



Ю.Е. ВОСКОБОЙНИКОВ

А.Ф. ЗАДОРЖНЫЙ

Л.А. ЛИТВИНОВ

Ю.Г. ЧЕРНЫЙ

ОСНОВЫ ВЫЧИСЛЕНИЙ И ПРОГРАММИРОВАНИЯ В ПАКЕТЕ MATHCAD



НОВОСИБИРСК 2012

МИНИСТЕРСТВО ОБРАЗОВАНИЯ
И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ
АРХИТЕКТУРНО-СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ (СИБСТРИН)

**Ю.Е. Воскобойников, А.Ф. Задорожный,
Л.А. Литвинов, Ю.Г. Черный**

ОСНОВЫ ВЫЧИСЛЕНИЙ И ПРОГРАММИРОВАНИЯ В ПАКЕТЕ MATHCAD

*Рекомендовано Новосибирским региональным
отделением УМО вузов Российской Федерации по образованию
в области строительства в качестве учебного пособия
для студентов, обучающихся по направлениям
270800.62 «Строительство»
и 230400.62 «Информационные системы и технологии»*

*Под редакцией д-ра физ.-мат. наук, профессора
Ю.Е. Воскобойникова*

НОВОСИБИРСК 2012

УДК 004.9

ББК 32.9

О 851

Основы вычислений и программирования в пакете MathCAD : учеб. пособие / Ю. Е. Воскобойников [и др.] ; под ред. Ю. Е. Воскобойникова ; Новосиб. гос. архитектур.-строит. ун-т (Сибстрин). – Новосибирск : НГАСУ (Сибстрин), 2012. – 212 с.

ISBN 978-5-7795-0589-5

В учебном пособии рассмотрены основные конструкции встроенного языка математического пакета. Изложены основы работы в пакете MathCAD версии 14 для создания документов, построения графиков, матричных и векторных операций, программирования основных типов вычислительных алгоритмов (линейных, разветвляющихся и циклов) и формирования файлов данных. Подробно рассматривается модульное программирование и его реализация в пакете MathCAD.

Изложение сопровождается большим числом примеров и задач, что способствует лучшему усвоению материала. В конце каждой темы приводятся вопросы и задания для самопроверки полученных знаний.

Учебное пособие предназначено для бакалавров по направлениям «Строительство», «Информационные системы и технологии» при изучении учебных курсов «Информатика», «Математические пакеты». Учебное пособие будет полезно аспирантам и инженерам, использующим в своих расчетах этот математический пакет.

Печатается по решению издательско-библиотечного совета
НГАСУ (Сибстрин)

Рецензенты:

- А.А. Воевода, д-р техн. наук, профессор (НГТУ);
- Г.Г. Асташенков, д-р техн. наук, профессор, завкафедрой инженерной геодезии (НГАСУ (Сибстрин));
- В.П. Умрихин, канд. техн. наук, доцент, завкафедрой вычислительной математики (НГАВТ)

ISBN 978-5-7795-0589-5

© Воскобойников Ю.Е.,
Задорожный А.Ф., Литвинов Л.А.,
Черный Ю.Г., 2012

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	8
РАЗДЕЛ 1. ВЫЧИСЛЕНИЯ В ПАКЕТЕ MATHCAD.....	6
ТЕМА 1. МАТЕМАТИЧЕСКИЕ ПАКЕТЫ. ОБЩИЕ ХАРАКТЕРИСТИКИ ПАКЕТА MATHCAD	9
1.1. Современные математические пакеты	9
1.2. Возможности и структура пакета MathCAD.....	11
1.3. Программное окно MathCAD.....	13
1.4. Панели инструментов и палитры инструментов	19
1.5. Работа с документами MathCAD	21
ТЕМА 2. ВЫЧИСЛЕНИЯ В ПАКЕТЕ MATHCAD.....	29
2.1. Константы, переменные, операторы присваивания и вывода....	29
2.2. Арифметические операции MathCAD	33
2.3. Встроенные функции и функции пользователя.....	37
2.4. Операторы математического анализа.....	40
ТЕМА 3. МАССИВЫ В ПАКЕТЕ MATHCAD	48
3.1. Массивы в пакете MathCAD	48
3.2. Создание массивов в MathCAD	51
3.3. Верхний индекс массива.....	53
3.4. Основные функции обработки массивов	54
3.5. Вычисления с массивами.....	60
3.6. Функции сортировки элементов векторов и матриц.....	65
ТЕМА 4. ГРАФИКА В ПАКЕТЕ MATHCAD.....	69
4.1. Основные инструменты для построения графиков.....	69
4.2. Построение графиков функции одной переменной в декартовой системе координат	70
4.3. Построение графиков функции одной переменной в полярной системе координат	79
4.4. Построение графиков функций двух переменных	82
4.5. Анимация в MathCAD	90
ТЕМА 5. СИМВОЛЬНЫЕ ВЫЧИСЛЕНИЯ В MATHCAD	95
5.1. Команды символьных вычислений строки меню Symbolics....	95
5.2. Символьные вычисления командами палитры Symbolic.....	102
5.3. Символьное решение уравнений и систем уравнений.....	110

ТЕМА 6. ЧТЕНИЕ И ЗАПИСЬ ДАННЫХ	118
6.1. Средства доступа к файлам	118
6.2. Мастер Ввода-Вывода файлов	119
6.3. Функции доступа к структурированным ASCII-файлам	122
6.4. Функции доступа к графическим файлам	124
 РАЗДЕЛ 2. ПРОГРАММИРОВАНИЕ В ПАКЕТЕ MATHCAD	130
 ТЕМА 7. БЕЗМОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ В ПАКЕТЕ MATHCAD	131
7.1. Программирование линейных алгоритмов	131
7.2. Программирование разветвляющихся алгоритмов	132
7.3. Программирование циклических алгоритмов	143
 ТЕМА 8. МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ В ПАКЕТЕ MATHCAD	154
8.1. Сущность и преимущества модульного программирования	154
8.2. Описание подпрограммы-функции и локальный оператор присваивания	156
8.3. Обращение к подпрограмме-функции MathCAD	159
8.4. Программирование линейных алгоритмов в подпрограмме-функции MathCAD	161
8.5. Программирование разветвляющихся алгоритмов в подпрограмме-функции MathCAD	161
8.6. Программирование циклических алгоритмов в подпрограмме-функции MathCAD	169
 ТЕМА 9. ПРОГРАММИРОВАНИЕ ТИПОВЫХ ЗАДАЧ В ПОДПРОГРАММАХ-ФУНКЦИЯХ MATHCAD	188
9.1. Программирование разветвляющихся алгоритмов	188
9.2. Программирование циклов типа арифметической прогрессии	191
9.3. Программирование итерационных циклов	202
9.4. Тестирование и отладка программ в пакете MathCAD	208
 ЗАКЛЮЧЕНИЕ	216
 БИБЛИОГРАФИЧЕСКИЙ СПИСОК	217

Не только боги изучают MathCAD,
но и студенты НГАСУ

Напутствие авторов пособия

ВВЕДЕНИЕ

При использовании вычислительной техники встает проблема реализации необходимых алгоритмов в виде так называемых программ. Для решения этой проблемы в различные годы использовались следующие средства:

- *программирование в машинных кодах* (включая языки типа Ассемблер);
- *программирование на языках высокого уровня* (включая объектно-ориентированное программирование);
- *системы компьютерной математики*.

Разработка программы (даже с использованием языков высокого уровня с приставками Visual) требует и *соответствующей подготовки* (назовем ее «*программистской*»), и *достаточно большого количества времени*. И то и другое часто отсутствует у обычного пользователя, который является специалистом в своей предметной области, но не в программировании. Поэтому, начиная с 90-х годов прошлого века, широкую известность и заслуженную популярность приобрели так называемые *системы компьютерной математики*, или, проще, *математические пакеты*. Сейчас эти системы получили широкое распространение.

Изучению основ вычислений и программирования в пакете MathCAD (версии 14) посвящается данное учебное пособие. Авторы надеются, что пакет MathCAD станет хорошим помощником при выполнении расчетов в курсовом и дипломном проектировании.

РАЗДЕЛ 1. ВЫЧИСЛЕНИЯ В ПАКЕТЕ MATHCAD

ТЕМА 1. МАТЕМАТИЧЕСКИЕ ПАКЕТЫ. ОБЩИЕ ХАРАКТЕРИСТИКИ ПАКЕТА MATHCAD

В этой теме делается краткое сравнение математических пакетов. Рассматриваются возможности, структура, программное окно пакета MathCAD.

1.1. Современные математические пакеты

В настоящее время для решения научно-инженерных задач все более широкое применение находят системы компьютерной математики, или математические пакеты. Наиболее популярными из них являются *Maple*, *MatLab*, *MathCAD*. Кратко охарактеризуем эти математические пакеты.

Пакет Maple ориентирован на достаточно широкий круг пользователей. Задание алгоритма вычислений осуществляется записью на входном языке пакета соответствующих математических формул. При вводе сложных выражений это вызывает определенные затруднения.

Пакет MatLab. MatLab – одна из старейших, тщательно проработанных и апробированных временем систем компьютерной математики, построенная на расширенном представлении и применении матричных операций (MatLab – Matrix Laboratory – матричная лаборатория). В настоящее время MatLab вышла за пределы специализированной матричной системы и является одним из наиболее мощных математических пакетов, сочетающим в себе удобную оболочку, редактор, вычислитель и графический процессор.

Пакет MathCAD. MathCAD является мощной системой компьютерной математики, сочетающей в себе визуально ориентированный входной язык, удобный редактор текста и формул, численный и символьный процессоры. Пакет достаточно прост в изучении, а наличие большого числа электронных книг

и «быстрых шпаргалок» существенно упрощают его применение для решения конкретных научно-инженерных задач.

Ниже приводятся фрагменты вычисления определенного интеграла вида

$\int_0^1 \frac{1}{\sqrt{1-x^2}} dx$ в разных математических пакетах.

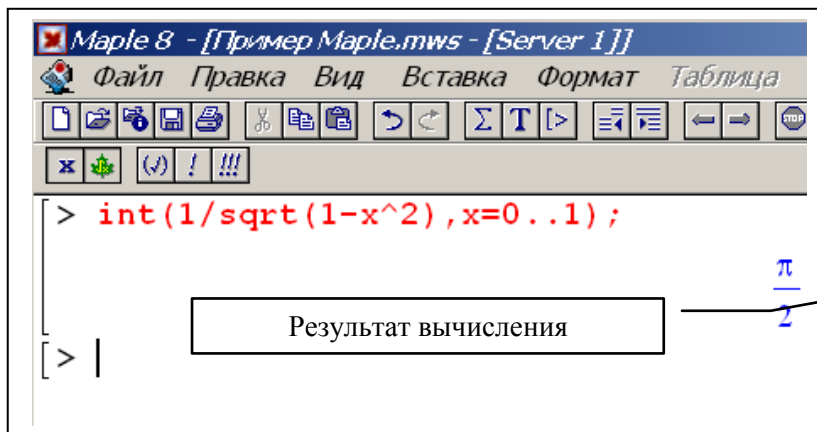


Рис. 1.1. Вычисление определенного интеграла в пакете Maple

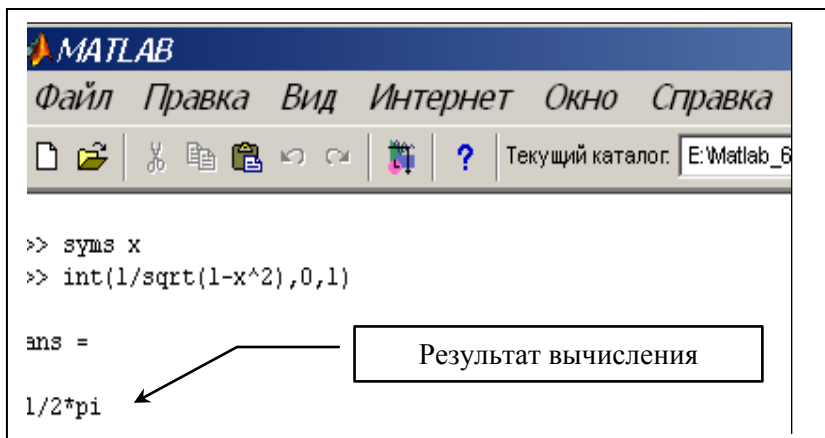


Рис. 1.2. Вычисление определенного интеграла в пакете MatLab

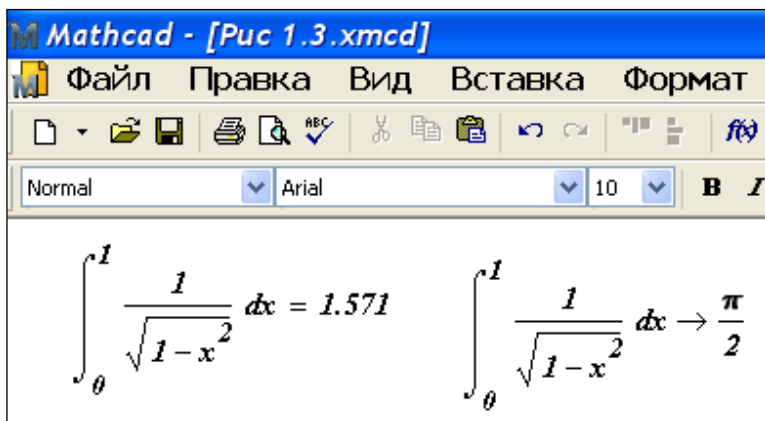


Рис. 1.3. Вычисление определенного интеграла
в пакете MathCAD

Сравнивая эти три рисунка, приходим к выводу, что запись операторов для вычисления определенного интеграла в пакете MathCAD очень близка к стандартному языку математических расчетов. Это обстоятельство упрощает постановку и решение задачи.

В отличие от других приведенных на рисунках математических пакетов (не говоря уже о языках программирования), MathCAD позволяет записывать на экране компьютера формулы в их привычном виде, как они представляются в печатном виде в книгах или как их записывают на листе бумаги. Эта и другие ниже рассматриваемые особенности делают пакет MathCAD почти идеальной вычислительной средой для решения научных и инженерных задач различного уровня сложности.

1.2. Возможности и структура пакета MathCAD

Пакет MathCAD можно отнести к *математически универсальным системам*. Объединение этих слов означает, что, помимо собственных вычислений (как численных, так и символьных), MathCAD позволяет подготавливать качественные тексты,

дипломные и курсовые проекты, диссертации, статьи с наглядным графическим представлением результатов вычислений (в том числе и анимации). К безусловным достоинствам пакета следует отнести возможность сохранения документов в формате Web-страниц, причем создание файлов с рисунками происходит автоматически.

Применение библиотек и пакетов расширений обеспечивает профессиональную ориентацию MathCAD на любую область науки, техники и образования.

Пакет MathCAD, в отличие от других современных математических пакетов, построен в соответствии с *принципом WYSIWYG* («*What You See Is What You Get*» – «*Что видите, то и получите*»). Он прост в использовании и по принципу обработки документа является программой-интерпретатором, т.е. после редактирования документа происходит его пересчет. Поэтому отсутствуют этапы трансляции, компиляции и загрузки программы.

MathCAD является интегрированной системой программирования, ориентированной на проведение математических, инженерно-технических, статистических и экономических расчетов.

MathCAD содержит: *текстовый редактор; вычислительный и графический процессор; справочную систему.*

Текстовый редактор служит для ввода и редактирования текстов. Вычислительный процессор системы MathCAD осуществляет численные вычисления по заданным математическим формулам, имеет обширный набор встроенных математических функций, обеспечивает вычисление рядов, сумм и произведений, определенных интегралов и производных. Символьный процессор позволяет получать результаты вычисления в символьном (аналитическом) виде. Второй интеграл на рис. 1.3 вычислен в символьном виде. Графический процессор служит для создания графиков функций одной и двух переменных.

Как отмечалось, входной язык пакета MathCAD позволяет записывать на экране компьютера формулы в их привычном ви-

де, как они представляются в печатном виде в книгах или как их записываем на листе бумаги.

Например, вычисление корня квадратного уравнения в электронной таблице Excel записывается в ячейку как:

$$=(-B1+КОРЕНЬ(B1*B1-4*A1*C1))/(2*A1).$$

В MathCAD то же самое уравнение выглядит таким, каким оно выглядит в тексте учебника или справочнике:

$$x_1 := \frac{-b + \sqrt{b^2 - 4ac}}{2a}.$$

Кроме того, уравнения и графики в MathCAD являются действующими. При изменении любых данных, переменных или уравнений MathCAD немедленно повторно пересчитает математические выражения и перерисует графики.

1.3. Программное окно MathCAD

Программное окно MathCAD, оформленное стандартным образом. Программное окно содержит следующие элементы: *строка заголовка; строка главного меню; панель инструментов; рабочая область; строка состояния.*

Строка заголовка содержит название программного пакета и имя открытого файла MathCAD. Если открыт новый документ, то в поле *Имя файла* находится фраза *Untitled (Без названия)*.

Строка меню располагается в верхней части программного окна ниже строки заголовка. В строке меню представлены следующие пункты:

- *File (Файл)* – работа с файлом (создать, открыть, закрыть, сохранить, печатать, отослать по почте);
- *Edit (Правка)* – редактирование документа; позволяет выполнить вырезать (cut), копировать (copy) и вставить (paste) часть документа, а также отменить (undo) предшествующую операцию или повторить (redo) отмененную операцию;

- *View (Вид)* – способы представления и обработки (toolbars) документа;
- *Format (Формат)* – изменение формата документа;
- *Tools (Инструменты)* – средства управления вычислениями (Calculate), анимации (Animation), отладки (Debug);
- *Symbolics (Символьные)* – выбор операций символьных вычислений;
- *Window (Окно)* – управление расположением окон документа (каскадное, вертикальное, горизонтальное), список открытых документов MathCAD;
- *Help (Справка)* – работа со справочной базой данных о системе, центром ресурсов и электронными книгами.

Команды пункта View. Пункт *View* строки меню содержит следующие команды:

- *Toolbars (Панели инструментов)* – список панелей и палитр, которые можно вывести на окно документа и скрыть;
- *Ruler (Линейка)* – установка мерной линейки;
- *Status bar (Строка состояния)* – установка строки состояния внизу документа;
- *Trace Window (Окно трассировки)* – скрывает или отображает окно трассировки в нижней части окна главного листа; используется для отладки программ; окно трассировки можно перемещать с помощью мыши, нажав и удерживая клавишу [Ctrl];
- *Header and Footer (Колонтитулы)* – элементы дизайна документа;
- *Regions (Области)* – создает цветной фон (по умолчанию серого цвета), в котором выделяет области документа;
- *Annotations (Аннотации)* – отображает цветные скобки вокруг всех областей ввода в документе;
- *Refresh (Обновить)* – обновляет содержимое экрана; устраняет остатки графических объектов и линий выделения;

- *Zoom (Масштаб)* – увеличивает или уменьшает размер документа при изменении процента масштаба.

Работа со справкой и информационным центром. Команды для работы со справочной системой сосредоточены в меню *Help (Справка)*, которое показано на рис. 1.4.

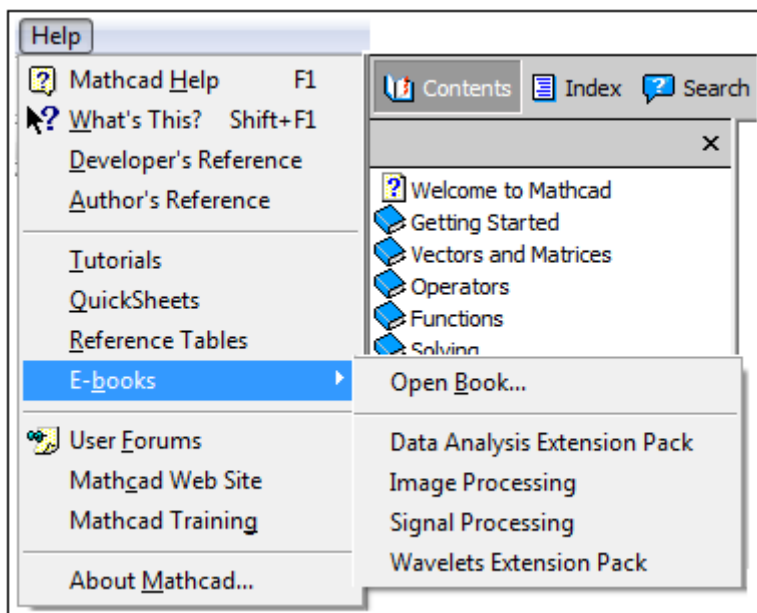


Рис. 1.4. Позиции строки меню Help

Для вызова справочной системы необходимо набрать команду MathCAD Help или нажать клавишу F1. Окно справочной системы имеет три вставки:

Contents (Содержимое) – оглавление справочной системы, которое представляет собой древовидную структуру с вложенными папками;

Index (Индекс) – предметный указатель, в котором по мере ввода для поиска требуемого слова в списке ниже будут появляться наиболее близкие по написанию слова;

Search (Поиск) – система поиска в базе данных справочной системы.

Справочная система широко использует гиперссылки. При активизации ссылки открывается связанная с ней часть справки.

Если курсор ввода установить на какую-либо функцию, то при нажатии на клавишу F1 открывается окно справочной системы с информацией именно об этой функции.

Центр информационных ресурсов. Центр информационных ресурсов включает в себя: *QuickSheets (Быстрые шпаргалки)*, *Reference Table (Справочные таблицы)*, *E-books (Электронные книги)* и *MathCAD Web site*.

«Быстрые шпаргалки» (*QuickSheets*) – это MathCAD документы, содержащие готовые примеры MathCAD, решающие широкий круг математических и научно-технических задач, от решения уравнений до построения диаграмм и выполнения процедур математического анализа. Содержание раздела *QuickSheets* показано на рис. 1.5.

<u>About QuickSheets</u>	<u>Data Analysis</u>
<u>Mathcad Techniques</u>	<u>Statistics</u>
<u>Vectors and Matrices</u>	<u>Using Mathcad with Other Applications</u>
<u>Solving Equations</u>	<u>Symbolic Math</u>
<u>Graphing and Visualization</u>	<u>Programming</u>
<u>Calculus and DiffEQs</u>	<u>Extra Math Symbols</u>
<u>Engineering Applications</u>	

Рис. 1.5. Содержание раздела *QuickSheets*

Теперь достаточно щелкнуть по ссылке нужной темы и появится документ с соответствующими примерами.

Справочные таблицы (*Reference Table*) – раздел, содержащий множество таблиц с данными о различных химических элементах, их физических свойствах, сведения о математических константах, таблицы производных и интегралов, формулы

по механике, электротехнике, радиотехнике и т.д. На рис. 1.6 показано окно с перечнем доступных таблиц.

Электронные книги (E-books) – документы с описанием работы основных функций MathCAD. Электронные книги объединены в тематические разделы. Документы вызываются гипермедиа- и гипертекстовыми ссылками друг на друга. Электронной книгой можно пользоваться для получения нужной информации и освоения приемов работы с MathCAD.

Reference Tables		
Basic Science	Properties of Liquids	Properties of Gases
Fundamental Constants	Density	Specific Gravity
Physics - Mechanics	Viscosity	Specific Heat
Periodic Table	Specific Gravity	Sound Velocity
Calculus	Sound Velocity	Molecular Weight
Derivative Formulas	Surface Tension	
Integral Formulas	Dielectric Constant	Properties of Metals
	Index of Refraction	Thermal Conductivity
Geometry	Molecular Weight	Specific Gravity
Areas and Perimeters		Linear Expansion Coefficient
Volumes and Surface Areas	Properties of Solids	Electrical Resistivity

Рис. 1.6. Окно списка справочных таблиц

Рабочее поле программного окна MathCAD находится под панелями инструментов и в нем располагается одно или несколько окон документов MathCAD. При работе с несколькими окнами документов используется пункт *Window (Окно)* строки меню. В нижней части программного окна располагается *Строка состояний*, в которой отображаются подсказки о назначении элементов на панели инструментов при подведении к этим элементам курсора мыши, режим вычислений MathCAD и номер текущей страницы.

В рабочем поле программного окна может находиться одно или несколько окон документов MathCAD. Каждый документ

MathCAD может содержать любое число математических областей и текстовых областей.

Математическая область содержит конструкции пакета MathCAD, необходимые для реализации того или иного алгоритма решения задачи. В этой же области строятся графики. Курсор в математической области имеет форму красного креста.

Текстовая область может включать любые символы (русские, латинские, греческие) в виде комментариев к вычислительным фрагментам. Для создания текстовой области необходимо обратиться к пункту *Insert*, команда *Text Region* (рис. 1.7).

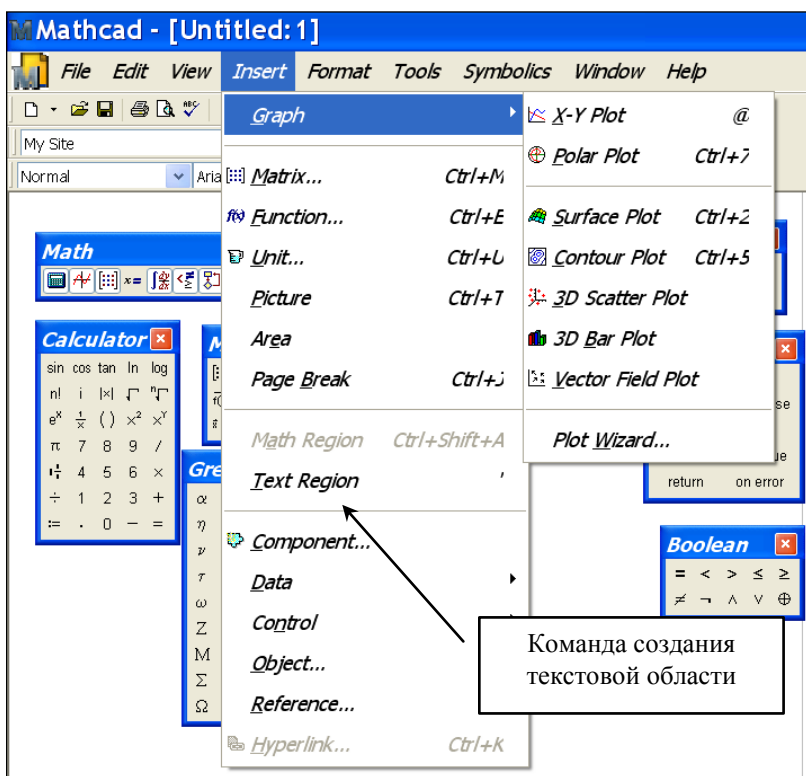


Рис. 1.7. Создание текстовой области

Работа в текстовой области осуществляется точно так же, как и в обычном текстовом редакторе (операции редактирования, форматирования и т.д.).

1.4. Панели инструментов и палитры инструментов

К панелям инструментов относятся три панели: **Стандартная**, **Форматирования** и панель **Math**. Первые две панели напоминают соответствующие панели текстового процессора Word и табличного процессора Excel. Если панель **Math** не отображена, она может быть вызвана из строки меню, пункт *View (Вид)*, в которой выполняется команда *Toolbars (Панели инструментов)* и в ней – *Math (Математика)*.

Панель инструментов **Math** содержит кнопки для вызова девяти *палитр инструментов*. Вид кнопок и названия палитр инструментов приведены на рис. 1.8.

Каждая палитра инструментов в свою очередь содержит кнопки для вызова конкретной функции или оператора. Например, палитра *Калькулятор* содержит арифметические операторы, а палитра *Матанализ* включает в себя операторы интегрального и дифференциального исчисления и т.д. Используя кнопки панели *Математика*, можно оперативно вызвать или убрать с экрана любую палитру инструментов. Значки инструментов всех девяти палитр показаны на рис. 1.9.



Рис. 1.8. Значки панели инструментов **Math**

Заметим, что вызвать любую палитру инструментов можно как через панель инструментов *Math*, так и через команду *Toolbars*.

С помощью представленных палитр можно вводить в документы MathCAD практически все известные математические символы и операторы. Поскольку их очень много, вывод всех палитр (как показано на рис. 1.9) обычно не нужен, так как в окне редактирования не остается место для подготовки документов. Поэтому рекомендуется неиспользуемые в данное время палитры закрывать.

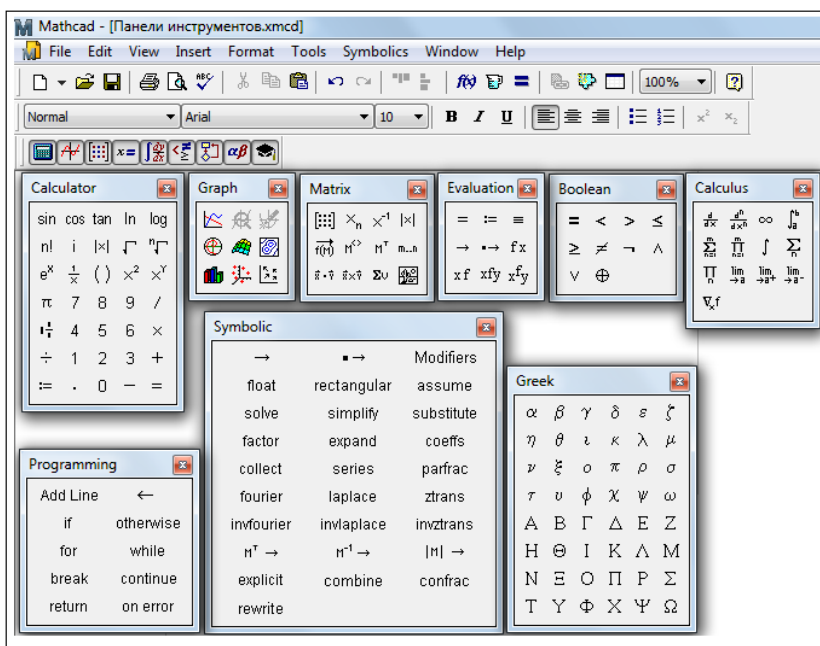


Рис. 1.9. Палитры инструментов

1.5. Работа с документами MathCAD

Создание документа MathCAD. Создание документа в MathCAD может включать ввод и редактирование математических формул, задание форматов отображения содержимого документа, ввод текстовых описаний и т.д. Документ одновременно является и листингом MathCAD-программы, и результатом выполнения этой программы, и отчетом, пригодным для распечатки на принтере или публикации в Web. Ввод выражений в документ MathCAD осуществляется в математические или в текстовые области.

Работа с текстовой областью документа. Отметим, что по умолчанию устанавливается *математическая область* ввода. *Текстовая область* может быть создана *тремя способами*. Первоначально установите курсор в нужном месте документа. Для этого щелкните левой клавишей мыши так, чтобы позиционировать красное перекрестие курсора в желаемой позиции.

Первый способ самый простой: просто начните вводить текст. Когда после первого слова будет набран пробел, MathCAD изменит текущую математическую область в текстовую. Вводимый текст будет заключен в прямоугольник, отображающий текстовую область.

Второй способ состоит в том, что необходимо ввести символ “ (одна двойная кавычка). Появится прямоугольная рамка, в которой можно вводить текст.

Третий способ – использовать кнопку *Insert (Вставка)* строки меню и выполнить команду *Text Region (Текстовая область)*. На месте курсора также появится прямоугольная рамка (см. рис. 1.7).

Текстовая область расширяется по мере ввода текста. Чтобы завершить режим ввода текста, необходимо щелкнуть мышью вне текстовой области. Линии, отображающие границы текстовой области, исчезнут.

Перемещение и копирование. Выделенные рамкой текстовые области можно переносить на другое место, зацепившись за рамку указателем мыши – он при этом превращается

в изображение кисти руки. Если до начала перемещения нажать и удерживать клавишу Ctrl, то будет выполняться копирование области с сохранением ее на первоначальном месте.

Редактирование. Для редактирования текстовой области необходимо щелкнуть на ней мышью. Появится курсор в виде красной вертикальной черты. Тип шрифта, размер, стиль и цвет могут быть изменены так же, как в большинстве офисных программ: с помощью главного меню, с помощью панели инструментов *Форматирования*. С помощью раскрывающегося списка выборы шрифтов на панели форматирования (рис. 1.10) можно установить шрифт, такой как, например, Arial или Times New Roman, установить размер в пикселях (например, 10 пикселей или 12 пикселей), стиль шрифта, например, Normal или *Italics*.

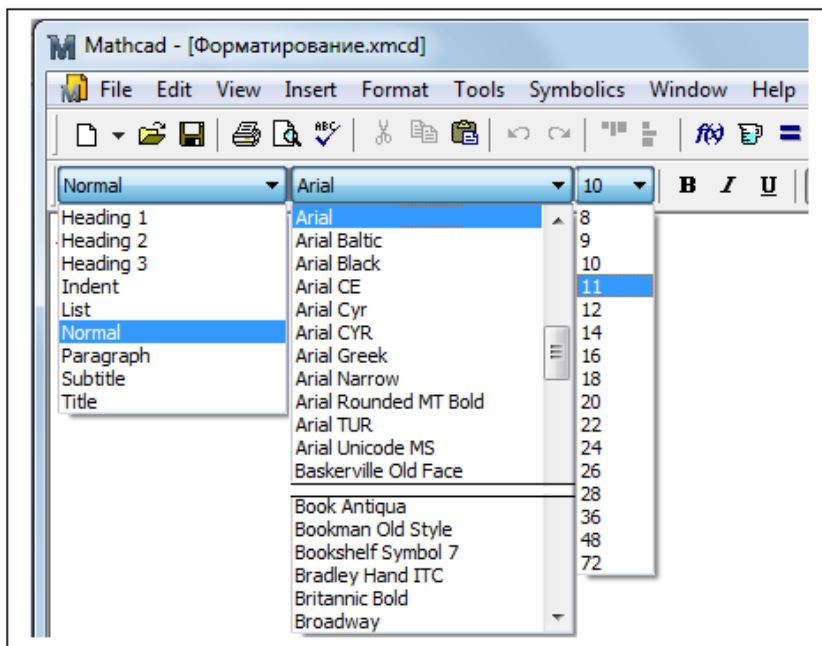


Рис. 1.10. Выбор шрифта и размера

При использовании главного меню выполните *Format* (Формат) – *Text* (Текст). Откроется диалоговое окно *Text Format* (рис. 1.11), в котором могут быть сделаны требуемые изменения в формате текста.

Диалоговое окно *Text Format* можно вызвать и другим способом: выделите часть текста, формат которого необходимо изменить, проведя мышью с зажатой левой кнопкой над текстом; удерживая курсор мыши над выделенным текстом, щелкните правой кнопкой; в раскрывшемся списке выберите кнопку *Font* (Шрифт).

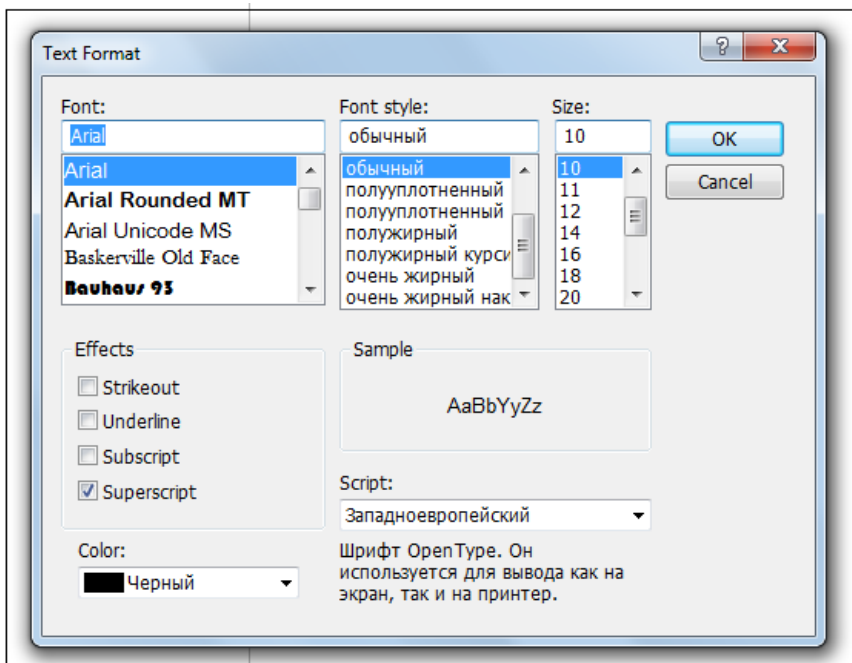


Рис. 1.11. Диалоговое окно *Text Format*

Отметим, что два последних метода предоставляют больше возможностей в форматировании текста, чем при использовании панели форматирования, например, позволяют создать над-

строчный (*Superscript*) или подстрочный (*Subscript*) индексы, перечеркнутый (*Strikeout*) и подчеркнутый (*Underline*) текст, а также позволяют изменять цвет текста.

Цвет фона можно изменить, если в главном меню выбрать: *Format (Формат) / Properties (Свойства)*. В открывшемся диалоговом окне (рис. 1.12) установить флажок *HighLight Region (Подсветка области)* и выбрать требуемый цвет. Это же окно можно открыть, щелкнув правой кнопкой мыши в любом месте текстовой области и затем выполнить команду контекстного меню *Properties (Свойства)*.

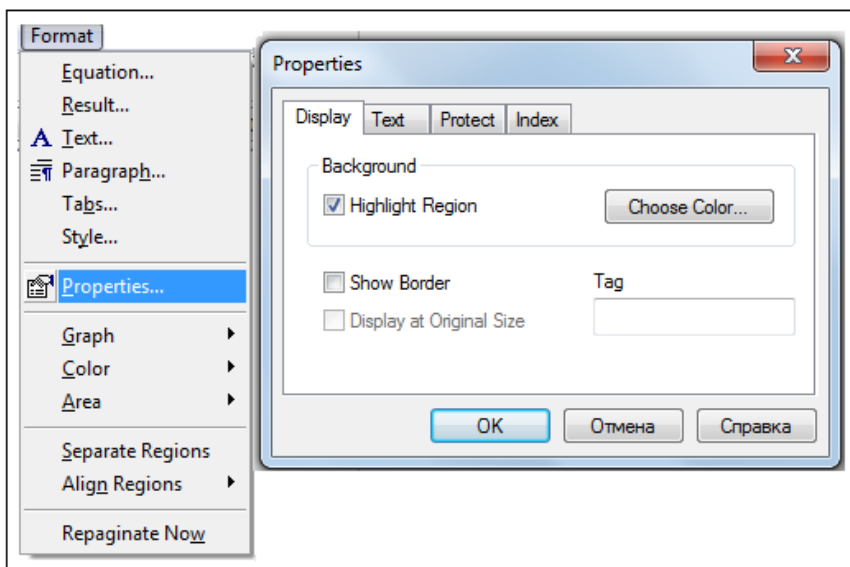


Рис. 1.12. Установка цвета фона

Выделение областей. Так как в рабочем окне MathCAD могут находиться несколько областей, то при их переносе или редактировании возможно частичное и даже полное наложение одних областей на другие. Некоторые области могут оказаться

скрытыми. В MathCAD имеется возможность выделения областей с помощью команд главного меню *View (Вид) / Regions (Области)*. При этом цвет фона области не изменится, а промежутки между областями закрасятся в серый цвет и области станут выделенными на затемненном фоне рабочего окна. Это позволяет видеть взаимное расположение областей и оценить степень их перекрытия.

Работа с математической областью документа. Математические формулы и выражения в документе MathCAD вводятся в *математическую область*. Математическая область устанавливается по умолчанию. Поэтому для ввода достаточно позиционировать указатель мыши в нужном месте документа и щелкнуть левой кнопкой. На экране появится маркер ввода в виде красного крестика. С этого места можно начинать набор формул. С началом ввода маркер превращается в *выделяющий уголок* синего цвета. Использование выделяющего уголка и управление им рассматривается в следующей теме (см. п. 2.2).

При необходимости выражения и функции математической области можно редактировать и форматировать, используя панель инструментов *Форматирование*.

Сохранение документа MathCAD. Для сохранения созданного документа MathCAD в файле необходимо использовать пункт *File (Файл)* строки меню и далее команду *Save as (Сохранить как)*. В появившемся диалоговом окне выбрать папку, где будет сохранен новый файл, и назначить ему имя.

При этом можно сохранить документ не только в формате MathCAD 14 (расширение файла *.xmcd), но и в форматах более поздних версий MathCAD (рис. 1.13), а также в формате web-страниц (расширение файла *.htm) и формате RTF (расширение файла *.rtf) для последующего редактирования в более мощных текстовых редакторах (например, Microsoft Word).

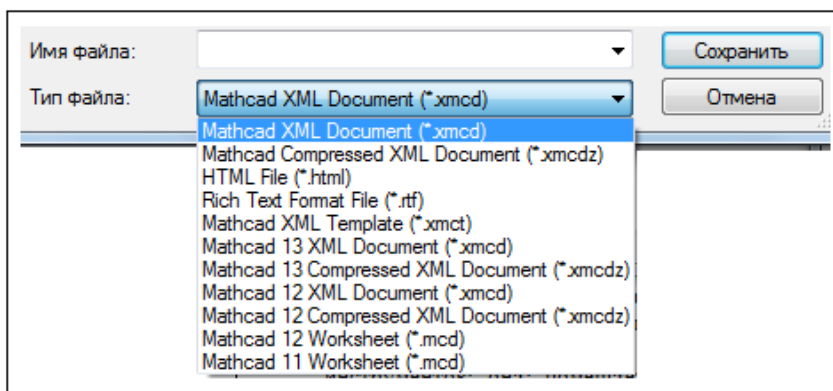


Рис. 1.13. Список форматов сохранения документа MathCAD

Символы как в математической, так и в текстовой областях можно форматировать, используя для этого панель инструментов *Форматирование*.

Открытие созданного документа MathCAD осуществляется точно так же, как в Word или Excel.

Для удаления, копирования и вставки фрагментов документа MathCAD можно использовать команды буфера обмена (инструменты панели *Стандартная* или контекстного меню).

Вопросы и задания для самопроверки

1. Основные особенности и возможности математического пакета MathCAD?
2. Что размещается в математической и текстовой областях документа MathCAD?
3. В каких форматах можно сохранить документ MathCAD?
4. Какие панели инструментов присутствуют в пакете MathCAD?
5. С использованием какой панели инструментов можно поменять размер шрифта в документе MathCAD?
6. Определите, какая палитра инструментов содержит наибольшее число кнопок инструментов?
7. Выполните следующие операции:
 - Загрузите пакет MathCAD.
 - Откройте программное окно на весь экран.
 - Расположите панели, как показано на рис. 1.1.
 - Присвойте файлу имя и сохраните его в заданной папке.
8. Выполните следующие операции:
 - Загрузите два документа MathCAD.
 - Расположите их на экране монитора один над другим, один с другим бок о бок.
9. Выполните следующие операции:
 - В справочнике найдите названия функций MathCAD, выполняющих сортировку.
 - В информационном центре найдите электронный документ с описанием применения этих команд.
 - Скопируйте текст описания в документ Word и выполните его перевод.
 - Найдите в информационном центре примеры использования команд сортировки.

10. Выполните следующие операции:

- Позиционируйте курсор на произвольном месте листа.
- Выполните команду *Insert (Вставка)/TextRegion (Текстовая область)*.
- В текстовой области введите произвольный текст, содержащий несколько строк.
- Измените размер, цвет и жирность текста.
- Включите флажок *View(Вид)/Regions(Области)*.
- Переместите текстовую область, для чего:
 - выделите область с текстом щелчком левой кнопкой мыши;
 - подведите курсор мыши к границе области. В этот момент он принимает форму руки;
 - удерживая левую кнопку мыши, переместите область в нужное место и отпустите кнопку мыши.
- Скопируйте текстовую область на первоначальное место, для чего:
 - выделите область с текстом щелчком левой кнопкой мыши;
 - подведите курсор мыши к границе области. В этот момент он принимает форму руки;
 - удерживая левую кнопку мыши и клавишу [Ctrl], переместите область в нужное место и отпустите кнопку мыши.
- Удалите текстовую область, для чего:
 - выделите область и выполните команду меню *Edit (Правка)/Delete (Удалить)*.

