



**Ю.Е. ВОСКОБОЙНИКОВ
А.Ф. ЗАДОРЖНЫЙ**

ВЫЧИСЛЕНИЯ И ПРОГРАММИРОВАНИЕ В ПАКЕТЕ MATHCAD PRIME 2.0

НОВОСИБИРСК 2013

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ АРХИТЕКТУР-
НО-СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ (СИБСТРИН)

Ю.Е. Воскобойников, А.Ф. Задорожный

ВЫЧИСЛЕНИЯ И ПРОГРАММИРОВАНИЕ В ПАКЕТЕ MATHCAD PRIME 2.0

*Рекомендовано Новосибирским региональным
отделением УМО вузов Российской Федерации
по образованию в области строительства в качестве
учебного пособия для студентов, обучающихся
по направлению 270800.62 «Строительство»*

НОВОСИБИРСК 2013

УДК 004.9
ББК 32.9
В 762

Воскобойников Ю. Е.

Вычисления и программирование в пакете MathCAD Prime 2.0 : учеб. пособие / Ю. Е. Воскобойников, А. Ф. Задорожный ; Новосиб. гос. архитектур.-строит. ун-т (Сибстрин). – Новосибирск : НГАСУ (Сибстрин), 2013. – 196 с.

ISBN 978-5-7795-0643-4

В учебном пособии рассмотрены основные конструкции встроенного языка математического пакета. Изложены основы работы в пакете MathCAD версии Prime 2.0 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 самоконтроля полученных знаний.

Учебное пособие предназначено для студентов, обучающихся по направлениям «Строительство», «Информационные системы и технологии», при освоении учебных курсов «Информатика», «Математические пакеты». Учебное пособие также 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);
- С.Н. Зеркаль, д-р техн. наук, профессор (НГАСУ (Сибстрин));
- Е.М. Григорьев, канд. техн. наук, доцент (НГАВТ)

ISBN 978-5-7795-0643-4

- © Воскобойников Ю.Е.,
Задорожный А.Ф., 2013
- © Новосибирский государственный
архитектурно-строительный
университет (Сибстрин), 2013

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	8
Раздел 1. ВЫЧИСЛЕНИЯ В ПАКЕТЕ MathCAD	
Тема 1. МАТЕМАТИЧЕСКИЕ ПАКЕТЫ. ОБЩАЯ ХАРАКТЕРИСТИКА ПАКЕТА MathCAD	
1.1. Современные математические пакеты	9
1.2. Возможности и структура пакета MathCAD	10
1.3. Программное окно MathCAD Prime 2.0	12
1.4. Лента и ее содержимое	16
1.5. Работа с документами MathCAD	20
Вопросы и задания для самоконтроля по теме 1	22
Тема 2. ВЫЧИСЛЕНИЯ В ПАКЕТЕ MathCAD	
2.1. Константы, переменные, операторы присваивания и вывода ...	24
2.2. Арифметические операции MathCAD	30
2.3. Встроенные функции и функции пользователя	33
2.4. Операторы математического анализа	35
Вопросы и задания для самоконтроля по теме 2	38
Тема 3. МАССИВЫ В ПАКЕТЕ MathCAD	
3.1. Массивы в пакете MathCAD	42
3.2. Создание массивов	43
3.3. Верхний индекс массива	46
3.4. Основные функции обработки массивов	47
3.5. Вычисления с массивами	52
3.6. Функции сортировки элементов векторов и матриц	56
Вопросы и задания для самоконтроля по теме 3	57
Тема 4. ГРАФИКА В ПАКЕТЕ MathCAD	
4.1. Основные инструменты для построения графиков	60
4.2. Построение графиков функции одной переменной в декартовой системе координат	61
4.3. Построение графиков функции одной переменной в полярной системе координат	69
4.4. Построение графиков функций двух переменных	71
4.5. Построение диаграмм	74
Вопросы и задания для самоконтроля по теме 4	77
Тема 5. СИМВОЛЬНЫЕ И ЧИСЛЕННЫЕ ВЫЧИСЛЕНИЯ В MathCAD	
5.1. Команды символьных вычислений	78
5.2. Выполнение символьных вычислений	82
5.3. Символьное решение уравнений и систем уравнений	88
5.4. Численное решение уравнений и систем уравнений	
5.4.1. Вычисление корней полиномов	95

