучитель Microsoft Internet Explorer 6.0	288 с.
Котеров Д. Самоучитель PHP 4	576 с.
Культин Н. Программирование на Object Pascal в Delphi 6 (с дискетой)	528 с.
Культин Н. Самоучитель. Программирование в Turbo Pascal 7.0 и Delphi, 2-е издание (с дискетой)	416 с.
Леоненков А. Самоучитель UML	304 с.

Матросов А., Чаунин М. Самоучитель Perl	432 с.
Омельченко П., Федоров А. Самоучитель Microsoft FrontPage 2002	576 с,
Омельченко Л., Федоров А. Самоучитель Windows 2000 Professional	528 с.
Омельченко Л., Федоров А. Самоучитель Windows Millennium	464 с.
Пекарев Л. Самоучитель 3D Studio MAX 4.0	370 с.
Полещук Н. Самоучитель AutoCad 2000 и Visual LISP, 2-е издание	672 с.
Полещук Н. Самоучитель AutoCAD 2002	608 с.
Понамарев В. Самоучитель Kylix	416 с.
Секунов Н. Самоучитель Visual C++ 6 {с дискетой)	960 с.
Секунов Н. Самоучитель C#	576 с.
Сироткин С. Самоучитель WML и WMLScript	240 с.
Тайц А. М., Тайц А. А. Самоучитель Adobe Photoshop 6 (с дискетой)	608 с.
Тайц А. М., Тайц А. А, Самоучитель CorelDRAW 10	640 с.
Тихомиров Ю. Самоучитель MFC (с дискетой)	640 с.
Хабибуллин И. Самоучитель Java	464 с.
Хомоненко А. Самоучитель Microsoft Word 2002	624 с.
Шапошников И. Интернет. Быстрый старт	272 с.
Шапошников И. Самоучитель HTML 4	288 с.
Шилдт Г. Самоучитель C++, 3-е издание (с дискетой)	512 с.

Серия "Компьютер и творчество"

Деревских В. Музыка на PC своими руками	352 с.
Дунаев В. Сам себе Web-дизайнер	512 с.
Дунаев В. Сам себе Web-мастер	288 с.
Людиновсков С. Музыкальный видеоклип своими руками	320 с.
Петелин Р., Петелин Ю. Аранжировка музыки на PC	272 с.
Петелин Р., Петелин Ю. Звуковая студия в PC	256 с.
Петелин Р., Петелин Ю. Музыка на PC. Cakewalk Pro Audio 9. Секреты мастерства	420 с.
Петелин Р., Петелин Ю. Музыка на PC. Cakewalk. "Примочки" и плагины	272 с.

Петелин Р., Петелин Ю. Музыкальный компьютер. Секреты мастерства	608 с.
Петелин Р., Петелин (О. Персональный оркестр в PC	240 с.

Серия "Учебное пособие"

Бенькович Е. Практическое моделирование динамических систем (с компакт-диском)	464 с.
Гомоюнов К. Транзисторные цепи	240 с.
Дорот В, Толковый словарь современной компьютерной лексики, 2-е издание	512 с.
Культин Н. С/С++ в задачах и примерах	288 с.
Культин Н. Turbo Pascal в задачах и примерах	256 с.
Порев В. Компьютерная графика	432 с.
Робачевский Г. Операционная система Unix	528 с.
Сафронов И. Бейсик в задачах и примерах	224 с.
Солонина А. и др. Алгоритмы и процессоры цифровой обработки сигналов	464 с.
Солонина А. и др. Цифровые процессоры обработки сигналов фирмы MOTOROLA	512 с.
Угрюмов Е. Цифровая схемотехника	528 с.
Шелест В. Программирование	592 с.

Серия "Знакомьтесь"

Надеждин Н. Карманные компьютеры	304 с.
Надеждин Н. Портативные компьютеры	288 с.
Надеждин Н. Знакомьтесь, цифровые фотоаппараты	304 с.

Серия "Быстрый старт"

Васильева В. Персональный компьютер. Быстрый старт	480 с.
Гофман В., Хомоненко А. Delphi. Быстрый старт	288 с.
Дмитриева М. JavaScript. Быстрый старт	336 с.
Культин Н. Microsoft Excel. Быстрый старт	208 с.
Хомоненко А., Гридин. В. Microsoft Access. Быстрый старт	304 с.