ТЕМА 2. ВЫЧИСЛЕНИЯ В ПАКЕТЕ MATHCAD

В этой теме будут рассмотрены основные (базовые) конструкции входного языка MathCAD для программирования простых вычислений в документе MathCAD.

2.1. Константы, переменные, операторы присваивания и вывода

Константы. Входной язык предусматривает работу со следующими часто используемыми типами констант:

- *целочисленные константы* (например, 12, -24, 0 и т.д.);
- *вещественные числовые константы*, которые могут записываться в одной из двух форм: с фиксированной точкой (например, 3.265) и с десятичным порядком (в экспоненциальной форме), записываемая в виде $a \cdot 10^p$, где a — целочисленная константа или вещественная константа с фиксированной точкой. Точка означает операцию умножения (клавиша [*]), p — десятичный порядок. Для ввода порядка нажать клавишу [^] — операция возведения в степень;
- *комплексные константы*, записываемые в виде $a + bi$, причем между величиной мнимой части b и мнимой единицей i не ставится знак операции умножения;
- *строковые константы* — любая последовательность символов (в том числе русские и греческие буквы), заключенные в кавычки (например, "Это строковая константа");
- *единицы измерения физических величин*.

Переменная. В пакете MathCAD можно использовать следующие переменные: *простые, индексированные (массивы), дискретные, глобальные, системные и размерные*.

Каждая *переменная MathCAD* имеет свое оригинальное имя. Имя переменной (идентификатор) – это набор из букв, цифр и символов, но обязательно начинающийся с буквы (латинской или греческой). **Использование в имени русских букв, а также пробелов запрещено.**

Греческие буквы вводятся с палитры инструментов *Греческий*. В конце имени переменной могут стоять нижние индексы, для ввода которых нужно нажать клавишу $[.]$ – десятичную точку.

Использование в имени строчных и прописных букв определяет имена разных переменных. Так, имена X_2 и x_2 соответствуют разным переменным MathCAD (часто эта особенность является причиной ошибок).

В отличие от алгоритмического языка Pascal переменную в MathCAD не нужно предварительно описывать – ее тип определяется автоматически при задании переменной конкретного значения. Не заданные переменные выделяются на экране *красным цветом*.

Значение переменной можно задать с помощью *оператора присваивания* ($:=$), имеющего следующий вид

$$\langle \text{Имя_переменной} \rangle := \langle \text{Выражение} \rangle$$


Для ввода знака присваивания « $:=$ » нажать клавишу с символом двоеточия $[:]$. Оператор присваивания, задающий значение некоторой переменной, должен стоять в документе *левее* или *выше* точки использования этой переменной. Оператор присвоения позволяет переназначить значение переменной. До оператора присвоения переменная в документе может иметь одно значение, а после (правее или ниже) новое значение.

Вывод значения переменной производится *оператором вывода* $[=]$.

Простая переменная – переменная, значение которой определяется одной величиной. Например, определение переменной x : $x := 2$. Для вывода значения переменной необходимо набрать: $x =$.

Дискретная переменная (в литературе часто используется термин *ранжированная переменная*) – переменная, принимающая ряд значений, меняющихся по закону арифметической прогрессии. Дискретная переменная определяется заданием начального значения, шага и конечного значения. Если шаг не задан, он предполагается равным 1 или –1. Например,

$$\langle Name \rangle := N1..N2$$

где *Name* – имя переменной; *N1* – начальное значение; *N2* – конечное значение. Символ двоеточия «:=» вводится нажатием клавиши [:=] или кнопкой  палитры инструментов *Калькулятор*. Если $N1 < N2$, шаг равен 1; если $N1 > N2$, шаг равен –1.

Задание дискретной переменной с произвольным шагом имеет вид:

$$\langle \text{Имя переменной} \rangle := \langle \text{Нач_знач} \rangle, \langle \text{След_знач} \rangle .. \langle \text{Кон_знач} \rangle$$

Шаг *h* задается неявно и вычисляется как:

$$h := \langle \text{След_знач} \rangle - \langle \text{Нач_знач} \rangle$$

Например:

- $x := 0.1, 0.3 .. 1.5$. Здесь шаг равен 0.2.
- $i := 1 .. 5 \quad j := 5 .. 1$. Здесь шаг у переменной *i* равен 1, а у переменной *j* равен –1.

Для вывода всех значений дискретных переменных необходимо набрать: $x = \quad i = \quad j =$.

Замечание 2.1.1. Дискретная переменная не является одномерным массивом и поэтому нельзя обратиться только к одному значению дискретной переменной.

Глобальная переменная. Как уже отмечалось, оператор присваивания, задающий значение некоторой простой или дискретной переменной, должен стоять в документе «левее или выше» «точки использования» переменной. Однако в MathCAD могут использоваться *глобальные переменные*, «доступные» в любой точке документа, вне зависимости в каком месте доку-

мента они были определены. Задание глобальной переменной имеет вид:

$$\langle \text{Имя_глобальной_переменной} \rangle \equiv \langle \text{Выражение} \rangle$$

Для ввода знака \equiv необходимо нажать клавишу [~].

Размерные переменные – числовые переменные и функции, обладающие размерностью и содержащие значения в системе измерения СИ физических величин. Сделано это для упрощения инженерных и физических расчетов, которые MathCAD выполняет с преобразованием единиц измерения физических величин. В MathCAD встроено большое количество единиц измерения, с помощью которых и создаются размерные переменные.

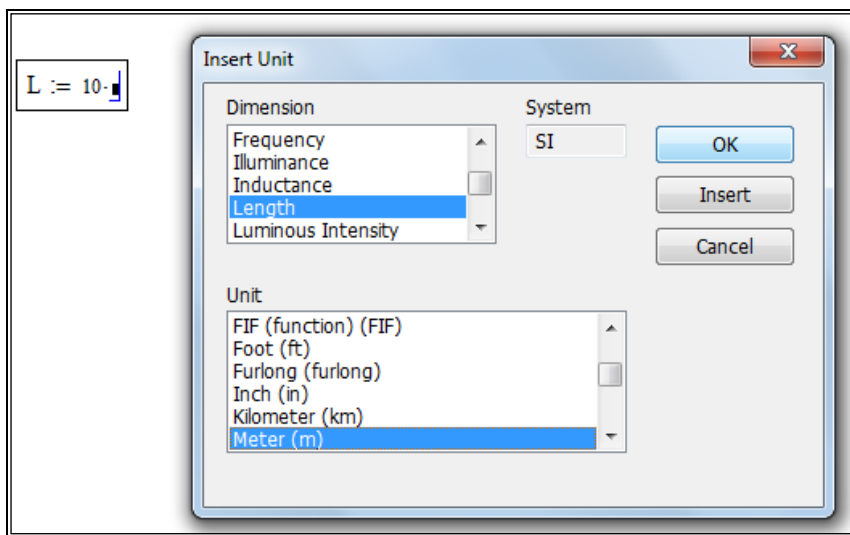


Рис. 2.1. Диалоговое окно задания единиц измерения

Диалоговое окно (рис. 2.1), в котором можно установить систему единиц измерения физических величин, открывается выбором в главном меню пункта *Insert (Вставка)* и выпол-

нением команды *Unit (Единицы_Измерения)* или щелчком по соответствующей кнопке с изображением мерного стакана на стандартной панели инструментов, либо с помощью комбинации клавиш <Ctrl>+<U>. Затем необходимо выбрать вид измерения (окно *Измерения (Dimension)*) и нужную единицу измерения (окно *Вставить Единицы (Unit)*).

Единицы измерения могут быть введены в математическое выражение и с клавиатуры. Чтобы создать размерную переменную, определяющую, например, длину в метрах, введите выражение, присваивающее переменной *L* значение 10: *L:=10*, и затем символ умножения <*>, а потом введите букву *m* либо с клавиатуры, либо из диалогового окна, изображенного на рис. 2.1.

Поскольку все символы, обозначающие единицы измерения, зарезервированы и имеют предустановленные значения (связанные с размерностью), то литера *m* будет распознана MathCAD как метр. Если ранее вы переопределили переменную *m*, присвоив ей какое-либо значение, то она уже не будет восприниматься как единица длины.

Системные переменные предназначены для хранения значения определенных параметров системы (например, переменная *ORIGIN* задает начальное значение индексных переменных у элементов массива).

Индексированные переменные соответствуют массивам MathCAD и они будут рассмотрены в теме 3.

2.2. Арифметические операции MathCAD

Арифметическими операциями в MathCAD являются: сложение, вычитание, умножение, деление и возведение в степень. Операции можно вводить либо с клавиатуры, либо с палитры инструментов *Calculator (Калькулятор)*. Обозначение некоторых операций на экране отличается от обозначения клавиши, нажимаемой для ввода этой операции (табл. 2.1).

Таблица 2.1

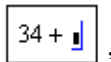
Операция	Клавиша	Последовательность набора	Вид на экране
Сложение	[+]	$x+2$	$x+2$
Вычитание	[-]	$x-y$	$x-y$
Умножение	[*]	$2*x$	$2 \cdot x$
Деление	[/]	$1/2$	$\frac{1}{2}$
Возведение в степень	[^]	x^2	x^2

Ввод математических выражений. Математические формулы и выражения в документе MathCAD вводятся в математическую область.

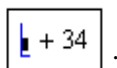
Математическая область устанавливается по умолчанию. Поэтому для запуска формульного редактора достаточно позиционировать указатель мыши в любом свободном месте окна и щелкнуть левой кнопкой. На экране появится маркер ввода в виде красного крестика. С этого места можно начинать набор формул. С началом ввода маркер превращается в выделяющий уголок синего цвета.

Выделяющий уголок имеет правую или левую ориентацию, которая указывает направление ввода математического выражения. Изменение ориентации уголка осуществляется нажатием клавиш [Insert]; перемещение уголка по математическому выражению осуществляется нажатием клавиш [←] и [→], а изменение размера сторон уголка – нажатием клавиши [пробел].

Например, если установлен правый уголок, то вводимая арифметическая операция (+) ставится справа от фрагмента:



если установлен левый уголок, то слева:



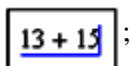
Фрагмент, охваченный уголком, является операндом для вводимой операции или функции.

Например, при вводе арифметического знака, действие которого относится к фрагменту математического выражения, необходимо с помощью клавиш [Insert] и [пробел] изменить размер уголка так, чтобы он охватывал данный фрагмент.

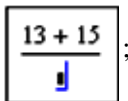
Покажем технику применения выделяющего уголка на примере ввода и вычисления арифметического выражения, соответствующего формуле:

$$\frac{13 + 15}{3 \cdot (27 - 8)} \cdot \frac{56 - 34}{19} = 0.569.$$

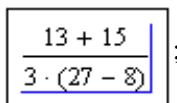
1) Набираем числитель первого сомножителя и, нажав клавишу [пробел], охватим его уголком:



2) к выделенному выражению применим арифметическую операцию **деления** (клавиша [/]):



3) набираем знаменатель и охватываем всю дробь правым уголком, так как предстоит ввести справа знак умножения:



- 4) вводим знак **умножения** (клавиша [*]):

$$\frac{13 + 15}{3 \cdot (27 - 8)} \cdot$$

- 5) аналогично пунктам 1–3 выполним набор значения второго сомножителя, затем все выражение охватим правым углом:

$$\frac{13 + 15}{3 \cdot (27 - 8)} \cdot \frac{56 - 34}{19}$$

- 6) введем оператор вывода – знак **равно** (клавиша [=]) и MathCAD произведет вычисление арифметического выражения:

$$\frac{13 + 15}{3 \cdot (27 - 8)} \cdot \frac{56 - 34}{19} = 0.569$$

Пример 2.2.1. Выполнить программирование и вычисление арифметических выражений, приведенных в документе, показанном на рис. 2.2.

Решение. Первоначально зададим значения простых переменных a , b , c , а затем запрограммируем нужные выражения и операторы.

$$\begin{aligned} a &:= 2.1 & b &:= 0.15 \cdot 10^{-3} & c &:= 3.5 + 4.78i \\ \frac{(|c|)^2 + a^2}{b} &= 2.634 \times 10^5 & y &:= \frac{e^{2a} + \frac{1}{a+b}}{\sqrt[3]{a+b}} \\ y &= 52.446 & z &:= \frac{a^{-y} + b^{-y}}{y} & z &= 6.728 \times 10^{198} \end{aligned}$$

Рис. 2.2. Программирование арифметических выражений

Пример 2.2.2. Используя оператор присваивания и размерные переменные, определить плотность стали, 2 см^3 которой имеют массу 15.7 г . Определить удельный вес стали.

Решение. Приведено в документе MathCAD на рис. 2.3. Заметим, что в последних версиях MathCAD в операторе присваивания для вывода вычисленного значения можно использовать оператор вывода (см. вывод вычисленного значения удельного веса – переменная γ).

<i>Дано</i>	$V := 2 \cdot \text{см}^3$	<i>объем</i>
	$M := 15.7 \cdot \text{г}$	<i>масса</i>
<i>Решение</i>	$\rho := \frac{M}{V}$	<i>плотность</i>
	$\gamma := \rho \cdot g = 7.698 \times 10^4 \frac{\text{кг}}{\text{м}^2 \cdot \text{с}^2}$	<i>удельный вес</i>
	$\rho = 7.85 \times 10^3 \frac{\text{кг}}{\text{м}^3}$	

Рис. 2.3. Использование размерных переменных

2.3. Встроенные функции и функции пользователя

Встроенные функции. Для вычисления значений наиболее распространенных математических функций в MathCAD определены так называемые *встроенные функции* (например, вычисление синуса, корня квадратного и т.д.). Для обращения к функции необходимо задать ее обозначение (ее имя или условный символ), а затем аргумент. Обозначение функции можно ввести с клавиатуры или с палитры инструментов (например, палитра *Калькулятор* содержит обозначения многих функций). Используя встроенные функции, можно решать сложные задачи (на-

пример, построение регрессии или решить дифференциальное уравнение в частных производных). Полный список функций и их синтаксис можно получить, щелкнув на кнопке f_x панели инструментов *Стандартная*.

Весь набор встроенных функций разбит на группы (категории) в соответствии с функциональным значением. Любую встроенную функцию можно вызвать, выбрав *Function* в строке меню *Insert (Вставка)*, чтобы открыть диалоговое окно *Insert Function (Вставка Функций)*, показанное на рис. 2.4.

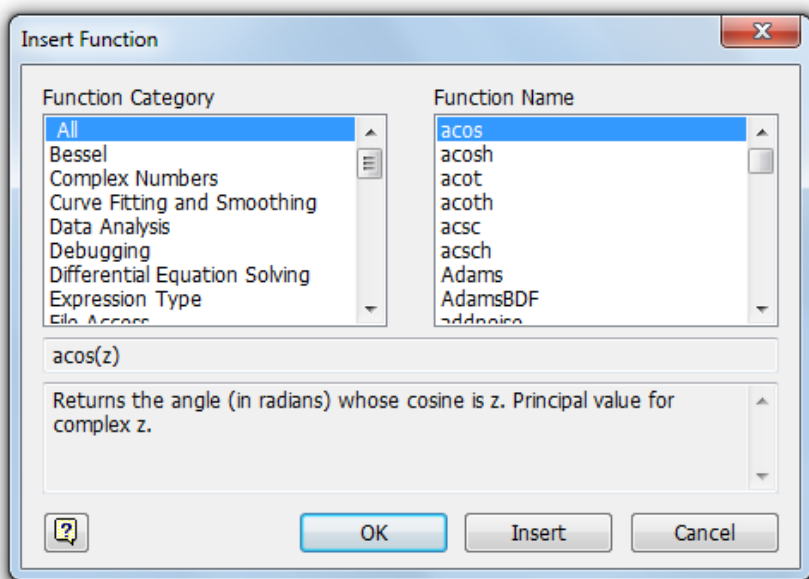



Рис. 2.4. Диалоговое окно вставки встроенных функций

В окне *Function Category (Категории Функций)* необходимо выбрать соответствующую категорию. При этом в окне *Function Name (Имя Функции)* будут представлены функции MathCAD, относящиеся к данной категории. Если требуемая функция в

выбранной категории не будет найдена, следует просмотреть категорию *All (Все)*, содержащую все встроенные функции. Если вы установили дополнительный Пакет расширения, то у вас есть дополнительные встроенные функции. Они также доступны из *Вставка* – функция меню под названием *Расширение Pack*.

Краткое описание выделенной функции показано в нижней части окна. Полное описание можно получить из Справки, которая вызывается по ссылке  в левом нижнем углу окна.

Кроме того, возможно вводить имена встроенных функций и непосредственно в документе MathCAD.

Функция пользователя. Несмотря на широкий набор встроенных функций, очень часто возникает необходимость дополнить вычисления новыми функциями, необходимыми для того или другого пользователя. Такие функции будем называть *функциями пользователя*. Использование функции пользователя предполагает два момента: *описание* функции и *обращение* к ней.

Описание размещается в документе перед обращением к функции и имеет вид:

$$\langle \text{Имя_функции} \rangle (\langle \text{Список_аргументов} \rangle) := \langle \text{Выражение} \rangle$$

Имя_функции – последовательность символов, удовлетворяющая тем же правилам, что и имя простой переменной.

Список аргументов – это перечень используемых в выражении переменных, записанных через запятую.

Выражение – это любое выражение (не только арифметическое), содержащее операции над аргументами. Переменные, входящие в список аргументов, по существу являются формальными параметрами и их значения *не задаются* до описания функции.

Пример 2.3.1. Составить описание функции пользователя, вычисляющее расстояние между точками *A, B* на плоскости.

Решение. Описание приведено на рис. 2.5. Напомним, что нижние индексы в именах простых переменных (например,

x_C, x_A и т.д.) вводятся после нажатия клавиши [.] – десятичная точка.

$Dist(x_A, y_A, x_B, y_B) := \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$	Описание функции
$x_C := 2 \quad y_C := 5 \quad Dist(x_C, y_C, 0.4, 5) = 1.6$	Обращения к функции
$d := Dist(2 \cdot x_C, \sqrt{y_C}, 0.4, 5) \quad d = 4.539$	

Рис. 2.5. Описание функции пользователя и обращение к ней

Для *обращения к функции* пользователя необходимо записать в каком-либо выражении имя функции и в круглых скобках указать конкретные значения аргументов функции.


В качестве аргумента могут использоваться не только константы и арифметические выражения, но и имена переменных, значения которых должны быть *определены до обращения к функции* (см. рис. 2.5).

2.4. Операторы математического анализа


В MathCAD имеется ряд операторов, позволяющих вычислить значения сумм, произведений, производных, определенного интеграла, пределов. Такие операторы часто называют *расширенными арифметическими операторами* или операторами математического анализа.

Обращение к этим операторам (значок оператора) можно ввести в документ MathCAD, используя палитру инструментов *Calculus (Матанализ)*. При вводе оператора задается шаблон для последующего заполнения числовыми или символьными значениями. Приведем операторы, часто используемые при решении научно-инженерных задач.


Оператор вычисления суммы. Для ввода оператора в доку-

мент необходимо щелкнуть на кнопке  и заполнить поля в появившемся шаблоне (рис. 2.6а). Назначение полей: 1 – имя переменной, являющейся параметром суммирования; 2 – нижний предел суммирования; 3 – верхний предел суммирования; 4 – выражение, зависящее от параметра суммирования.

Оператор вычисления произведения. Для ввода оператора

в документ необходимо щелкнуть на кнопке  и заполнить поля в появившемся шаблоне (рис. 2.6б). Назначение полей те же, что в операторе суммы.

Операторы вычисления производных. Значения производных вычисляются численным методом и поэтому необходимо задавать значения аргументов функции, при которых будут вычисляться производные.

Для ввода оператора дифференцирования в документ необходимо щелкнуть на кнопке  и заполнить поля в появившемся шаблоне (рис. 2.7а). Назначение полей: 1 – переменная дифференцирования; 2 – дифференцируемая функция или выражение.

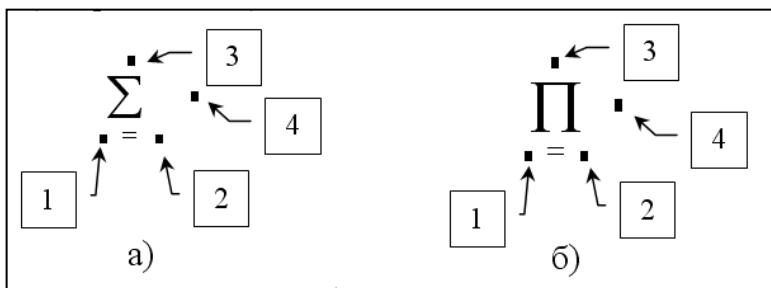


Рис. 2.6. Шаблоны операторов суммы и произведения

Для ввода оператора вычисления производной n -го ($n \leq 5$) порядка необходимо щелкнуть на кнопке $\frac{d^n}{dx^n}$ и заполнить поля в появившемся шаблоне (рис. 2.7б). Назначение полей: 1 – переменная дифференцирования; 2 – порядок производной; 3 – порядок производной (дублируется при заполнении поля 2); 4 – дифференцируемая функция или выражение.

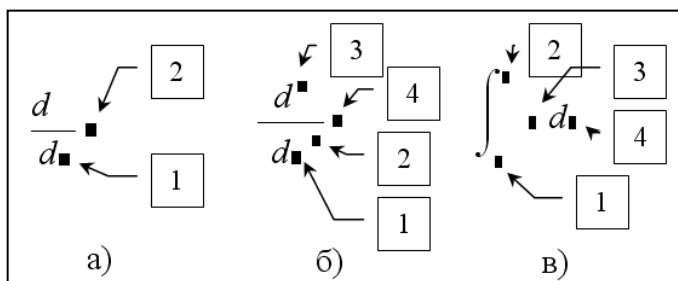


Рис. 2.7. Шаблоны операторов дифференцирования и интегрирования

Оператор вычисления определенных интегралов. Для ввода оператора в документ необходимо щелкнуть на кнопке \int_a^b и заполнить поля в появившемся шаблоне (рис. 2.7в). Назначение полей: 1, 2 – нижний и верхний пределы интегрирования; 3 – интегрируемое выражение; 4 – переменная интегрирования.

Пример 2.4.1. Используя рассмотренные операторы математического анализа, вычислите суммы, произведения, производные и интегралы, приведенные на рис. 2.8.

Решение. Необходимые конструкции используются в документе MathCAD, который показан на рис. 2.8. Обратите внимание, что ряд вычислений оформлены в виде функций пользователя. Это дает определенные удобства использования операторов математического анализа.

Символьное вычисление с использованием операторов математического анализа. Используя операторы математического анализа, результат можно получать не только в виде чисел, но и в символьном виде, т.е. в виде формул. Такие вычисления будем называть *символьными*.

Оператор вычисления сумм

$$\sum_{i=1}^{10} \sqrt{i} = 22.468 \quad S(n, m) := \sum_{i=1}^n \sum_{j=1}^m \frac{1}{i+j} \quad S(5, 7) = 5.896$$

Оператор вычисления произведений

$$Pr(a, m, n) := \prod_{i=m}^n (i+a) \quad Pr(2, 2, 10) = 7.983 \times 10^7$$

Операторы вычисления производных

$$x := 0.2 \quad \frac{d}{dx} (x^2 + \sin(x)) = 1.38$$

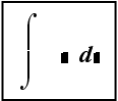

$$f(x, y) := \frac{x^2 + y^2}{\sqrt{|x+y|}} \quad df(x, y) := \frac{d^2}{dx^2} \frac{d^3}{dy^3} f(x, y) \quad df(3, -5) = 18.29.$$

Оператор вычисления определенных интегралов

$$I(a, b) := \int_{0.1}^a \int_0^b f(x, y) dy dx \quad I(1, 0) = 0 \quad I(1, 5) = 19.626$$

Рис. 2.8. Использование операторов математического анализа

Подробно символьные вычисления рассмотрены в теме 5. Здесь рассматриваются символьные вычисления с помощью оператора символьного вывода \rightarrow , который вводится с палитры инструментов *Символьный*. Этот оператор ставится после записи соответствующего оператора математического анализа.

Для вычисления первообразной функции используется оператор , который вводится нажатием кнопки  палитры инструментов *Calculus (Матанализ)*.

Пример 2.4.2. Используя рассмотренные операторы математического анализа, вычислите суммы, произведения, производные и интегралы, приведенные на рис. 2.9, в символьном виде.

Решение. Необходимые конструкции показаны в документе MathCAD, фрагмент который показан на рис. 2.9.

$$\sum_{i=1}^n i^2 \rightarrow \frac{1}{3} \cdot (n+1)^3 - \frac{1}{2} \cdot (n+1)^2 + \frac{1}{6} \cdot n + \frac{1}{6} \quad \prod_{n=1}^{\infty} \frac{1}{n^2+1} \rightarrow$$

$$\frac{d}{dx} \sin(x) \rightarrow \cos(x) \quad \int \ln(x) dx \rightarrow x \cdot \ln(x) - x$$

$$\int_a^{\infty} \frac{1}{x^3} dx \rightarrow \frac{1}{2 \cdot a^2} \quad \int_a^b \frac{1}{x^3} dx \rightarrow \frac{-1}{2 \cdot b^2} + \frac{1}{2 \cdot a^2}$$

Рис. 2.9. Использование операторов математического анализа

На рис. 2.10 приведен пример вычисления не только первообразной, но и проверка правильности полученной функции с использованием для этой цели, оператора дифференцирования.

$$\int \frac{-1 + \cos(x)}{(x - \sin(x))^2} dx \rightarrow \frac{1}{x - \sin(x)}$$

$$\frac{d}{dx} \frac{1}{x - \sin(x)} \rightarrow \frac{\cos(x) - 1}{(x - \sin(x))^2} \quad \text{Проверка}$$

Рис. 2.10. Проверка правильности вычисления первообразной функции

Вопросы и задания для самопроверки

1. Какие переменные можно использовать в алгоритмах, реализованных в пакете MathCAD?
2. В чем отличие записи имен переменных MathCAD и Pascal?
3. Что такое дискретная переменная и как выглядит задание дискретной переменной, принимающей значения: 0.3, 0.6, ..., 1.8?
4. Какие арифметические операции определены в MathCAD?
5. Как записывается и вводится оператор присваивания?
6. Используя встроенные функции палитры инструментов *Калькулятор*, введите и вычислите приведенные ниже выражения (сравните полученные результаты с приведенными):

$$\frac{23.1 \sqrt{3\pi - \frac{\ln(54)}{16}}}{5 - \pi^{4!}} = -5.661 \times 10^{-11}$$

$$\left(\frac{\cos\left(\frac{2\pi}{6}\right)\sin\left(\frac{\pi}{3}\right)}{\tan\left(\frac{2\pi}{9}\right) - \log(44)} \right)^3 = -0.156$$

$$e^{\sqrt{\frac{22-12.7}{44.8}}} - e^{\left[\left(\frac{1}{3}\right)^6 - 2\right]} = 1.575.$$

7. Используя оператор присваивания, задайте переменным x, y, z значения приведенных в пункте 6 арифметических выражений (т.е. первое выражение определяет переменную x , второе – переменную y и т.д.).

8. Как описывается функция пользователя и как выглядит обращение к функции пользователя?

9. Составьте описание функции, вычисляющей расстояние между двумя точками в трехмерном евклидовом пространстве.

10. Используя функцию пользователя предыдущего пункта, вычислите расстояние между точками $C(2.3, 4, 5.6)$ и $D(-4.0, 3.1, -0.0)$.

11. Составить функцию пользователя, вычисляющую факториал числа n

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1,$$

и протестировать эту функцию для различных значений n .

12. Используя операторы математического анализа, выполните следующие численные вычисления:

$$\text{a) } \int_0^2 \frac{x}{x^2-1} dx; \quad \text{b) } \int_1^4 \int_3^5 (2x^4 + 13y^5) dy dx; \quad \text{c) } \sum_{i=3}^{20} e^{-\frac{i}{2}}.$$

13. Используя операторы математического анализа, выполните следующие символьные вычисления:

$$\text{a) } \int_{-\infty}^0 e^x x^2 dx; \quad \text{b) } \sum_{i=0}^{\infty} e^{-\frac{i}{2}};$$

$$\text{c) } \int_3^5 2\pi(x^2 - 1) \sqrt{1 + \left[\frac{d}{dx}(x^2 - 1) \right]^2} dx.$$

14. Вычислите момент инерции стержня массой $M = 5 \text{ kg}$ и длиной $L = 2 \text{ m}$, вращающегося относительно оси, проходящей через его конец и перпендикулярной к нему:

$$I := \int_0^L x^2 \frac{M}{L} dx; \quad I = 13.333 m^2 \text{ kg}.$$

15. Вычислите момент инерции прямоугольной пластины массой $M = 5 \text{ kg}$, длиной $L = 2 \text{ m}$ и шириной $h = 10 \text{ cm}$, относительно оси, перпендикулярной к ней и проходящей через ее центр:


$$I := \frac{M(h^2 + L^2)}{12}; \quad I = 3.342 m^2 \text{ kg}.$$

ТЕМА 3. МАССИВЫ В ПАКЕТЕ MATHCAD

В этой теме будет рассмотрено создание массивов, основные операции над массивами и часто используемые функции обработки массивов.

3.1. Массивы в пакете MathCAD

При работе с массивами используются такие понятия, как *скаляр*, *вектор* и *матрица*. Одиночное число называется *скаляром* (в программировании – простая переменная). Общий термин для вектора или матрицы – *массив*. В *одномерном массиве* (векторе) положение элемента определяется одним индексом (например, x_i). В *двумерном массиве* (матрице), который имеет вид прямоугольной таблицы, положение элемента определяется двумя индексами (например, $A_{i,j}$). Первый индекс определяет номер строки, а второй – номер столбца, на пересечении которых находится данный элемент. Количество строк и столбцов матрицы задают размер матрицы. Вектор можно рассматривать как матрицу, содержащую один столбец.

Для обращения к элементу массива вводится имя массива, нажимается клавиша «левая квадратная скобка» (т.е. « [») или кнопка  в палитре инструментов *Matrix* (рис. 3.3), а затем вводятся нужные индексы (или индексные выражения). После ввода индексов нужно выйти из строки нижних индексов, нажав клавишу \rightarrow .

Пример 3.1.1. Необходимо сформировать вектор x , вторая проекция которого равна 10, и матрицу A с элементом $A_{2,3} = 30$.

Решение. Операторы присваивания, формирующие необходимые элементы, приведены на рис. 3.1. Здесь выведены сформированные массивы. Однако вместо ожидаемого вектора s

двумя проекциями и матрицы размером 2×3 получены массивы с размерностями большими на 1. Это объясняется нулевым начальным значением системной переменной *ORIGIN*. Ее нулевое значение также выведено на рис. 3.1.

$x_2 := 10$	$A_{2,3} := 30$	$ORIGIN = 0$
$x = \begin{pmatrix} 0 \\ 0 \\ 10 \end{pmatrix}$	$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 30 \end{pmatrix}$	

Рис. 3.1. Формирование массивов с $ORIGIN=0$

Системная переменная *ORIGIN*. Вектор и элементы матрицы обычно нумеруются, начиная с нулевой строки и нулевого столбца. Начальное значение нумерации хранится в системной переменной *ORIGIN*, которая по умолчанию равна нулю. Обычно в матричной алгебре значения индексов начинается с 1. Замену нулевого значения на другое можно выполнить несколькими способами. Рассмотрим следующие два способа.


Способ 1. Заменить значения *ORIGIN* во всем документе с помощью оператора глобального присвоения (\equiv). Для этого в любом месте рабочего документа надо ввести оператор присваивания $ORIGIN \equiv N$, где N – начальное значение нумерации массива. Например, если введем $ORIGIN \equiv 1$, то MathCAD будет вести отсчет номеров строк и столбцов от 1 по всему документу.


Способ 2. Переопределить *ORIGIN* можно с помощью оператора присвоения ($:=$). В этом случае новое значение будет действовать левее и ниже этого оператора присваивания до следующего переопределения переменной *ORIGIN*. Этот способ иллюстрируется рис. 3.2.


$$\begin{array}{l}
 \mathbf{ORIGIN} := 1 \quad \mathbf{x}_2 := 10 \quad \mathbf{A}_{2,3} := 30 \\
 \mathbf{x} = \begin{pmatrix} 0 \\ 10 \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 30 \end{pmatrix}
 \end{array}$$


Рис. 3.2. Формирование массивов с $\mathbf{ORIGIN}=1$

Для работы с векторами и матрицами система MathCAD используют палитру инструментов *Matrix (Матрицы)*, изображенную на рис. 3.3. Назначение инструментов палитры:

 – создание массива;


 – задание индекса у элемента массива;

 – вычисление обратной матрицы;


 – выделение столбца матрицы;

 – вычисление длины вектора или определителя матрицы;

 – транспонирование вектора или матрицы;

 – оператор векторизации, т.е. поэлементная обработка массива;

 – суммирование элементов вектора;

 – вычисление скалярного произведения;

 – вычисление векторного произведения.

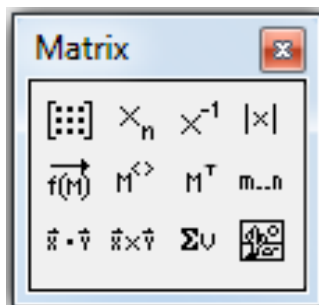


Рис. 3.3. Палитра инструментов *Matrix*

Замечание 3.1.1. В пакете MathCAD максимальное число индексов у элементов массива не превосходит 2. Однако каждый элемент массива может являться в свою очередь вектором или матрицей. Это позволяет создавать массивы практически любой размерности.

3.2. Создание массивов в MathCAD

Создание массива с использованием шаблона. Первоначально вводится имя создаваемого массива (которое аналогично имени простой переменной), знак присваивания и, используя инструмент $\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}$, вызывается окно *Insert Matrix* (*Вставить матрицу*), в котором указывается размер создаваемого массива. Напомним, что при создании вектора число столбцов (параметр *columns*) задается равным 1. После нажатия кнопки *Ok* в документе появляется шаблон массива, в поля которого вводятся соответствующие элементы массива.

Например, чтобы определить матрицу, необходимо выполнить следующее:

- введите имя матрицы и знак присваивания (двоеточие);
- щелкните по значку $\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}$ в палитре инструментов *Matrix*.

В появившемся диалоговом окне введите число строк и столбцов матрицы (рис. 3.4);

- щелкнуть на кнопке *Ok* и ввести в поля шаблона нужные значения элементов массива.

Переход от одного поля шаблона к другому осуществляется с помощью клавиши [Tab] или щелкнув мышью непосредственно на нужном поле. Значением элемента матрицы может быть число или выражение. При выводе элементов массива введенные выражения вычисляются.

На рис. 3.4 показано задание размера 3×3 матрицы A и заполнение шаблона соответствующего размера.

Создание массива вводом его элементов с клавиатуры.

Ввод элементов вектора осуществляется в цикле, параметром которого является дискретная переменная. Ввод элементов матрицы осуществляется в двойном цикле с соответствующими параметрами внешнего и внутреннего цикла. Ввод элементов осуществляется через запятую.

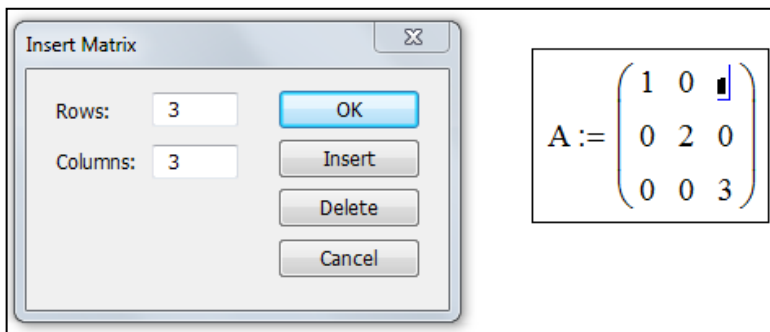


Рис. 3.4. Создание и заполнение шаблона матрицы

Пример 3.2.2. Сформировать следующие массивы:

$$z = \begin{vmatrix} 2 \\ 5 \\ 2 \\ 1 \end{vmatrix}, \quad B = \begin{vmatrix} 3 & 2 & 3 & 1 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 \end{vmatrix}.$$

Решение. Конструкции, осуществляющие формирование требуемого вектора и матрицы, показаны на рис. 3.5.

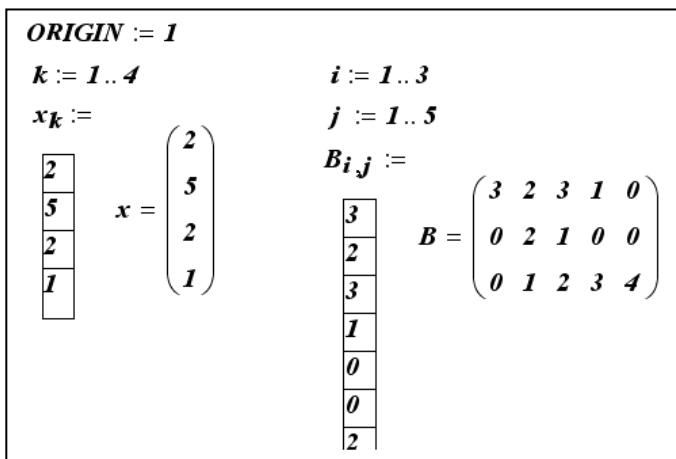


Рис. 3.5. Ввод элементов массивов с клавиатуры

Заметим, что ввод элементов матрицы осуществляется построчно, т.е. первоначально вводятся элементы первой строки, затем – второй и т.д. Это обусловлено тем, что переменная i – номер строки является параметром внешнего цикла. Заметим, что на рис. 3.5 показаны не все вводимые элементы массива B (вводятся 15 элементов).

3.3. Верхний индекс массива

В некоторых задачах обработки массивов возникает необходимость выделить определенный столбец. Для эффективного выполнения такого выделения массивы MathCAD могут иметь *верхний индекс*. Верхний индекс позволяет обращаться к отдельному столбцу массива.

Чтобы ввести верхний индекс, необходимо ввести имя массива, а затем нажать клавиши $[Ctrl + 6]$ или щелкнуть на кнопке



палитры инструментов *Matrix*. Затем в появившихся угловых скобках ввести номер выделяемого столбца (с учетом значения системной переменной *ORIGIN*).

Пример 3.3.1. Из второго и четвертого столбца матрицы *B* примера 3.2.2 сформировать векторы *c*, *d*.

Решение. Конструкции, формирующие эти векторы, приведены на рис. 3.6.

$$c := B^{\langle 2 \rangle} \quad d := B^{\langle 4 \rangle} \quad c = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \quad d = \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix}$$

Рис. 3.6. Использование верхнего индекса массива

3.4. Основные функции обработки массивов

Пакет MathCAD содержит большой набор встроенных функций, позволяющих возвращать различные характеристики массивов, создавать новые матрицы и преобразовывать существующие, обрабатывать массивы и облегчающих решение широкого круга задач линейной алгебры.

Вызов функций осуществляется выполнением пунктов меню **Insert/Functions** и в появившемся диалоговом окне выбирается нужная категория функций (для обработки массивов это *Vector and Matrix*), а затем из правого списка *Function Name* выбирается конкретная функция обработки массива (рис. 3.7). В нижней части окна выводится краткая информация о параметрах выбранной функции и ее назначении. Заметим, что обращение к функции можно ввести с клавиатуры.

Ниже приводится список встроенных функций, часто используемых для обработки массивов и разделенных на группы по назначению функций.

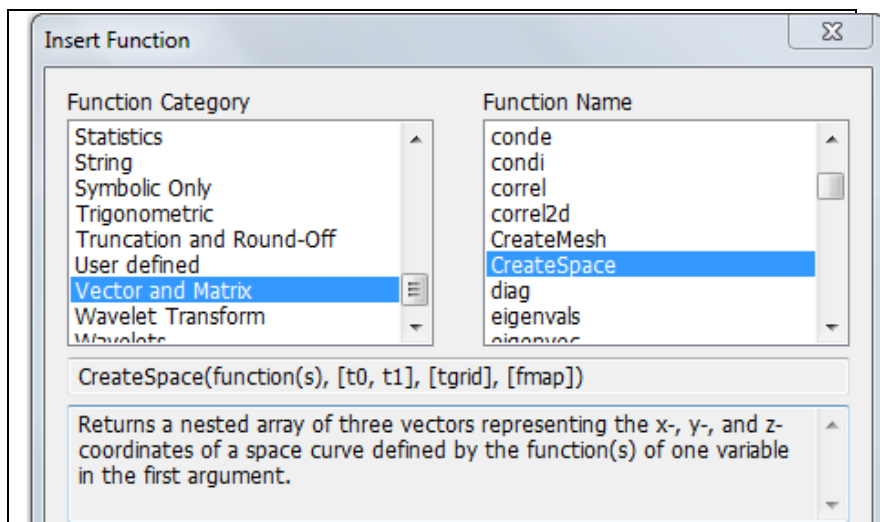


Рис. 3.7. Встроенные векторные и матричные функции

Функции определения характеристик массива. Эти функции вычисляют размеры массива и диапазон изменения его элементов. Имена и назначение функций приведены в табл. 3.1.

Таблица 3.1

<i>cols</i> (<i>M</i>)	вычисляет число столбцов матрицы <i>M</i>
<i>rows</i> (<i>M</i>)	вычисляет число строк матрицы <i>M</i>
<i>max</i>(<i>M</i>)	определяет значение максимального элемента в массиве <i>M</i>
<i>min</i>(<i>M</i>)	определяет значение минимального элемента в массиве <i>M</i>
<i>length</i>(<i>v</i>)	вычисляет число элементов в векторе <i>v</i>
<i>last</i>(<i>v</i>)	вычисляет индекс последнего элемента в векторе <i>v</i>

Пример 3.4.1. Использование функций из табл. 3.1 показано на рис. 3.8.

Матрица $M :=$	$\begin{pmatrix} 0 & 13 \\ 5 & 32 \\ 6 & -2 \end{pmatrix}$	$\text{rows}(M) = 3$	Возвращают число строк и столбцов
		$\text{cols}(M) = 2$	
		$\text{min}(M) = -2$	Возвращают наибольшее и наименьшее значение элементов
		$\text{max}(M) = 32$	
Вектор $v :=$	$\begin{pmatrix} 18 \\ 5 \\ 70 \end{pmatrix}$	$\text{length}(v) = 3$	Возвращают число элементов и индекс последнего элемента
		$\text{last}(v) = 2$	

Рис. 3.8. Вычисление характеристик массивов

Формирование матриц специального типа. Встроенные функции MathCAD, осуществляющие формирование матриц специального вида, приведены в табл. 3.2.