5.4.2. Решение систем линейных уравнений	95
5.4.3. Решение систем нелинейных уравнений	97
5.4.4. Поиск экстремальных значений	101
5.5. Решение дифференциальных уравнений	102
5.6. Решение систем обыкновенных дифференциальных уравнений	105
Вопросы и задания для самоконтроля по теме 5	108
Тема 6. ЧТЕНИЕ И ЗАПИСЬ ДАННЫХ	
6.1. Средства доступа к файлам	110
6.2. Функции доступа к структурированным ASCII-файлам	113
6.3. Функции доступа к графическим файлам	116
Вопросы и задания для самоконтроля по теме 6	118
Раздел 2. ПРОГРАММИРОВАНИЕ В ПАКЕТЕ MathCAD	120
Тема 7. БЕЗМОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ В ПАКЕТЕ MathCAD	
7.1. Программирование линейных алгоритмов	121
7.2. Программирование разветвляющихся алгоритмов	122
7.3. Программирование циклических алгоритмов	133
Вопросы и задания для самоконтроля по теме 7	141
Тема 8. МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ В ПАКЕТЕ MathCAD	
8.1. Сущность и преимущества модульного программирования ...	143
8.2. Описание подпрограммы-функции и локальный оператор присваивания	145
8.3. Обращение к подпрограмме-функции MathCAD	148
8.4. Программирование линейных алгоритмов в подпрограмме-функции MathCAD	149
8.5. Программирование разветвляющихся алгоритмов в подпрограмме-функции MathCAD	150
8.6. Программирование циклических алгоритмов в подпрограмме-функции MathCAD	158
Вопросы и задания для самоконтроля по теме 8	173
Тема 9. ПРОГРАММИРОВАНИЕ ТИПОВЫХ ЗАДАЧ В ПОДПРОГРАММАХ-ФУНКЦИЯХ MathCAD	
9.1. Программирование разветвляющихся алгоритмов	176
9.2. Программирование циклов типа арифметической прогрессии	179
9.3. Программирование итерационных циклов	191
Вопросы и задания для самоконтроля по теме 9	195
ЗАКЛЮЧЕНИЕ	197
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	198

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. Для выполнения научно-инженерных расчетов любой сложности в настоящее время часто используют пакет MathCAD. В 2012 году фирмой PTC разработана новая версия этого пакета – MathCAD Prime 2.0, которая имеет принципиально новый интерфейс и новые функциональные возможности.

Изучению основ вычислений и программирования в пакете MathCAD (версии Prime 2.0) и посвящается данное учебное пособие. Авторы надеются, что пакет MathCAD Prime 2.0 станет хорошим помощником при выполнении расчетов в курсовом и дипломном проектировании.

Раздел 1. ВЫЧИСЛЕНИЯ В ПАКЕТЕ MathCAD

Тема 1. МАТЕМАТИЧЕСКИЕ ПАКЕТЫ. ОБЩАЯ ХАРАКТЕРИСТИКА ПАКЕТА MathCAD

В этой теме делается краткое сравнение математических пакетов. Рассматриваются возможности, структура, программное окно пакета MathCAD Prime 2.0.

1.1. Современные математические пакеты

В настоящее время существует множество *систем компьютерной математики* или, проще, *математических пакетов*.

Наиболее популярными из них являются *Maple*, *MatLab*, *MathCAD*. Для выполнения научно-инженерных расчетов любой сложности в настоящее время часто используют пакет MathCAD. Чем объяснить популярность этого пакета?

Пакет 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 существенно упрощает его применение для решения конкретных научно-инженерных задач. В 2012 году фирмой PTC разработана новая версия этого пакета – MathCAD Prime 2.0, которая имеет принципиально новый интерфейс и новые функциональные возможности.

Фрагмент документа вычисления определенного интеграла вида $\int_0^1 \frac{1}{\sqrt{1-x^2}} dx$ в версии MathCAD Prime 2.0 показан на рис. 1.1.