Таблица 3.2

<i>identity</i> (n)	Формирует единичную квадратную матрицу размером $n \times n$
<i>diag</i> (v)	Формирует диагональную матрицу, на главной диагонали которой стоят соответствующие проекции вектора v . Размер матрицы определяется размерностью вектора v
<i>matrix</i> (m, n, f)	Формирует матрицу, в которой (i, j) -й элемент равен значению функции $f(i, j)$
<i>Re</i> (M)	Формирует матрицу действительных частей матрицы M с комплексными элементами
<i>Im</i> (M)	Формирует матрицу мнимых частей матрицы M с комплексными элементами

$$\text{identity}(3) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \leftarrow \text{Единичная матрица} \quad \mathbf{M} := \begin{pmatrix} 2 & 4 & 6 \\ 4 & 5 & 6 \\ 2 & 7 & 12 \end{pmatrix}$$

Матрица B: Вещественная часть: Мнимая часть:

$$\mathbf{B} := \begin{bmatrix} 5 + 2i & 4 \\ 2.54 & 1 + 6i \\ 3 + (4 + 8i) & 3i \end{bmatrix} \quad \text{Re}(\mathbf{B}) = \begin{pmatrix} 5 & 4 \\ 2.54 & 1 \\ 7 & 0 \end{pmatrix} \quad \text{Im}(\mathbf{B}) = \begin{pmatrix} 2 & 0 \\ 0 & 6 \\ 8 & 3 \end{pmatrix}$$

Вектор b:

$$\mathbf{b} := \mathbf{M}^{\langle 1 \rangle} \quad \mathbf{b} = \begin{pmatrix} 4 \\ 5 \\ 7 \end{pmatrix} \quad \text{diag}(\mathbf{b}) = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 7 \end{pmatrix} \quad \leftarrow \text{Диагональная матрица, полученная из вектора b}$$

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

$$\mathbf{C} := \text{matrix}(3, 4, f) \quad \mathbf{C} = \begin{pmatrix} 0 & 1 & 4 & 9 \\ 1 & 4 & 9 & 16 \\ 4 & 9 & 16 & 25 \end{pmatrix} \quad \leftarrow \text{Матрица, где } C_{i,j} = f(i, j)$$

Заметим, что значение системной переменной *ORIGIN* в этом фрагменте равно 0. Докажите это, используя определение матрицы *M* и формирование вектора *b*.

57

Таблица 3.3

augment (A, B)	Формирует массив расположением A и B бок о бок, причем массивы A и B должны иметь одинаковое число строк
stack (A, B)	Формирует массив расположением A над B , причем массивы A и B должны иметь одинаковое число столбцов
submatrix (A, ir, jr, ic, jc)	Формирует подматрицу, содержащую строки с ir по jr и столбцы с ic по jc матрицы A

Пример 3.4.3. Используя функции табл. 3.3, сформировать новые матрицы из матриц, приведенных на рис. 3.10.

Решение. Первоначально сформируем исходные матрицы, показанные на рис. 3.10. Затем сформируем новые матрицы, приведенные на рис. 3.11.

$$M := \begin{pmatrix} 1 & 7 & 1 \\ 5 & 8 & 2 \\ 6 & 9 & 3 \end{pmatrix} \quad A := \begin{pmatrix} -1 & -2 \\ -3 & -7 \\ -4 & -9 \end{pmatrix} \quad B := \begin{pmatrix} 11 & 12 \\ 13 & 17 \\ 14 & 19 \end{pmatrix}$$

Рис. 3.10. Формирование исходных матриц

Специальные характеристики матрицы. К специальным характеристикам матрицы можно отнести функции, определяющие такие характеристики матриц, как *след*, *ранг*, *нормы* и другие величины, через элементы исходной матрицы. Некоторые функции MathCAD приведены в табл. 3.4.



Рис. 3.11. Формирование новых матриц

Таблица 3.4

<i>rank</i> (M)	Вычисляет ранг матрицы M , т.е. число линейно независимых строк или столбцов матрицы
<i>tr</i> (M)	Вычисляет след (сумму диагональных элементов) квадратной матрицы M
<i>mean</i> (M)	Вычисляет среднее значение элементов массива M
<i>norme</i> (M)	Вычисляет евклидову норму квадратной матрицы M , т.е. величину $\sqrt{\sum_{i=1}^n \sum_{j=1}^m M_{i,j}^2}$

Пример 3.4.4. Вычислить специальные характеристики матриц D, M , элементы которых приведены на рис. 3.12.

Решение. Вычисление характеристик показано на рис. 3.12.

$$D := \begin{pmatrix} 2 & 1 & 5 & 9 & -3 \\ 1 & 3 & -5 & 5 & 9 \\ 2 & 1 & 5 & 8 & 4 \end{pmatrix} \quad M := \begin{pmatrix} 3 & 0 & 1 \\ 6 & 0 & 2 \\ 1 & -4 & 8 \end{pmatrix}$$

$$\text{rank}(D) = 3 \quad \text{norme}(D^T \cdot D) = 262.414 \quad \text{mean}(D) = 3.133$$

$$\text{tr}(M) = 11 \quad \text{rank}(M) = 2 \quad \text{norme}(M) = 11.446$$

Рис. 3.12. Вычисление специальных характеристик матриц

3.5. Вычисления с массивами

Над векторами и матрицами можно производить вычисления в точности так же, как и над числами. Некоторые из операторов MathCAD имеют особые значения в применении к векторам и матрицам. Например, символ умножения означает просто умножение, когда применяется к двум числам, но он же означает скалярное произведение, когда применяется к векторам, и умножение матриц – когда применяется к матрицам.

В табл. 3.5. представлены векторные и матричные операторы MathCAD. Многие из этих операторов доступны из соответствующих палитр инструментов. Отметим, что операторы, которые имеют в качестве аргумента вектор, требуют *вектор-столбец*, а не *вектор-строку*.

Операторы, не перечисленные в этой таблице, не будут работать для векторов и матриц. При попытке использовать такой оператор с вектором или матрицей MathCAD будет отмечать это сообщением об ошибке «*неверная операция с массивом*», или «*нескалярная величина*». Используя оператор векторизации,

можно выполнить любую скалярную операцию или функцию поэлементно с вектором или матрицей.

Введем следующие обозначения: для матриц – A и M , для векторов – v и u и для скалярных величин – z .

Таблица 3.5

Операция	Обозначение	Клавиши	Описание
1	2	3	4
Умножение матрицы на скаляр	$A \cdot z$	*	Умножает каждый элемент A на скаляр z
Скалярное произведение	$u \cdot v$	*	Возвращает $\sum_{i=1}^n v_i \cdot u_i$. Векторы должны иметь одинаковое число элементов
Матричное умножение	$A \cdot B$	*	Возвращает произведение матриц A и B , число столбцов в A должно быть равно числу строк в B
Умножение матрицы на вектор	$A \cdot v$	*	Вычисляет произведение матрицы A и v , число столбцов в A должно быть равно числу строк в v
Деление	$\frac{A}{v}$	/	Делит каждый элемент массива на скаляр z
Сложение векторов и матриц	$A + M$	+	Массивы A и M должны иметь одинаковое число строк и столбцов

Продолжение табл. 3.5

1	2	3	4
Скалярная сумма	$A + z$	+	Добавляет z к каждому элементу A
Векторное и матричное вычитание	$A - M$	–	Вычитает соответствующие элементы массива M из элементов массива A , массивы A и M должны иметь одинаковые размеры
Скалярное вычитание	$A - z$	–	Вычитает z из каждого элемента A
Изменение знака	$-A$	–	Умножает все элементы A на -1
Степень матрицы	M^n	\wedge_n	Вычисляет n -ю степень квадратной матрицы M , где n – целое число
Обращение матрицы	M^{-1}	\wedge_{-1}	Вычисляет матрицу, обратную к M
Длина вектора	$ v $		Вычисляет величину $\sqrt{\sum_{i=1}^n v_i^2}$
Детерминант	$ M $		Возвращает детерминант квадратной матрицы M
Транспонирование массива	A^T	[Ctrl+1]	Меняет строки на столбцы, а столбцы на строки. Массив A может быть вектором или матрицей

Окончание табл. 3.5

1	2	3	5
Векторное произведение	$\mathbf{u} \times \mathbf{v}$	[Ctrl+8]	Вычисляет векторное произведение, где \mathbf{u} и \mathbf{v} векторы с тремя проекциями
Комплексное сопряжение	\overline{A}	"	Меняет знак мнимой части каждого элемента комплексной матрицы A
Суммирование элементов	$\sum \mathbf{v}$	[Ctrl+4]	Суммирует элементы вектора \mathbf{v} ; возвращает скаляр
Векторизация	\vec{A}	[Ctrl + -]	Выполнение заданной операции с каждым элементом массива A

Пример 3.5.1. Даны матрица M и вектор \mathbf{u} (рис. 3.13). Выполнить векторные и матричные операции над этими массивами.

Решение. На рис. 3.13 показан фрагмент документа MathCAD, в котором выполнены операции. Объясните:

- почему векторное произведение векторов \mathbf{z} и \mathbf{u} дало нулевой вектор;
- почему произведение матрицы M на обратную дало единичную матрицу, в которой на главной диагонали стоят 1, а все остальные элементы равны 0.

Пример 3.5.2. Дана система линейных алгебраических уравнений:

$$\begin{cases} y + 2 \cdot z = 13 \\ 3x + 2 \cdot z = -3 \\ 5 \cdot x + y + 3 \cdot z = 50. \end{cases}$$

Необходимо найти решение этой системы и выполнить проверку найденного решения.

Решение. Первоначально сформируем матрицу \mathbf{K} из коэффициентов системы и вектор \mathbf{q} из значений правой части системы (рис. 3.14). Получаем систему уравнений $\mathbf{Kd} = \mathbf{q}$, где вектор \mathbf{d} имеет проекции x, y, z , т.е. является искомым решением системы. Используя обратную матрицу, вычисляем решение системы и делаем проверку, а именно: произведение \mathbf{Kd} равно вектору \mathbf{q} (получено тождество левой и правой части исходной системы уравнений), а разность $\mathbf{Kd} - \mathbf{q}$ есть нулевой вектор, так как величины порядка 10^{-14} в нашем примере можно считать нулевыми (см. рис. 3.14).

$$\begin{array}{l}
 \mathbf{M} := \begin{pmatrix} 2 & 4 & 2 \\ 1 & 5 & -3 \\ 6 & 3 & 1 \end{pmatrix} \quad \mathbf{y} := \begin{pmatrix} 2 \\ 0 \\ 3 \end{pmatrix} \quad \text{исходные массивы} \\
 2 \cdot \mathbf{M} = \begin{pmatrix} 4 & 8 & 4 \\ 2 & 10 & -6 \\ 12 & 6 & 2 \end{pmatrix} \quad \mathbf{M} - 2 = \begin{pmatrix} 0 & 2 & 0 \\ -1 & 3 & -5 \\ 4 & 1 & -1 \end{pmatrix} \quad \mathbf{z} := 3 \cdot \mathbf{y} = \begin{pmatrix} 6 \\ 0 \\ 9 \end{pmatrix} \\
 \mathbf{M} \cdot \mathbf{y} = \begin{pmatrix} 10 \\ -7 \\ 15 \end{pmatrix} \quad \mathbf{M}^2 = \begin{pmatrix} 20 & 34 & -6 \\ -11 & 20 & -16 \\ 21 & 42 & 4 \end{pmatrix} \quad \mathbf{M} \cdot \mathbf{M}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
 \mathbf{z} - 3 = \begin{pmatrix} 3 \\ -3 \\ 6 \end{pmatrix} \quad |\mathbf{M}| = -102 \quad \mathbf{z} \cdot \mathbf{y} = 39 \quad \mathbf{z} \times \mathbf{y} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}
 \end{array}$$

Рис. 3.13. Операции над массивами

$$\begin{aligned}
 K &:= \begin{pmatrix} 0 & 1 & 2 \\ 3 & 0 & 2 \\ 5 & 1 & 3 \end{pmatrix} & q &:= \begin{pmatrix} 13 \\ -3 \\ 50 \end{pmatrix} & d &:= K^{-1} \cdot q = \begin{pmatrix} 11 \\ 49 \\ -18 \end{pmatrix} \\
 K \cdot d &= \begin{pmatrix} 13 \\ -3 \\ 50 \end{pmatrix} & K \cdot d - q &= \begin{pmatrix} 0 \\ 7.105 \times 10^{-15} \\ 1.421 \times 10^{-14} \end{pmatrix} & & \text{проверка} \\
 & & & & \text{решения}
 \end{aligned}$$

Рис. 3.14. Решение системы алгебраических уравнений

3.6. Функции сортировки элементов векторов и матриц

В системе MathCAD используются функции сортировки – перестановки элементов векторов и матриц, которые приведены табл. 3.6.

Таблица 3.6

<i>sort</i> (v)	Сортировка элементов векторов в порядке возрастания их значений
<i>reverse</i> (v)	Перестановка элементов в обратном порядке
<i>csort</i> (M,n)	Перестановка строк матрицы <i>M</i> таким образом, чтобы отсортированным оказался <i>n</i> -й столбец
<i>rsort</i> (M,n)	Перестановка строк матрицы <i>M</i> таким образом, чтобы отсортированной оказалась <i>n</i> -я строка

Пример 3.6.1. Используя функции табл. 3.6, выполнить различную сортировку вектора y и матрицы M (см. рис. 3.15).

$$\begin{array}{l}
 \text{ORIGIN} = 0 \qquad y := \begin{pmatrix} 5 \\ 1 \\ 3 \end{pmatrix} \quad \text{sort}(y) = \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix} \quad \text{reverse}(y) = \begin{pmatrix} 3 \\ 1 \\ 5 \end{pmatrix} \\
 M := \begin{pmatrix} 4 & 15 & 8 \\ 3 & 6 & 2 \\ 7 & 12 & 9 \end{pmatrix} \quad \text{csort}(M, 0) = \begin{pmatrix} \boxed{3} & 6 & 2 \\ 4 & 15 & 8 \\ 7 & 12 & 9 \end{pmatrix} \quad \text{rsort}(M, 2) = \begin{pmatrix} 4 & 8 & 15 \\ 3 & 2 & 6 \\ \boxed{7} & \boxed{9} & \boxed{12} \end{pmatrix}
 \end{array}$$

Рис. 3.15. Сортировка элементов массивов

Решение. На рис. 3.15 приведен документ MathCAD, где показаны результаты работы различных функций сортировки элементов вектора и матрицы. Столбец и строка матрицы, являющиеся результатами сортировки, выделены рамкой.

Вопросы и задания для самопроверки

1. Как выглядит обращение к элементу массива MathCAD?
2. Что обозначают нижние индексы?
3. К какому типу переменных относятся нижние индексы?
4. Какой размерности массив представляет переменная с одним нижним индексом? С двумя нижними индексами?
5. Что обозначает верхний индекс в матричных операциях?
6. Найдите определитель и обратную матрицу для матриц:

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 11 \\ 7 & 8 & 9 \end{pmatrix} \quad B := \begin{pmatrix} 11 & -11 & 3 \\ 2 & -2 & 7 \\ 9 & 3 & -3 \end{pmatrix}.$$

7. Используя операции со столбцами, получите матрицу C перестановкой 2-го и 3-го столбцов матрицы A .

8. Создайте две матрицы Q_1, Q_2 размером 3×4 . Ранг первой матрицы должен быть равен 3, а второй – 2. Для вычисления ранга используйте соответствующую функцию MathCAD.

9. Вычислите матрицу $2*A*B - 3*C*D$, где:

$$A := \begin{pmatrix} 2 & 3 & -1 & -2 & 4 \\ 0 & 3 & -2 & -1 & 3 \\ 5 & 1 & 1 & -4 & 1 \end{pmatrix} \quad B := \begin{pmatrix} 0 & -1 \\ 1 & -2 \\ -1 & 0 \\ 1 & 2 \\ -2 & 3 \end{pmatrix}$$
$$C := \begin{pmatrix} 1 & -2 & 1 & 4 \\ -3 & 1 & 0 & 5 \\ 2 & 0 & 3 & -1 \end{pmatrix} \quad D := \begin{pmatrix} 1 & -1 \\ -2 & 0 \\ 3 & -1 \\ 2 & -1 \end{pmatrix}.$$

10. Решите систему линейных уравнений и проверьте найденное решение.

$$\begin{cases} 1 \cdot x_1 - 4 \cdot x_2 + 3 \cdot x_3 - 7 \cdot x_4 = 3 \\ -8 \cdot x_1 - 2 \cdot x_2 - 9 \cdot x_3 - 3 \cdot x_4 = -2 \\ 3 \cdot x_1 - 6 \cdot x_2 + 5 \cdot x_3 - 3 \cdot x_4 = 5 \\ 4 \cdot x_1 - 8 \cdot x_2 - 3 \cdot x_3 - 4 \cdot x_4 = -3 \end{cases}.$$

11. Выясните, являются ли линейно-независимыми векторы p, q, r :

$$p = \begin{pmatrix} -4 \\ -5 \\ -13 \end{pmatrix} \quad q = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} \quad r = \begin{pmatrix} 2 \\ 3 \\ 6 \end{pmatrix}.$$

Напомним, что векторы линейно независимы, если смешанное произведение $\overline{(p \times q)} \cdot r$ равно нулю.

12. Из первых двух строк матрицы D (см. задание 9) сформировать матрицу R_1 , а из двух последних строк – матрицу R_2 . Выяснить, являются ли эти две матрицы перестановочными, т.е. выполняется ли матричное равенство $R_1 \cdot R_2 = R_2 \cdot R_1$. Для проверки этого тождества вычислить норму матрицы $R_1 \cdot R_2 - R_2 \cdot R_1$. Если норма матрицы имеет значимую величину (например, $\geq 10^{-10}$), то матрицы R_1, R_2 не являются перестановочными.

ТЕМА 4. ГРАФИКА В ПАКЕТЕ MATHCAD

В этой теме рассматривается представление данных в виде двумерных графиков в декартовых и полярных координатах, трехмерных графиков поверхностей, линий уровня поверхностей, двух- и трехмерных гистограмм, изображения векторных полей, пространственных кривых.

4.1. Основные инструменты для построения графиков

По количеству аргументов у отображаемой функции графики можно разбить на две группы: графики функций одной переменной (или двумерные графики 2D) и графики функций двух переменных (или трехмерные графики 3D).

Задание типа графика осуществляется с использованием кнопок палитры *Graph (График)* (см. рис. 1.1):



X-Y Plot (*X-Y Зависимость*) (или клавиша [*@*]). Служит для построения графика функции $y = f(x)$ или для отображения точек на плоскости (x_i, y_i) .



Polar Plot (*Полярные координаты*) (или клавиши [*Ctrl+7*]). Служит для построения графика функции $\rho(\varphi)$, заданной в полярных координатах, где полярный радиус ρ зависит от полярного угла φ .



Surface Plot (*Поверхности*) (или клавиши [*Ctrl+2*]). Служит для представления функции $z = f(x, y)$ в виде поверхности в трехмерном пространстве, а также для отображения значений элементов матрицы.



Contour Plot (*Контурный график*). Строит диаграмму линий уровня функции вида $z = f(x, y)$, т.е. отображает точки, в которых данная функция принимает фиксированное значение $z = \text{const}$.



3D Scatter Plot (3D Точечный). Служит для точечного представления элементов матрицы или отображения значений функции $z = f(x, y)$ в заданных точках (x_i, y_j) . Эта команда может также использоваться для построения пространственных кривых.



3D Bar Plot (3D Диаграммы). Служит для представления значений матрицы или отображения значений функции $z = f(x, y)$ в виде трехмерной столбчатой диаграммы.




Vector Field Plot (Поле векторов). Служит для представления двумерных векторных полей. При помощи этой команды можно построить поле градиента функции $f(x, y)$.

Задание типа графика можно также осуществить к пункту меню **Insert (Вставить)**, команда **Graph (График)**, а затем в выпадающем меню выбрать нужный тип графика. Однако в дальнейшем будем пользоваться кнопками палитры инструментов **Graph (График)**.

4.2. Построение графиков функции одной переменной в декартовой системе координат

Построение графика одной переменной включает следующие этапы:

1. Задайте функцию или выражение, значение которых будут представлены графиком (например, $f(x) := \sin(x)^3$). Используя дискретную переменную, задайте значения аргумента в нужном диапазоне (например, $x := 0, 0.1 \dots 10$). Чем меньше шаг дискретной переменной, тем более плавной будет кривая графика.

2. Введите шаблон графика с щелкнув по кнопке  палитры инструментов **Graph (График)**. На экране появляется шаблон с шестью полями (рис. 4.1): в поле 1 вводятся аргументы; в поле 2 – обращения к функциям или выражения, зависящие от указанных аргументов. Поля 3, 4 содержат границы ин-

тервала значений по оси абсцисс, поля 5, 6 – границы по оси ординат.

3. После того как заполнены поля 1, 2, нужно щелкнуть мышью в любом месте документа вне графика. Поля 3–6 будут использованы позже. Построенный график будет иметь вид, показанный на рис. 4.2.

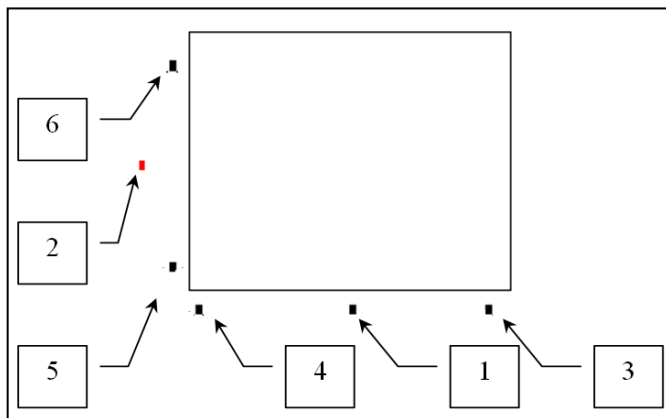


Рис. 4.1. Поля шаблона графика функции одной переменной

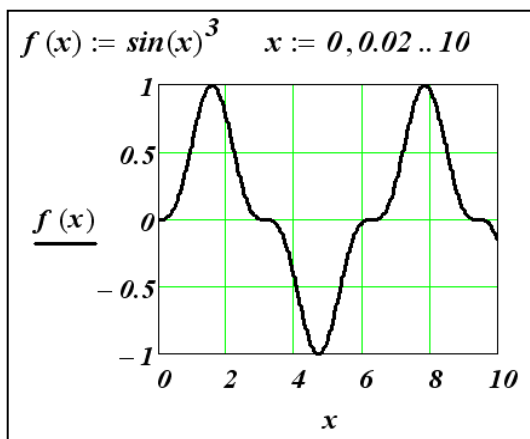


Рис. 4.2. Построение графика функции одной переменной

Изменение диапазонов по оси X и Y позволяет изменять масштаб, чтобы представить график в удобном для пользователя виде. Для этого достаточно щелкнуть левой кнопкой мыши на построенном графике и изменить значения полей 3–6, задав нужные значения границ диапазонов.

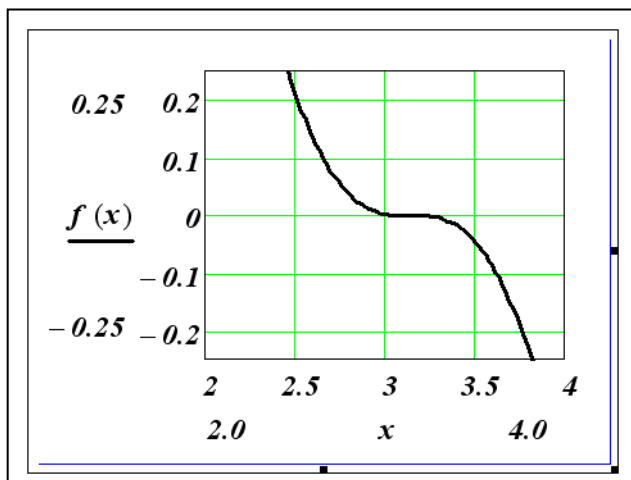


Рис. 4.3. Масштабирование графика

Например, для просмотра поведения функции в области, определяемой прямоугольником со сторонами: по оси X $[2.0, 4.0]$, а по оси Y $[-0.25, 0.25]$. Построенный график показан на рис. 4.3. Обратите внимание на значения полей 3–6, в которых задаются границы соответствующих диапазонов.

Форматирование графиков. Другие свойства графика могут быть установлены в диалоговом окне форматирования, которое можно вызвать командой *Format* контекстного меню или сделать двойной щелчок левой кнопкой на построенном графике. Окно форматирования показано на рис. 4.4.

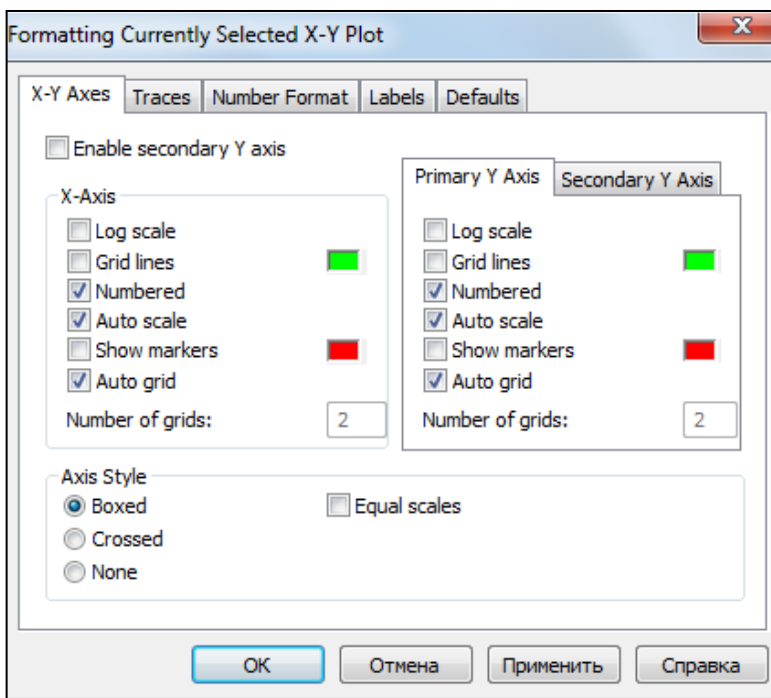


Рис. 4.4. Окно форматирования графика

Как видно на рисунке, окно форматирования имеет следующие вкладки:

- ***X-Y Axis (Оси X-Y)*** – задание параметров форматирования осей;
- ***Traces (Трассировка)*** – задание параметров форматирования линий графика;
- ***Labels (Метки)*** – задание параметров форматирования меток осей (установление надписей);
- ***Defaults (По умолчанию)*** – назначение установленных параметров форматирования параметрами по умолчанию.

На вкладке **X-Y Axis** содержатся следующие основные параметры, относящиеся к осям **X** и **Y** (*Axis X* и *Axis Y*):

- *Log scale* (*Логарифмический масштаб*) – задание логарифмического масштаба оси;
- *Grid Lines* (*Линии сетки*) – проведение линий масштабной сетки;
- *Нумерация* (*Numbered*) – оцифровка оси;
- *Auto scale* (*Автомасштаб*) – автоматическое масштабирование графика;
- *Show markers* (*Показ маркеров*) – установка делений по осям;
- *Auto grid* (*Автосетка*) – автоматическая установка масштабных линий;
- *Number of grids* (*Количество узлов*) – установка заданного числа масштабных линий.

Группа Axis Style (Стиль осей) позволяет задать стиль отображения координатных осей:

- *Boxed* – оси в виде прямоугольника;
- *Crossed* – оси в виде креста;
- *None* – отсутствие осей;
- *Equal scale* – установка одинакового масштаба по осям графика.

Вкладка Traces служит для управления отображением линий, из которых строится график. На этой вкладке представлены следующие параметры:

- *Legend Label* – выбор типа линии в легенде;
- *Symbol Frequency* – задает частоту отрисовки символа, из которого строится график. Например, если задать 2, то будет отрисовываться каждая вторая точка графика.
- *Symbol* – выбор символа, который помещается на линию, для отметки базовых точек графика;
- *Symbol Weight* – установка толщины символа;
- *Line* – установка типа линии;
- *Line Weight* – установка толщины линии;
- *Color* – установка цвета линии и базовых точек;

- *Type* – установка типа графика.

Узловые точки (точки, для которых вычисляются координаты) графиков часто требуется выделить какой-нибудь фигурой. Список столбца *Symbol* позволяет выбрать отметки для базовых точек графика каждой из функций такие как: пусто – без отметки; $x's$ – наклонный крестик; $+x$ – прямой крестик; квадрат – квадрат и другие.

Список в столбце *Line* позволяет выбрать типы линий: непрерывная, пунктирная, штрихпунктирная и др.

Раскрывающийся список столбца *Type* позволяет выбрать следующие типы линий графика:

- *line* (линия) – построение линиями;
- *points* (точки) – построение точками;
- *bar* (столбец) – построение в виде столбцов гистограммы;
- *step* (ступенька) – построение ступенчатой линией;
- *stem* (стебель) – построение вертикальными черточками с кружком на месте значения;
- *solidbar* (заполненные столбцы) – построение в виде закрашенных столбцов гистограммы.

Вкладка *Labels* позволяет вводить в график дополнительные надписи. Для установки надписей служат поля ввода:

- *Title* (Заголовок) – установка титульной надписи к рисунку;
- *X-Axis* (Ось X) – установка надписи по оси X;
- *Y-Axis* (Ось Y) – установка надписи по оси Y.

В группе *Заголовок* имеются переключатели *above* и *снизу* для установки титульной надписи либо над графиком, либо под ним.

Вкладка *Default* позволяет назначить установленные на других вкладках параметры форматирования параметрами по умолчанию. Для этого служит флажок установки «использовать по умолчанию». Щелкнув на кнопке «вернуть значения по умолчанию» можно вернуть стандартные параметры графика.

Построение нескольких графиков на одном рисунке.

В MathCAD можно на одном рисунке отобразить несколько функций. Для этого необходимо выполнить следующие действия:

- в поле 2 шаблона (см. рис. 4.1) ввести имена функций через запятую. При этом первое выражение уходит вверх, а под ним появляется место ввода для следующей функции;

- в поле 1 шаблона ввести имена независимых переменных функций через запятую в той же последовательности, как введены функции; если у всех функций одинаковая переменная, то достаточно ввести ее один раз.

Пример 4.2.1. Построим на одном рисунке графики двух функций $\sin(x)$ и $\cos(x) \cdot e^{-0.2x}$ для $x \in [0, 10]$.

Решение. Построение содержит следующие шаги:

1. Задание дискретной переменной x и определение функции пользователя $g(x) = \cos(x) \cdot e^{-0.2x}$ (рис. 4.5).

2. Вызов шаблон графика, ввод в поле 1 переменной x , ввод в поле 2 через запятую имен отображаемых функций.

3. Щелкнуть указателем мыши за пределами графика.

4. Используя вкладку окна форматирования *Traces*, изменить тип, цвет и толщину кривых так, как это показано на рис. 4.5.

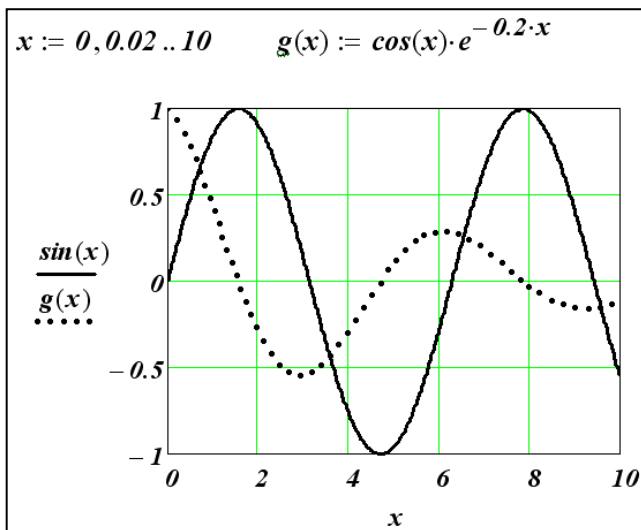


Рис. 4.5. Построение на одном рисунке двух кривых
Графическое представление элементов одномерных массивов (проекций вектора). Для этого в поле 1 вводится переменная, являющаяся индексом у элемента одномерного массива, а в поле 2 стоит обращение к элементу соответствующего массива.

Пример 4.2.2. Отобразить на одном рисунке элементы двух векторов с проекциями:

$$x_i = -4\pi + \frac{\pi}{10} \cdot i, \quad y_i = x_i \cdot \sin(x_i), \quad i = 0, 1, \dots, 80.$$

Решение. Выполнив те же шаги, что и в примере 4.2.1, получаем графики, приведенные на рис. 4.6. Здесь в поле 1 вводится переменная i – индексная переменная, а поле 2 – обращение к элементам массивов x, y .

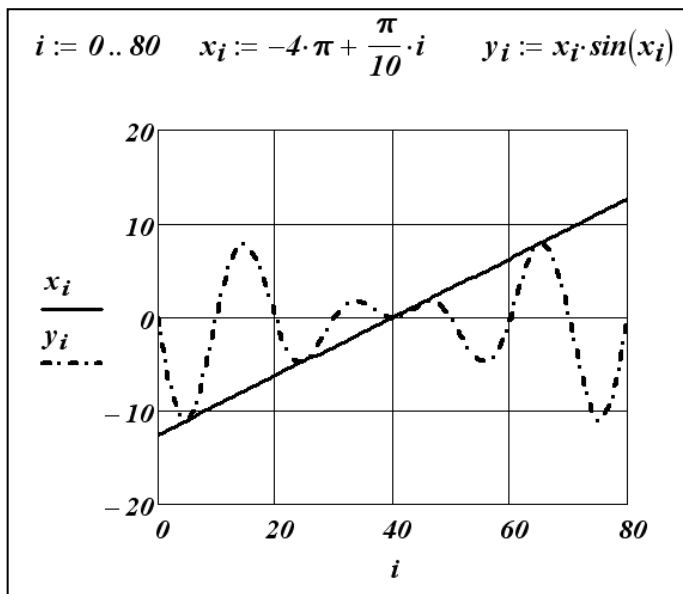


Рис. 4.6. Отображение элементов одномерных массивов

Построение графиков параметрических функций. Функция $y(x)$ называется параметрической, если x, y являются функциями некоторого параметра, например, параметра t , т.е. $x = x(t)$, $y = y(t)$. В этом случае значения параметра задаются дискретной переменной, в поле 1 вводится зависимость $x(t)$, в поле 2 – зависимость $y(t)$.

Пример 4.2.3. Построить график параметрической функции, заданной следующими уравнениями:

$$x(t) = 2\sin(t), \quad y(t) = \cos(t) \cdot e^{-0.05t}, \quad t \in [0, 15.0].$$

Решение. Построение графика показано на рис. 4.7. Здесь шаг изменения дискретной переменной равен 0.02, что обеспечивает «гладкость» кривой графика.

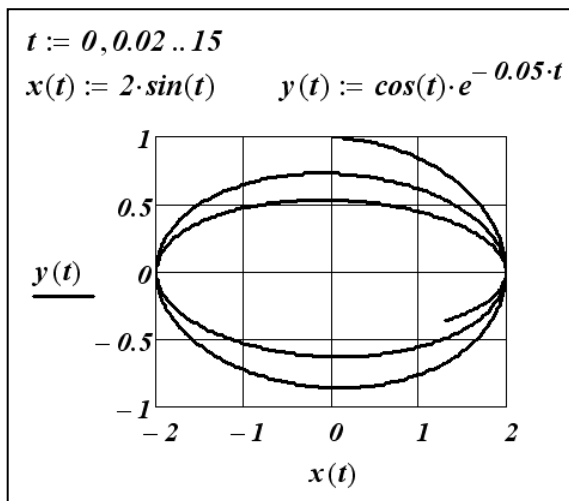



Рис. 4.7. Построение графика параметрической функции

4.3. Построение графиков функции одной переменной в полярной системе координат

В полярной системе координат положение точки на плоскости определяется двумя координатами: углом и радиусом, например $\rho(\varphi)$, где φ – угол, ρ – радиус точки. Шаблон графика в полярной системе координат вызывается кнопкой  (*Polar Plot*) палитры инструментов **Graph** (показан на рис. 4.8).

В поле 1 вводится полярный угол φ , в поле 2 – зависимость радиуса от угла, в поле 3 – максимальный радиус для масштабирования графика по радиусу, поле 4 – минимальный радиус для масштабирования графика по радиусу. Заметим, что в поле 1 можно ввести через запятую несколько зависимостей радиуса от угла, которые будут отображены соответствующими кривыми. Это иллюстрирует следующий пример.

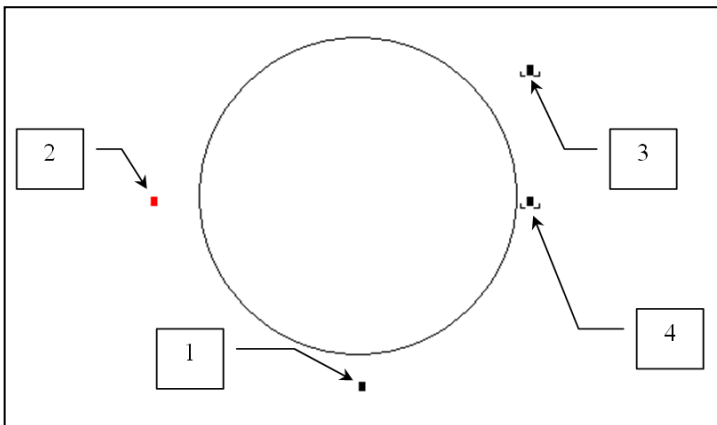


Рис. 4.8. Шаблон графика в полярной системе координат

Пример 4.3.1. Построить графики в полярной системе координат следующих функций:

$$\rho(\varphi) = 1 + \sin\left(2\varphi + 3\frac{\pi}{2}\right), \quad r(\varphi) = 1 + \frac{\sin(3\varphi + \pi)}{2}$$

для $\varphi \in [0, 2\pi]$.

Решение. На рис. 4.9 приведен фрагмент документа, в котором находятся необходимые конструкции MathCAD. Заметим, что в поле 2 кроме двух функций $\rho(\varphi)$, $r(\varphi)$ введена константа 2, которая отображена на рисунке окружностью, радиуса 2. В поле 3 был введен максимальный радиус 2.5.

Форматирование графиков в полярной системе координат. Для этого необходимо выделить график (щелчком левой кнопки мыши) и выполнить команду *Format* контекстного меню или выполнить двойной щелчок на графике. При этом откроется окно *Formatting Currently Selected Polar Plot* (Форматирование полярного графика). Это окно содержит вкладки, которые аналогичны вкладкам окна форматирования графиков в декартовой системе координат.

Вкладка **Polar Axis** содержит следующие элементы:

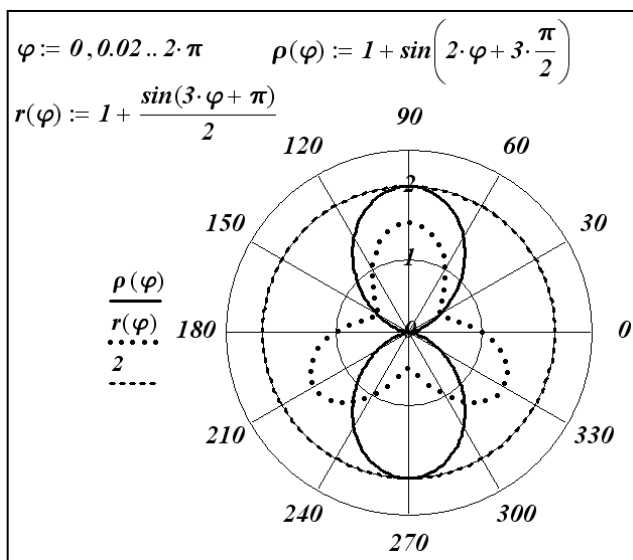


Рис. 4.9. Графики функций в полярных координатах

- *Log Scale* (Логарифмическая шкала) – создает логарифмическую шкалу по радиусу.
- *Grid Lines* (Линии сетки) – создает сетку линий, соответствующих уравнениям $\rho = \text{const}$ и $\varphi = \text{const}$.
- *Numbered* (Нумерованная) – линии $\rho = \text{const}$ и $\varphi = \text{const}$ снабжаются соответствующими значениями констант. Напомним, что, используя поля 3, 4 (см. рис. 4.8), можно задать максимальный и минимальный радиусы для масштабирования графика.
- *Auto Grid* (Автосетка) – при включении этой опции число линий сетки определяет автоматически.
- *Show Markers* (Показать метки) – при помощи этой опции можно снабдить график двумя дополнительными пунктирными

ными окружностями $\rho = \text{const}$. Для этого надо ввести нужные значения радиусов в появившиеся поля.

Остальные значения опций этой вкладки и других вкладок те же, что и в окне форматирования графиков в декартовой системе координат.

4.4. Построение графиков функций двух переменных

Для создания графика функции двух переменных (3D-графика) необходимо щелкнуть на одной из следующих кнопок



палитры *Graph (Графика)* с изображением требуемого типа графика. В документе появится шаблон графика с тремя осями и пустым полем. В это поле вводится либо имя массива, либо имя функции двух переменных. **В первом случае** предварительно необходимо сформировать матрицу из значений функции в узлах прямоугольной сетке. Этот способ позволяет отобразить элементы матрицы, содержащей результаты вычислений, например, значения температурного поля в разных точках плоскости. **Во втором случае** предварительно надо описать функцию от двух переменных.

Пример 4.4.1. Отобразить в виде поверхности значения функции $f(x,y) = \sin(x^2 + y^2)$ для $x \in [-1.5, 1.5]$, $y \in [-1.5, 1.5]$, предварительно сформировав матрицу из значений функции в узлах сетки $[-1.5, 1.5] \times [-1.5, 1.5]$.

Решение. Необходимые конструкции показаны в документе, приведенном на рис. 4.10. Заметим, что в одномерных массивах x, y находятся узлы по осям x, y , в которых вычисляются значения элементов матрицы M .

Форматирование графиков. Для вызова окна форматирования необходимо щелкнуть по графику правой кнопкой мыши и в контекстном меню выбрать *Properties (Свойства)*. Окно форматирования (рис. 4.11) имеет следующие вкладки:

- *General (Общие)* – установка общих параметров форматирования;
- *Axis (Оси)* – установка параметров форматирования координатных осей;
- *Appearance (Вид)* – установка вида графика;

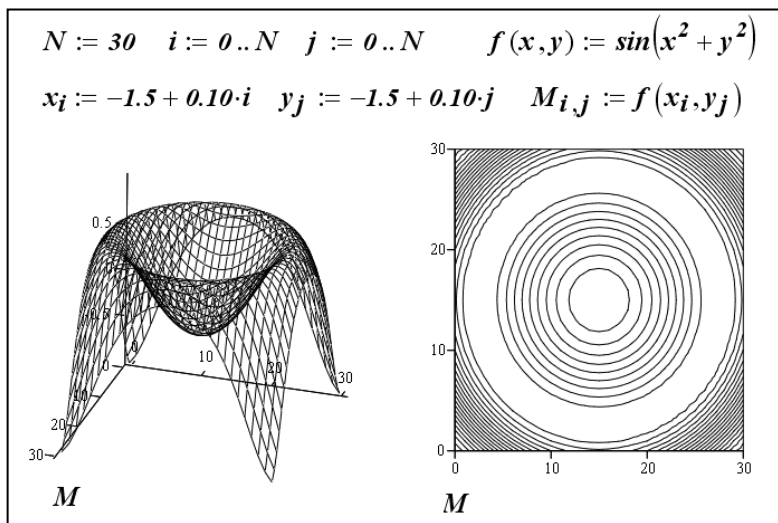


Рис. 4.10. Графики функции двух переменных

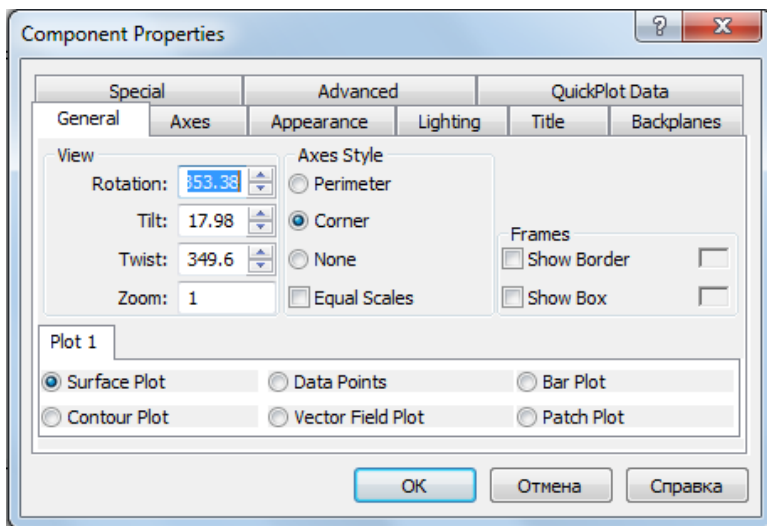


Рис. 4.11. Окно форматирования графика

- *Lighting (Освещение)* – задание условий освещения и выбор схемы освещения;
- *Title (Название)* – задание титульных надписей и их параметров;
- *Backplanes (Задний план)* – установка параметров форматирования граней;
- *Special (Специальный)* – задание специальных эффектов форматирования;
- *Advanced (Дополнительные)* – установка дополнительных параметров;
- *QuickPlotData (Данные QuickPlot)* – параметры быстрого построения графика.

Описание вкладок окна форматирования. Каждая из вкладок состоит из разделов устанавливающих характеристики графика.

1. Вкладка **General** содержит параметры углов представления фигуры, стили осей и внешнее оформление.

В разделе **View** (*Вид*) имеются следующие поля:

- *Поворот* – задание угла поворота (от 0 до 360 градусов);
- *Наклон* – задание угла наклона (от 0 до 180 градусов);
- *Кручение* – задание угла вращения (от 0 до 360 градусов);
- *Масштаб* – задание относительного размера.

Раздел **Axis Style** (*Стиль осей*) позволяет задать стиль отображения осей:

- *Периметр* – по периметру;
- *Угол* – в углу;
- *Ничего* – без вывода осей;
- *Равные деления* – равные масштабы по всем осям.

В разделе **Frames** (*Границы графика*) устанавливают параметры рисунка:

- *Границы* – показать рамку вокруг рисунка;
- *Каркас* – показать обрамляющий рисунок в виде параллелепипеда.

В разделе **Plot1** (*График*) устанавливается тип графика: *SurfacePlot* – график поверхности, *ContourPlot* – график изолиний, *DataPoints* – точечный график, *VectorFieldPlot* – график векторного поля.

2. Вкладка **Axis** служит для установки параметров координатных осей трехмерного графика. Внутри этой вкладки имеются еще три идентичные вкладки *X-ось*, *Y-ось*, *Z-ось*, которые позволяют установить параметры каждой из координатных осей *X*, *Y*, *Z*.

Раздел **Grids** (сетки) устанавливает формат координатной сетки:

- *Рисовать линии* – вывод линий сетки;
- *Рисовать метки* – вывод делений на осях;
- *Автосетка* – автоматический выбор числа линий;
- *Цвет линий сетки* – вывод цветных линий сетки (при включенной опции *Рисовать линии*);
- *Число* – установка количества делений;
- *Ширина линий* – установка ширины линий сетки.

Раздел Формат осей устанавливаются параметры самих координатных осей.

Раздел Границы осей задается предел изменения координат:

- *Автомасштаб* – автоматическая установка масштаба;
- *Минимум значение* – минимальное значение координаты;
- *Максимум значение* – максимальное значение координаты.

3. **Вкладка Внешний вид** включает в себя три раздела:

- **Установки заливки** – установка параметров заливки поверхностей;

- **Установки линий** – установка режима отображения линий и их окраска;

- **Установки точек** – установка режима отображения точек разными символами и их окраска.

Пример 4.4.2. Построить поле изолиний функции $f(x,y) = \sin(x^2 + y^2)$ для $x \in [-1.5, 1.5]$, $y \in [-1.5, 1.5]$, предварительно сформировав матрицу из значений функции в узлах сетки $[-1.5, 1.5] \times [-1.5, 1.5]$.

Решение. Документ MathCAD включает те же конструкции, что и в примере 4.4.1, но для вывода изолиний на вкладке **General** задается формат графика *ContourPlot*. Результирующий график показан правой фигурой на рис. 4.10.

Как отмечалось, построение графика функции двух переменных можно осуществить без явного формирования матрицы, имя которой указывается в поле шаблона. Для этого достаточно в поле шаблона задать только имя функции. Однако этот способ построения обладает одним недостатком: по умолчанию график строится на прямоугольной области $[-5,5] \times [-5,5]$ с числом узлов по каждой переменной 20. Для изменения этих параметров построения необходимо открыть окно форматирования и активизировать вкладку *QuickPlotData* (см. рис. 4.12) и изменить границы области и число узлов по осям.

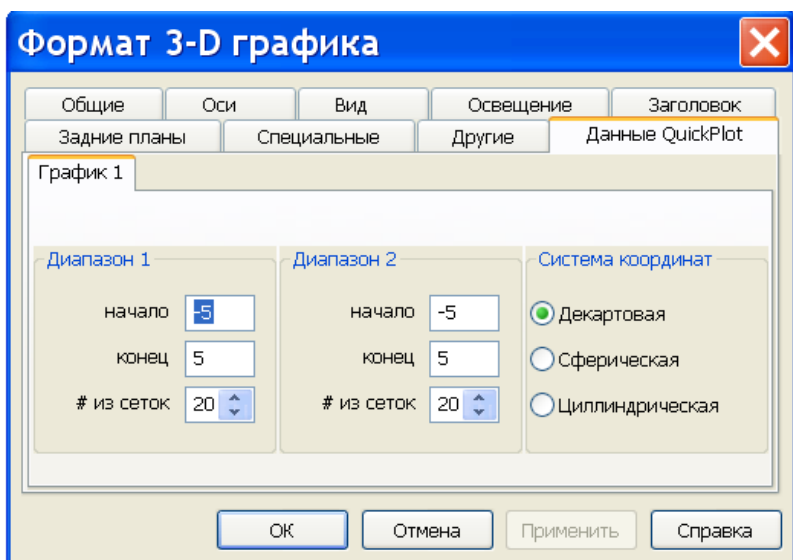


Рис. 4.12. Изменение параметров вкладки *QuickPlotData*

Пример 4.4.3. Построить график поверхности функции $f(x,y) = \sin(x^2 + y^2)$ для $x \in [-1.5, 1.5]$, $y \in [-1.5, 1.5]$ без формирования матрицы значений этой функции.

Решение. Построение графика показано в документе MathCAD, приведенном на рис. 4.13. Первоначально в поле шаблона было введено имя функции f . Результат построения показан на левом графике. Видно, что параметры графика, устанавливаемые по умолчанию, не позволили получить приемлемый масштаб графика. Для изменения масштаба в окне *QuickPlotData* (см. рис. 4.12) были установлены граничные точки: по оси X – 1.5 и 1.5, по оси Y – 1.5 и 1.5. Число узлов задавалось 30 по оси X и оси Y . Результат показан на правом графике, который идентичен графику примера 4.4.1 (см. левый график рис. 4.10).

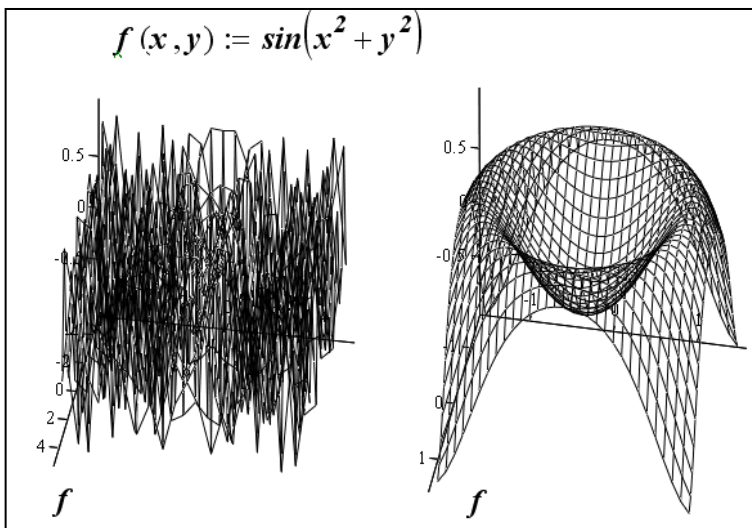


Рис. 4.13. Построение графиков без формирования матрицы

Функция *CreateMesh* (Создать сетку). Эта функция позволяет построить график функции двух переменных без вычисления матрицы с предварительным заданием диапазонов изменений аргументов и нужного числа узлов. Обращение к функции имеет вид:

$$\text{CreateMesh}(f, x_n, x_k, y_n, y_k, N_x, N_y),$$

где f – имя функции; x_n, x_k – границы интервала изменения аргумента x ; y_n, y_k – границы интервала изменения аргумента y ; N_x, N_y – количество узлов по x и y соответственно. Результатом работы является матрица, имя которой вводится в поле шаблона.

Пример 4.4.4. Построить график поверхности гиперболического параболоида $F(x, y) = x^2 - y^2$ для $-2 \leq x \leq 2$ и $-3 \leq y \leq 3$ и числом узлов 50 по каждой переменной.

Решение. Конструкции MathCAD, необходимые для построения графика, показаны на рис. 4.14.

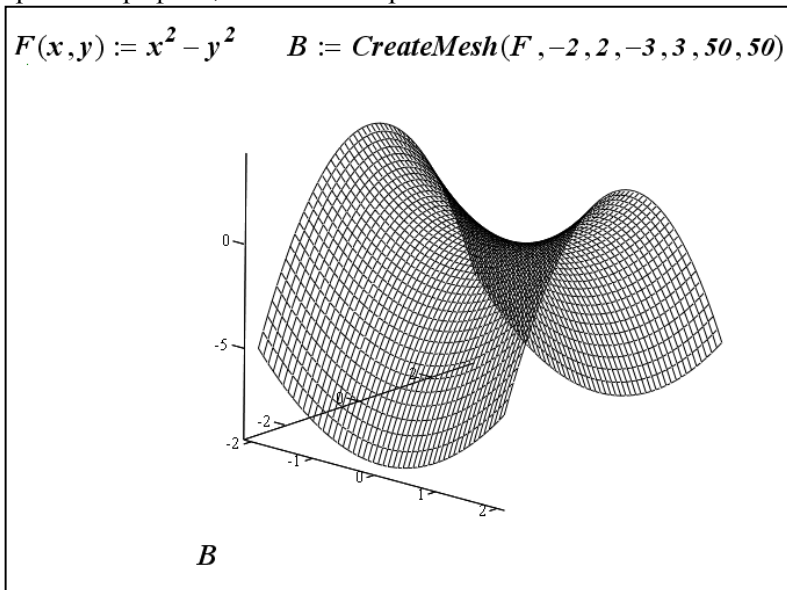


Рис. 4.14. Построение графика с помощью функции *CreateMesh*

Вращение трехмерных графиков. Вид трехмерной фигуры сильно зависит от того, под какими углами относительно осей X , Y и Z ее рассматривают. Вращая фигуру, можно заглянуть внутрь впадины или посмотреть, что находится за пиком. Для этого надо просто поместить указатель мыши в область графика, нажать левую кнопку мыши и, удерживая ее, начать перемещать мышь в том или ином направлении. Если при этом еще удерживать клавишу [Ctrl], то объект можно удалять или приближать к наблюдателю. Если же проделать те же действия с нажатой клавишей [Shift], то после отпускания левой кнопки можно вообще наблюдать анимированную (живую) картину вращения объекта в любом предварительно заданном направлении. Для остановки вращения надо щелкнуть левой кнопкой мыши.

Мини-задание. Попробуйте применить вращение графика на правом графике, приведенном на рис. 4.13 (пример 4.4.3).

4.5. Анимация в MathCAD

Для создания анимации в MathCAD имеется системная переменная *FRAME* (счетчик кадров), которая принимает целочисленные значения (по умолчанию от 0 до 9) с шагом 1. В функцию, для которой строится анимация, переменная *FRAME* вводится как дополнительный параметр. Диапазон изменений *FRAME* задается в диалоговом окне команды *Animation*.

Для построения анимационного графика необходимо выполнить следующие действия:

1) задать системную переменную как дискретную, например: $t := 2 \cdot \text{FRAME}$;

2) задать функцию, одним из параметров которой будет ранжированная переменная, определяющая счетчик кадров *FRAME*;

3) построить график функции, используя палитру *Graph*, отформатировать график;

4) вызвать диалоговое окно анимации, выбирая последовательно из строки меню **Tools / Animation / Record** (рис. 4.14);

5) отодвинуть окно анимации от графика, чтобы он не загромождал его;

6) выделить мышью построенный график, заключив его в маркировочный прямоугольник;

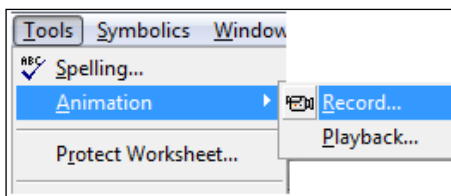


Рис. 4.14. Выбор диалогового окна анимации

7) установить в диалоговом окне параметры анимации:

- задать минимальное и максимальные значения параметра *FRAME* (поля **From** и **To**);

– задать в поле **At** количество воспроизводимых кадров в секунду;

8) выполнить щелчок на кнопке **Animate**. При этом в области просмотра окна анимации появятся последовательности анимационных кадров, а под этой областью можно наблюдать изменение переменной **FRAME**;

9) для просмотра анимированных рисунков нужно щелкнуть по кнопке **Play** в появившемся окне **Playback (Прои**грывать);

10) при помощи команды **Save As** можно сохранить анимацию в файле с расширением **AVI**.

Пример 4.5.1. Создать анимацию графика в полярной системе координат.

Решение. Необходимо выполнить следующие шаги:

- установить значение размерной переменной $t := \text{FRAME}$;
- вызвать шаблон графика в полярной системе координат;
- ввести в поля шаблона значение полярного угла, содержащего размерную переменную t ($7 \cdot \varphi - t$) и полярный радиус равный ρ ;
- вызвать диалоговое окно, установить параметры и сформировать последовательности кадров;
- включить просмотр анимации на проигрывателе.

На рис. 4.15 показан документ MathCAD создания анимации.

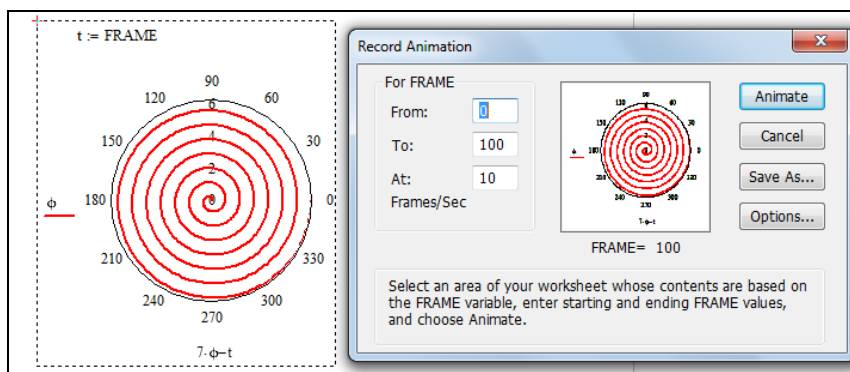


Рис. 4.15. Формирование анимационных последовательностей

Вопросы и задания для самопроверки

1. Какие типы графиков можно построить в MathCAD?
2. Задайте дискретную переменную x , меняющуюся от 0 до $\pi/2$ с шагом 0.1; определите функцию $f(x) = x \cdot \sin(2x)^2$, постройте ее график.
3. Постройте на одном рисунке графики функций: $y_1 = 2\cos(x)$, $y_2 = \sin(x)^2$ и $y_3 = x$ для $x \in [0, 5.0]$ с шагом 0.02. Используя вкладку окна форматирования *Trace (Трассировка)*, измените стиль отображения и цвет кривых.
4. Постройте график функции $y = \cos(x) + 3$, где x изменяется от 10 до 20, а y от 0 до 10. Изменяя диапазон по y , увеличьте построенный график и немного сместите его. В итоге на рисунке должны быть увеличенный и перемещенный график.
5. Постройте график функции $f(x) = \cos(x) + \sin(x)^2$. Задайте свой масштаб, изменив диапазон значений аргумента x .
6. Постройте график функции $p(x) = 5x^6 - 3$, задав свой цвет и стиль кривой.
7. Для индекса i , меняющегося от нуля до 15, вычислите массивы $x_i = i/10$, $y_i = x_i \cdot \sin(2x_i)^2$, постройте график функции $y_i(x_i)$.
8. Постройте на разных рисунках графики следующих функций в полярной системе координат (угол φ меняется от 0 до 2π):

- $\rho(\varphi) := \cos(\varphi) \cdot \sin(\varphi)$;
- $\rho(\varphi) := \varphi$;
- $\rho(\varphi) := e^\varphi$.

Выполнить необходимое форматирование графиков.

9. Постройте график лемнискаты Бернулли

$$r(\varphi) := 3 \cdot \sqrt{2 \cdot \cos(2\varphi)}$$

в полярной системе координат для $\varphi \in [0, 2\pi]$.

10. Постройте анимационный полярный график $R(\varphi) = \cos(FRAME \cdot \varphi)$ (число кадров равно 20, число кадров в секунду – 4). Просмотрите на *Плере* получившуюся анимацию.

11. Постройте график поверхности функции $g(x, y) = x^2 - y^2$, где переменные x и y меняются от -5 до 5 с шагом 0.25 . График поверхности строится по вычисленной матрице M (см. пример 4.4.2).

12. Постройте график поверхности функции $g(x, y) = x^2 - y^2$, где переменные x и y меняются от -5 до 5 с шагом 0.25 без вычисления матрицы M (см. пример 4.4.3). Используя вкладку *QuickPlotData*, задайте необходимое количество узлов по переменным x и y . Сравните с графиком примера 4.4.4.

13. Постройте график поверхности, используя конструкции следующего документа:

$$\begin{aligned} f(x, y) &:= \sin(x^2 + y^2) \\ N &:= 20 \quad x := 0..N \quad y := 0..N. \\ M_{x,y} &:= f\left[\frac{(x-10)}{5}, \frac{(y-10)}{5}\right] \end{aligned}$$

Определить диапазоны значений аргументов x и y . Заметим, что в приведенном фрагменте переменные x и y выполняют двойную функцию: являются индексами у формируемых элементов матрицы M и аргументами функции $f(x, y)$.

ТЕМА 5. СИМВОЛЬНЫЕ ВЫЧИСЛЕНИЯ В MATHCAD

Пакет MathCAD позволяет получать результат некоторых вычислений в символьном виде, т.е. в виде аналитического выражения. Такие вычисления называются символьными. В отличие от численных вычислений, которые дают частный (численный) результат, при символьных вычислениях полученные аналитические выражения обладают высокой общностью результатов. Символьные вычисления можно осуществлять с помощью:

- команд строки меню *Symbolics* (Символика);
- оператора символьного вывода « \rightarrow » и команд символьного процессора, которые вводятся с палитры инструментов *Symbolic* (Символические).

5.1. Команды символьных вычислений строки меню *Symbolics*

Команды меню *Symbolics* (рис 5.1) более удобны для отображения аналитического результата выражения в целом или его части, не сохраняя сам ход вычислений. Чтобы символьные команды выполнялись, необходимо выделить ту часть выражения, над которым будем производить преобразование, или выделить переменную, относительно которой выполняется символьная операция.

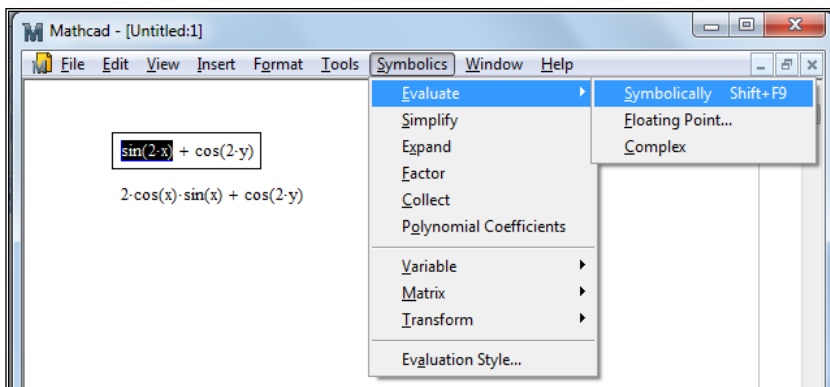


Рис. 5.1. Команды строки меню *Symbolics*

Команды, выполняемые при выделении выражения:

- *Evaluate (Выполнить)* – преобразовать выражение с выбором вида преобразования (см. рис. 5.1).
- *Simplify (Упростить)* – упростить выделенное выражение.
- *Expand (Расширить)* – разложить выражение по степеням.
- *Factor (Факторизовать)* – разложить выражение по степеням на множители.
- *Collect (Разложить по подвыражениям)* – собрать слабые, подобные выделенному выражению.
- *Polynomial Coefficients (Полиномиальные коэффициенты)* – найти коэффициенты полинома по заданной переменной.

Пример 5.1.1. Дана формула $\sin(2 \cdot x) + \cos(2 \cdot y)$. Необходимо получить разложение на сомножители в символьном виде только выражения $\sin(2 \cdot x)$.

Решение. Необходимо: выделить **$\sin(2 \cdot x)$** как часть формулы (см. рис. 5.1); выбрать в главном меню пункты *Symbolics Evaluate / Symbolically (Символика / Вычислить / Символически)*.

После этого результат разложения выражения появится чуть ниже формулы в виде еще одной строки:

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

Такой же результат получим и при выполнении в главном меню пунктов *Symbolics / Expand (Символика / Разложить)*.

Команды, выполняемые при выделении переменной. Эти команды находятся в подменю *Variable* пункта *Symbolics* и они показаны на рис. 5.2.

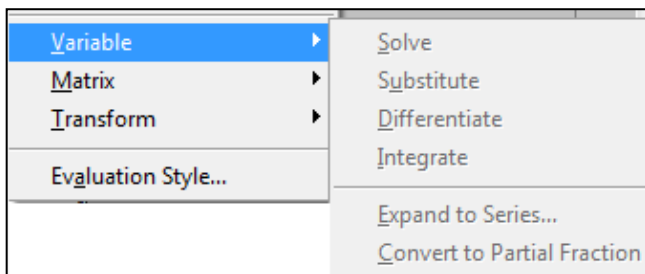


Рис. 5.2. Команды подменю *Variable*

Приведем эти команды:

- *Solve* (*Решить*) – решить уравнение или неравенство относительно выделенной переменной.
- *Substitute* (*Подставить*) – заменить выделенную переменную значением из буфера обмена.
- *Differentiate* (*Дифференцировать*) – дифференцировать выражение относительно выделенной переменной.
- *Integrate* (*Интегрировать*) – интегрировать выражение по выделенной переменной.
- *Expand to Series* (*Разложить в ряд*) – разложить выражение в ряд Тейлора относительно выделенной переменной.
- *Convert to Partial Fraction* (*Разложить на элементарные дроби*) – разложить выражение на элементарные дроби.

Пример 5.1.2. Вычислить производную выражения $(1-e^x) \cdot \sin(x)$ в символьном виде.

Решение. Необходимо:

- выделить переменную x в выражении (см. рис. 5.3);
- выбрать в главном меню пункты *Symbolics / Variable / Differentiate* (*Символика / Переменная / Дифференцировать*).

После этого результат разложения выражения появится чуть ниже формулы в виде еще одной строки.

$$\cdot (1 - e^x) \cdot \sin(x)$$

$$\cdot -\cos(x) \cdot (e^x - 1) - e^x \cdot \sin(x)$$

Рис. 5.3. Операция относительно выделенной переменной

Команды, выполняемые с выделенными матрицами.

Эти команды находятся в подменю команды *Matrix* (см. рис. 5.4):

- *Transpose* (Транспонировать) – получить транспонированную матрицу.
- *Invert* (Обратить) – получить обратную матрицу.
- *Determinant* (Определитель) – вычислить определитель матрицы.

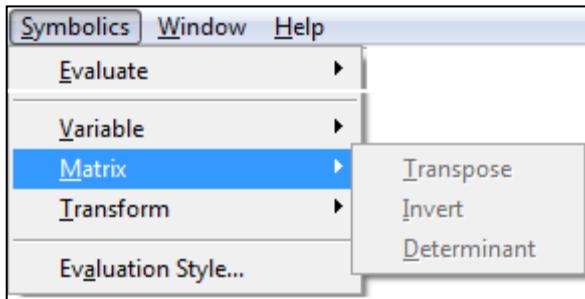


Рис. 5.4. Подменю команды *Matrix*

Пример 5.1.3. Вычислить в символьном виде определитель матрицы размером 3×3 , показанной на рис. 5.5.

Решение. Для этого необходимо:

- создать матрицу 3×3 и заполнить ее символьными значениями (см. рис. 5.5);

- выделить матрицу;
- выбрать в главном меню пункты *Symbolics / Matrix / Determinant* (Символика / Матрица / Определитель).

После этого результат символьного вычисления определителя появится чуть ниже матрицы в виде строки.

The image shows a software window with a light gray background. On the left, a 3x3 matrix is displayed within a white box with a thin black border. The matrix elements are: top row (q, w, e), middle row (r, t, y), and bottom row (u, i, o). Below the matrix, the determinant is calculated and shown as a single line of text: $o \cdot q \cdot t - i \cdot q \cdot y - o \cdot r \cdot w + u \cdot w \cdot y + i \cdot r \cdot e - t \cdot u \cdot e$. The text is in a monospaced font, with some characters in different colors (blue, red, green) to indicate signs and variables.

Рис. 5.5. Символьное вычисление определителя матрицы

Команды выполнения интегральных преобразований.

Эти команды находятся в подменю команды *Transform* и они показаны на рис. 5.6.

Приведем эти команды:

- *Fourier* (Преобразование Фурье) – выполнить прямое преобразование Фурье относительно выделенной переменной.
- *Inverse Fourier* (Обратное преобразование Фурье) – выполнить обратное преобразование Фурье относительно выделенной переменной.
- *Laplace* (Преобразование Лапласа) – выполнить прямое преобразование Лапласа относительно выделенной переменной.

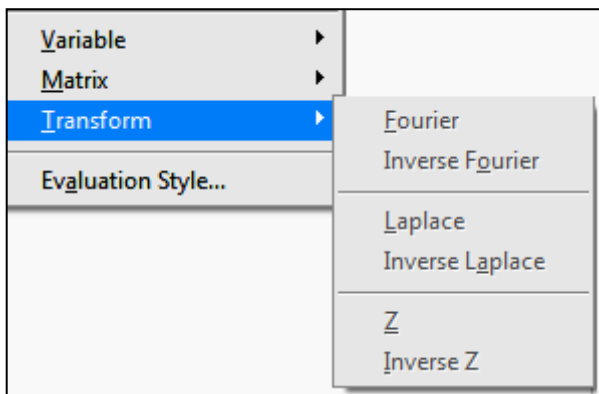


Рис. 5.6. Подменю *Transform*

- *Inverse Laplace* (Обратное преобразование Лапласа) – выполнить обратное преобразование Лапласа относительно выделенной переменной.
- *Z* (*Z-преобразование*) – выполнить прямое *Z*-преобразование относительно выделенной переменной.
- *Inverse Z* (Обратное *Z-преобразование*) – выполнить обратное *Z*-преобразование относительно выделенной переменной.

Прямое преобразование Фурье позволяет получить в аналитическом виде функцию частоты $F(w)$ от временной функции $f(t)$:

$$F(w) = \int_{-\infty}^{\infty} f(t) \cdot e^{-iwt} dt.$$

Соответственно, обратное преобразование Фурье позволяет по функции $F(w)$ найти функцию $f(t)$:

$$f(t) = \int_{-\infty}^{\infty} F(w) \cdot e^{iwt} dw.$$

Прямое преобразование Лапласа позволяет получить в передаточную функцию $F(s)$ от временной функции $f(t)$:

$$F(s) = \int_0^{\infty} f(t) \cdot e^{-st} dt ,$$

где s — параметр интегрального преобразования, который, в общем случае, является комплексной величиной $s = \sigma + i\omega$.

Обратное преобразование Лапласа позволяет по передаточной функции $F(s)$ найти временную функцию $f(t)$:

$$f(t) = \int_{\sigma - i\infty}^{\sigma + i\infty} F(s) \cdot e^{st} ds.$$

Z-преобразованием называют преобразование дискретного сигнала $f(n)$ (последовательность вещественных чисел) в аналитическую функцию комплексной частоты $z = e^{i\omega n}$. Прямое Z-преобразование реализуется выражением:

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

Обратное Z-преобразование вычисляется контурным интегралом:

$$f(n) = \frac{1}{2\pi i} \int F(z) \cdot z^{n-1} dz.$$

Пример 5.1.4. Над функцией e^{-at} выполнить все 3 типа преобразований (Фурье, Лапласа и Z-преобразование).

Решение. Для выполнения команд, осуществляющих эти преобразования, необходимо:

- записать исходное выражение;
- выделить переменную, относительно которой будет производиться преобразование;
- выбрать в главном меню *Symbolics / Transform /* необходимую команду (см. рис. 5.5).

Результат преобразований показан на рис. 5.7.

Фурье	Лапласа	Z- преобразование	
$e^{-a \cdot t}$	$e^{-a \cdot t}$	$e^{-a \cdot t}$	Функция
$2 \cdot \pi \cdot \Delta(\omega + a \cdot i)$	$\frac{1}{a + s}$	$\frac{z}{z - e^{-a}}$	Прямое
$e^{-a \cdot t}$	$e^{-a \cdot t}$	$(e^{-a})^t$	Обратное

Рис. 5.7. Преобразования Фурье, Лапласа и Z-преобразование

5.2. Символьные вычисления командами палитры *Symbolic*

Способ с использованием оператора символьного вывода « \rightarrow » и команд, которые вводятся с палитры инструментов Symbolic (рис. 5.8), более нагляден, так как позволяет записывать выражения в традиционной математической форме и сохранять символьные вычисления в документах MathCAD.

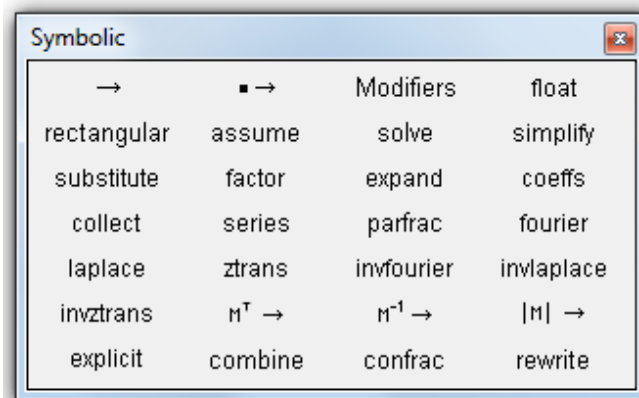


Рис. 5.8. Палитра инструментов *Symbolic* (Символические)

Палитра инструментов *Symbolic* (Символические) вызывается из панели инструментов *Math* (Математика). В палитре *Symbolic* находятся кнопки, соответствующие специфическим командам символьных преобразований. В состав палитры *Symbolic* также входят команды, аналогичные командам меню *Symbolics*, рассмотренные выше.

Команды вводятся после вычисляемого выражения. Команда заканчивается оператором символьного вывода « \rightarrow » и может иметь пустое поле, которое отделяется от команды запятой. В этом поле задает имя переменной, относительно которой выполняется то или иное символьное преобразование.

Палитра инструментов *Symbolic* (Символические) вызывается из панели инструментов *Math* (Математика). В палитре *Symbolic* находятся кнопки, соответствующие специфическим командам символьных преобразований. В состав палитры *Symbolic* также входят команды, аналогичные командам меню *Symbolics*, рассмотренные выше.

Команды вводятся после вычисляемого выражения. Команда заканчивается оператором символьного вывода « \rightarrow » и может иметь пустое поле, которое отделяется от команды запятой. В этом поле задает имя переменной, относительно которой выполняется то или иное символьное преобразование.

Пример 5.2.1. Выполнить разложение выражения $\cos(2y)$ относительно переменной y .

Решение. Выполним следующую последовательность действий (рис. 5.9):

- введите выражение $\cos(2 \cdot y)$;
- нажмите кнопку *Expand* (Разложить) на панели *Symbolic*;
- введите через запятую переменную, относительно которой производится вычисление;
- щелкните мышью в стороне от выражения.

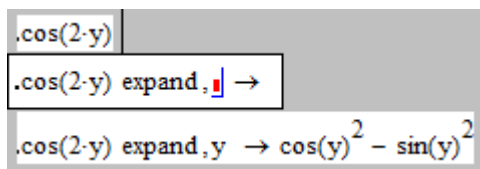
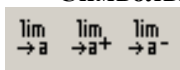


Рис. 5.9. Символьное разложение выражения

По умолчанию оператор символьного вывода « \rightarrow » выполняет функцию упрощения (*simplify*), т.е. берет выражение с левой стороны и помещает его упрощенную версию с правой.

Символьное вычисление пределов. Кнопки



палитры инструментов *Calculus (Исчисления)* используются для символьного вычисления пределов. Назначение этих кнопок понятно из их обозначения. Примеры вычисления пределов приведены на рис. 5.10.

$$\lim_{x \rightarrow \infty} \frac{(1 + 3 \cdot x)}{x} \rightarrow 3 \qquad \lim_{x \rightarrow 0^+} \frac{1}{x} \rightarrow \infty$$

Рис. 5.10. Символьное вычисление пределов

Символьные вычисления операций сложения, умножения, дифференцирования и интегрирования. Эти вычисления осуществляются с использованием соответствующих кнопок палитры инструментов *Матанализ*.

Пример 5.2.3. Выполнить символьные вычисления выражений, приведенных на рис. 5.11.

На рис. 5.11 показаны необходимые символьные вычисления с использованием операторов суммы, произведения, дифференцирования и интегрирования.

Символьные алгебраические преобразования. Символьный процессор MathCAD может выполнять основные алгебраические преобразования, такие как упрощение выражения, разложение их на множители, вычисление коэффициентов полинома и т.д. Рассмотрим некоторые часто встречающиеся преобразования.

$$\sum_{i=1}^n i^2 \rightarrow \frac{1}{3} \cdot (n+1)^3 - \frac{1}{2} \cdot (n+1)^2 + \frac{1}{6} \cdot n + \frac{1}{6} - \frac{1}{3} \cdot 1^3 + \frac{1}{2} \cdot 1^2 - \frac{1}{6} \cdot 1$$

$$\prod_{n=1}^{\infty} \frac{1}{n^2 + 1} \rightarrow 0 \qquad \frac{d}{dx} \sin(x) \rightarrow \cos(x)$$

$$\int \ln(x) dx \rightarrow x \ln(x) - x \qquad \int_a^b \frac{1}{x^3} dx \rightarrow \frac{-1}{2 \cdot b^2} + \frac{1}{2 \cdot a^2}$$

Рис. 5.11. Примеры символьных вычислений

Замечание 5.2.1. Если перед вычисляемым алгебраическим выражением задать численные значения переменным, входящим в алгебраическое выражение, то символьный процессор будет «обрабатывать» уже соответствующие числовые значения.

Simplify – команда, осуществляющая упрощение алгебраического выражения.

Пример 5.2.4. Упростите алгебраическое выражение, приведенное на рис. 5.12.

Решение. Сначала введем преобразуемые выражения, а затем выполним необходимые упрощения. В первой строке для этого используется оператор символьного вывода (по «умолча-

нию» упрощение); во второй – оператор символьного вывода и команда *Simplify*.

$$\begin{aligned} & (3x + 2) - (5x - 3) \rightarrow -2 \cdot x + 5 \\ & (3x^2 - 2x + 1) + (4 \cdot x^3 - 5x^2 + 1) \text{ simplify} \rightarrow -2 \cdot x^2 - 2 \cdot x + 2 + 4 \cdot x^3 \end{aligned}$$

Рис. 5.12. Упрощение алгебраических выражений (*simplify*)

Factor – команда, осуществляющая разложение алгебраического выражения на множители.

Пример 5.2.5. Выполнить разложение выражения, приведенное на рис. 5.13.

Решение. Первоначально введем преобразуемое выражение, к которому затем применим команду **factor**.

$$a^2 \cdot b + a \cdot b^2 + 2 \cdot a \cdot b \cdot c + b^2 \cdot c + a^2 \cdot c + a \cdot c^2 + b \cdot c^2 \text{ factor} \rightarrow (b + c) \cdot (a + c) \cdot (a + b)$$

Рис. 5.13. Разложение алгебраического выражения (*factor*)

Parfrac – команда, осуществляющая разложение алгебраического выражения на простые дроби. Эта команда может иметь поле, отделяемое от команды запятой, в котором задается переменная или выражение относительно которых будет осуществляться разложение

Пример 5.2.6. Выполнить разложение на простые дроби выражения, приведенное на рис. 5.14.

Решение. Первоначально введем преобразуемое выражение, к которому затем применим команду *parfrac* (первая строка документа MathCAD). Во второй строке показан вызов команды, в поле которой указана переменная x . Естественно, что для дан-

ного выражения (в котором только одна переменная) результаты этих двух преобразований совпадают (см. рис. 5.14).

$$\frac{x^2 - 3 \cdot x + 7}{(x - 1)^2 \cdot (x^2 + x + 1)} \text{ parfrac} \rightarrow \frac{6 \cdot x + 10}{3 \cdot (x^2 + x + 1)} - \frac{2}{x - 1} + \frac{5}{3 \cdot (x - 1)^2}$$

$$\frac{x^2 - 3 \cdot x + 7}{(x - 1)^2 \cdot (x^2 + x + 1)} \text{ parfrac ,x} \rightarrow \frac{6 \cdot x + 10}{3 \cdot (x^2 + x + 1)} - \frac{2}{x - 1} + \frac{5}{3 \cdot (x - 1)^2}$$

Рис. 5.14. Разложение на простые множители

Expand – команда, осуществляющая раскрытие скобок и приведение подобных членов. В команду можно вставить поле (через запятую), в котором указывается переменная, относительно которой выполняется преобразование, а именно: **expand ,*n*** →

Пример 5.2.7. Выполнить раскрытие скобок и приведение подобных членов выражения, приведенное на рис. 5.15.

Решение. Первоначально введем преобразуемое выражение, а затем обратимся к команде **expand**. Первая строка документа на рис. 5.15 содержит преобразования относительно переменной *x*. Во второй строке приведено преобразование, выполненное относительно константы 2.

$$x \cdot (z + 1)^2 - 2 \cdot x \cdot (x + z) \text{ expand ,x} \rightarrow x \cdot z^2 - 2 \cdot x^2 + x$$

$$x \cdot (z + 1)^2 - 2 \cdot x \cdot (x + z) \text{ expand ,2} \rightarrow x \cdot (z + 1)^2 - 2 \cdot x^2 - 2 \cdot x \cdot z$$

Рис. 5.15. Приведение подобных членов алгебраического выражения

Coeffs – команда осуществляет преобразование выражения, являющегося полиномом относительно некоторой переменной (например, x), к виду:

$$a_0 + a_1x + a_2x^2 + \dots$$

В команду можно вставить поле (через запятую), в котором указывается переменная, относительно которой выполняется преобразование. Найденные коэффициенты полинома представляются в виде вектора.

Пример 5.2.8. Вычислить коэффициенты полинома, приведенного на рис. 5.16.

Решение. В первой строке документа MathCAD осуществляется вычисление коэффициентов полинома относительно переменной xx , во второй строке – относительно переменной z .

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

Рис. 5.16. Вычисление коэффициентов полинома

Series – команда осуществляет разложение заданного выражения или функции в ряд Тейлора. Напомним, что рядом Тейлора (относительно точки x_0) называется ряд вида:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 + \dots + \frac{f^{(n)}(x_0)}{n}(x - x_0)^n + \dots$$

Как правило, в вычислениях сохраняют конечное число слагаемых этого ряда, например, n слагаемых. Возможные

формы записи команды *series* показаны на рис. 5.17 и обозначены латинскими буквами.

■ <i>series</i> → <i>a</i>	■ <i>series</i> , <i>x</i> = <i>a</i> → <i>b</i>	■ <i>series</i> , <i>n</i> → <i>c</i>	■ <i>series</i> , <i>x</i> = <i>a</i> , <i>n</i> → <i>d</i>
-------------------------------	---	--	--

Рис. 5.17. Обращение к команде *series*

Поясним эти формы:

- a) подразумевает $x_0 = 0$ (ряд Маклорена) и число членов разложения равно 6;
- b) подразумевает $x_0 = a$ (ряд Тейлора) и число членов разложения равно 6;
- c) подразумевает $x_0 = 0$ (ряд Маклорена) и число членов разложения равно n ;
- d) подразумевает $x_0 = a$ (ряд Тейлора) и число членов разложения равно n .

Пример 5.2.9. Выполнить разложение функции $f(x) = e^{-x}$, используя различные формы записи команды *series*.

Решение. На рис. 5.18 приведен фрагмент документа MathCAD, в котором используются различные формы команды *series*.

$f(x) := e^{-x}$	
$f(x) \text{ series } \rightarrow 1 - x + \frac{x^2}{2} - \frac{x^3}{6} + \frac{x^4}{24} - \frac{x^5}{120}$	
$f(x) \text{ series , } x = 0.4, 4 \rightarrow 0.938 + -0.67 \cdot x + 0.335 \cdot (x - 0.4)^2 + -0.112 \cdot (x - 0.4)^3$	
$f(x) \text{ series , } 3 \rightarrow 1 - x + \frac{x^2}{2}$	$f(x) \text{ series , } 4 \rightarrow 1 - x + \frac{x^2}{2} - \frac{x^3}{6}$

Рис 5.18. Различные записи функции *series*

5.3. Символьное решение уравнений и систем уравнений

Для вычисления решения уравнения или систем уравнений используется команда *solve*. После команды через запятую можно задать переменную, относительно которой ищется решение.

Решение нелинейного уравнения. Напомним, что нелинейным уравнением относительно переменной x является запись вида

$$f(x) = 0.$$

Корнями нелинейного уравнения называются такие значения x_j^* , $j = 1, \dots, m$, для которых $f(x_j^*) = 0$. В зависимости от вида функции $f(x)$ может иметь несколько корней, т.е. $m > 1$.

Пример 5.3.1. Используя команду *solve*, найдите решение уравнения $x - a + 5 = 0$ в символьном виде.

Решение. Обращение к команде *solve* показано в документе MathCAD, приведенном на рис. 5.19. Заметим, что при первом обращении к команде не была указана переменная, относительно которой ищется решение, и это вызвало ошибку, диагностика которой также показана на рисунке.

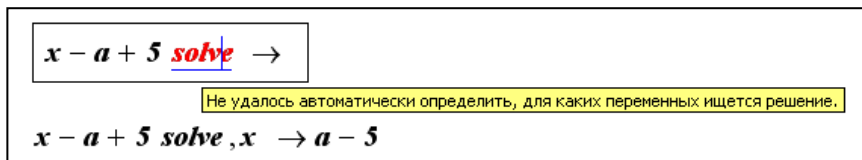


Рис. 5.19. Решение нелинейного уравнения примера 5.3.1

Пример 5.3.2. Используя команду *solve*, найдите решение уравнения $N = N_0 e^{-\lambda h}$ в символьном виде относительно переменных λ и N_0 .