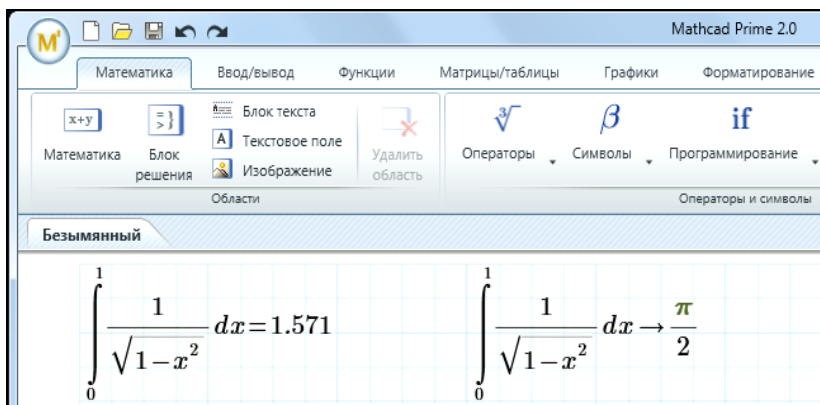


Рис. 1.1. Вычисление интеграла в пакете MathCAD Prime 2.0

Из рисунка видно, что запись операторов для вычисления определенного интеграла в пакете MathCAD Prime 2.0 (как и в более ранних версиях пакета) очень близка к стандартному языку математических расчетов. Для краткости в дальнейшем изложении пакет MathCAD Prime 2.0 будем называть MathCAD. MathCAD позволяет записывать на экране компьютера формулы так, 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 математическим 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 с анимацией).

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, символьный и графический процессоры, справочную систему.

Текстовый редактор служит для ввода и редактирования текстов.

Вычислительный процессор осуществляет численные вычисления по заданным математическим формулам, имеет обширный набор встроенных математических функций, обеспечивает вычисления рядов, сумм и произведений, определенных интегралов и производных.

Символьный процессор позволяет получать результаты вычисления в символьном (аналитическом) виде. Второй интеграл на рис. 1.1 вычислен в символьном виде.

Графический процессор служит для создания графиков функций одной и двух переменных.

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

1.3. Программное окно MathCAD Prime 2.0

При запуске MathCAD Prime 2.0 открывается пустой документ, в который можно добавить текст, уравнения, графики и изображения. Пример окна приведен на рис. 1.1.

Программное окно MathCAD Prime 2.0 содержит следующие элементы документа и интерфейса пользователя: *кнопка Mathcad*, *лента*, *панель быстрого доступа*, *контекстные меню*, *рабочая область*, *строка состояния*.

Кнопка Mathcad  – содержит команды для работы с файлами, такие как: Открыть (Open), Сохранить (Save) и Печать (Print).

Лента – содержит вкладки, на которых сгруппированы связанные команды. Рабочую область можно настроить, уменьшив или увеличив ленту и добавив часто используемые команды на панель быстрого доступа. Дополнительные сведения о командах ленты можно получить из всплывающих подсказок и контекстной справки. На вкладках ленты отображаются контекстные меню, всплывающие подсказки и сочетания клавиш.

Панель быстрого доступа – содержит часто используемые команды. Команды можно добавлять и удалять с панели. Панель быстрого доступа может находиться под или над лентой.

Контекстные меню – если щелкнуть документ правой кнопкой мыши, можно получить доступ к часто используемым командам.

Строка состояния – содержит номера страниц документа, цветной кружок, показывающий состояние кнопок «Найти», «Заменить на» (Replace), «Параметры поиска» (Options), «Регулятор масштабирования» (Zoom Slider), «Представление страницы» (Page View) и «Режим черновика» (Draft View).

Работа со справкой и информационным центром. Справочный центр предоставляет доступ к справке и примерам MathCAD Prime 2.0. Команды для работы со справочной системой сосредоточены в меню *Help (Справка)*, которое показано на рис. 1.2.