Решение. Обращение к команде показано в документе MathCAD, приведенном на рис. 5.20. Заметим, что в левом поле команды *solve* записано нелинейное уравнение, в котором стоит «жирный» знак $=$.

$$N = N_0 \cdot e^{-\lambda \cdot h} \text{ solve, } \lambda \rightarrow -\frac{\ln\left(\frac{N}{N_0}\right)}{h} \quad N = N_0 \cdot e^{-\lambda \cdot h} \text{ solve, } N_0 \rightarrow N \cdot e^{\lambda \cdot h}$$

Рис. 5.20. Решение нелинейного уравнения примера 5.3.2

Пример 5.3.3. Используя команду *solve*, вычислить все корни алгебраического уравнения

$$f(x) = x^3 - 6x^2 + 21x - 52 = 0.$$

Решение. Известно, что рассматриваемое уравнение имеет три корня. На рис. 5.21 показаны два обращения к команде *solve*. В первом обращении в левом поле команды стоит запись нелинейного уравнения, во втором – левая часть нелинейного уравнения.

$$\begin{aligned} f(x) := x^3 - 6 \cdot x^2 + 21 \cdot x - 52 \quad f(x) = 0 \text{ solve, } x &\rightarrow \begin{pmatrix} 4 \\ 1 + 2i \cdot \sqrt{3} \\ 1 - 2i \cdot \sqrt{3} \end{pmatrix} \\ x^3 - 6 \cdot x^2 + 21 \cdot x - 52 \text{ solve, } x &\rightarrow \begin{pmatrix} 4 \\ 1 + 2i \cdot \sqrt{3} \\ 1 - 2i \cdot \sqrt{3} \end{pmatrix} \end{aligned}$$

Рис. 5.21. Решение нелинейного уравнения примера 5.3.3

Для учета априорной информации или ограничений относительно искомых решений дополнительно к команде *solve* используется команда *assume*, в которой через запятую вводятся определенные требования к искомому решению. Часто используемые ограничения приведены в следующем примере.

Пример 5.3.4. Дано нелинейное алгебраическое уравнение

$$f(x) = (x^3 - 1)(x^2 - 2) = 0.$$

Используя команду *solve*, вычислить:

1. Все корни этого алгебраического уравнения.
2. вещественные корни этого алгебраического уравнения.
3. вещественные корни из интервала $(0, 2)$.

Решение. Вычисление корней в соответствии с вышеперечисленными требованиями показано на рис. 5.22.

Решение систем нелинейных уравнений. Напомним, что системой из n нелинейных уравнений относительно m неизвестных называется запись:

$$\left\{ \begin{array}{l} f_1(x_1, x_2, \dots, x_m) = 0 \\ f_2(x_1, x_2, \dots, x_m) = 0 \\ \dots\dots\dots \\ f_n(x_1, x_2, \dots, x_m) = 0. \end{array} \right.$$

Решением этой системы являются значения $x_1^*, x_2^*, \dots, x_m^*$, которые обращают каждое уравнение системы в тождество. Как правило, число уравнений равно числу неизвестных, т.е. $m = n$.

$$\begin{aligned} (x^3 - 1) \cdot (x^2 - 2) &= 0 \text{ solve} \rightarrow \begin{pmatrix} 1 \\ \sqrt{2} \\ -\sqrt{2} \end{pmatrix} \\ (x^3 - 1) \cdot (x^2 - 2) &= 0 \left| \begin{array}{l} \text{solve} \\ \text{assume}, x = \text{real} \end{array} \right. \rightarrow \begin{pmatrix} 1 \\ \sqrt{2} \\ -\sqrt{2} \end{pmatrix} \\ (x^3 - 1) \cdot (x^2 - 2) &= 0 \left| \begin{array}{l} \text{solve} \\ \text{assume}, x = \text{RealRange}(0, 2) \end{array} \right. \rightarrow \begin{pmatrix} 1 \\ \sqrt{2} \end{pmatrix} \end{aligned}$$

Рис. 5.22. Решение нелинейного уравнения примера 5.3.4

Для нахождения решения системы уравнений в символьном виде также используется команда *solve*, но в левом поле этой команды стоит система уравнений, вводимая с помощью шаблона вектора, число проекций которого равно числу уравнений системы. В правом поле команды указывается вектор, составленный из имен переменных, относительно которых ищется решение.

Пример 5.3.5. Вычислить решение следующей системы уравнений:

$$\begin{cases} x + 13y = 2; \\ 2x - y = 1. \end{cases}$$

Решение. Фрагмент документа MathCAD, вычисляющий решение системы приведен на рис. 5.23. Вычисленное решение системы выводится в виде строки.

$$\left(\begin{matrix} x + 13 \cdot y = 2 \\ 2 \cdot x - y = 1 \end{matrix} \right) \text{solve}, \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} \frac{5}{9} & \frac{1}{9} \end{pmatrix}$$

Рис. 5.23. Решение нелинейного уравнения примера 5.3.5

Пример 5.3.6. Вычислить решение следующей системы уравнений:

$$\begin{cases} bx + 13y = 2; \\ 2x - ay = 1. \end{cases}$$

Решение представить в виде функции пользователя.

Решение. Фрагмент документа MathCAD, вычисляющий решение системы, приведен на рис. 5.24. Вычисленное решение системы зависит от двух коэффициентов a, b , которые являются аргументами функции пользователя *Sol*. Обращение к этой функции с аргументами $a=1, b=1$ дало решение предыдущей задачи 5.3.4.

$$\begin{aligned}
 \text{Sol}(a, b) &:= \begin{pmatrix} b \cdot x + 13 \cdot y = 2 \\ 2 \cdot x - a \cdot y = 1 \end{pmatrix} \text{solve}, \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \left(\frac{2 \cdot a + 13}{a \cdot b + 26} - \frac{b - 4}{a \cdot b + 26} \right) \\
 \text{Sol}(1, 1) &= \begin{pmatrix} 5 \\ 9 \\ 9 \end{pmatrix}
 \end{aligned}$$

Рис. 5.24. Решение нелинейного уравнения примера 5.3.6

Для проверки найденного решения можно использовать команду *substitute*, с помощью которой осуществляется подстановка значений, заданных в правом поле команды в систему уравнений, заданных в левом поле команды. Результатом выполнения команды будет вектор, количество проекции которого равно количеству уравнений решаемой системы. Каждая проекция вектора представляет собой результаты вычисления левой и правой части соответствующего уравнения системы, связанные «жирным» знаком $=$. Если слева и справа стоят одинаковые значения, то подставленные значения обращают уравнение в тождество. В противном случае подставленные значения не являются решением данного уравнения.

Для задания значений нескольким переменным необходимо ввести соответствующее число команд *substitute*.

Пример 5.3.7. Используя команду *substitute*, проверить правильность решения системы, вычисленного в примере 5.3.5.

Решение. В первой проверке (см. рис. 5.25) были подставлены значения, найденные в примере 5.3.3 и являющиеся решением рассматриваемой системы уравнений. Получили вектор, проекции которого содержат равенства, и это говорит о том, что подставленные значения являются решением системы. Во второй проверке были подставлены другие значения, которые не являются решением системы. Это видно из проекций результирующего вектора.

$\begin{pmatrix} x + 13 \cdot y = 2 \\ 2 \cdot x - y = 1 \end{pmatrix}$	$\begin{array}{l} \text{substitute } x = \frac{5}{9} \\ \text{substitute } y = \frac{1}{9} \end{array} \rightarrow \begin{pmatrix} 2 = 2 \\ 1 = 1 \end{pmatrix}$
$\begin{pmatrix} x + 13 \cdot y = 2 \\ 2 \cdot x - y = 1 \end{pmatrix}$	$\begin{array}{l} \text{substitute } x = \frac{5}{9} \\ \text{substitute } y = \frac{2}{9} \end{array} \rightarrow \begin{pmatrix} \frac{31}{9} = 2 \\ \frac{8}{9} = 1 \end{pmatrix}$

Рис. 5.25. Проверка решений нелинейного уравнения

Решение неравенств. Используя команду *solve* можно решить неравенство или систему неравенств. Это иллюстрируется следующим примером.

Пример 5.3.8. Используя команду *solve*, решить следующее неравенство:

$$\frac{4x^2 + x + 1}{3 - 2x - x^2} < 1.$$

Решение. Фрагмент документа MathCAD показан на рис. 5.26. Найдены три интервала (заданных с использованием знаков отношений) значений величины x , удовлетворяющих заданному неравенству:

$$(-\infty, -3), [-1, 0.4], (1, \infty).$$

$\frac{4 \cdot x^2 + x + 1}{3 - 2 \cdot x - x^2} \leq 1 \text{ solve } ,x \rightarrow 1 < x \vee x < -3 \vee -1 \leq x \leq \frac{2}{5}$
--

Рис. 5.26. Решение неравенства примера 5.3.8

Задания и вопросы для самоконтроля

1. Что такое символьные вычисления в пакете MathCAD?
2. С помощью каких средств интерфейса пакета MathCAD можно осуществить символьные вычисления?
3. Какие символьные преобразования и вычисления осуществляют следующие команды: *evaluate*, *simplify*, *expand*, *factor*, *collect polynomial coefficients*?
4. Упростите приведенные ниже выражения:

a)
$$\frac{x-3}{4 \cdot x^2 + 24 \cdot x + 36} \cdot \left(\frac{x}{3 \cdot x - 9} - \frac{3}{x^2 + 3 \cdot x} + \frac{x^2 + 9}{27 - 3 \cdot x^2} \right)^{-1};$$

b)
$$\frac{3 \cdot x^2 - 2 \cdot x}{6 - 7 \cdot x - 3 \cdot x^2};$$

c)
$$\frac{a - \frac{4 \cdot a - 4}{a}}{\frac{2}{a} - 1}.$$

5. Разложите на множители выражения, приведенные ниже:

a) $x^4 - 10 \cdot x^3 + 35 \cdot x^2 - 50 \cdot x + 24;$

b) $24 \cdot a^3 \cdot b - 6 \cdot a \cdot b \cdot x^3 - 8 \cdot a^3 \cdot x^2 + 2 \cdot a \cdot x^5 - 12 \cdot x \cdot b \cdot a^2 +$
 $+ 3 \cdot x^4 \cdot b + 4 \cdot x^3 \cdot a^2 - x^6.$

6. Разложите на простейшие дроби, приведенное ниже выражение:

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

7. Выполните операцию «Раскрыть скобки и привести подобные» для приведенных ниже выражений:

a) $(2 \cdot x^2 + 3 \cdot x^2 + 4 \cdot x + 1) \cdot (x - 1) \cdot (x - 2) \cdot (x - 3);$

b) $(a \cdot x^2 + 3 \cdot x^2 + 4 \cdot x + 1) \cdot (a \cdot x + b \cdot x^2) - a \cdot x^3.$

8. Определите коэффициенты следующих полиномов:

a) $3 \cdot b \cdot x^4 - \pi \cdot x^2 + \frac{2}{3} \cdot x - 3 \cdot a \cdot b$ относительно переменной x ;

b) $\sin(x) + 2 \cdot \sin(x)^2$ относительно $\sin(x)$.

9. Вычислить решения следующих нелинейных уравнений и выполнить проверку найденных решений:

a) $x^3 - 2.92 \cdot x^2 + 1.4355 \cdot x + 0.791 = 0;$

b) $5^{2 \cdot x - 3} = 5$; c) $\sqrt{x - 1} + \sqrt{2x + 6} = 6.$

10. Разложить в ряд Тейлора с точностью до 7 степени функции:

a) косинуса; b) тангенса.

11. Решите следующие системы нелинейных уравнений:

$$\text{a) } \begin{cases} x \cdot (y + 1) = 12 \\ \frac{x}{y + 1} = 3 \end{cases} ; \quad \text{b) } \begin{cases} \frac{4}{x - y} + \frac{12}{x + y} = 3 \\ \frac{8}{x - y} - \frac{18}{x + y} = -1 \end{cases}.$$

ТЕМА 6. ЧТЕНИЕ И ЗАПИСЬ ДАННЫХ

В этой теме рассматриваются различные средства MathCAD для работы с файлами.

6.1. Средства доступа к файлам

В MathCAD имеется возможность доступа к дисковым файлам для чтения и записи данных. Данные могут быть представлены в различных форматах. MathCAD обеспечивает чтение и запись смешанных текстово-числовых файлов, файлов формата Microsoft Excel, а также чтение и запись файлов двоичного формата, в том числе и графических файлов.

Записывая файлы данных, можно экспортировать результаты MathCAD в текстовые процессоры (например, в Блокнот), в электронные таблицы Excel, в бинарные и другие прикладные программы. Числа в файлах данных могут быть целыми числами или числами с плавающей запятой. В качестве разделителя между целой и дробной частью вещественного числа используется только десятичная точка (*в отличие от таблиц Excel*).

Файлы данных по их структурной организации разделяются на две группы:

- структурированные файлы;
- неструктурированные файлы.

В структурированном файле данные располагаются в виде матрицы, т.е. каждая строка (так называемая запись) имеет одинаковое число элементов. Числа отделяются друг от друга разделителями – пробелами, запятыми, возвратами каретки и другими, задаваемыми пользователем.

В неструктурированном файле данные располагаются либо последовательно (только одна запись), либо в нескольких записях, но с разным числом элементов в них.

В функциях работы с файловыми данными аргументом является *Имя файла*, в качестве которого может выступать:

- строковая константа, содержащая полное имя файла или только имя файла (если он находится в текущем каталоге);

- строковая переменная, получившая значение строковой константы, определяющей имя файла.

Импорт и экспорт данных в MathCAD выполняется двумя способами:

1. С помощью встроенной компоненты – *Мастер Ввода-Вывода файлов*, позволяющей осуществить перенос данных в нужном формате в диалоговом режиме с подсказками.

2. С помощью функций, вводимых в формулы с клавиатуры или вызываемых из *Мастера функций*. Эти функции позволяют устанавливать нужный формат данных, номера считываемых строк и столбцов и выбор разделителей между данными.

Оба способа подразумевают возможность импорта файлов данных самых разных форматов: текстовых с разнообразными символами-разделителями, а также файлы формата xls (Microsoft Excel).

6.2. Мастер Ввода-Вывода файлов

Для ввода и вывода файла используются компоненты MathCAD – *File Input*, *File Output* и *Data Import Wizard*.

Вызов компонент производится из пункта *Data* строки меню *Insert* (см. рис. 6.1).

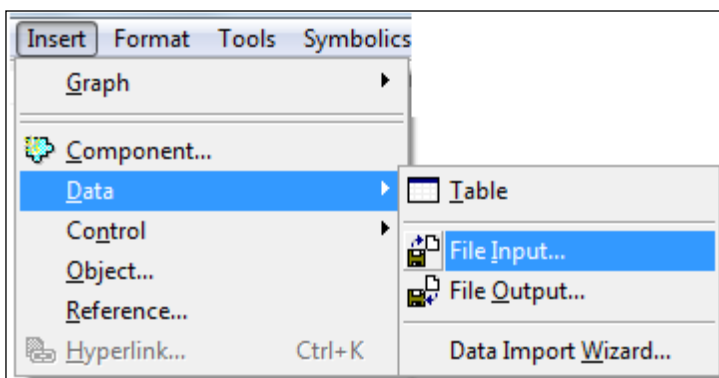


Рис. 6.1. Вызов компонент работы с файлами

Мастер работы с файлами позволяет:

- читать файлы с широким набором форматов – включая бинарные данные, данные фиксированной ширины и данные форматов баз данных;
- просматривать данные;
- выбирать разделители;
- выделять номера строк и столбцов для импорта данных.

Рассмотрим реализацию первой возможности на примере считывания данных из файла.

1. Введите команду меню *Insert / Component (Вставка / Компонент)*, а затем выберите в списке тип компонента *Data Import Wizard (Мастер импорта данных)*. В результате появится окно Мастера, которое в пошаговом диалоговом режиме позволит осуществить считывание нужной информации (рис. 6.2).

2. Выберите в раскрывающемся списке *File format (Формат файла)* желаемый формат файла, из которого вы осуществляете импорт. Если вы затрудняетесь с его точной идентификацией, лучшим решением будет задание типа *Delimited Text (Текст с разделителями)*, что позволит возложить распознавание типа данных и формат их записи на MathCAD.

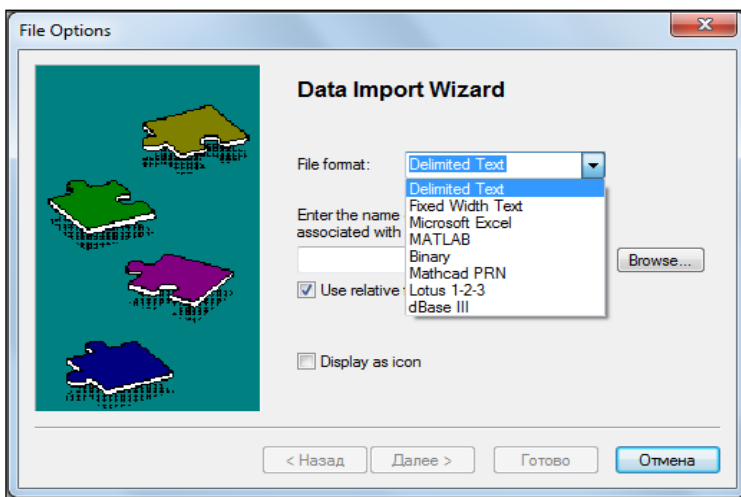


Рис. 6.2. Окно компоненты ввода данных из файла

3. Нажмите кнопку *Browse* (*Пролистать*) и отыщите в открывшемся диалоговом окне местоположение нужного вам файла.

4. Если вы уверены, что данные будут считаны правильно, то можете сразу нажать кнопку *Finish* (*Завершить*). Если вы хотите просмотреть и, при необходимости, изменить те или иные опции импорта, нажмите кнопку *Next* (*Вперед*). В следующем окне (рис. 6.3) в диалоговом режиме можно будет отредактировать параметры импорта (такие как тип разделителя между данными, интервалы импорта и т.д.).

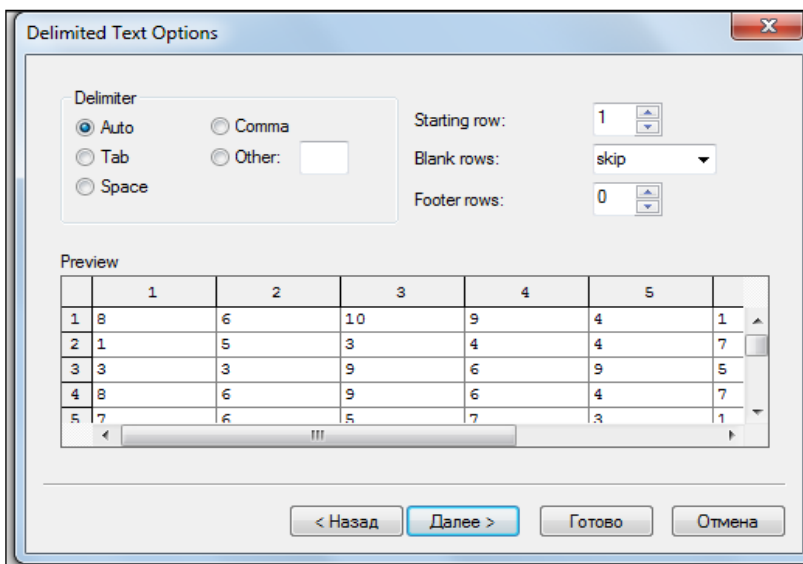


Рис. 6.3. Окно компоненты ввода данных из файла

5. После нажатия кнопки *Finish* (*Завершить*) на рабочую область документа MathCAD переносится содержимое файла в виде таблицы с заголовком (рис. 6.4), содержащим номера столбцов и дополнительным столбцом, содержащим номера строк. Введите

в место-заполнитель, появившийся слева от таблицы импортированных данных, желаемое имя переменной. В дальнейших расчетах ее можно будет использовать как обычную матрицу.

$A :=$

	0	1	2	3	4	5
0	8	6	10	9	4	1
1	1	5	3	4	4	7
2	3	3	9	6	9	5
3	8	6	9	6	4	7
4	7	6	5	7	3	1

$$A = \begin{pmatrix} 8 & 6 & 10 & 9 & 4 & 1 \\ 1 & 5 & 3 & 4 & 4 & 7 \\ 3 & 3 & 9 & 6 & 9 & 5 \\ 8 & 6 & 9 & 6 & 4 & 7 \\ 7 & 6 & 5 & 7 & 3 & 1 \end{pmatrix}$$

Рис. 6.4. Импортированные данные в документе MathCAD

6.3. Функции доступа к структурированным ASCII-файлам

Для чтения и записи структурированных ASCII-данных в MathCAD используются функции READPRN, WRITEPRN и APPENDPRN.

Эти команды считывают целую матрицу из файла со строками и столбцами данных или записывают матрицу из MathCAD в файл, т.е. выполняют действия со структурированными данными.

Форматы функций доступа к файлам:

READPRN(*«file»*) – читает структурированный файл данных. Возвращает матрицу. Каждая строка в файле данных становится строкой в матрице. Число элементов в каждой строке должно быть одинаковым.

WRITEPRN(*«file»*) – записывает матрицу в файл данных. Каждая строка матрицы становится строкой в файле. Если файл содержит какие-либо данные, то они будут заменены новыми.

APPENDPRN(*«file»*) – дописывает матрицу к существующему файлу. Каждая строка в матрице становится новой строкой в файле данных. Существующий файл должен иметь столько же столбцов, как и дописываемая матрица.

Создаваемый в MathCAD файл данных можно просмотреть и отредактировать в текстовом редакторе, отображающем ASCII-данные. Примерами таких редакторов может служить программа Блокнот, входящая в состав Windows, программная оболочка FAR и другие.

Параметр *«file»* – полное имя файла, которое указывается в кавычках. Если не указано расширение, то по умолчанию подразумевается расширение *.pm*.

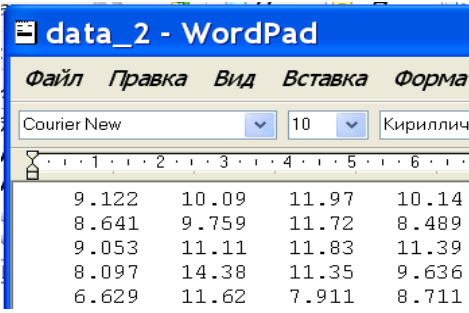
Пример 6.3.1. Записать в файл матрицу B , k -й столбец которой есть выборка из нормального распределения с математическим ожиданием $\mu_0 = 10$ и среднеквадратическим отклонением $\sigma_0 = 2$ (дисперсия, соответственно, $\sigma_0^2 = 4$).

Решение. Фрагмент документа, осуществляющий запись структурированного файла, приведен на рис. 6.5. В окне программы Блокнот хорошо видна структура сформированного файла (число строк $n = 5$, число столбцов $m = 4$).

```

ORIGIN := 1
μ0 := 10   σ0 := 2
n := 5     m := 4
k := 1..m
B(k) := rnorm(n, μ0, σ0)
file2 := "data_2"
WRITEPRN(file2) := B(k)

```



9.122	10.09	11.97	10.14
8.641	9.759	11.72	8.489
9.053	11.11	11.83	11.39
8.097	14.38	11.35	9.636
6.629	11.62	7.911	8.711

Рис. 6.5. Запись структурированного файла

Пример 6.3.2. К матрице B , созданной и записанной в файл (см. приме 6.3.1), дописать еще заданную матрицу C размером 3×4 . Прочитать из файла новую матрицу (обозначим как D) и вывести ее элементы.

Решение. Фрагмент документа, осуществляющий «дозапись» матрицы C в структурированный файл, приведен на рис. 6.6. Здесь же выведены элементы новой матрицы. Заметим, что в операторе чтение имя файла содержит явно заданное расширение *prn*. Если расширение не задать, файл не откроется для чтения.

$$\begin{array}{l}
 C := \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix} \\
 APPENDPRN (file2) := C \\
 D := READPRN ("D:\data_2.prn") \quad D = \begin{pmatrix} 9.122 & 10.09 & 11.97 & 10.14 \\ 8.641 & 9.759 & 11.72 & 8.489 \\ 9.053 & 11.11 & 11.83 & 11.39 \\ 8.097 & 14.38 & 11.35 & 9.636 \\ 6.629 & 11.62 & 7.911 & 8.711 \\ 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix}
 \end{array}$$

Рис. 6.6. Дополнительная запись и чтение структурированного файла

6.4. Функции доступа к графическим файлам

Функции MathCAD, показанные в табл. 6.1, позволяют читать и записывать файлы изображений в различных форматах.

Таблица 6.1

<i>READ_IMAGE</i> («file»)	Создает массив, содержащий оттенки серого представляющий файл изображения
<i>READBMP</i> («file»)	Создает массив, содержащий представление серого растровых изображений файлов
<i>READRGB</i> («file»)	Создает массив, содержащий красный, зеленый, синий упакованное матричное представление файла изображения
<i>READ_RED</i> («file»), <i>READ_GREEN</i> («file»), <i>READ_BLUE</i> («file»)	Создают массив, содержащий только красный, зеленый или синий компонент из цветного файла образа
<i>WRITEBMP</i> («file», [M])	Создает файл серого растровых изображений
<i>WRITERGB</i> («file», [M])	Создает файл растрового изображения 16 миллионов цветов
<i>READ_HLS</i> («file»)	Создает массив, содержащий тона яркости и насыщенности для представления файла изображения
<i>READ_HLS_HUE</i> («file»), <i>READ_HLS_LIGHT</i> («file»), <i>READ_HLS_SAT</i> («file»)	Извлекает только оттенок, яркость или насыщенность из цветного изображения
<i>READ_HSV</i> («file»)	Создает массив, содержащий оттенок, насыщенность и значения файла изображения
<i>READ_HSV_HUE</i> («file»), <i>READ_HSV_SAT</i> («file»), <i>READ_HSV_VALUE</i> («file»)	Извлекают только оттенок, насыщенность или компоненту значений из цветного изображения
<i>WRITE_HSV</i> («file», [M])	Создает файл растрового изображения 16 миллионов цветов
<i>WRITE_HLS</i> («file», [M])	Создает файл растрового изображения 16 миллионов цветов

Аргументы функций:

- *«file»* – строка содержит имя файла или полный путь к файлу и имя файла. Признанные графические форматы – BMP, GIF, JPG, PCX или TGA;

- M – матрица.

Отметим следующие особенности:

- Имена функций вводятся прописными символами.

- Независимо от исходного формата файла изображения, изображения в MathCAD являются матрицами со значениями от 0 (черный) до 255 (белый), которые могут быть отображены с помощью операторов меню *Insert > Picture*. Каждый элемент матрицы представляет интенсивность одного пиксела. Цвета RGB изображения хранятся как три упакованные $m \times n$ блок о блок матрицы (функция *augment*):

$$\mathbf{M} = \begin{bmatrix} \text{red} & \text{red} & \text{red} & \text{green} & \text{green} & \text{green} & \text{blue} & \text{blue} & \text{blue} \\ \text{red} & \text{red} & \text{red} & \text{green} & \text{green} & \text{green} & \text{blue} & \text{blue} & \text{blue} \\ \text{red} & \text{red} & \text{red} & \text{green} & \text{green} & \text{green} & \text{blue} & \text{blue} & \text{blue} \\ \text{red} & \text{red} & \text{red} & \text{green} & \text{green} & \text{green} & \text{blue} & \text{blue} & \text{blue} \end{bmatrix}$$

$\mathbf{I} = \begin{bmatrix} \text{red} & \text{green} & \text{blue} \end{bmatrix}$

Первые n столбцов представляют красную компоненту $n \times t$ изображения. Во втором и третьем наборах n столбцов представляют зеленый и синий компоненты изображения. HSV или HLS изображения хранятся в упакованном формате.

Пример 6.4.1. Считывание и обработка ВМР файла.

1. Считаем с диска данные графического файла в MathCAD с помощью команды READBMP(). Данные сформируют матрицу M оттенков серого цвета размерностью 255×255 .

2. Будем обрабатывать только центральную часть изображения размерностью 100×100 . Для этого функцией *submatrix()* выделим из матрицы *M* подматрицу *S*.

3. Выведем изображение, представленное матрицей S , с помощью компоненты строки меню *Insert/Picture*. В появившемся шаблоне введем S в место-заполнитель.

Результат выполненных действий показан на рис. 6.7.

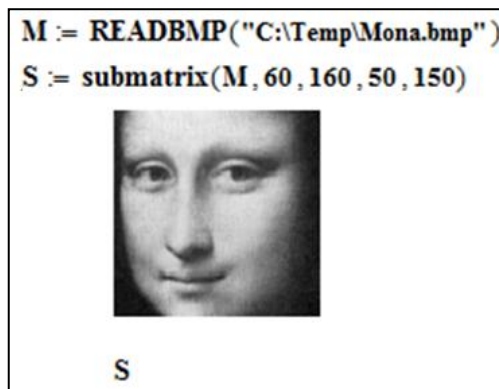


Рис. 6.7. Чтение графического файла

Пример 6.4.2. Продолжим обработку графического файла, прочитанного в примере 6.4.1.

1. Изменим уровень интенсивности серого оттенка в каждом пикселе, разделив матрицу S на число 2 и добавив к результату число 120. Получим матрицу $S1$. Выведем полученное изображение.

2. Создадим инверсию графического изображения в матрице $S2$. Для этого значение интенсивности в каждом элементе матрицы $S2$ вычислим как $S2_{ij} = 255 - S_{ij}$. Выведем полученное изображение, представленное матрицей $S2$.

3. Поменяем строки матрицы так, чтобы номера строк шли по убыванию (функция *reverse()*) – матрица $S3$. Изображение должно перевернуться.

4. Проведем контрастирование изображения. Для этого все значения ниже 120 переназначим в 0, а выше в 255, т.е. оттенки будут только черные и белые. Преобразованные с помощью функции $f(x)$ значения записаны в матрицу $S4$.

Результаты выполненных преобразований показаны на рис. 6.8.

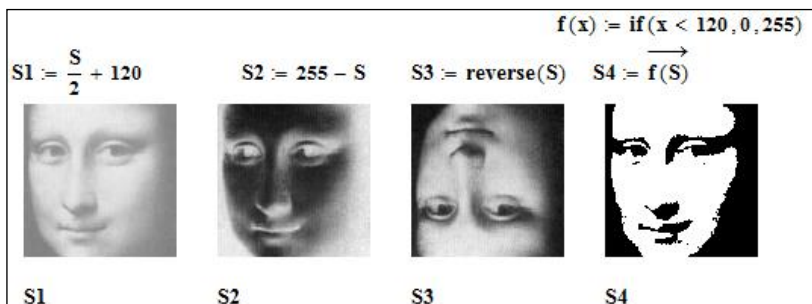


Рис. 6.8. Примеры обработки графического файла

Вопросы и задания для самопроверки

1. Какой файл является структурированным?
2. Какой файл называется ASCII-файлом?
3. К каким типам файлов обеспечивается доступ в MathCAD?
4. Какая компонента в MathCAD выполняет импорт и экспорт данных в файлы?
5. Какие форматы графических файлов поддерживают функции в MathCAD?
6. Какие значения имеют элементы массива, содержащего представление серого в растровых изображений файлов? Что означают эти значения?
7. Составить фрагмент документа MathCAD, формирующий ASCII-файл, содержащий два столбца:
 - первый столбец содержит значения $x_i, i = 0, 1, \dots, n$ определяемые по формуле:

$$x_i = a + \frac{b-a}{n};$$

- второй столбец содержит значения функции

$$\psi(x_i) = e^{-x_i^2} \cdot \cos(5x_i), \quad i = 0, 1, \dots, n.$$

8. Обработка графического файла. Выполните следующие действия:

- Зайдите в MathCAD\Nahdbook\improcl и по указанию преподавателя выберите BMP-файл. Перепишите его в C:\TEMP.
- считайте его данные в матрицу документа MathCAD. Определите размер полученной матрицы.
- Выделите из нее подматрицу.
- Отобразите полученное графическое изображение.
- С помощью матричных операций создайте негативное изображение и черно-белое изображение.
- Склейте два BMP-файла бок о бок и снизу-сверху.
- Отобразите полученные изображения.

РАЗДЕЛ 2. ПРОГРАММИРОВАНИЕ В ПАКЕТЕ MATHCAD

В этом разделе рассматриваются конструкции пакета MathCAD, позволяющие реализовать следующие типы алгоритмов: линейный, разветвляющийся и циклический (проще – цикл). При этом будут изучены два способа программирования:

- программирование в пакете MathCAD без использования программных модулей;
- программирование с использованием программных модулей.

Первый способ (в дальнейшем для простоты названный безмодульным программированием) реализуется записью соответствующих конструкций непосредственно в математических областях документа MathCAD, и он приемлем для сравнительно простых алгоритмов.

Второй способ (называемый для простоты модульным программированием) предполагает реализацию отдельных независимых алгоритмов вычисления (например, решение нелинейного уравнения методом «деления отрезка пополам») в виде отдельных программных модулей, которые будем называть подпрограммами-функциями (сокращенно П-Ф). Первое слово «подпрограмма» указывает на свойство «изолированности» этого модуля от других вычислений в документах MathCAD, а второе слово «функция» – на способ вызова модуля и механизмы передачи вычисленных в модуле значений.

Заметим, что принцип модульного программирования в свое время (70–80 годы XX века) существенно повысил производительность труда программистов, разрабатывающих программы с использованием алгоритмических языков высокого уровня (подробнее см. п. 8.1). Применение его в пакете MathCAD позволяет:

- «распараллелить» разработку программы между несколькими исполнителями;
- создать проблемно-ориентированные библиотеки П-Ф для решения научно-технических задач с размещением библиотек на сайтах Интернета;
- уменьшить затраты на разработку и сопровождение программ для пакета MathCAD.

ТЕМА 7. БЕЗМОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ В ПАКЕТЕ MATHCAD

Рассматриваются конструкции пакета, позволяющие реализовать линейный, разветвляющийся и циклический алгоритмы непосредственно в математических областях документа MathCAD.

7.1. Программирование линейных алгоритмов

Характерной особенностью линейных алгоритмов является *строго последовательное выполнение всех операций алгоритма* без пропусков и повторов вычислений. Поэтому конструкции, реализующие такой алгоритм, записываются в документе MathCAD в нужном порядке их выполнения, т.е. «слева направо – сверху вниз».

Пример 7.1.1. Составить программу для вычисления корней квадратного уравнения: $ax^2 + bx + c = 0$ по известной формуле:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \quad (7.1.1)$$

Алгоритм (7.1.1) является линейным (убедитесь в этом), и фрагмент документа MathCAD содержит конструкции, приведенные на рис. 7.1.

$a := 2$	$b := 5$	$c := 8$	$d := \sqrt{b^2 - 4 \cdot a \cdot c}$
$x_1 := \frac{-b + d}{2 \cdot a}$	$x_2 := \frac{-b - d}{2 \cdot a}$		
$x_1 = -1.25 + 1.561i$	$x_2 = -1.25 - 1.561i$		
Проверка найденных корней			
$a \cdot x_1^2 + b \cdot x_1 + c = 0.00000$	$a \cdot x_2^2 + b \cdot x_2 + c = 0.00000$		

Рис. 7.1. Пример программирования линейного алгоритма

В этом фрагменте x_1 , x_2 являются простыми переменными, цифры 1, 2 являются нижними индексами в именах (а не индексными выражениями у элементов массива) и поэтому вводятся после нажатия клавиши [.] – «десятичная точка».

Задание 7.1.1. Подобрать коэффициенты a , b , c , получить две пары вещественных корней уравнения и две пары комплексных корней. Выполнить их проверку.

Очевидно, что в реализации линейного алгоритма могут использоваться обращения к встроенным функциям MathCAD и функциями пользователя.

Задание 7.1.2. Составить программу вычисления площади треугольника по формуле: $S = \sqrt{p(p-a)(p-b)(p-c)}$, где $p = \frac{a+b+c}{2}$ – полупериметр; a , b , c – стороны треугольника.

Исходные данные: $a = 1.6$; $b = 2.03$; $c = 0.5$.

Задание 7.1.3. Составить программу вычисления величины z по следующей формуле:

$$z(x, y) = \left(\frac{ay}{a+b} + \frac{c}{ax^2 + bx} \right)^3 + \sin yt,$$

где $y = \left(\sqrt{\frac{ax+b}{c+dx}} + \sqrt{\arctg x} \right)^{2/3} - e^{2x}$. Определить какие переменные являются исходными и задать их значения.

7.2. Программирование разветвляющихся алгоритмов

Характерной чертой разветвляющихся алгоритмов является наличие в них нескольких возможных ветвей вычислений. Выбор конкретной ветви зависит от выполнения (или невыполнения) заданных условий на значения переменных алгоритма.

Пример 7.2.1. Значение переменной y зависит от значений переменной x и определяется выражением:

$$y = \begin{cases} x^2, & \text{если } x \leq 0; \\ \sqrt{x}, & \text{в противном случае.} \end{cases} \quad (7.2.1)$$

Выбор одной из двух ветвей вычислений определяется текущим x . На рис. 7.2. представлена блок-схема этого алгоритма, хорошо подтверждающая название алгоритма – «разветвляющийся».

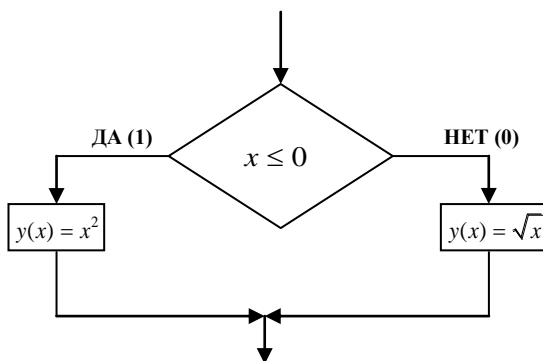


Рис. 7.2. Блок-схема разветвляющегося алгоритма (7.2.1)

Возникает вопрос: какие конструкции необходимы для реализации разветвляющегося алгоритма? Анализ алгоритма (7.2.1) и «программистская интуиция» подсказывают необходимость использования:

- конструкций, проверяющих выполнение заданных условий (чаще гораздо более сложных, чем условие алгоритма (7.2.1));
- конструкций, выбирающих нужную ветвь вычислений в зависимости от результатов проверки заданных условий.

Для проверки заданных условий в MathCAD используется: *выражение отношений, логические операции и логические выражения.*

Выражением отношений (или просто *отношением*) называется конструкция вида:

$\langle \text{выр.1} \rangle \langle \text{операция отношения} \rangle \langle \text{выр.2} \rangle$,

где $\langle \text{выр.1} \rangle$, $\langle \text{выр.2} \rangle$ – произвольные арифметические выражения, $\langle \text{операция отношения} \rangle$ – любая из следующих операций: $\left| \left| \left| \left| \left| \left| \right| \right| \right| \right| \right| \right|$ (здесь вертикальные черточки являются разделительным символом при перечислении).

Смысл этих операций понятен и не нуждается в пояснении. Для ввода знаков операций отношений можно использовать палитру **Логический** (приведенную на рис. 7.3) или использовать клавиши, обозначения которых приведены в табл. 7.1.

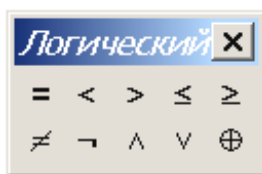


Рис. 7.3. Палитра инструментов **Логический**

Внимание! Не следует путать знак операции сравнения $=$ с похожим знаком вывода значений переменных. Знак операции $=$ имеет больший размер и более жирное начертание.

Таблица 7.1

Знаки операции	Клавиши
$<$	[<]
\leq	[Ctrl] + [9]
$>$	[>]
\geq	[Ctrl] + [0]
$=$	[Ctrl] + [=]
\neq	[Ctrl] + [3]

Выражение отношений принимает одно из двух значений: 1 – если заданное отношение выполняется, 0 – в противном случае. Значение 1 можно интерпретировать как значение ИСТИНА, а 0 – как ЛОЖЬ.

Пример 7.2.1. На рис. 7.4 приведен фрагмент документа, в котором выполнено вычисление некоторых выражений отношений.

Внимание! Использование оператора вывода позволяет сразу увидеть результат вычисления.

$x := 3$	$y := -2$	$z := 6$
$x > y = 1$	$x - 2 \cdot y \geq 3 \cdot z = 0$	
$0 \leq x \leq 5 = 1$	$-4 \leq x - y \leq 3 = 0$	

Рис. 7.4. Примеры записи выражений отношений

Задание 7.2.1. Пусть значение целой переменной $x = 7$. Определить значение следующих выражений отношений:

а) $x \geq 4$; б) $x + 1 \geq 4$; в) $x - 4 > 1$.

Для проверки более сложных условий используются *четыре логические операции*, обозначения которых приведены в табл. 7.2.

Таблица 7.2

Название операции	Знак
Логическое отрицание (NOT)	\neg
Логическое ИЛИ (OR)	\vee
Логическое И (AND)	\wedge
Исключающее ИЛИ (XOR)	\oplus

Знаки этих операций вводятся с палитры **ЛОГИЧЕСКИЙ**.
Результат выполнения этих операций приведен в табл. 7.3.

Таблица 7.3

NOT \neg	AND \wedge	OR \vee	XOR \oplus
$0 \neg = 1$	$0 \wedge 0 = 0$	$0 \vee 0 = 0$	$0 \oplus 0 = 0$
$1 \neg = 0$	$0 \wedge 1 = 0$	$0 \vee 1 = 1$	$1 \oplus 0 = 1$
	$1 \wedge 0 = 0$	$1 \vee 0 = 1$	$0 \oplus 1 = 1$
	$1 \wedge 1 = 1$	$1 \vee 1 = 1$	$1 \oplus 1 = 0$

Логическим выражением называется конструкция, состоящая из выражений отношений, логических операций и круглых скобок. Логическое выражение принимает только одно из двух значений: 1 или 0; вычисляется слева направо с учетом приоритета входящих в выражение операций. Наивысший приоритет – круглые скобки, а затем по убыванию: AND, OR и XOR – одинаковый приоритет и самый низкий приоритет выражения отношений.

Замечание 7.2.1. Поскольку выражения отношений имеют самый низкий приоритет, то *их необходимо заключать в круглые скобки*.

Задание 7.2.2. Определите порядок вычисления значений логических выражений в документе MathCAD, приведенных на рис. 7.5.

$$x := 0.2 \quad y := -4$$

$$(x > 3) \wedge (y > 0) = \blacksquare$$

$$\neg(x > 3) \wedge (y < 0) = \blacksquare$$

Рис. 7.5. Примеры логических выражений

В MathCAD имеется ряд встроенных функций, которые возвращают результат, зависящий от знака или величины аргу-

мента, и могут использоваться при программировании разветвляющихся алгоритмов. Приведем некоторые из них:

- ***ceil* (x)** – наименьшее целое, большее или равное x ;
- ***trunc* (x)** – целая часть вещественного числа x ;
- ***floor* (x)** – наибольшее целое, меньшее или равное x ;
- ***round* (x, n)** – округленное значение вещественного x с точностью до n знаков после десятичной точки;

- **$\Phi(x)$** – функция Хевисайда – равна 0 при $x < 0$ и 1 в противном случае;

- ***sign* (x)** – функция знака (равна 0 если $x = 0$; -1 , если $x < 0$ и 1 , если $x > 0$);

- ***signum* (x)** – возвращает 1, если $x = 0$ и $\frac{x}{|x|}$ в остальных случаях.

Пример 7.2.3. Используя функцию Хэвисайда $\Phi(x)$, запрограммировать вычисление функции $f(x)$, график которой показан на рис. 7.6а.

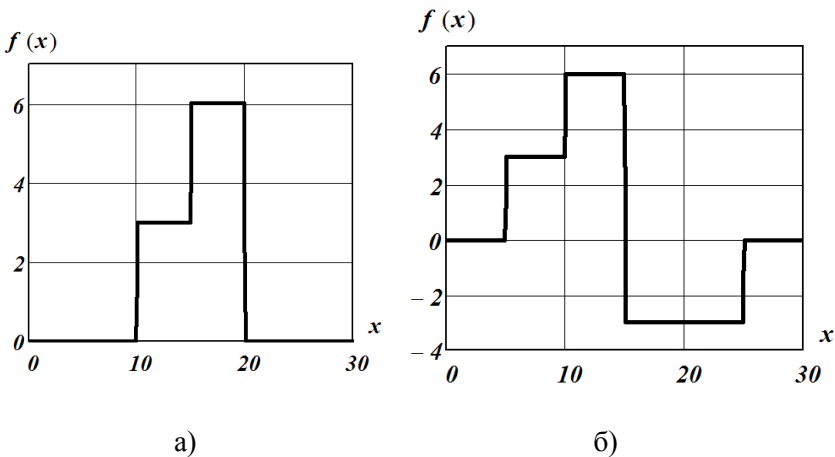


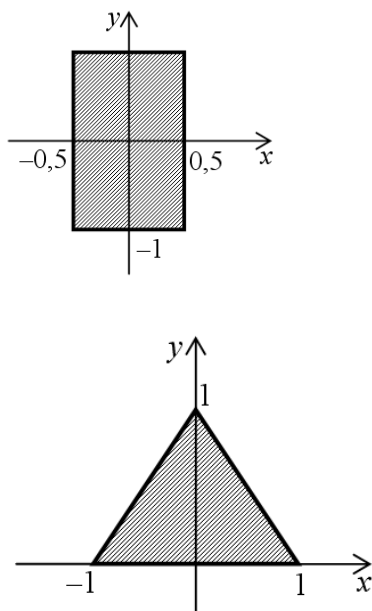
Рис. 7.6. Графики функций к примеру 7.2.3 и заданию 7.2.3

Решение показано в следующем фрагменте документа MathCAD.

$$f(x) := 3 \cdot (\Phi(x - 10) + \Phi(x - 15) - 2 \cdot \Phi(x - 20))$$

Задание 7.2.3. Используя функцию Хэвисайда $\Phi(x)$, запрограммировать вычисление функции $f(x)$, график которой показан на рис. 7.6б.

Пример 7.2.4. Написать логическое выражение принимающее значение 1, при попадании точки с координатами (x, y) в заштрихованные области плоскости, показанные на рис. 7.7а.



а)

б)

Рис. 7.7. Изображение областей к примеру 7.2.4

Запрограммированные алгоритмы вычислений в двух вариантах (логические выражения и функции пользователя) приведены в следующем фрагменте документа MathCAD.

$$\begin{array}{l} x := 0.2 \quad y := -0.8 \\ (-0.5 \leq x \leq 0.5) \wedge (-1 \leq y \leq 1) = 1 \\ f(x, y) := (-0.5 \leq x \leq 0.5) \wedge (-1 \leq y \leq 1) \quad f(-1, 0.3) = 0 \end{array}$$

Задание 7.2.4. Написать логическое выражение, принимающее значение 1 при попадании точки с координатами (x, y) в заштрихованные области плоскости, показанные на рис. 7.7б.

Для выбора нужной ветви разветвляющегося алгоритма используется конструкция, названная *условной функцией if*, записываемая в виде:

if (<логическое выражение>, <выр. 1>, <выр. 2>),

где имя функции *if* вводится с клавиатуры. Если логическое выражение равно 1, то значение функции определяется *выр. 1*, в противном случае – *выр. 2*. Блок-схема этой функции приведена на рис. 7.8.

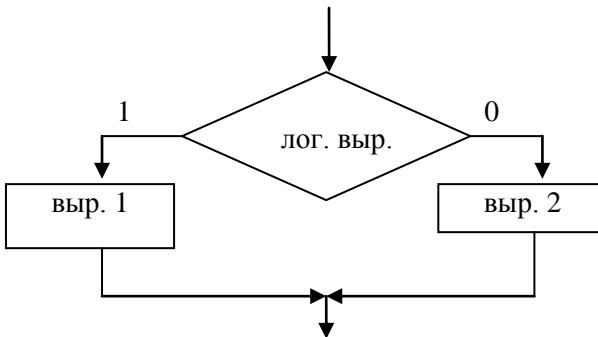


Рис. 7.8. Блок-схема функции *if*

Условная функций *if* реализует логическую структуру, которая в программировании называется «**ЕСЛИ-ТО-ИНАЧЕ**».

Пример 7.2.5. Используя условную функцию *if*, вычислить значение переменной *y*, определяемой следующим выражением:

$$y = \begin{cases} \ln x, & \text{если } x > 0, \\ e^x, & \text{если } x \leq 0. \end{cases}$$

Решение. Блок-схема алгоритма показана на рис. 7.9. Условные сокращения: Д – да; Н – нет. Фрагмент документа MathCAD, реализующий этот разветвляющийся алгоритм, приводится ниже. Здесь же выполнено тестирование запрограммированной функции.

Замечание. Стрелка в нижней части блок-схемы, направленная вниз, обозначает, что результат работы разветвляющегося алгоритма будет использоваться ниже (это может быть оператор вывода или другие конструкции MathCAD).

$y(x) := \text{if}(x > 0, \ln(x), e^x)$	
$y(-3) = 0.05$	$y(2.71) = 0.997$

При программировании разветвляющихся алгоритмов с тремя и более вычислительными ветвями в качестве *выр. 1* и *выр. 2* вновь может использоваться функция *if*. Такая структура называется вложенными условными функциями.

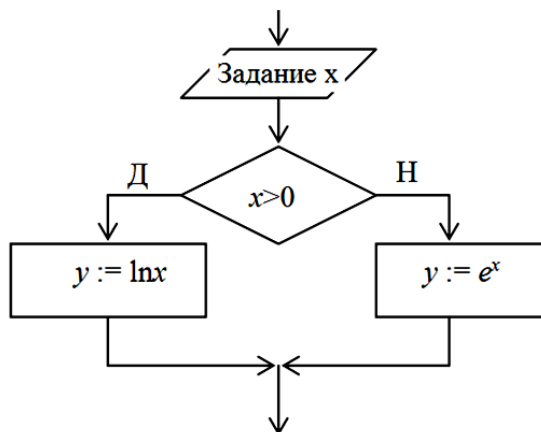


Рис. 7.9. Блок-схема разветвляющегося алгоритма примера 7.2.5

Пример 7.2.6. Вычислить значение функции $y(x)$, задаваемой следующим разветвляющимся алгоритмом, состоящим из трех ветвей:

$$y = \begin{cases} \frac{1}{2}\sqrt{x}, & \text{если } x \geq 1, \\ \frac{1}{3}\sqrt[3]{x}, & \text{если } 0 < x < 1, \\ \frac{1}{4}\sqrt[4]{|x|}, & \text{если } x \leq 0. \end{cases}$$

Решение. Блок-схема алгоритма приведена на рис. 7.10. Так как алгоритм содержит три вычислительных ветви, то будем использовать вложенные условные функции, как показано в следующем фрагменте документа MathCAD.

$$y(x) := \text{if}\left(x \geq 1, \frac{1}{2} \cdot \sqrt{x}, \text{if}\left(x > 0, \frac{1}{3} \cdot \sqrt[3]{x}, \frac{1}{4} \cdot \sqrt[4]{|x|}\right)\right)$$

$$y(2) = 0.707 \quad y(0.5) = 0.265 \quad y(-2) = 0.297$$

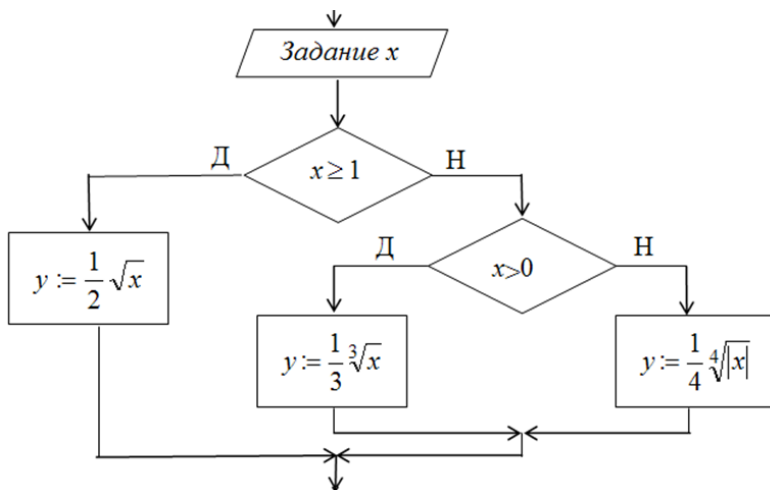


Рис. 7.10. Блок-схема разветвляющегося алгоритма примера 7.2.6

Задание 7.2.5. Используя условную функцию *if*, запрограммировать следующий разветвляющийся алгоритм:

$$z(x) = \begin{cases} 30, & \text{если } x \leq -1; \\ |x|, & \text{если } -1 < x \leq 1; \\ x^2 - 30, & \text{если } x > 1. \end{cases}$$

Пример 7.2.7. Запрограммировать алгоритм, вычисляющий величину x , по следующему правилу: если $x < 2$, то значение x оставить без изменения, в противном случае величину x увеличить на 2.

Решение. Блок-схема алгоритма приведена на рис. 7.11. Изображенный алгоритм относится к логической структуре «ЕСЛИ-ТО». В таком алгоритме выполняется только одна ветка (если заданное логическое выражение равно 1).

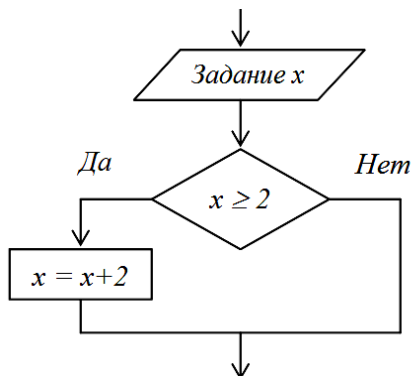


Рис. 7.11. Блок-схема структуры «ЕСЛИ-ТО» (к примеру 7.2.6)

Возникает вопрос: как запрограммировать структуру «ЕСЛИ-ТО» в документе MathCAD. Пример реализации структуры «ЕСЛИ-ТО» демонстрируется в следующем фрагменте документа MathCAD:

$x := 3$ $x := \text{if}(x \geq 2, x + 2, x) \qquad x = 5$
--

Видно, что при $x < 2$ переменная x не меняет своего значения.

7.3. Программирование циклических алгоритмов

Циклическим алгоритмом (или просто *циклом*) называется алгоритм, содержащий вычисления, повторяющиеся при различных значениях некоторой переменной, названной параметром цикла, а сами повторяющиеся вычисления составляют тело цикла. Обобщенная блок-схема циклического алгоритма представлена на рис. 7.12. Для правильной организации цикла необходимы четыре блока: номер 1 (см. рис. 7.12) – подготовка к циклу (обычно задание начального значения параметру цикла, 2 – тело цикла (содержит повторяющиеся вычисления); 3 – подготовка к повторению цикла (меняется значение параметра цикла); 4 – проверка условия окончания или условия продолжения цикла.

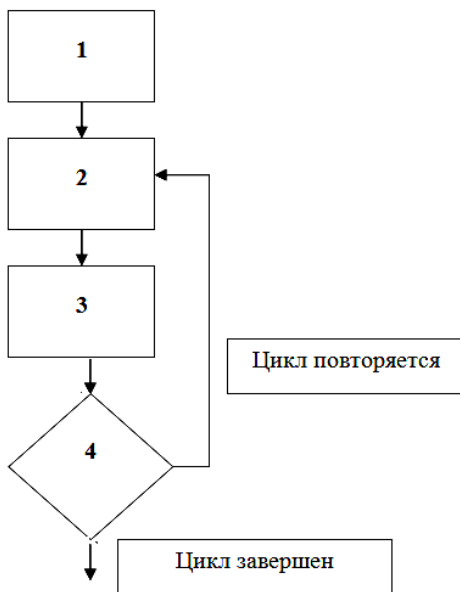


Рис. 7.12. Обобщенная блок-схема циклического алгоритма

Если цикл не завершен, то вновь повторяется тело цикла, но при измененном (в блоке 3) значении параметра цикла. В ряде алгоритмов блоки 3, 4 могут быть переставлены местами. **Заметим, что в программных реализациях блоки 1, 3, 4 могут быть объединены в одной конструкции.**

Типы циклов. По способам организации цикла можно выделить:

- а) *цикл типа арифметической прогрессии;*
- б) *итерационный цикл.*

Особенностью цикла типа арифметической прогрессии является изменение параметра цикла по закону арифметической прогрессии (т.е. от начального значения (x_0) до конечного (x_k) с заданным шагом (d_x)). В этом случае априори (т.е. до выполнения цикла) можно определить количество n повторений тела цикла по формуле:

$$n = \left[\frac{x_k - x_0}{d_x} \right] + 1, \quad (7.3.1)$$

где $[z]$ – означает целую часть вещественной величины z .

Пример 7.3.1. Сформировать вектор z из n элементов, определяемых соотношением:

$$z_i = \frac{1}{(i + 4)}, \quad i = 1, \dots, n; \quad n = 10.$$

В этом примере i – параметр цикла, а тело цикла (вычисление элементов z_i будет повторено $n = 10$ раз, т.е. алгоритм является циклом типа арифметической прогрессии.

В итерационных циклах параметр меняется по более сложному закону и поэтому невозможно определить количество повторений тела цикла, не выполняя вычислений. Итерационные циклы используются в таких вычислительных алгоритмах, в которых вместо точного решения рассматриваемой задачи ищется приближенное решение, вычисляемое последовательно. При этом чем больше итераций (т.е. повторений тела цикла), тем с большей точностью получаем приближенное решение.

Пример 7.3.2. Вычислить приближенное значение \sqrt{a} , используя итерационную процедуру:

$$x_{n+1} = 0.5 \cdot \left(x_n + \frac{a}{x_n} \right), \quad n = 0, 1, \dots; \quad x_0 = a. \quad (7.3.2)$$

На первой итерации в правую часть формулы (7.3.2) подставляется x_0 и вычисляется x_1 . На второй итерации подставляется уже x_1 , а вычисляется x_2 и т.д. В качестве приближенного значения принимают такое значение x_{n+1} , для которого выполняется условие: $|x_{n+1}^2 - a| \leq \varepsilon$, где $\varepsilon > 0$ достаточно малая величина (порядка $10^{-8} \div 10^{-6}$).

На рис. 7.13 показан график зависимости погрешности $\varepsilon_n = |x_n - \sqrt{a}|$ при $a = 9$ от числа итераций n . Из графика видно,

что погрешность вычисления с каждой итерацией существенно (на два и более порядка) уменьшается. В этом случае говорят о сильной сходимости итерационного процесса.

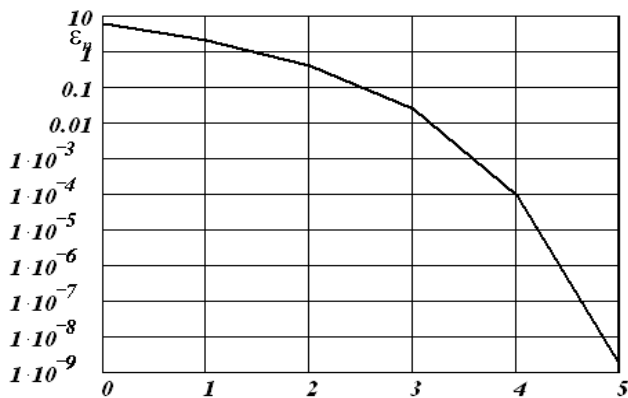


Рис. 7.13. График сходимости итерационного процесса (7.3.1)

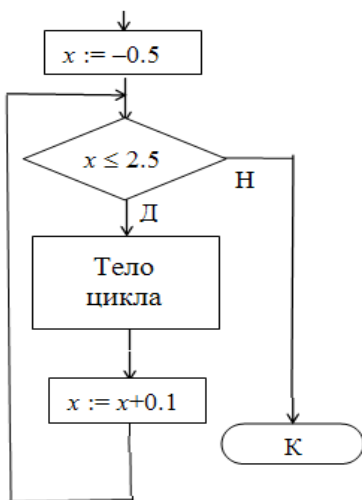
Программирование цикла типа арифметической прогрессии. Параметр такого цикла задается дискретной переменной (называемой также ранжированной переменной), и тогда конструкции, входящие в тело цикла, располагаются, начиная от этого описания и до конца документа MathCAD или до конструкции, переопределяющей дискретную переменную – параметр цикла.

Пример 7.3.3. Вычислить значения функции

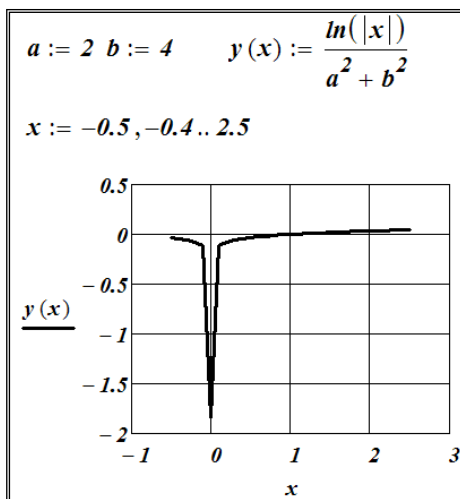
$$y(x) = \frac{\ln|x|}{a^2 + b^2}$$

и построить график этой функции для всех x , изменяющихся в интервале $[-0.5, 2.5]$ с шагом $\Delta x = 0.1$. a, b – заданные вещественные числа.

Решение. В данной задаче переменная x является управляющей переменной цикла. На рис. 7.14а приведена блок-схема изменения значений этой переменной.



а)



б)

Рис. 7.14. Блок-схема вычисления функции и ее график

Условные сокращения: Д – да; Н – нет; К – конец цикла. В программной реализации (см. рис. 7.14б) параметр цикла задается дискретной переменной, которая объединяет функции блоков 1, 3, 4 (см. рис 7.12). Тело цикла начинается после задания дискретной переменной и включает вычисление очередного значения функции $y(x)$ и построение графика. Описание функции пользователя $y(x)$ и задание переменных a, b размещаются в документе MathCAD до начала цикла.

Фрагмент документа, в котором реализован алгоритм решения задачи примера 7.3.1, приведен на рис. 7.15. Здесь параметром цикла является переменная i , которая задается дискретной переменной.

$n := 10 \qquad i := 1..10$ $z_i := \frac{1}{i + 4}$											
$z^T =$	1	2	3	4	5	6	7	8	9	10	
	1	0.2	0.167	0.143	0.125	0.111	0.1	0.091	0.083	0.077	0.071

Рис. 7.15. Формирование вектора z примера 7.3.1

Пример 7.3.4. Для переменной x , меняющейся от 1 до 2 с шагом 0,1 сформировать вектор q из соответствующих значений функции $y(x) = x^2 + 1$, т.е.

$$q_1 = y(1.0), q_2 = y(1.1), \dots, q_n = y(2.0).$$

Решение. В этом циклическом алгоритме присутствуют две переменные, которые могут быть взяты в качестве параметра цикла: номер j проекции q_j формируемого вектора и значение аргумента x . Примем в качестве параметра цикла переменную j , которая должна меняться от 1 до n с шагом 1, а в теле цикла значения аргумента x будем вычислять по формуле $x_j = 1 + (j - 1) \cdot 0.1$. Для вычисления неизвестного значения n обратимся к формуле (7.3.1) и получим $n = 11$. Фрагмент докумен-

та, реализующий рассмотренный алгоритм формирования вектора q приводится ниже.

$$\begin{aligned} \text{ORIGIN} &:= 1 & n &:= \text{trunc}\left(\frac{2-1}{0.1}\right) + 1 = 11 \\ j &:= 1..n \\ x_j &:= 1 + (j-1) \cdot 0.1 & q_j &:= (x_j)^2 + 1 \\ q^T &= (2 \quad 2.21 \quad 2.44 \quad 2.69 \quad 2.96 \quad 3.25 \quad 3.56 \quad 3.89 \quad 4.24 \quad 4.61 \quad 5) \end{aligned}$$

Замечание. По условиям задачи нет необходимости формировать массив из изменяющихся от 1 до 2 значений аргумента x . Однако безмодульное программирование циклов в MathCAD содержит требование: если в правой части оператора присваивания присутствует параметр цикла, то в левой части должен стоять элемент массива с таким же параметром. Этим объясняется необходимость формирования массива x .

Циклы, рассмотренные в этих примерах, относятся к *простым циклам*, в которых присутствует только один параметр цикла. В *кратных циклах* один цикл (внутренний) вкладывается в тело другого цикла, называемого внешним циклом. Поэтому в кратных циклах присутствуют несколько параметров циклов.

Пример 7.3.5. Сформировать матрицу (двумерный массив) В по следующему правилу:

$$B_{i,j} = \frac{1}{i+j+1}; \quad i=1, \dots, 4; \quad j=1, \dots, 6$$

Решение. Очевидно, что формирование матрицы надо осуществлять в цикле с двумя параметрами: i – параметр, определяющий номер строки; j – параметр, определяющий номер столбца матрицы. Цикл с двумя параметрами называется *двойным* циклом. В двойном цикле вложенный цикл называется

внутренним. Параметр внутреннего цикла всегда меняется быстрее, чем параметр внешнего. Это означает, что для каждого значения внешнего параметра цикла меняются все значения внутреннего параметра.

Примем в качестве внешнего параметра цикла i , а в качестве внутреннего – переменную j . На рис. 7.16 приведен фрагмент документа, формирующий матрицу B в соответствии с принятыми параметрами цикла.

$$\begin{array}{l}
 n := 3 \quad m := 6 \\
 i := 1..n \quad j := 1..m \\
 B_{i,j} := \frac{1}{i + j + 1} \quad B = \begin{pmatrix} 0.333 & 0.25 & 0.2 & 0.167 & 0.143 & 0.125 \\ 0.25 & 0.2 & 0.167 & 0.143 & 0.125 & 0.111 \\ 0.2 & 0.167 & 0.143 & 0.125 & 0.111 & 0.1 \end{pmatrix}
 \end{array}$$

Рис. 7.16. Формирование матрицы B примера 7.3.4

Заметим, что в приведенных на рис. 7.15 и 7.16 фрагментах системная переменная $ORIGIN=1$.

Программирование итерационных циклов. Для этого в версиях MathCAD до *MathCAD 2001i* использовалась функция *until*, записываемая в виде:

$$until (<выр. 1>, <выр. 2>).$$

Эта функция принимает значение равное *выр. 2*, если *выр. 1* больше или равно 0. Как только *выр. 1* принимает отрицательное значение, функция *until* принимает значение 0 и дальнейшее выполнение функции прекращается. Это свойство позволяет применять функцию *until* для программирования итерационных циклов.

Пример 7.3.6. Используя функцию *until*, реализовать итерационную процедуру вычисления корня квадратного, рассмотренную в примере 7.3.2.

Решение. На рис. 7.17 приведен фрагмент документа, реализующий алгоритм примера 7.3.2 (*ORIGIN*=0). Последний, 7-й элемент массива x (значение индекса равно 6) равен нулю из-за того, что выражение $|(x_i)^2 - a| - \varepsilon$ стало меньше нуля, а значение функции *until* = 0.

<i>ORIGIN</i> := 0		
<i>a</i> := 9	$x_0 := a$	$\varepsilon := 10^{-6}$
<i>i</i> := 0 .. 10		
$x_{i+1} := \text{until} \left[(x_i)^2 - a - \varepsilon, \frac{x_i + \frac{a}{x_i}}{2} \right]$		
$x^T = (\text{9.0000} \quad \text{5.0000} \quad \text{3.4000} \quad \text{3.0235} \quad \text{3.0001} \quad \text{3.0000} \quad \text{0.0000})$		

Рис. 7.17. Реализация итерационного алгоритма примера 7.3.5

Поэтому в качестве приближенного значения корня необходимо принять предыдущий элемент, номер и значение которого определяются в следующем фрагменте:

$n_{solve} := \text{last}(x) - 1 \quad n_{solve} = 5 \quad x_{n_{solve}} = 3.0000000$

Следует отметить, что такой способ программирования *не эффективен при большом числе итераций* из-за необходимости хранить в памяти все предыдущие значения x_i . На самом деле

необходимо хранить в памяти только два приближения: x_i («старое») и x_{i+1} («новое»).

Замечание. В версиях *MathCAD11*, *MathCAD12*, *MathCAD13*, *MathCAD14* обращение к этой функции **вызывает ошибку** и вместо этой функции рекомендуется для программирования итерационных циклов использовать подпрограмму-функцию с оператором *while*.

В заключение этого параграфа заметим, что рассмотренные способы безмодульного программирования можно рекомендовать для реализации достаточно простых алгоритмов. Более сложные алгоритмы целесообразно реализовать в виде подпрограммы-функции, которая может содержать конструкции, присутствующие в других алгоритмических языках высокого уровня (условные операторы, операторы цикла и т.д.). В следующих темах будут рассмотрены конструкции пакета MathCAD, позволяющие реализовывать различные алгоритмы в подпрограмме-функции.

Вопросы и задания для самоконтроля

1. Какая характерная черта безмодульного программирования?
2. Какая характерная особенность линейного алгоритма?
3. Составить блок-схему и реализовать в документе MathCAD алгоритм вычисления площади треугольника по длинам его сторон:

$$S = \sqrt{p(a-p)(b-p)(c-p)},$$

где $p = \frac{a+b+c}{2}$ – полупериметр сторон треугольника.

4. Какая характерная особенность разветвляющегося алгоритма?

5. Условная функция *if* и результат ее выполнения?

6. Составить блок-схему и реализовать в документе MathCAD алгоритм вычисления функции:

$$q(x) = \begin{cases} 2e^{x+2}, & \text{если } x \leq 0; \\ \sqrt{x+3}, & \text{если } x > 0. \end{cases}$$

7. Составить блок-схему и реализовать в документе MathCAD алгоритм вычисления функции:

$$y(x) = \begin{cases} 3x, & \text{если } x \leq -2; \\ |x|, & \text{если } -2 < x \leq 2; \\ \sqrt{x} - 3, & \text{если } x > 1. \end{cases}$$

8. В чем заключается характерная особенность простого цикла?

9. Составить блок-схему и реализовать в документе MathCAD алгоритм формирования вектора, состоящего из $n = 10$ элементов, по правилу:

$$x_i = \frac{2+i}{i^3 + \ln(i)}, i = 1, 2, \dots, n.$$

ТЕМА 8. МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ В ПАКЕТЕ MATHCAD

В этой теме будут рассмотрены описание подпрограммы-функции и ее вызов, а также будут даны конструкции для программирования линейных, разветвляющихся и циклических алгоритмов в подпрограмме-функции MathCAD.

8.1. Сущность и преимущества модульного программирования

Вопросам модульного программирования было посвящено большое число публикаций (например, [4, 5]). Основная идея модульного программирования заключается:

- в разбиении алгоритма решения той или иной задачи на слабо зависимые друг от друга фрагменты вычислений (на подзадачи) и реализацию каждого такого фрагмента в виде *программных модулей*;
- в вызове в нужных местах «основной» программы соответствующих модулей с передачей необходимых данных.

Программа, реализующая алгоритм вычислений в виде модулей и обращения к ним, получила название *модульной программы*.

Что же дает модульное программирование? К основным преимуществам можно отнести следующее:

- Слабая зависимость модулей между собой позволяет при необходимости распараллелить разработку программы, поручив это программистам с разной квалификацией.
- Модульную программу легче отлаживать, так как модули могут быть подвергнуты «автономному» тестированию и отладке, т.е. каждый модуль может проходить тестирование и отладку отдельно на подготовленном наборе тестовых данных. Считается, что усилия, затрачиваемые на отладку модуля, пропорциональны квадрату числа операторов, реализующих этот модуль. Поэтому автономное тестирование (при правильно оп-

ределенных размерах модуля) дает возможность протестировать все ветви вычислительного алгоритма, реализуемого в модуле.

- Модульную программу легче сопровождать и модифицировать. В модуль можно ввести изменения, переписать его или заменить без внесения изменений в другие модули (это отчасти является признаком слабой зависимости между модулями).

- Но самое главное – использование библиотек «готовых» модулей, ориентированных на решение определенного класса научно-технических задач, например тепловой расчет строительных конструкций. Использование модулей таких библиотек существенно снижает затраты на программирование, и если эти модули прошли достаточно полное тестирование, то их применение уменьшает количество ошибок в разработанном программном обеспечении.

При реализации модульного программирования большое значение имеет правильное определение размера модуля. Приводятся разные рекомендации по размеру модуля: от нескольких десятков до нескольких сотен операторов. Однако ограничение на размер модуля *не гарантирует модульность разработанного ПО*. Поэтому размер модуля должен определяться исходя из компромисса между ограничением на *число операторов* (скажем, не более 60) и *высокой степенью независимости модуля*. Высокая степень независимости определяется *сильными внутренними связями в модуле* и *слабыми взаимосвязями между модулями*.

Слабую взаимосвязь между модулями можно охарактеризовать инвариантностью модулей к изменению алгоритма. Если изменение алгоритма приведет к изменению сравнительно большого числа модулей, то это говорит о сильной взаимосвязи между модулями. В этом случае можно объединить зависимые между собой модули, получив один более крупный модуль, либо алгоритм реализовать в виде отдельного модуля, вынося его из других модулей. Отсутствие сильной взаимосвязи внутри модуля часто объясняется наличием в нем нескольких независимых (или слабо зависимых) вычислительных алгоритмов. Каж-

дый из этих алгоритмов может быть реализован в виде отдельных модулей.

В MathCAD программный модуль может быть реализован в виде *подпрограммы-функции*.

8.2. Описание подпрограммы-функции и локальный оператор присваивания

Перед тем как использовать подпрограмму-функцию (П-Ф), нужно ее задать, т.е. выполнить описания. Описание П-Ф размещается в рабочем документе перед ее вызовом и включает в себя имя подпрограммы-функции, список формальных параметров (который может отсутствовать) и тело подпрограммы-функции. Для ввода конструкций в тело П-Ф используется палитра инструментов **ПРОГРАММИРОВАНИЕ**, приведенная на рис. 8.1.

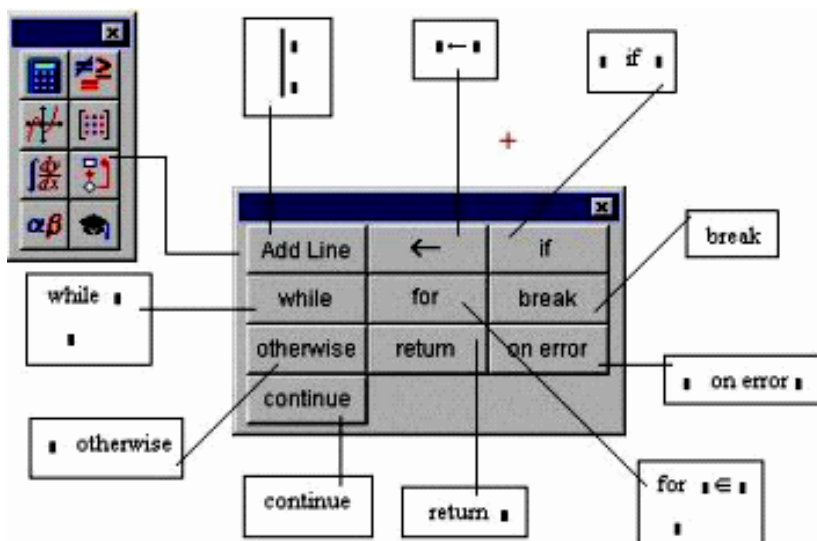


Рис. 8.1. Палитра **ПРОГРАММИРОВАНИЕ**

Каждая П-Ф MathCAD имеет *оригинальное имя*, посредством которого осуществляется обращение к ней. Через это же имя (*и только через это имя*) «возвращается» результат выполнения П-Ф.

После имени П-Ф идет *список формальных параметров*, заключенный в круглые скобки. Через формальные параметры «внутри» П-Ф «передаются» данные, необходимые для выполнения вычислений внутри программы, т.е. *все формальные параметры являются входными*.

В качестве формальных параметров могут использоваться имена простых переменных, массивов и функций. Формальные параметры отделяются друг от друга запятой.

Замечание 8.1. П-Ф может не иметь *формальных параметров*, и тогда данные передаются через имена переменных, определенных выше описания П-Ф.

Тело подпрограммы-функции включает любое число операторов: локальных операторов присваивания, условных операторов и операторов цикла, а также вызов других П-Ф и функций пользователя.

Порядок описания подпрограммы-функции MathCAD. Для ввода в рабочий документ описания П-Ф необходимо выполнить следующие действия:

- ввести имя П-Ф и список формальных параметров, заключенный в круглые скобки (см. замечание 8.2.);
- ввести символ “:=” — на экране отображается как “:=”;
- открыть палитру **ПРОГРАММИРОВАНИЕ** и щелкнуть кнопкой Add line. На экране появится вертикальная черта и вертикальный столбец с двумя полями для ввода операторов, образующих тело П-Ф (см. рис. 8.2);

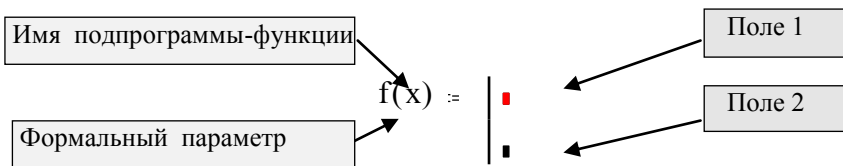


Рис. 8.2. Структура подпрограммы-функции

- перейти в поле 1 (щелкнув на нем мышью или нажав клавишу [Tab]) и ввести первый оператор тела П-Ф. Так как самое нижнее поле всегда предназначено для возвращаемого П-Ф значения, то поля ввода для дополнительных операторов открываются с помощью щелчка на кнопке **Add line** палитры **ПРОГРАММИРОВАНИЕ** (см. рис. 8.3). При этом поле ввода добавляется внизу выделенного к этому моменту оператора. Для удаления того или иного оператора или поля ввода из тела П-Ф нужно заключить его в выделяющий уголок и нажать клавишу [Delete];
- заполнить самое нижнее поле ввода (поле 2), введя туда выражение, определяющее возвращаемое через имя П-Ф (см. рис. 8.4).

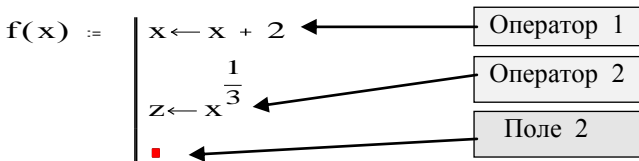


Рис. 8.3. Добавление операторов в тело подпрограммы-функции

f(x) :-	x ← x + 2
	$z \leftarrow x^{\frac{1}{3}}$
	z

Рис. 8.4. Окончательная структура подпрограммы-функции

В приведенном примере формальным параметром является простая переменная x , тело П-Ф включает два локальных оператора присваивания (см. следующий пункт) и значение переменной z , которая определяет возвращаемый через имя функции результат выполнения П-Ф.

Замечание 8.2. Если результатом работы П-Ф являются несколько величин, то из них в теле П-Ф необходимо сформировать массив и его имя поместить в последней строке тела П-Ф.

Локальный оператор присваивания. Для задания внутри программы значения какой-либо переменной используется так называемый *локальный оператор присваивания*, имеющий вид:

< имя переменной > ← < выражение >

Внимание! Использование «обычного» оператора присваивания в теле П-Ф (на экране отображается : =) приводит к синтаксической ошибке.

8.3. Обращение к подпрограмме-функции MathCAD

Для выполнения П-Ф необходимо обратиться к ее имени с указанием *списка фактических параметров* (если в описании программы присутствует список формальных параметров), т.е.:

< имя П-Ф > (< список фактических параметров >)

Фактические параметры указывают, при каких конкретных значениях осуществляются вычисления в теле П-Ф. Фактические параметры отделяются друг от друга запятой.

Очевидно, что между фактическими и формальными параметрами должно быть *соответствие по количеству, порядку следования и типу*. Последнее соответствие означает:

- если формальным параметром является простая переменная, то в качестве фактического может использоваться константа, переменная, арифметическое выражение;
- если формальным параметром является вектор или матрица, то фактическим должен быть вектор или матрица;
- если формальным параметром является имя встроенной функции или другой программы, то и фактическим параметром должен являться тот же объект.

Замечание 8.3. Обращение к П-Ф должно находиться после ее описания, и *к моменту обращения фактические параметры должны быть определены.*

Пример 8.3.1. Обращение к программе $f(x)$, приведенной на рис. 8.4, может иметь следующий вид:

$x := 2$	$f(x) = 1.587$	$f(-3.23) = 0.536 + 0.928i$
$z := f(x + 2.5)$	$z = 1.866$	

Заметим, что переменная z никак не связана с «локальной» переменной z , используемой внутри тела П-Ф.

Замечание 8.4. Передать данные внутрь П-Ф можно используя внутри подпрограммы переменные, определенные до описания П-Ф (см. пример на рис. 8.5).



Рис. 8.5. Подпрограмма-функция без формальных параметров

Хотя значение переменной x изменилось внутри П-Ф, вне описания П-Ф эта переменная сохранила свое прежнее значение.

Замечание 8.5. Имена фактических параметров при вызове П-Ф могут либо совпадать, либо не совпадать с именами ее формальных параметров.

8.4. Программирование линейных алгоритмов в подпрограмме-функции MathCAD

Напомним, что под линейным алгоритмом понимается вычислительный процесс, в котором необходимые операции выполняются строго последовательно (см. п. 7.1). Операторы, реализующие этот алгоритм, в теле П-Ф также размещаются последовательно и выполняются все, начиная с первого и заканчивая последним.

Пример 8.4.1. Оформим в виде П-Ф вычисление корней квадратного уравнения $ax^2 + bx + c = 0$ по формуле

$$x_{1,2} = \frac{-b \mp (b^2 - 4ac)^{1/2}}{2a}.$$

Решение. Описание П-Ф `root_poly2` и обращение к ней приведено на рис. 8.6. П-Ф имеет три входных формальных параметра – коэффициенты квадратного уравнения. Выходом является вектор с двумя компонентами. Заметим, что величины x_1 , x_2 являются простыми переменными, а не элементами одномерного массива. Поэтому нижние индексы в их именах вводятся после нажатия клавиши `[.]` – «десятичная точка». Подпрограмма-функция реализует *линейный алгоритм – все операторы выполняются всегда строго последовательно.*

8.5. Программирование разветвляющихся алгоритмов в подпрограмме-функции MathCAD

Напомним, что в разветвляющихся алгоритмах присутствует несколько ветвей вычислительного процесса. Выбор конкретной ветви зависит от выполнения (или невыполнения) заданных условий на значения переменных алгоритма.

$$\begin{array}{l}
 \text{root_poly2} (a, b, c) := \left\{ \begin{array}{l}
 d \leftarrow \sqrt{b^2 - 4 \cdot a \cdot c} \\
 x_1 \leftarrow \frac{-b + d}{2 \cdot a} \\
 x_2 \leftarrow \frac{-b - d}{2 \cdot a} \\
 \left(\begin{array}{c} x_1 \\ x_2 \end{array} \right)
 \end{array} \right. \\
 z := \text{root_poly2} (12, 7, 10) \quad z = \left(\begin{array}{c} -0.292 + 0.865i \\ -0.292 - 0.865i \end{array} \right)
 \end{array}$$

Рис. 8.6. Программирование линейного алгоритма
(к примеру 8.4.1)

Для программирования разветвляющихся алгоритмов в подпрограмме-функции MathCAD можно использовать *условную функцию и условный оператор if*. Используя эти конструкции, можно «изменить» последовательное выполнение операторов. Условная функция *if* была рассмотрена в п. 7.2, поэтому перейдем к изучению условного оператора *if*.

Условный оператор. Этот оператор используется *только в теле П-Ф* и для его ввода необходимо щелкнуть на кнопке **if** палитры **ПРОГРАММИРОВАНИЕ**. На экране появляется конструкция с двумя полями ввода, изображенная на рис. 8.7.

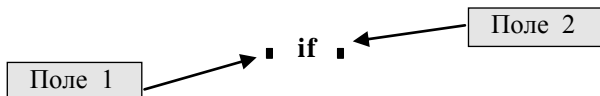


Рис. 8.7. Структура условного оператора *if*

В поле 2 вводится логическое выражение УСЛ (в простейшем случае это выражение отношений). В поле 1 вводится конструкция ВЫР1 (как правило, арифметическое выражение), которая выполняется, если проверяемое логическое выражение принимает значение 1. Если УСЛ = 0, то ВЫР1 не выполняется. Это соответствует условной структуре, называемой **ЕСЛИ-ТО**.

Для получения условной структуры **ЕСЛИ-ТО-ИНАЧЕ** используется оператор *otherwise*, вводимый с палитры **ПРОГРАММИРОВАНИЕ**, в поле которого размещается конструкция ВЫР2, которая выполняется, если проверяемое логическое выражение в вышестоящем условном операторе принимает значение 0 (см. рис. 8.8). Оператор *otherwise* непосредственно следует после условного оператора *if*.

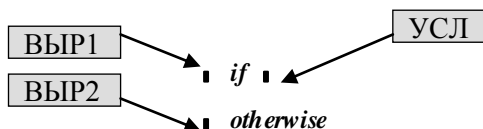


Рис. 8.8. Реализация структуры **ЕСЛИ-ТО-ИНАЧЕ**

Для ввода ВЫР2 в поле оператора *otherwise* необходимо:

- выделить поле, стоящее после оператора *if*;
- щелкнуть на кнопке *otherwise* палитры **ПРОГРАММИРОВАНИЕ**;
- в появившееся поле оператора *otherwise* ввести необходимую конструкцию ВЫР2.

Пример 8.5.1. Составим описание П-Ф, вычисляющей функцию $y(x)$. Описание и вызов П-Ф приведены на рис. 8.9.

$$\begin{array}{l}
 y(x) := \left\{ \begin{array}{ll} x^2 & \text{if } x \leq 0 \\ \sqrt{x} & \text{otherwise} \end{array} \right. \\
 y(2) = 1.414 \qquad y(-4) = 16
 \end{array}$$

Рис. 8.9. Реализация разветвляющегося алгоритма примера 8.3

Таким образом, конструкция, стоящая перед словом *otherwise*, выполняется только в том случае, если не выполнено заданное перед этим условие. В программе можно использовать несколько следующих друг за другом условных операторов с одним оператором *otherwise*.