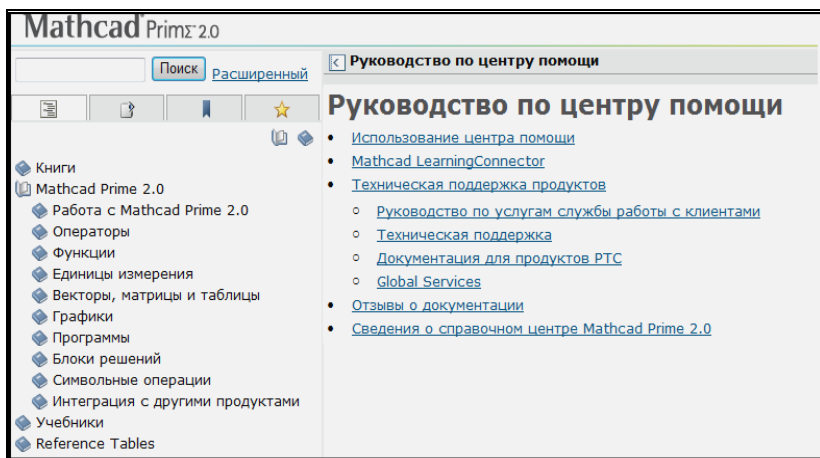


Рис. 1.2. Позиции строки меню *Help*

Центр помощи открывается щелчком значка  в правом верхнем углу ленты MathCAD или нажатием клавиши [F1]. Окно справочной системы имеет четыре вставки: *Книги*, *Mathcad Prime 2.0*, *Учебники*, *Справочные Таблицы* (*Reference Tables*).

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.

MathCAD Prime 2.0 содержит практические примеры, разработанные для того, чтобы продемонстрировать использование MathCAD, а также встроенных функций и операторов. Эти примеры позволяют лучше понять принципы работы с функциями или изучить различные типы графиков, которые могут улучшить визуализацию и анализ результатов расчета. Содержание подраздела *Индекс тем примеров* раздела *Руководство по центру помощи* MathCAD Prime 2.0 показано на рис. 1.3.

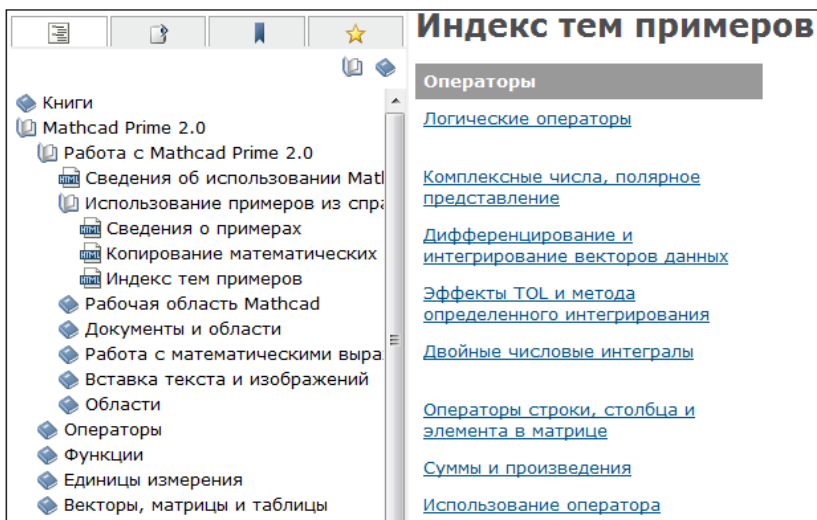


Рис. 1.3. Содержание подраздела *Индекс тем примеров*

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. 1.4 показано окно с перечнем доступных таблиц.

Рабочее поле программного окна MathCAD находится под лентой и в нем располагается одно или несколько окон документов MathCAD. Можно открыть одновременно несколько документов. Каждый открытый документ помещается на новую вкладку под лентой. Чтобы перейти в нужный документ, необходимо щелкнуть его вкладку.

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.