Пример 8.5.2. Составим описание П-Ф для вычисления переменной $z(t)$ по формуле

$$z(t) = \begin{cases} t^3, & t < -3; \\ \ln(t), & t > 4; \\ t^2, & -3 \leq t \leq 4. \end{cases}$$

Решение. Описание П-Ф и ее вызов приведены на рис. 8.10.

$$\begin{array}{l}
 z(t) := \left\{ \begin{array}{ll} t^3 & \text{if } t \leq -3 \\ t^2 & \text{if } -3 \leq t \leq 4 \\ \ln(t) & \text{otherwise} \end{array} \right. \\
 z(-8) = -512 \qquad z(2) = 4 \qquad z(7) = 1.946
 \end{array}$$

Рис. 8.10. Реализация разветвляющегося алгоритма примера 8.5.2

Из описания видно, что функция $z(t)$ получит значение $\ln(t)$ только тогда, когда не выполняются условия, записанные в двух вышестоящих условных операторах.

Внимание! Если в строке 3 ввести просто $\ln(t)$, то это выражение *будет вычисляться всегда* вне зависимости от выполнения заданных выше условных операторов.

Задание 8.5.1. Составьте описания П-Ф, реализующих следующие разветвляющиеся алгоритмы:

$$\text{А. } y = \begin{cases} \frac{1}{2\sqrt{x}}, & \text{если } x \geq 1; \\ \frac{1}{3}\sqrt[3]{x}, & \text{если } 0 \leq x \leq 1; \\ \frac{1}{4}\sqrt[4]{|x|}, & \text{если } x < 0. \end{cases}$$

$$\text{Б. } \mu(x, \varepsilon) = \begin{cases} \frac{1}{2\sqrt{x+1}}, & \text{если } |x - y| < \varepsilon; \\ \frac{1}{3}\sqrt[3]{x+y}, & \text{если } |x - y| \geq \varepsilon, \end{cases}$$

где $y = |x|$.

Рассмотрим использование условного оператора *if* при программировании более сложных разветвляющихся алгоритмов. Выделим следующие варианты.

Вариант 1. При выполнении заданного условия УСЛ необходимо выполнить несколько конструкций MathCAD.

В этом случае необходимо выделить *поле 1* условного оператора (см. рис. 8.11), щелкнуть на кнопке **Add line** палитры **ПРОГРАММИРОВАНИЕ** нужное число раз и заполнить появившиеся поля (рис. 8.21).

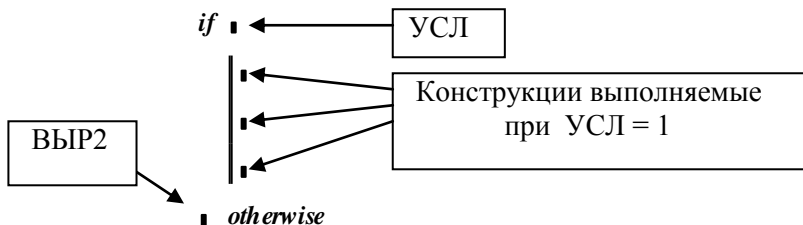


Рис. 8.11. Вариант 1 реализации условного оператора

Вариант 2. При невыполнении заданного условия УСЛ необходимо выполнить несколько конструкций MathCAD.

В этом случае необходимо выделить *поле* оператора *otherwise*, щелкнуть на кнопке **Add line** палитры **ПРОГРАММИРОВАНИЕ** нужное число раз и заполнить появившиеся поля (рис. 8.12).

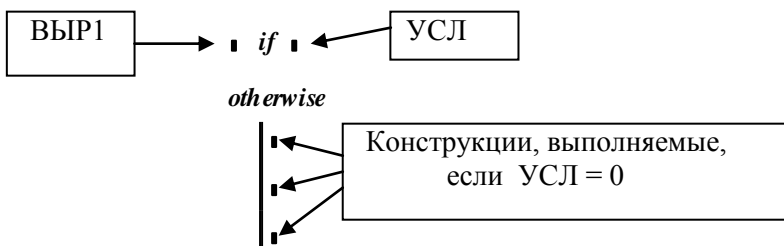


Рис. 8.12. Вариант 2 реализации условного оператора

Пример 8.5.3. Составьте описание П-Ф, вычисляющей значения двух полиномов $x(t)$, $y(t)$ нулевой или первой степени. Порядок полиномов задается переменной n . Если $n < 0$ или $n > 1$, то значения полиномов равны 0.

Решение. Описание П-Ф приведено на рис. 8.13. Обратите внимание, что коэффициенты полиномов передаются через массивы a , b , а в полях условных операторов находятся не арифметические выражения, а локальные операторы присваивания.

$poly(n, a, b, t) :=$	$if\ n = 0$ $\left \begin{array}{l} x \leftarrow a_0 \\ y \leftarrow b_0 \end{array} \right.$ $if\ n = 1$ $\left \begin{array}{l} x \leftarrow a_0 + a_1 \cdot t \\ y \leftarrow b_0 + b_1 \cdot t \end{array} \right.$ $otherwise$ $\left \begin{array}{l} x \leftarrow 0 \\ y \leftarrow 0 \end{array} \right.$ $\left(\begin{array}{c} x \\ y \end{array} \right)$
-----------------------	---

Рис. 8.13. Реализация алгоритма примера 8.5.3

Пример 8.5.4. Даны два числа x , y . Составить описание П-Ф, которая переменной x присваивает максимальное значение из этих двух чисел, а y – минимальное.

Решение. Описание приведено на рис. 8.14. Выходным параметром является массив v , нулевой элемент которого содержит новое значение x , а первый элемент – новое значение y . Вызов этой П-Ф показан на рис. 8.15.

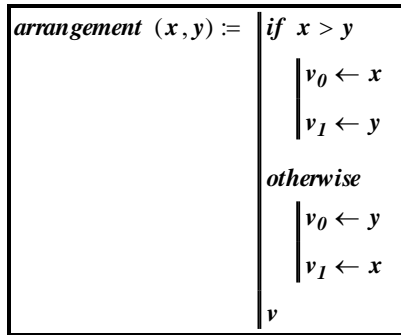


Рис. 8.14. Реализация алгоритма примера 8.5.4

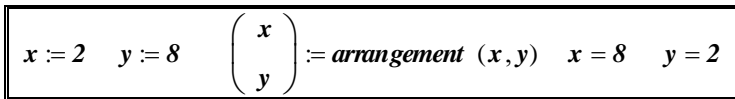


Рис. 8.15. Вызов подпрограммы-функции *arrangement*

Задание 8.5.1. Даны три числа a, b, c . Составить П-Ф, реализующую следующий алгоритм. Если $a \leq b \leq c$, то все числа заменить их квадратами, если $a > b > c$, то каждое число заменить максимальным значением из этих трех чисел, в противном случае – сменить знаки у чисел.

Задание 8.5.2. Координаты точки на плоскости задаются двумя числами x, y . Составить П-Ф, вычисляющую номер четверти на плоскости, в которую попала точка.

Задание 8.5.3. Длина сторон треугольника задается числами a, b, c . Составить П-Ф, вычисляющую значение целой переменной n по следующему правилу: $n = 3$, если три стороны равны; $n = 2$, если любые две стороны равны; $n = 1$, если все три стороны имеют разную длину.

8.6. Программирование циклических алгоритмов в подпрограмме-функции MathCAD

Напомним (см. п. 7.3), что циклические алгоритмы (циклы) содержат повторяющиеся вычисления, зависящие от некоторой переменной. Такая переменная называется *параметром цикла*, а сами повторяющиеся вычисления составляют *тело цикла*. Циклы можно условно разделить на две группы:

- *циклы типа арифметической прогрессии;*
- *итерационные циклы.*

Особенности этих циклов были рассмотрены в п. 7.3.

Программирование цикла типа арифметической прогрессии. Для программирования таких циклов используется оператор цикла *for* (часто называемый *оператором цикла с параметром*).

Для ввода этого оператора необходимо выполнить следующие действия:

- щелкнуть на кнопке **for** палитры **ПРОГРАММИРОВАНИЕ**. На экране появятся поля ввода, изображенные на рис. 8.16;

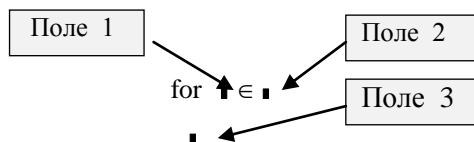


Рис. 8.16. Поля оператора цикла *for*

- в поле ввода 1 ввести имя переменной, являющейся параметром цикла;
- в поле 2 – закон изменения параметра цикла, используя для этого описание дискретной переменной или *описание массива*;

- в поле 3 – операторы, составляющие тело цикла. Если одной строки недостаточно, то дополнительные поля ввода (дополнительные строки) создаются щелчком на кнопке **Add line** палитры **ПРОГРАММИРОВАНИЕ**, и тогда слева от тела цикла появляется вертикальная черта, охватывающая тело цикла.

Пример 8.6.1. Составить описание П-Ф, реализующей алгоритм примера 7.3.1.

Решение. Описание П-Ф и ее вызов приведены на рис. 8.17. Заметим, что значение системной переменной *ORIGIN* (начальное значение индексного выражения) задается равным 1.

$form_vec1(n) := \left \begin{array}{l} \text{for } i \in 1..n \\ z_i \leftarrow \frac{1}{i^2 + 1} \end{array} \right _z$	$form_vec1(5) = \begin{pmatrix} 0.5 \\ 0.2 \\ 0.1 \\ 0.059 \\ 0.038 \end{pmatrix}$
---	---

Рис. 8.17. Подпрограмма-функция формирования вектора

Пример 8.6.2. Для x , меняющегося от -2 до 2 с шагом 0.5, вычислить значение $f(x) = e^{-x} \cos(2x)$ и сформировать из этих значений вектор y , т.е. $y_1 = f(-2)$, $y_2 = f(-1.5)$ и т.д.

Решение. В этом примере количество повторений тела цикла определяется по формуле:

$$\left\lceil \frac{x_k - x_0}{d} \right\rceil + 1,$$

где x_k, x_0 – конечное и начальное значения параметра цикла; d – шаг его изменения.

Подставив значения, получаем $(2 - (-2))/0.5 + 1 = 9$. Следовательно, сформированный вектор y будет содержать 9 элементов. Необходимо отметить, что алгоритм формирования содержит два параметра цикла: первый параметр – переменная x , определяющая текущее значение аргумента функции $f(x)$; второй – это переменная i , определяющая текущее значение индекса y элемента формируемого вектора. Но между этими двумя параметрами имеется однозначное соответствие, поэтому цикл нужно организовывать только по одному параметру, например, по переменной x , а значение второго параметра будет вычисляться в теле цикла.

Описание П-Ф и ее вызов приведены на рис. 8.18. Видно, что в теле цикла выполняется два оператора. Первый оператор формирует элемент массива y , а второй изменяет на 1 значение индекса.

$$\begin{array}{l}
 \text{form_vec2} (x_0, x_k, d) := \left| \begin{array}{l} i \leftarrow 1 \\ \text{for } x \in x_0, x_0 + d .. x_k \\ \quad \left| \begin{array}{l} y_i \leftarrow e^{-x} \cdot \cos(2 \cdot x) \\ i \leftarrow i + 1 \end{array} \right. \\ y \end{array} \right. \quad z := \text{form_vec2} (0, 0.8, 0.1) \\
 z^T = (\ 1 \ 0.887 \ 0.754 \ 0.611 \ 0.467 \ 0.328 \ 0.199 \ 0.084 \ -0.013 \)
 \end{array}$$

Рис. 8.18. Формирование вектора примера 8.6.2

Отметим несколько особенностей оператора *for*, которые выгодно отличают его от аналогичных конструкций других языков программирования (например, языка Pascal).

Первая особенность состоит в том, что переменная цикла может принимать вещественные значения. Эту особенность хорошо иллюстрирует рассмотренный пример 8.5.2.

Вторая особенность заключается в том, что значения переменной цикла могут задаваться вектором или матрицей, а также последовательностью чисел. Это иллюстрируется примерами 8.5.3 и 8.5.4.

Третья особенность состоит в том, что переменная цикла может принимать значения различных типов – численное, текстовое и т.д. Эту особенность иллюстрирует рассмотренный пример 8.5.5.

Пример 8.6.3. Составить описание П-Ф, вычисляющей сумму вектора с проекциями $|2, 5, 7|$.

Решение. На рис. 8.19 приведено описание П-Ф, в которой слагаемые задаются значениями параметра цикла.

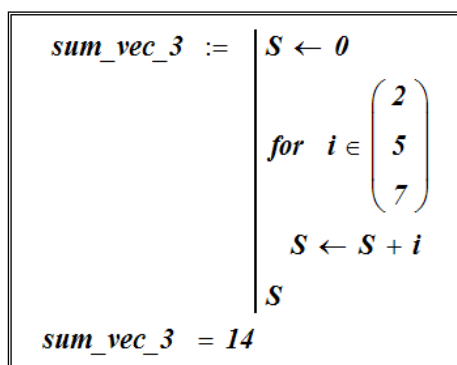


Рис. 8.19. Описание П-Ф примера 8.6.3

Пример 8.6.4. Составить описание П-Ф, вычисляющей сумму следующих слагаемых $2, 3, 5, 9, 11, 17, 21$.

Решение. На рис. 8.20 приведено описание П-Ф, в которой слагаемые задаются значениями параметра цикла, определяемого элементами последовательности.

$sum_1 :=$	$S \leftarrow 0$ for $i \in 2, 3, 5, 9, 11, 17, 21$ $S \leftarrow S + i$ S
$sum_1 = 68$	

Рис. 8.20. Описание П-Ф примера 8.6.4

Пример 8.6.5. Запрограммировать П-Ф, которая формирует двумерный массив из объектов разных типов.

Решение. На рис. 8.21 приведено описание П-Ф, а также показан результат ее работы. Видно, что переменная цикла i_c принимает значения различных типов: вещественное, комплексное и строковое.

$ORIGIN := 1$				
$Form :=$	$i \leftarrow 1$ for $i_c \in \left(\begin{array}{cc} 100.25 & "Good" \\ 20 + 55i & "Yes" \end{array} \right)$ <table style="margin-left: 40px;"> <tr> <td style="padding-right: 10px;">$V_i \leftarrow i_c$</td> <td rowspan="2" style="padding-left: 20px; vertical-align: middle;"> $Form = \left(\begin{array}{c} 100.25 \\ 20 + 55i \\ "Good" \\ "Yes" \end{array} \right)$ </td> </tr> <tr> <td>$i \leftarrow i + 1$</td> </tr> </table> V	$V_i \leftarrow i_c$	$Form = \left(\begin{array}{c} 100.25 \\ 20 + 55i \\ "Good" \\ "Yes" \end{array} \right)$	$i \leftarrow i + 1$
$V_i \leftarrow i_c$	$Form = \left(\begin{array}{c} 100.25 \\ 20 + 55i \\ "Good" \\ "Yes" \end{array} \right)$			
$i \leftarrow i + 1$				

Рис. 8.21. Описание П-Ф примера 8.6.5

Задание 8.6.1. Составьте описание П-Ф формирования вектора u примера 8.6.2, приняв в качестве параметра цикла переменную i .

Задание 8.6.2. Составьте описание П-Ф формирования вектора q примера 7.3.4.

Программирование итерационных циклов. Для программирования таких циклов используется оператор цикла *while*. Для ввода этого оператора необходимо выполнить следующие действия:

- щелкнуть на кнопке **while** палитры **ПРОГРАММИРОВАНИЕ**. На экране появляются элементы, показанные на рис. 8.22;

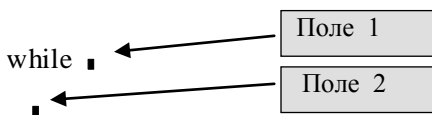


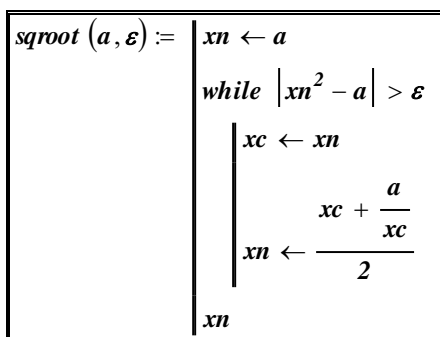
Рис. 8.22. Структура оператора цикла *while*

- в поле 1 ввести условие выполнения цикла;
- в поле 2 ввести операторы тела цикла. В теле цикла должны присутствовать операторы, которые *могут изменить значение условия цикла*, иначе цикл будет продолжаться бесконечно.

Оператор цикла *while* выполняется следующим образом: обнаружив оператор *while*, MathCAD проверяет указанное в операторе условие. Если оно равно 1 (т.е. выполняется), то выполняется тело цикла и снова проверяется условие. Если условие принимает значение 0, то цикл заканчивается.

Пример 8.6.6. Составим П-Ф, реализующую итерационную процедуру приближенного вычисления корня квадратного, описанную в примере 7.3.2.

Решение. Описание П-Ф приведено на рис. 8.23. Как видно из текста П-Ф, нет необходимости хранить в памяти все приближенные решения x_0, x_1, x_2, \dots (как это делалось при использовании функции *until* – см. рис. 7.17). Достаточно хранить предыдущее («старое») значение (обозначим его как xc) и следующее («новое») значение xn .



Вызов подпрограммы-функции

$sqroot(9, 0.1) = 3.00009$

$sqroot(9, 0.0001) = 3.00000$

Рис. 8.23. Реализация итерационного цикла примера 8.6.6

К сожалению, организация итерационного цикла с помощью оператора *while* без дополнительных средств контроля может привести к заикливанию, т.е. повторению тела цикла «бесконечное» число раз. Например, задав при обращении к П-Ф $\varepsilon < 0$, получаем заикливание. Поэтому в MathCAD имеется специальный оператор *break*, который позволяет выйти из цикла или приостановить исполнение программы при выполнении заданного в операторе *break* условия.

Для ввода оператора *break* необходимо щелкнуть на кнопке **break** панели **ПРОГРАММИРОВАНИЕ** (нельзя вводить этот оператор с клавиатуры по символам). Оператор *break* используется в левом поле ввода условного оператора *if*, а в правом размещается условие, при выполнении которого происходит прекращение работы цикла или программы. Поэтому *первоначально вводится оператор if*, а затем заполняются поля этого оператора. Следующий пример показывает написание подпрограммы без «зацикливания» с использованием оператора *break*.

Пример 8.6.7. Составим П-Ф, реализующую итерационную процедуру вычисления корня квадратного без «зацикливания».

Решение. Описание П-Ф приведено на рис. 8.24. В этой подпрограмме число повторений тела ограничено 10000. Если за это число итераций приближенное значение корня с заданной точностью не найдено, то параметр *ierr* – индикатор ошибки получает значение 1, что говорит об ошибке в вычислительном процессе.

Для передачи двух вычисленных значений *xn*, *ierr* используется вектор. Если *ierr* = 1, то переменной *xn* присваивается значение строковой константы «*not solve*» для привлечения внимания пользователя к ошибочному завершению работы П-Ф. Заметим, что после обращения к П-Ф необходимо обязательно проверить значение индикатора ошибки *ierr* и, в зависимости от его значения, либо продолжить работу с вычисленным значением корня квадратного, либо прекратить дальнейшие вычисления и определить причину появления ошибки.

Пример 8.6.8. Составить П-Ф, осуществляющую суммирование ряда с бесконечным числом слагаемых. Накопление суммы прекращается, как только очередное слагаемое по абсолютной величине становится меньше заданной погрешности ε .

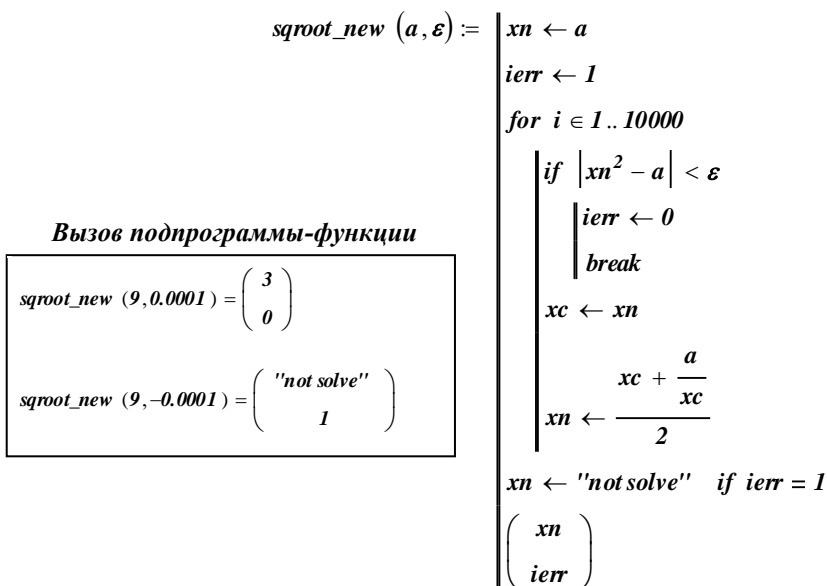


Рис. 8.24. Реализация итерационного цикла без «зацикливания»

Решение. Описание П-Ф и ее вызов показаны на рис. 8.25. Заметим, что вторым формальным параметром является имя функции пользователя, определяющей зависимость величины члена ряда от его номера. При вызове этот формальный параметр заменяется фактическим – именем функции пользователя, описанной до обращения к П-Ф.

Программирование двойных циклов. Напомним, что элементы матриц (двумерных массивов) имеют два индекса: первый индекс определяет номер строки, а второй – номер столбца, на пересечении которых находится указанный элемент. Например, запись $A_{2,4}$ – означает обращение к элементу, стоящему на пересечении второй строки и четвертого столбца.

*Описание слагаемых
суммируемых рядов*

$$b(i) := \frac{1}{i^2} \quad d(i) := \frac{1}{\sqrt{i}}$$

$$sum_conv(\varepsilon, a) := \left\| \begin{array}{l} sum \leftarrow 0 \\ ierr \leftarrow 1 \\ \text{for } i \in 1..10000 \\ \quad \left\| \begin{array}{l} \text{if } |a(i)| < \varepsilon \\ \quad ierr \leftarrow 0 \\ \quad break \\ \quad sum \leftarrow sum + a(i) \end{array} \right. \\ \end{array} \right\|$$

$$sum \leftarrow "not\ convergence" \quad \text{if } ierr = 1$$

$$\left\| \begin{pmatrix} sum \\ ierr \end{pmatrix} \right\|$$

Вызов подпрограммы-функции

$$sum_conv(0.00001, b) = \begin{pmatrix} 1.64177 \\ 0 \end{pmatrix},$$

$$sum_conv(0.00001, d) = \begin{pmatrix} "not\ convergence" \\ 1 \end{pmatrix}.$$

Рис. 8.25. Реализация итерационного цикла примера 8.6.7

Поэтому в общем случае для работы с матрицами необходимо использовать так называемый двойной цикл, у которого тело одного цикла, называемого внутренним, «вкладывается» в тело другого – «внешнего» цикла. Каждый из этих циклов имеет свой параметр: параметр внешнего цикла и параметр внутреннего цикла. Следовательно, для организации такого двойного цикла необходимо использовать два оператора цикла, один из которых размещен в теле другого (различные варианты вложений операторов цикла приведены на рис. 8.26).

for $i \in I$ for $i \in I$ while $i \in I$
 for $j \in J$ while $j \in J$ for $j \in J$
 S S S

Рис. 8.26. Варианты вложений операторов цикла

Пример 8.6.9. Составить описание П-Ф, формирующей матрицу по следующему правилу:

$$B_{i,j} = \frac{1}{i+j+1}, \quad i=1,\dots,n; \quad j=1,\dots,m.$$

Решение. Описание и вызов П-Ф приведены на рис. 8.27. В этой П-Ф параметром внешнего цикла является переменная i , а параметром внутреннего – переменная j .

$$\begin{array}{l}
 \text{ORIGIN} := 1 \quad \text{form_mat1}(n, m) := \left| \begin{array}{l} \text{for } i \in 1..n \\ \text{for } j \in 1..m \\ B_{i,j} \leftarrow \frac{1}{i+j+1} \end{array} \right| \\
 \text{form_mat1}(4, 2) = \left(\begin{array}{cc} 0.333 & 0.25 \\ 0.25 & 0.2 \\ 0.2 & 0.167 \\ 0.167 & 0.143 \end{array} \right) \quad B
 \end{array}$$

Рис. 8.27. Реализация двойного цикла примера 8.6.9

Пример 8.6.10. Дана матрица A размерности $n \times m$. Составить описание П-Ф, вычисляющей индексы и значение элемента матрицы, который по модулю максимально близок к заданному числу c .

Решение. Описание П-Ф и ее вызов приведены на рис. 8.28. Напомним, что системная переменная **ORIGIN** = 1. Заметим, что в последней строке тела П-Ф стоит вектор, состоящий из иско-
мых числовых характеристик.

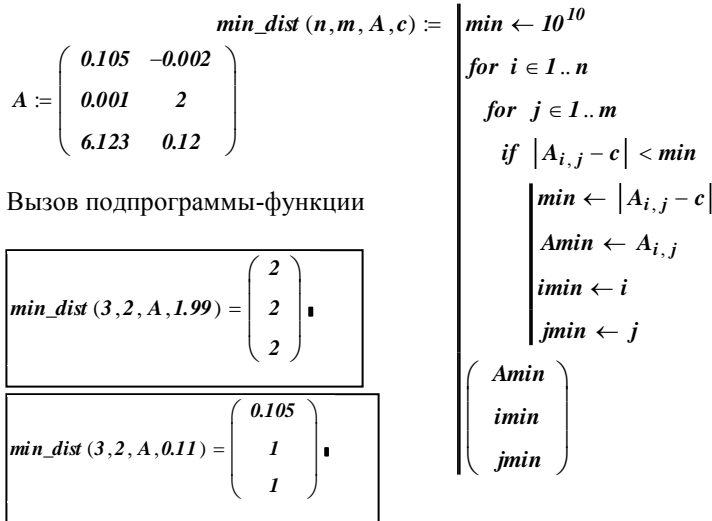


Рис. 8.28. Реализация двойного цикла примера 8.6.8

Дополнительные операторы, используемые при программировании циклов. Рассмотрим несколько операторов, которые могут быть полезны при программировании циклов.

Оператор *continue*. Обычно используется для продолжения выполнения цикла путем возврата в начало тела цикла. Следующий пример поясняет работу этого оператора.

Пример 8.6.11. Составить описание П-Ф, формирующей новый вектор из положительных проекций исходного вектора.

Решение. Описание П-Ф приведено на рис. 8.29. В теле П-Ф используется функция *last(v)*, определяющая значение индекса

последнего элемента массива v . Если очередной элемент v_i не больше нуля, то пропускаются все нижележащие операторы тела цикла (в нашем случае – два оператора, формирующие очередную проекцию вектора w), и тело цикла повторяется при новом значении параметра цикла i .

Сравнение исходного вектора x с результатом работы П-Ф – вектором z показывает правильность работы запрограммированного алгоритма.

$$\begin{array}{lcl}
 x := \begin{pmatrix} 2 \\ -4 \\ 3 \\ 2 \\ -11 \end{pmatrix} & \text{form_vec2}(v) := & \begin{array}{l} i \leftarrow 0 \\ j \leftarrow 0 \\ \text{while } i < \text{last}(v) \\ \quad \begin{array}{l} i \leftarrow i + 1 \\ \text{continue if } v_i \leq 0 \\ j \leftarrow j + 1 \\ w_j \leftarrow v_i \end{array} \end{array} \\
 z := \text{form_vec2}(x) & z = & \begin{pmatrix} 2 \\ 3 \\ 2 \end{pmatrix}
 \end{array}$$

Рис. 8.29. Реализация алгоритма примера 8.6.11

Оператор *return*. Прерывает выполнение П-Ф и возвращает значение операнда, стоящего в поле 1 (см. рис. 8.30).

return ■ ← Поле 1

Рис. 8.30. Структура оператора *return*

Следующий пример поясняет работу этого оператора.

Пример 8.6.12. Составить описание П-Ф, находящей первую положительную проекцию исходного вектора.

Решение. Возможны два варианта программной реализации алгоритма решения этой задачи (приведены на рис. 8.31). Вариант В представляется более простым и «элегантным».

Вариант А	Вариант В
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> $pol_1(v) := \begin{array}{l} i \leftarrow 1 \\ \text{while } v_i \leq 0 \\ \quad i \leftarrow i + 1 \\ v_i \end{array}$ </div>	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> $pol_2(v) := \text{for } i \in 1..last(v) \\ \quad \text{return } v_i \text{ if } v_i > 0$ </div>

Рис. 8. 31. Два варианта реализации алгоритма примера 8.5.12

Оператор *on error*. Этот оператор является обработчиком возникающих при выполнении тех или иных вычислений ошибок и записывается в виде:

$\langle \text{конструкция } 1 \rangle \text{ on error } \langle \text{конструкция } 2 \rangle$

Оператор выполняется следующим образом. Если при выполнении $\langle \text{конструкция } 2 \rangle$ возникает ошибка, то выполняется $\langle \text{конструкция } 1 \rangle$. Если ошибка не возникает, то выполняется $\langle \text{конструкция } 2 \rangle$.

Пример 8.6.13. Используя оператор *on error*, необходимо предотвратить появление ошибки «деление на ноль» при вычислении функции $angl(x,y)$.

Решение. Фрагмент представлен на рис. 8.32.

$$angl(x, y) := \frac{x}{y}$$

$$angl(2, 0) = \blacksquare$$

Ошибка
«деление на ноль»

$$angl(x, y) := 10^{10} \text{ on error } \frac{x}{y}$$

$$angl(2, 0) = 1 \times 10^{10} \quad angl(2, 2) = 1$$

Рис. 8.32. Использование функции *on error* в примере 8.6.13

Функция *error*. Используется для вывода диагностических сообщений при возникновении в вычислениях ошибки и записывается в виде:

error (" < диагностическое сообщение пользователя >")

Внимание! Имя функции вводится с клавиатуры.

Функция используется в левом поле условного оператора *if*, как показано в следующем примере.

Пример 8.6.14. Запрограммируем вывод диагностического сообщения при попытке спроецировать вектор *v* на нулевой вектор *w*. Описание П-Ф и ее вызовы приведены на рис. 8.33.

$$proj(v, w) := \begin{cases} error("You \text{ cannot onto the } 0 \text{ vector"}) & \text{if } |w| < 10^{-10} \\ \frac{w}{|w|} \cdot (v \cdot w) & \text{otherwise} \end{cases}$$

$$x := \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} \quad w := \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix} \quad proj(x, w) = \begin{pmatrix} 16.036 \\ 10.69 \\ 5.345 \end{pmatrix} \blacksquare$$

Рис. 8.33. Использование в П-Ф оператора *error*

Следует заметить, что взятое в кавычки диагностическое сообщение появится на экране только после щелчка мышью на выделенном красным цветом обращении к П-Ф, при выполнении которой произошла ошибка (см. рис. 8.34).

$$x := \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} \quad w := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$


Рис. 8.34. Применение функции *error* в примере 8.6.14

Вопросы и задания для самоконтроля

1. Суть и преимущества модульного программирования?
2. Структура подпрограммы-функции (П-Ф) MathCAD?
3. Какие объекты могут использоваться в качестве формальных параметров П-Ф?
4. Как передаются результаты работы П-Ф в вызывающую программу?
5. Как обратиться к П-Ф?
6. Какие объекты могут использоваться в качестве фактических параметров?
7. Какие соответствия должны выполняться между фактическими и формальными параметрами?
8. Какое значение переменной z будет выведено в конце документа MathCAD, приведенного на рис. 8.35? Объясните почему вывелось такое значение.

$z := 2.1$
$f(x) := \begin{cases} z \leftarrow x^2 \\ z \leftarrow z + 2 \\ z \end{cases}$
$f(2) = 6 \quad z = ?$

Рис. 8.35. Фрагмент документа к вопросу № 8

9. Запрограммировать П-Ф, вычисляющую значение функции $z(x)$, определяемую выражением:

$$z(x) = a \cdot y^{5+x} + b \cdot \cos(|y|),$$

где $y = b \cdot \sin^2(x) + b^2 \cdot \sin^4(x^2)$.

10. Составить П-Ф для вычисления корней квадратного уравнения: $ax^2 + bx + c = 0$ по известной формуле (см. пример 7.1.1):

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Протестируйте П-Ф при различных значениях коэффициентов.

11. Какой оператор используется для программирования разветвляющихся алгоритмов в П-Ф?

12. Нарисуйте блок-схему работы условного оператора *if* и объясните, как он работает.

13. Какая конструкция используется для реализации в MathCAD логической структуры «ЕСЛИ-ТО-ИНАЧЕ»?

14. Составить П-Ф для вычисления переменной y как минимальное значение из трех различных чисел, т.е. $y = \min(a, b, c)$. Протестировать составленную П-Ф.

15. Составить П-Ф, вычисляющей значение y в соответствии с формулой:

$$y = \begin{cases} x, & \text{если } n = 1, \\ 2\sqrt{|x|}, & \text{если } n = 2 \text{ или } n = 3, \\ e^x, & \text{если } n = 4. \end{cases}$$

16. Составить П-Ф, вычисляющую значение y по следующему алгоритму:

$$y = \begin{cases} \sin x, & \text{если } 1 \leq x < 2, \\ e^{-x}, & \text{если } 2 \leq x < 3, \\ \ln x, & \text{если } 3 \leq x < 4, \\ \operatorname{tg} x, & \text{если } 4 \leq x < 5. \end{cases}$$

Если значение x не принадлежит рассматриваемым промежуткам, то переменной y присвоить значение строковой константы: « x не принадлежит заданным интервалам».

17. Какая конструкция используется для программирования цикла типа арифметической прогрессии в П-Ф?

18. Какое значение будет выведено в конце фрагмента (оператор вывода со знаком ?), приведенного на рис. 8.36?

$Pr := 3$	$Pr :=$	$P \leftarrow 1$
		$\text{for } i \in \begin{pmatrix} 3 \\ 1 \\ 2 \\ -1 \end{pmatrix}$
		$P \leftarrow P \cdot i$
$Pr = ?$		P

Рис. 8.36. Фрагмент документа к заданию 18

19. Дан вектор x , состоящий из n проекций. Составить П-Ф, вычисляющую сумму положительных проекций, имеющих четные номера. Рекомендация: для проверки четности используйте функцию MathCAD $\text{mod}(x, d)$, которая определяет остаток от целочисленного деления величины x на величину d . Эти величины могут быть как целыми, так и вещественными. Примеры использования функции mod приведены на рис. 8.37.

$\text{mod}(5, 2) = 1$	$\text{mod}(4, 2) = 0$	$\text{mod}(4.5, 2) = 0.5$
------------------------	------------------------	----------------------------

Рис. 8.37. Примеры использования функции mod

20. Матрица A размером $n \times k$ содержит экзаменационные оценки n студентов по k учебным дисциплинам. Запрограммировать П-Ф, формирующую вектор из n проекций x_i , где x_i равно среднему баллу i -го студента.

ТЕМА 9. ПРОГРАММИРОВАНИЕ ТИПОВЫХ ЗАДАЧ В ПОДПРОГРАММАХ-ФУНКЦИЯХ MATHCAD

В этой теме будет рассмотрено программирование некоторых типовых задач, которые могут входить в состав более сложных алгоритмов.

9.1. Программирование разветвляющихся алгоритмов

Будет рассмотрено несколько примеров программирования разветвляющихся алгоритмов, в некоторых из которых используются встроенные функции MathCAD.

Пример 9.1.1. Составить описание П-Ф, формирующей новый вектор из первых n элементов исходного вектора.

Решение. Описание П-Ф и ее вызов приведены на рис. 9.1. В этой П-Ф использованы две функции:

- $length(v)$ – вычисляет общее число элементов массива v ;
- $submatrix(A, i1, i2, j1, j2)$ – формирует новую матрицу, состоящую из элементов исходной матрицы A , находящихся на пересечении строк с индексами с $i1$ по $i2$ и столбцов с индексами с $j1$ по $j2$. Значение системной переменной $ORIGIN = 0$.

$form_vec3(n, v) :=$	$N \leftarrow length(v)$ $v \quad \text{if } n \geq N$ $submatrix(v, 0, n-1, 0, 0) \quad \text{if } n \geq 1 \quad \text{otherwise}$
-----------------------	--

$x := \begin{pmatrix} 2 \\ 5 \\ 7 \\ 9 \\ 2 \\ 1 \end{pmatrix}$

$form_vec3(3, x) = \begin{pmatrix} 2 \\ 5 \\ 7 \end{pmatrix}$

$form_vec3(10, x) = \begin{pmatrix} 2 \\ 5 \\ 7 \\ 9 \\ 2 \\ 1 \end{pmatrix}$

Обращение к подпрограмме-функции

Рис. 9.1. Реализация алгоритма примера 9.1.1

Пример 9.1.2. Составить описание П-Ф, вычисляющей значения прямоугольных сигналов с периодом 2 и амплитудой +1 и −1 (график приведен на рис. 9.2).

Решение. Описание П-Ф приведено на рис. 9.2. Здесь использовались функции:

- $\text{mod}(y, \text{modul})$ – вычисляет остаток от деления значения вещественной переменной x на modul ;
- $\text{ceil}(y)$ – вычисляет наименьшее целое, большее или равное y .

$$f(x) := \begin{cases} 1 & \text{if } \text{mod}(\text{ceil}(x), 2) = 1 \\ -1 & \text{if } \text{mod}(\text{ceil}(x), 2) = 0 \end{cases}$$

$$x := 0, 0.05 \dots 6$$

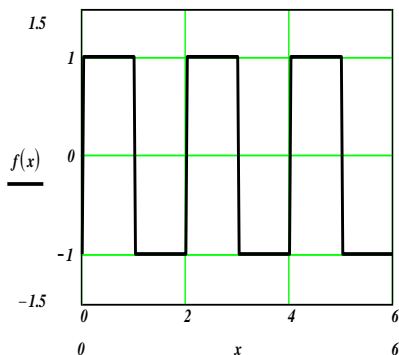


Рис. 9.2. Реализация алгоритма примера 9.1.2

Пример 9.1.3. Составить описание П-Ф, вычисляющей прямоугольный сигнал амплитудой A на интервале $[a, b]$. Вне этого интервала значение равно B .

Решение. Описание П-Ф имеет вид:

$$\boxed{f(x, a, b, A, B) := B + A \cdot (\Phi(x - a) - \Phi(x - b))}$$

а график сигнала при $a = 2, b = 4, A = 6, B = -2$ приведен на рис. 9.3. Здесь использовалась функция $\Phi(x)$, равная нулю при $x < 0$ и 1 для всех других значений аргумента.

Пример 9.1.4. Составить фрагмент документа MathCAD для вычисления значения $z = x_{\max} + y_{pol}$, где x_{\max} – максимальный по модулю корень уравнения $x^2 + 5.45x - 8.12 = 0$; y_{pol} – любой положительный корень уравнения $y^2 + 25.3y - 9.5 = 0$.

Решение. Фрагмент приведен на рис. 9.4. В этом фрагменте вызывается П-Ф вычисления корней квадратного уравнения, описание которой приведено на рис. 8.1 (см. пример 8.1.1).

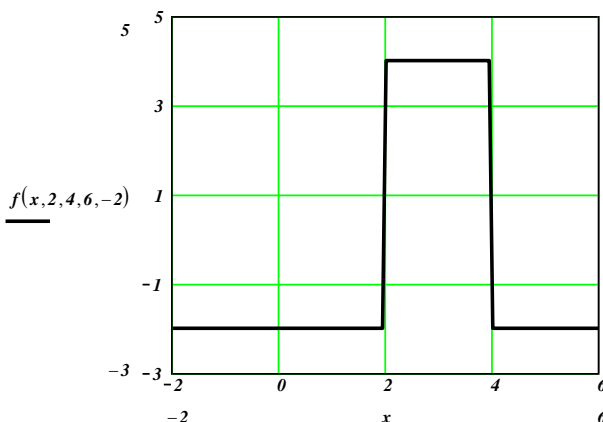


Рис. 9.3. График сигнала примера 9.1.3

$$\begin{aligned}
 X &:= \text{root_poly2}(1, 5.45, -8.12) & X &= \begin{pmatrix} 1.218 \\ -6.668 \end{pmatrix} \\
 Y &:= \text{root_poly2}(1, 25.3, -9.5) & Y &= \begin{pmatrix} 0.37 \\ -25.67 \end{pmatrix} \\
 x_{\max} &:= \text{if}(|X_0| > |X_1|, X_0, X_1) & x_{\max} &= -6.668 \\
 y_{pol} &:= \text{if}(Y_0 > 0, Y_0, Y_1) & y_{pol} &= 0.37 \\
 z &:= x_{\max} + y_{pol} & z &= -6.298
 \end{aligned}$$

Корни уравнений

Рис. 9.4. Реализация алгоритма примера 9.1.4

Задание 9.1.1. Составить описание П-Ф, вычисляющей значения прямоугольных сигналов такой же формы, как на рис. 9.2, но с периодами: а) равным 1; б) равным 8. Построить графики этих сигналов (см. пример 9.1.2).

Задание 9.1.2. Составить описание П-Ф, реализующей алгоритм, приведенный на рис. 9.4а. Составить фрагмент для проверки правильности работы П-Ф.

Задание 9.1.3. Составить описание П-Ф, вычисляющей площадь S геометрической фигуры по ее номеру $n = 1, 2, 3$. Расчетные соотношения приведены на рис 9.4б, где h, a, b, r, n – исходные данные. Составить фрагмент для проверки правильности работы П-Ф.

$$\begin{aligned} & y = \max(a, b), \\ \text{где} \\ & a = \sqrt[3]{2\sin^2 x + 3\cos(x^2 + 2)} \\ & b = \sqrt{\frac{e^{\sin 2x} - e^{\cos 3x}}{\ln x^2 + d^2}} \end{aligned}$$

а)

$$S = \begin{cases} ab, & \text{если } n = 1; \\ \pi r^2, & \text{если } n = 2; \\ h \frac{a+b}{2}, & \text{если } n = 3. \end{cases}$$

б)

Рис. 9.4. Разветвляющиеся алгоритмы

9.2. Программирование циклов типа арифметической прогрессии

В этом параграфе будут рассмотрены четыре типа циклических алгоритмов, входящих в более сложные алгоритмы. Во всех нижеприведенных фрагментах MathCAD системная переменная $ORIGIN=1$.

Тип 1 (формирование элементов массива, удовлетворяющих заданным условиям)

Пример 9.2.1. Составить П-Ф, формирующую матрицу A по следующему правилу:

$$a_{i,j} = \begin{cases} \sin(i + j), & \text{если } i = j; \\ \sin i + \cos j, & \text{если } i > j; \\ \sin j + \cos i, & \text{если } i < j. \end{cases}$$

Решение. Описание П-Ф приведено на рис. 9.5. Элементы матрицы формируются внутри двойного цикла. Переменная i является параметром внешнего цикла, переменная j – параметром внутреннего цикла.

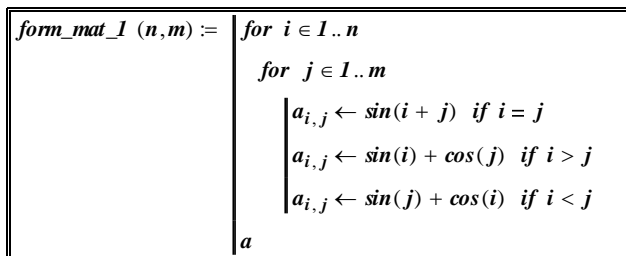


Рис. 9.5. Реализация алгоритма примера 9.2.1

Задание 9.2.2. Даны два вектора x и y , состоящие из n элементов. Составить П-Ф, формирующую вектор q размерности n , i -й элемент которого равен 1, если точка с координатами (x_i, y_i) принадлежит кругу радиуса r с центром в начале координат, и равен 0 в противном случае.

Задание 9.2.3. Дана матрица A размерности $n \times n$. Составить П-Ф, заменяющую все элементы матрицы, с четной суммой индексов на 1, а другие элементы – на 0.

Задание 9.2.4. Даны два вектора x, y , состоящие из n элементов. Преобразовать эти векторы по правилу: большее из чисел x_i и y_i принять в качестве нового значения x_i , а меньшее – в качестве нового значения y_i .

Тип 2 (подсчет числа элементов массива или последовательности, удовлетворяющих заданным условиям)

Пример 9.2.2. Дан массив y , состоящий из n элементов. Составить П-Ф, определяющую:

- число элементов, удовлетворяющих условию $a \leq |y_i| \leq b$;
- число положительных элементов.

Решение. Описание П-Ф приведено на рис. 9.6.

$ \begin{aligned} & \text{calc_num}(n, y, a, b) := \\ & \quad k1 \leftarrow 0 \\ & \quad k2 \leftarrow 0 \\ & \quad \text{for } i \in 1..n \\ & \quad \quad \left \begin{array}{l} k1 \leftarrow k1 + 1 \text{ if } a \leq y_i \leq b \\ k2 \leftarrow k2 + 1 \text{ if } y_i > 0 \end{array} \right. \\ & \quad \left(\begin{array}{c} k1 \\ k2 \end{array} \right) \end{aligned} $
--

Рис. 9.6. Реализация алгоритма примера 9.2.2

Пример 9.2.3. Дана квадратная матрица C размерности $n \times n$. Составить П-Ф, вычисляющую число элементов, находящихся ниже главной диагонали и удовлетворяющих условию $a \leq |C_{i,j}| \leq b$.

Решение. Заметим, что для элементов, лежащих ниже главной диагонали, справедливо неравенство $i > j$. Поэтому возможны два варианта выбора элементов, лежащих ниже главной диагонали: а) проверять все элементы матрицы на выполнение этого условия; б) организовать двойной цикл так, чтобы в нем было обращение только к элементам, удовлетворяющим этому условию. В П-Ф, приведенной на рис. 9.7, реализован второй вариант.

$\text{calc_num2}(n, C, a, b) :=$	$k \leftarrow 0$ <i>for</i> $i \in 2..n$ <i>for</i> $j \in 1..i-1$ $k \leftarrow k + 1$ <i>if</i> $a \leq C_{i,j} \leq b$ k
------------------------------------	---

Рис. 9.7. Реализация алгоритма примера 9.2.3

Пример 9.2.4. Дана последовательность

$$a_i = \cos\left(\frac{i}{10} + i\right),$$

где $i = 1, 2, \dots, n$; $n = 10$.

Составить П-Ф, подсчитывающую число отрицательных членов последовательности и число членов, принадлежащих отрезку $[0.5, 1]$.

Решение. При составлении описания учтем два момента:

а) для универсальности П-Ф элементы последовательности задаются функцией пользователя, имя которой будет параметром П-Ф;

б) для подсчета числа элементов необязательно внутри П-Ф формировать массив из элементов последовательности –

достаточно работать со значением текущего элемента последовательности. Описание П-Ф и ее вызов приведены на рис. 9.8.

$$a(l) := \cos\left(\frac{l}{10} + l\right)$$

$$calc_post_1(n, a, a1, a2) := \left| \begin{array}{l} k1 \leftarrow 0 \\ k2 \leftarrow 0 \\ \text{for } i \in 1..n \\ \quad \left| \begin{array}{l} c \leftarrow a(i) \\ k1 \leftarrow k1 + 1 \text{ if } c < 0 \\ k2 \leftarrow k2 + 1 \text{ if } a1 \leq c \leq a2 \end{array} \right. \\ \left(\begin{array}{l} k1 \\ k2 \end{array} \right) \end{array} \right. calc_post_1(10, a, 0.5, 1) = \left(\begin{array}{l} 5 \\ 2 \end{array} \right) \blacksquare$$

Рис. 9.8. Реализация алгоритма примера 9.2.4

Замечание 9.1. В предыдущих примерах искомые характеристики вычислялись для всего заданного массива. Поэтому начальные значения искоемых характеристик задавались до начала цикла (в случае матрицы – до начала внешнего цикла). В ряде задач обработки матриц необходимо вычислить некоторые характеристики для каждой строки или каждого столбца матрицы. В этом случае начальные значения задаются в теле внешнего цикла до начала внутреннего цикла, в теле которого вычисляются характеристики. Этот важный момент иллюстрируется следующим примером.

Пример 9.2.5. Дана матрица D размерности $n \times m$. Составить П-Ф, формирующую вектор, i -й элемент которого равен количеству положительных элементов в i -й строке матрицы D .

Решение. Описание П-Ф приведено на рис. 9.9. Здесь иско-
мая характеристика (количество положительных элементов) вы-
числяется для каждой строки матрицы. Поэтому нулевое на-
чальное значение для переменной k задается в теле внешнего
цикла до начала внутреннего цикла.

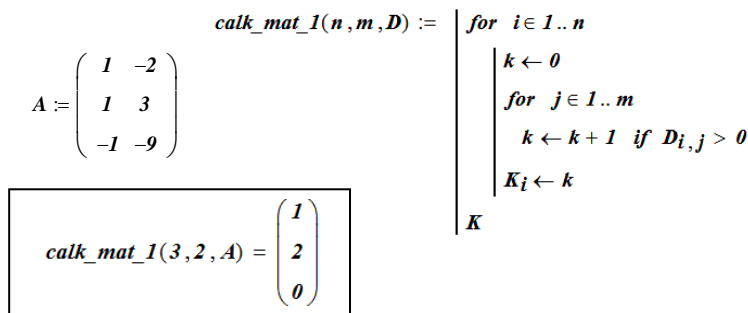


Рис. 9.9. Реализация алгоритма примера 9.2.5

Задание 9.2.4. Даны две матрицы A, B размерности $n \times m$. Составить П-Ф, подсчитывающую число случаев $A_{i,j} \geq B_{i,j}$.

Задание 9.2.5. Дана матрица D размерности $n \times m$. Составить П-Ф, которая вместо последнего элемента i -й строки записывает количество элементов i -й строки матрицы D , удовлетворяющих условию $|D_{i,j}| \geq \delta$, где δ – задаваемая величина.

Тип 3 (подсчет суммы или произведения элементов массива, удовлетворяющих заданным условиям)

Пример 9.2.6. Дан массив u , состоящий из n элементов. Составить П-Ф, определяющую:

- произведение строго положительных элементов массива;
- сумму элементов, находящихся в интервале $[a, b]$.

Решение. Описание П-Ф приведено на рис. 9.10. Обратите внимание, что начальные значения для переменных sum , pr задаются до начала цикла, в котором происходит «накопление» значений этих переменных. Обратите внимание на задание переменной pr начального значения равного 1. Почему не равного нулю?

$calc_sum1(n, y, a, b) :=$	$sum \leftarrow 0$
	$pr \leftarrow 1$
	$for\ i \in 1..n$
	$\quad \left \begin{array}{l} sum \leftarrow sum + y_i\ \text{if}\ a \leq y_i \leq b \\ pr \leftarrow pr \cdot y_i\ \text{if}\ y_i > 0 \end{array} \right.$
	$\left(\begin{array}{c} sum \\ pr \end{array} \right)$

Рис. 9.10. Реализация алгоритма примера 9.2.6

Пример 9.2.7. Дана матрица A размерности $n \times m$. Составить П-Ф, которая вместо первого элемента i -й строки записывает сумму всех элементов этой строки матрицы.

Решение. Описание П-Ф приведено на рис. 9.11.

Задание 9.2.6. Дан вектор x , состоящий из n элементов. Составить П-Ф, вычисляющую наибольшую из двух сумм элементов с четными и нечетными индексами.

Задание 9.2.7. Дан вектор x , состоящий из n элементов. Составить П-Ф, вычисляющую среднее арифметическое элементов вектора, а затем преобразующую исходный вектор по правилу: те элементы, которые меньше среднего арифметического, заменить нулями.

$$\begin{array}{l}
 \text{sum_mat_2}(n, m, A) := \begin{array}{l} \text{for } i \in 1..n \\ \quad \text{sum} \leftarrow 0 \\ \quad \text{for } j \in 1..m \\ \quad \quad \text{sum} \leftarrow \text{sum} + A_{i,j} \\ \quad A_{i,1} \leftarrow \text{sum} \end{array} \\
 \\
 \boxed{\text{sum_mat_2}(3, 2, A) = \begin{pmatrix} -1 & -2 \\ 4 & 3 \\ -10 & -9 \end{pmatrix}}
 \end{array}$$

Рис. 9.11. Реализация алгоритма примера 9.2.7

Задание 9.2.8. Дана матрица A размерности $n \times m$. Составить П-Ф, формирующую вектор, i -й элемент равен среднему арифметическому i -й строки матрицы.

Задание 9.2.9. Дана матрица A размерности $n \times m$. Составить П-Ф, вычисляющую сумму и количество только тех элементов матрицы, которые удовлетворяют условию $|A_{i,j}| \geq \varepsilon$, где ε – задаваемая величина.

Задание 9.2.10. Имеется экзаменационная ведомость студенческой группы из $n = 20$ человек по $m = 5$ дисциплинам. Оценки из этой ведомости занесены в матрицу размерности $n \times m$. Составить П-Ф, вычисляющую число студентов, получивших только четверки и пятерки.

Задание 9.2.11. Имеется экзаменационная ведомость студенческой группы из $n = 20$ человек по $m = 5$ дисциплинам. Оценки из этой ведомости занесены в матрицу размерности $n \times m$. Составить П-Ф, вычисляющую средний балл по каждой дисциплине.

*Тип 4 (вычисление максимального или минимального
элемента массива и его индекса)*

Пример 9.2.8. Дан массив y , состоящий из n элементов. Составить П-Ф, меняющую максимальный и минимальный элемент местами.

Решение. Описание П-Ф приведено на рис. 9.12. Начальные значения для искомых числовых характеристик (значения максимального и минимального элементов и их индексов) задаются из предположения, что первый элемент массива имеет максимальное и минимальное значение.

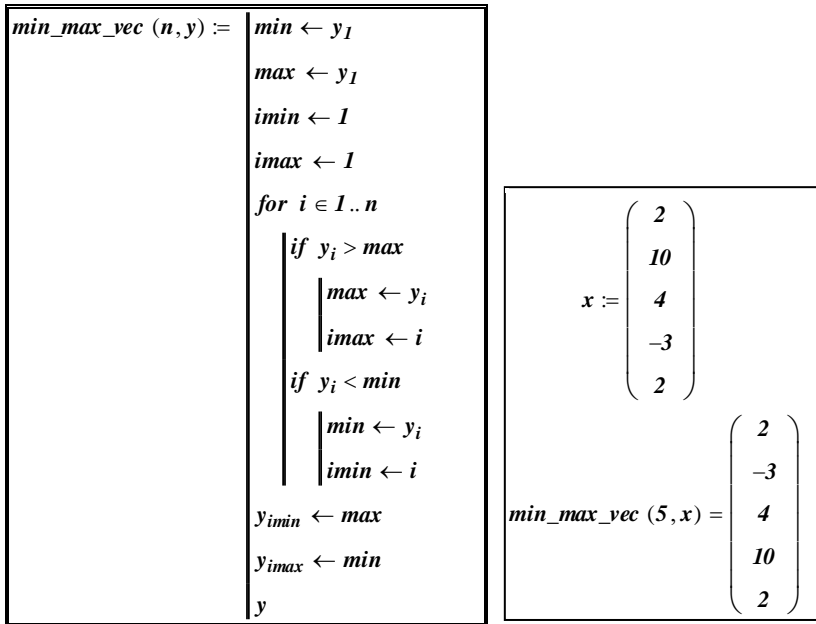


Рис. 9.12. Реализация алгоритма примера 9.2.8

Замечание 9.2. Для задания начальных значений при вычислении максимального или минимального элемента массива можно использовать следующее правило: начальному значению при вычислении максимального элемента задается заведомо наименьшее значение (например, 10^{-20}), а начальному значению при вычислении минимального элемента – заведомо наибольшее значение (например, 10^{20}).

Пример 9.2.9. Дана функция $f(x)$ и вектор x , состоящий из n элементов. Составить описание П-Ф, определяющей: а) максимальное значение этой функции на элементах этого вектора; б) номер индекса и значение элемента массива, на котором функция достигает максимального значения.

Решение. Описание П-Ф и обращение к ней приведены на рис. 9.13. Заметим, что в качестве третьего параметра П-Ф используется имя функции пользователя. Формальным параметром является функция $f(x)$, а фактическим – функция $\psi(x)$, описание которой должно находиться до обращения к П-Ф.

$$\begin{array}{l}
 \psi(x) := e^{\frac{-(x-3.5)^2}{3}} \\
 \boxed{\max_fun(6, x, \psi) = \begin{pmatrix} 0.92 \\ 3 \\ 3 \end{pmatrix}} \\
 \end{array}
 \quad
 \begin{array}{l}
 \max_fun(n, x, f) := \left\{ \begin{array}{l}
 x_{\max} \leftarrow x_1 \\
 i_{\max} \leftarrow 1 \\
 f_{\max} \leftarrow f(x_{\max}) \\
 \text{for } i \in 1..n \\
 \quad \text{if } f(x_i) > f_{\max} \\
 \quad \quad \left\{ \begin{array}{l}
 f_{\max} \leftarrow f(x_i) \\
 i_{\max} \leftarrow i \\
 x_{\max} \leftarrow x_i
 \end{array} \right. \\
 \end{array} \right. \\
 \left(\begin{array}{l} f_{\max} \\ x_{\max} \\ i_{\max} \end{array} \right)
 \end{array}
 \quad
 x := \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}$$

Рис. 9.13. Реализация алгоритма примера 9.2.9

Пример 9.2.10. Дана матрица A размерности $n \times m$. Составить П-Ф, формирующую вектор, i -й элемент равен номеру столбца, в котором на i -й строке матрицы стоит минимальный элемент. Описание П-Ф приведено на рис. 9.14.

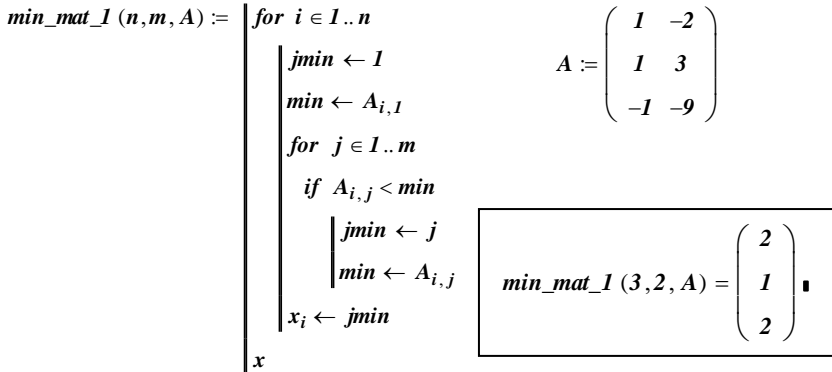


Рис. 9.14. Реализация алгоритма примера 9.2.10

Задание 9.2.12. Даны две матрицы A, B размерности $n \times m$. Вычислить величину $\max = \max A + \max B$, где $\max A, \max B$ – максимальные значения матриц A, B соответственно. Для вычисления значения максимального элемента составить П-Ф.

Задание 9.2.13. Дана матрица A размерности $n \times m$. Составить П-Ф, меняющую местами максимальный и минимальный элементы матрицы.

Задание 9.2.14. Дана матрица A размерности $n \times m$. Составить П-Ф, вычисляющую номер столбца, который содержит наименьший по модулю элемент этой матрицы.

Задание 9.2.15. Имеется экзаменационная ведомость студенческой группы из $n = 20$ человек по $m = 5$ дисциплинам. Оценки из этой ведомости занесены в матрицу размерности $n \times m$. Составить П-Ф, вычисляющую номер дисциплины (т.е. номер столбца), имеющей максимальный средний балл.

Задание 9.2.16. Имеется экзаменационная ведомость студенческой группы из $n = 15$ человек по $m = 4$ дисциплинам. Оценки из этой ведомости занесены в матрицу размерности $n \times m$. Составить П-Ф, вычисляющую номер студента (т.е. номер строки), имеющего максимальный средний балл по учебным дисциплинам.

Пример 9.2.11. Составить описание П-Ф, вычисляющей факториал числа $n \geq 0$ по формуле

$$n! = 1 \cdot 2 \cdot 3 \cdots n.$$

Решение. Описание П-Ф с использованием оператора произведения приведено на рис. 9.15.

$FI(n) := \begin{cases} 0 & \text{if } n \leq 0 \\ \prod_{i=1}^n i & \text{otherwise} \end{cases}$	$FI(5) = 120$
$FI(0) = 0$	$FI(-3) = 0$

Рис. 9.15. Вычисление факториала числа n

9.3. Программирование итерационных циклов

Напомним, что одной из особенностей итерационных циклов является невозможность априори (до выполнения программы) определить число повторений тела цикла (подробно см.

п. 7.3). Для организации таких циклов используют оператор *while* (см. п. 8.5).

Пример 9.3.1. Дан вектор v , состоящий из n элементов. Составить П-Ф, определяющую первый элемент этого массива, который удовлетворяет условию $|v_j - x| \leq \varepsilon$, где x, ε – заданные величины.

Решение. Описание П-Ф приведено на рис. 9.16. Однако приведенная П-Ф имеет существенный недостаток: при неправильном задании исходных данных (например, $\varepsilon < 0$) произойдет «зацикливание» – цикл по условию $|v_j - x| \leq \varepsilon$ не завершится.

Для устранения этого недостатка можно использовать оператор цикла *for* совместно с оператором *break*, который завершит цикл, как только выполнится условие $|v_j - x| \leq \varepsilon$. Описание такой П-Ф приведено на рис. 9.17.

$$search(v, x, \varepsilon) := \left| \begin{array}{l} j \leftarrow 1 \\ \text{while } |v_j - x| > \varepsilon \\ \quad j \leftarrow j + 1 \\ \left(\begin{array}{c} j \\ v_j \end{array} \right) \end{array} \right. \quad \boxed{\begin{array}{l} j := 1..500 \quad w_j := \text{ceil}(1000 \cdot \cos(j)) \\ \\ search(w, 400, 10^{-3}) = \left(\begin{array}{c} 397 \\ 400 \end{array} \right) \end{array}}$$

Рис. 9.16. Реализация алгоритма примера 9.3.1

$$search_1(v, x, \varepsilon) := \left| \begin{array}{l} \text{for } j \in 1..last(v) \\ \quad \text{break if } |v_j - x| \leq \varepsilon \\ \left(\begin{array}{c} j \\ v_j \end{array} \right) \end{array} \right. \quad \boxed{\begin{array}{l} search_1(w, 400, 10^{-3}) = \left(\begin{array}{c} 397 \\ 400 \end{array} \right) \blacksquare \end{array}}$$

Рис. 9.17. Реализация алгоритма примера 9.3.1

В предыдущих примерах была показана возможность замены оператора *while* (используемого при программной реализации итерационного цикла) другими конструкциями. Это обусловлено тем, что в этих примерах параметры цикла изменялись по закону арифметической прогрессии. Однако встречаются циклы, параметры которых изменяются по более сложным соотношениям. Особенно это характерно для алгоритмов решения нелинейных уравнений. В примере 9.3.2 рассматривается один из таких алгоритмов – метод последовательных предложений.

Пример 9.3.2. Дано нелинейное уравнение

$$x^2 - e^x = 0. \quad (9.3.1)$$

Необходимо вычислить с точностью $\varepsilon = 10^{-5}$ вещественный корень, лежащий в интервале $[-4, 0]$, используя метод последовательных приближений (называемый также методом простой итерации).

Решение. Не останавливаясь на теоретическом обосновании, приведем основные расчетные соотношения, необходимые для программной реализации этого метода.

Первоначально исходное уравнение $f(x) = 0$ заменяется эквивалентным ему уравнением

$$x = \varphi(x). \quad (9.3.2)$$

Затем выбираем начальное значение x_0 из некоторого интервала $[a, b]$, внутри которого находится только один корень уравнения $f(x) = 0$ (в нашем примере это интервал $[-4, 0]$). Подставляя это значение в правую часть (9.3.2), получаем первое приближение $x_1 = \varphi(x_0)$. Затем значение x_1 вновь подставляем в правую часть (9.3.2) и получаем $x_2 = \varphi(x_1)$.

Повторяя этот процесс, получаем последовательность чисел

$$x_n = \varphi(x_{n-1}) \quad (9.3.3)$$

(это оправдывает название – метод последовательных приближений). Возможны два случая.

Случай 1. Последовательность $x_0, x_1, x_2, \dots, x_n, \dots$ **сходится** к некоторому числу x^* , т.е. имеет предел, и тогда этот предел является решением нелинейного уравнения, т.е. $f(x^*) = 0$.

Случай 2. Последовательность $x_0, x_1, x_2, \dots, x_n, \dots$ **расходится**, т.е. не имеет предела, и в этом случае метод не применим.

Приведем условие, являющееся достаточным для сходимости метода последовательных приближений.

Пусть на отрезке $[a, b]$ имеется единственный корень уравнения $x = \varphi(x)$, и во всех точках этого отрезка справедливо неравенство

$$|\varphi'(x)| \leq q < 1. \quad (9.3.4)$$

Если при этом выполняется и условие

$$a \leq \varphi(x) \leq b, \quad (9.3.5)$$

то итерационный процесс (9.3.3) сходится, а за нулевое приближение можно брать любое $x_0 \in [a, b]$.

Заметим, что чем меньше величина q в (9.3.4), тем быстрее сходится последовательность $x_0, x_1, x_2, \dots, x_n, \dots$ к своему пределу x^* . В качестве приближенного значения корня можно принять приближение x_n , удовлетворяющее условию

$$|x_n - x_{n-1}| \leq \frac{\varepsilon \cdot (1 - q)}{q} = \eta, \quad (9.3.6)$$

где ε – заданная точность вычисления корня, входящая в неравенство

$$|x_n - x^*| \leq \varepsilon, \quad (9.3.7)$$

где x^* – точное значение корня.

Вернемся к уравнению нашего примера и преобразуем его к виду (9.3.2). Получаем $x = -e^{\frac{x}{2}}$ (знак минус появился из-за отрицательности искомого корня), т.е. $\varphi(x) = -e^{\frac{x}{2}}$. Проверим выполнение условия (9.3.4) графическим способом, используя графику MathCAD (см. рис. 9.18).

$$\phi(x) := -e^{\frac{x}{2}} \quad d\phi(x) := \frac{d}{dx} \phi(x) \quad x := -4, -3.95 \dots 0$$

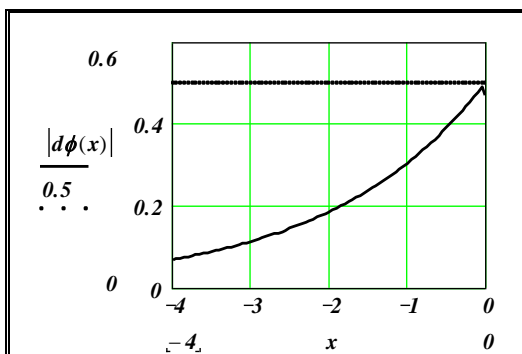


Рис. 9.18. Проверка условия (9.3.4)

Из графика видно, что в качестве величины q в условии (9.3.4) можно принять значение 0.5, которое удовлетворяет условию (9.3.4).

Для проверки условия (9.3.5) построим график функции $\varphi(x)$ на интервале $[-4, 0]$ (см. рис. 9.1). Видно, что условие (9.3.5) также выполняется – на всем интервале $[-4, 0]$ функция $\varphi(x)$ удовлетворяет условию $-4 < \varphi(x) < 0$.

Вывод: метод последовательных приближений для любого $x_0 \in [-4, 0]$ сходится к точному решению нелинейного уравнения (9.3.1).

Определим величину η в условии окончания итераций (9.3.6):

$$\eta := \frac{10^{-6} \cdot (1 - 0.5)}{0.5} \quad \eta = 1 \times 10^{-6}$$

Описание П-Ф, реализующей метод последовательных приближений, приведено на рис. 9.20. Здесь же показаны вычисленное приближенное значение корня и его проверка. Имя функции $\phi(x)$ является формальным параметром. Для предотвращения «зацикливания» итерационной процедуры (9.3.3) используется оператор цикла *for* (обсуждение см. в примере 5.3.4). Заметим, что переменные x_0, x_n , используемые в теле П-Ф, являются простыми переменными, имена которых имеют нижние индексы (вводимые после нажатия клавиши $[\cdot]$ – «десятичная точка»).

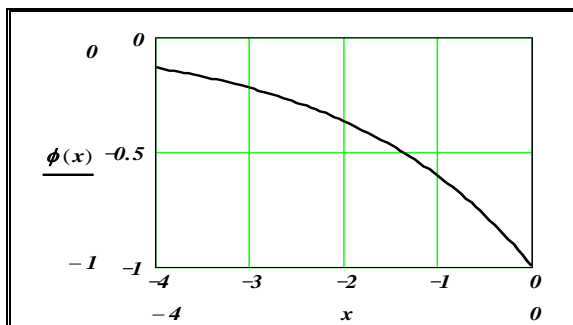


Рис. 9.19. Проверка условия (9.3.5)

9.4. Тестирование и отладка программ в пакете MathCAD

Этап тестирования и отладки разработанного программного обеспечения (ПО) является весьма важным и ответственным, от полноты выполнения которого напрямую зависит «жизнь» (этап эксплуатации ПО). Поэтому все традиционные среды программирования, как правило, имеют минимум три инструмента – трассировку программы, пошаговое выполнение (пооператорное) выполнение программы и наблюдение за значениями переменных при выполнении программы.

Одной из слабых сторон средств программирования ранних версий пакета MathCAD (до 13 версии) является полное отсутствие каких-либо инструментов для отладки. Отчасти это можно объяснить сравнительно малым объемом программы MathCAD, достаточной наглядностью и хорошей читаемостью текста программы.

$posl_prib(x_0, \phi, \eta) :=$	$ \begin{array}{l} ierr \leftarrow 1 \\ \text{for } n \in 1..10000 \\ \quad \left \begin{array}{l} x_n \leftarrow \phi(x_0) \\ \text{if } x_n - x_0 \leq \eta \\ \quad \left \begin{array}{l} ierr \leftarrow 0 \\ break \end{array} \right. \\ x_0 \leftarrow x_n \end{array} \right. \\ x_n \leftarrow \text{"not solve"} \text{ if } ierr = 1 \\ x_n \end{array} $
$f(x) := x^2 - e^x \quad \phi(x) := -e^{\frac{x}{2}} \quad x_0 := -2 \quad \eta := 10^{-6}$	
$x_{app} := posl_prib(x_0, \phi, \eta) \quad x_{app} = -0.70347 \quad f(x_{app}) = -3.01 \times 10^{-7}$	

Рис. 9.20. Программная реализация метода
последовательных приближений

Начиная с версий MathCAD 13/14, появились следующие инструменты отладки программы:

- две встроенных функции *trace* и *pause*;
- окно трассировки программы;
- панель с четырьмя кнопками отладки.

Обращение к этим инструментам отладки показаны на рис. 9.21.

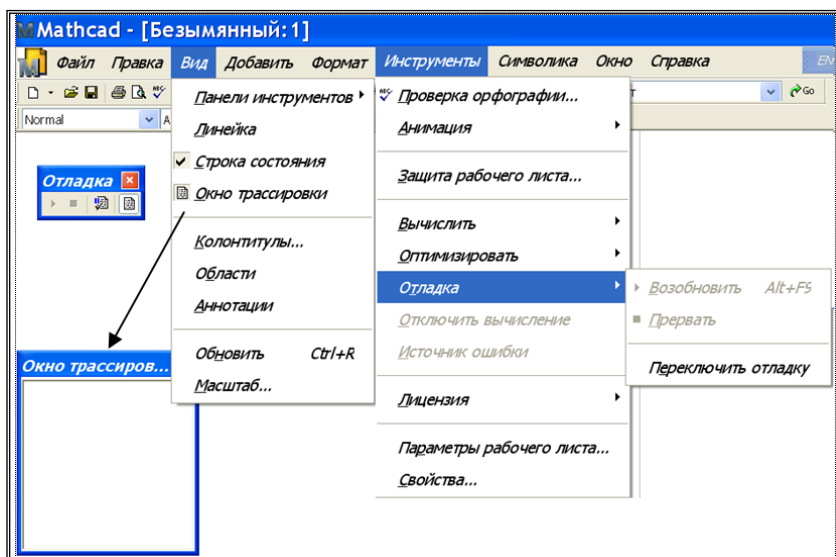


Рис. 9.21. Инструменты отладки пакета MathCAD

Встроенная функция *trace*. Эта функция возвращает строку, содержащую значения аргументов x , y , z , которая выводится в *Окне трассировки* (см. рис. 9.21). Обращение к этой функции имеет вид:

$$\text{trace}(S, x, y, z).$$

Параметр S задает очередность вывода информации в окне трассировки и сопровождающий текст. Если выводится значение

только одной переменной, то параметр S в обращении к функции можно опустить.

Пример 9.4.1. Запрограммировать в виде П-Ф алгоритм вычисления суммы только положительных элементов одномерного массива. Осуществить вывод значения переменных, входящих в тело цикла накопления суммы.

Решение. Описание П-Ф приведено на рис. 9.22. В тело цикла вставлена функция *trace*, в которой задаются имена выводимых переменных и порядок их вывода (первый параметр этой функции). В окне трассировки показаны значения переменных и заданные пояснения.

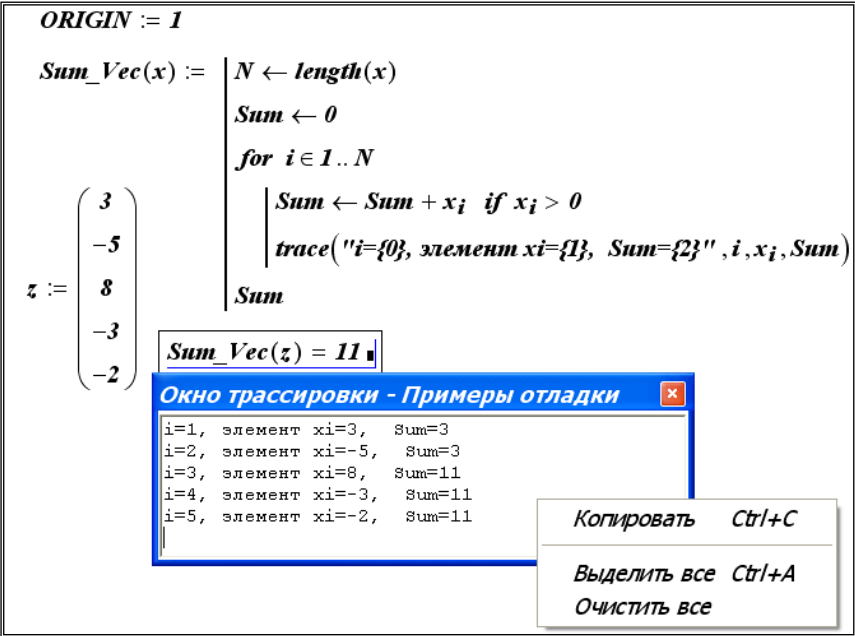


Рис. 9.22. Использование *Окна трассировки* (к примеру 9.4.1)

Замечание 9.4.1. Отметим следующие моменты, важные при работе с функцией *trace*:

- для получения данных в *Окне трассировки* необходимо пересчитать весь документ MathCAD (нажать клавиши [Ctrl+F9] даже при установленном автоматическом режиме вычислений);
- для работы с данными *Окна трассировки* можно использовать команды контекстного меню, показанные на рис. 9.22;
- в параметре *S* порядковый номер вывода значений (номер задается в фигурных скобках $\{\}$) обязательно должен начинаться с нулевого значения (независимо от значения системной переменной *ORIGIN*);
- пояснения к выводимым значениям, задаваемые в параметре *S*, могут содержать пробелы и другие символы (включая русские буквы), что повышает читаемость информации, выводимой в *Окне трассировки*.

Встроенная функция *pause*. Эта функция возвращает строку (также как функция *trace*), но после этого выполнение вычисления прерывается. Для продолжения вычислений необходимо нажать кнопку 1 палитры инструментов *Отладка* (см. рис. 9.22). Обращение к этой функции имеет вид:

$$pause(S,x,y,z).$$

Формальные параметры имеют то назначение, что и параметры функции *trace*.

Пример 9.4.2. Запрограммировать в виде П-Ф алгоритм вычисления суммы только положительных элементов одномерного массива. Осуществить вывод значения переменных, входящих в тело цикла накопления суммы, с использованием функции *pause*.

Решение. Описание П-Ф приведено на рис. 9.23. В тело цикла вставлена функция *pause*. В окне трассировки показаны значения переменных после двух выполнений тела цикла.

Палитра инструментов Отладка. Эта палитра (показана на рис. 9.23) содержит четыре инструмента, которые пронумерованы следующим образом:

1. *Возобновить вычисления* (при нажатии кнопки процесс вычисления продолжается – часто используется с функцией *pause*).

2. *Прервать вычисления* (прекращается процесс вычисления – для продолжения нажать клавиши [Ctrl+F9]).

3. *Переключение режима отладки* (нажать на кнопку для включения режима отладки).

4. *Включение или выключение окна трассировки* (при нажатии кнопки появляется окно трассировки).

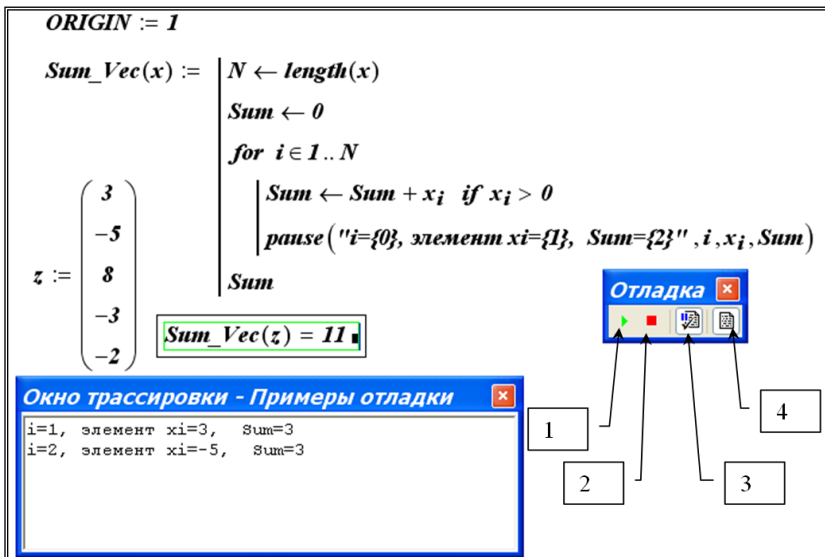


Рис. 9.23. Использование функции *pause*

Заметим, что использование окна трассировки существенно облегчает отладку итерационных циклов. Это иллюстрирует следующий пример.

Пример 9.4.3. Используя функцию *root*, найти корень нелинейного уравнения $f(x) = x^2 - 4 = 0$ на интервале локализации $[0, 5]$ с промежуточной выдачей приближенных значений корня и соответствующих значений $f(x)$.

Решение. Напомним, что одно из двух возможных обращений к функции *root* имеет вид $root(f(x), x, a, b)$, где $f(x)$ – задает нелинейное уравнение, переменные a, b – границы интервала локализации, внутри которого должен находиться только один корень нелинейного уравнения $f(x) = 0$. Фрагмент документа, содержащий обращение к функции показан на рис. 9.24.

$$\begin{array}{l}
 ff(x) := x^2 - 4 \\
 f(x) := \left| \begin{array}{l} trace("x=\{0\}, f(x)=\{1\} \text{ "}, x, ff(x)) \\ x^2 - 4 \end{array} \right. \\
 x := root(f(x), x, 0, 5) = 2 \quad ff(x) = 1.776 \times 10^{-15}
 \end{array}$$

Рис. 9.24. Обращение к функции *root* (к примеру 9.4.3)

Для организации вывода промежуточных значений в П-Ф, задающей $f(x)$, вставлена функция *trace*. Результат работы этой функции – окно трассировки – показан на рис. 9.25.

Используя этот прием, можно организовать вывод промежуточных значений во многих функциях MathCAD, реализующих итерационные алгоритмы.

```
Окно трассировки - Примеры отладки
x=0, f(x)=-4
x=5, f(x)=21
x=2.5, f(x)=2.25
x=1.90396043932073, f(x)=-0.374934645501612
x=2.20198021966037, f(x)=0.84871688777551
x=1.99972478128844, f(x)=-0.00110079910089977
x=2.1008525004744, f(x)=0.413581228749552
x=1.9999998333151, f(x)=-6.66739558230489e-07
x=2.05042616689475, f(x)=0.204247465886709
x=1.99999999997416, f(x)=-1.03341335488949e-10
x=2.02521308343446, f(x)=0.101488033314109
x=2, f(x)=-4.44089209850063e-15
x=2.01260654171723, f(x)=0.0505850917629829
x=2, f(x)=-8.88178419700125e-16
x=2.00630327085861, f(x)=0.025252814657974
x=2, f(x)=-8.88178419700125e-16
x=2.00315163542931, f(x)=0.0126164745231065
```

Рис. 9.25. Результаты работы функции *trace* (к примеру 9.4.3)

Вопросы и задания для самоконтроля

1. Когда используется оператор цикла *while*? Нарисовать блок-схему этого оператора цикла.
2. Для чего используется оператор *break*?
3. Как работает функция *error*?
4. В каких задачах обработки матриц начальные значения искомых характеристик матрицы задаются до внешнего цикла? Привести пример такой задачи.
5. Матрица A размером $n \times k$ содержит экзаменационные оценки n студентов по k учебным дисциплинам. Запрограммировать П-Ф, вычисляющую номер студента, имеющего максимальный средний балл, и величину максимального среднего балла. Выполнить тестирование составленной П-Ф.

6. В каких задачах обработки матриц начальные значения искомых характеристик матрицы задаются в теле внешнего цикла до начало внутреннего цикла? Привести пример такой задачи.

7. Дана матрица A размерности $n \times m$. Составить П-Ф, которая вместо последнего элемента i -й строки записывает сумму всех элементов этой строки матрицы.

8. Составить П-Ф, реализующую итерационный цикл суммирования следующего сходящегося ряда

$$\sum_{n=1}^{\infty} (-1)^{n+1} \frac{\sin(nx)}{n} \quad \text{при } x = 0.75$$

с точностью $\varepsilon = 10^{-6}$. Считается, что заданная точность суммирования достигнута, если последнее слагаемое a_n , включенное в сумму, удовлетворяет условию $|a_n| > \varepsilon$, а последующее $|a_{n+1}| \leq \varepsilon$. Определить число итераций, необходимых для вычисления суммы с заданной точностью. Осуществить тестирование составленной П-Ф.

Рекомендация: для прекращения итерационного цикла используйте команду **break**.

9. Какие средства отладки предусмотрены в версии MathCAD15?

10. Чем отличается выполнение функций *trace* и *pause*?

ЗАКЛЮЧЕНИЕ

В данном пособии авторы стремились изложить основы вычислений и программирования в пакете MathCAD. К сожалению, ограниченный объем учебного пособия не позволил продемонстрировать возможности пакета для решения часто встречающихся задач. Это, прежде всего, относится к решению задач вычислительной математики, обработке экспериментальных данных в пакете MathCAD, включая задачи математической статистики и регрессионного анализа. Некоторым оправданием этого может служить хорошее освещение этих разделов в литературе, в том числе приводимой ниже:

- решение задач математического анализа в пакете MathCAD [4, 7, 8, 9];
- решение задач вычислительной математики в пакете MathCAD [3, 10, 11];
- решение задач теории вероятностей и математической статистики в пакете MathCAD [4, 12];
- решение задач регрессионного анализа экспериментальных данных и моделирования в пакете MathCAD [3, 13, 14];
- решение других задач в пакете MathCAD [3, 7, 8, 9].

Авторы желают читателям (особенно студентам, магистрантам и аспирантам), чтобы пакет MathCAD стал для них дружелюбным и незаменимым помощником в решении широкого круга учебных заданий и научно-инженерных задач.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Очков В. Ф. MathCAD 14 для студентов и инженеров / В. Ф. Очков. – СПб. : БХВ-Петербург, 2009. – 512 с.
2. Воскобойников Ю. Е. Основы работы в пакете MathCAD : учеб. пособие / Ю. Е. Воскобойников, А. Ф. Задорожный ; Новосиб. гос. архитектур.-строит. ун-т. – Новосибирск : НГАСУ (Сибстрин), 2006. – 116 с. (электронная версия – <ftp://ftp.sibstrin.ru/prikl/archives/mathcad2006.zip>).
3. Воскобойников Ю. Е. Программирование и решение задач в пакете MathCAD : учеб. пособие / Ю. Е. Воскобойников, В. Ф. Очков ; Новосиб. гос. архитектур.-строит. ун-т. – Новосибирск : НГАСУ (Сибстрин), 2002. – 136 с. (электронная версия – <ftp://ftp.sibstrin.ru/prikl/archives/math03.zip>).
4. Кирьянов Д. В. Самоучитель MathCAD 12 / Д. В. Кирьянов. – СПб. : БХВ-Петербург, 2004. – 576 с.
5. Очков В. Ф. Советы пользователям MathCAD / В. Ф. Очков. – М. : Изд-во МЭИ, 2006. – 196 с.
6. Ершова Е. Е. Лабораторный практикум по современным компьютерным технологиям. Ч. 3. MathCAD : учеб. пособие / Е. Е. Ершова, И. В. Ершов ; Новосиб. гос. архитектур.-строит. ун-т (Сибстрин). – Новосибирск : НГАСУ (Сибстрин), 2007. – 52 с. (электронная версия – <ftp://ftp.sibstrin.ru/prikl/archives/skt-3.zip>).
7. Гурский Д. Вычисления в MathCAD 12 / Д. Гурский. – СПб. : Питер, 2006. – 544 с.
8. Дьяконов В. П. MathCAD 8 Professional в математике, физике и Internet / В. П. Дьяконов, И. В. Абраменконова. – М. : Нолидж, 1999. – 512 с.

9. Черняк В. А. Математика для экономистов на базе MathCAD / В. А. Черняк, В. А. Новиков. – СПб. : БХВ-Петербург, 2003. – 496 с.
10. Поршнев С. В. Численные методы на базе MathCAD / С. В. Поршнев. – СПб. : БХВ-Петербург, 2005. – 464 с.
11. Бедарев И. А. Численные методы решения инженерных задач в пакете MathCAD : учеб. пособие / И. А. Бедарев, О. Н. Белоусова, Н. Н. Федорова ; Новосиб. гос. архитектур.-строит. ун-т. – Новосибирск : НГАСУ (Сибстрин), 2005. – 96 с. (электронная версия – <ftp://ftp.sibstrin.ru/prikl/archives/mathcad2005.zip>).
12. Ивановский Р. И. Теория вероятностей и математическая статистика с примерами и задачами в среде MathCAD / Р. И. Ивановский. – СПб. : БХВ-Петербург, 2008. – 528 с.
13. Воскобойников Ю. Е. Регрессионный анализ данных в пакете MathCAD / Ю. Е. Воскобойников. – СПб. : Лань, 2011. – 286 с.
14. Воскобойников Ю. Е. Построение регрессионных моделей в пакете MathCAD : учеб. пособие / Ю. Е. Воскобойников ; Новосиб. гос. архитектур.-строит. ун-т. – Новосибирск : НГАСУ (Сибстрин), 2009. – 220 с.