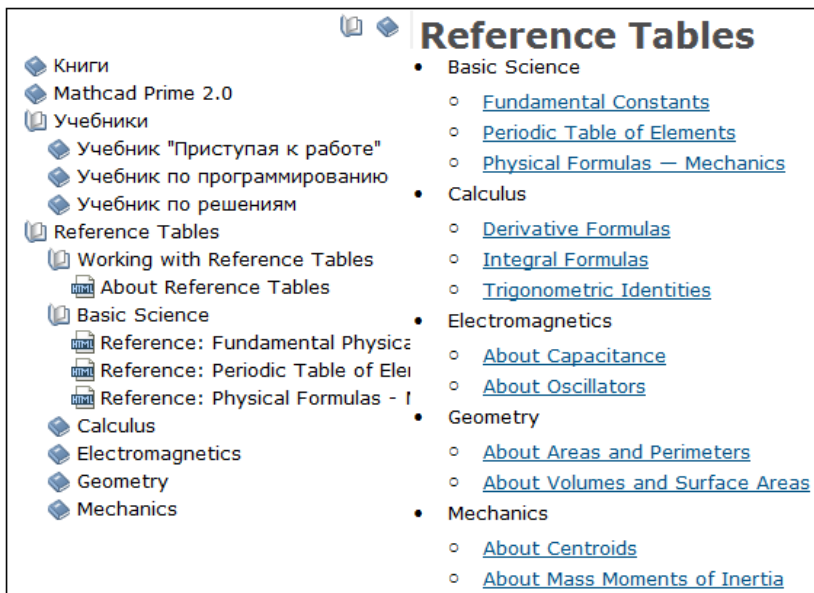


Рис. 1.4. Окно списка справочных таблиц

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 *синего креста*.

Текстовая область может включать любые символы (русские, латинские, греческие) в виде комментариев к вычислительным фрагментам. Для создания текстовой области необходимо обратиться в ленте к пункту *Документ* (рис. 1.5).

В документ можно добавлять текст и изображения. Доступны следующие типы текстовых областей:

Блок текста. Располагается по всей ширине страницы. Эта область не перекрывается с другими областями документа. При перемещении блока текста или добавлении либо удалении строк другие области документа смещаются вверх или вниз. Блоки

текста используются для вставки больших параграфов содержимого на страницу.

Текстовое поле. Эту область можно перемещать по документу, и она может перекрываться другими областями. Текстовые окна используются для размещения кратких примечаний, аннотаций и подзаголовков для математических выражений.

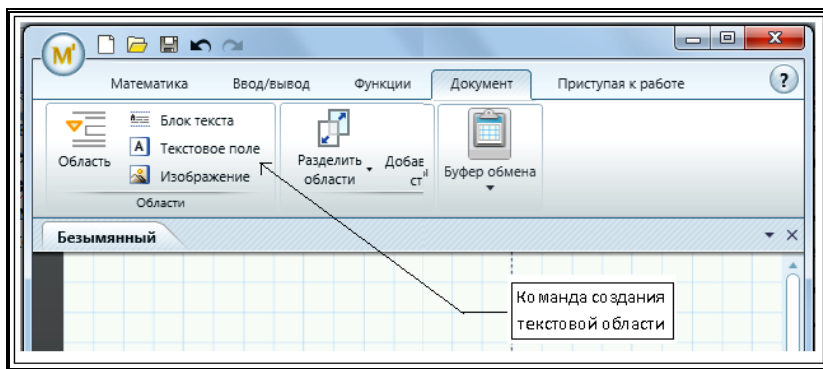


Рис. 1.5. Создание текстовой области

Если в документе содержится много страниц, можно воспользоваться функциями *Найти* и *Заменить на* для поиска текстовых и математических строк.

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

1.4. Лента и ее содержимое

Лента содержит строку меню с вкладками: *Математика*, *Ввод и вывод*, *Матрицы/таблицы*, *Графики*, *Форматирование*, *Расчет*, *Документ* (рис. 1.6).

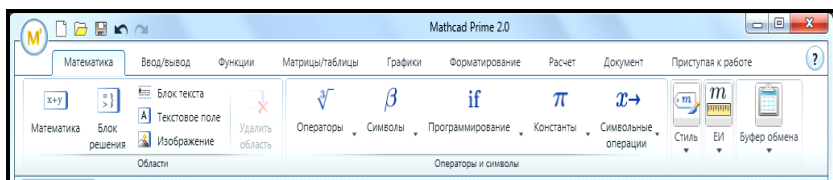


Рис. 1.6. Внешний вид ленты

Вкладка *Математика* содержит пять разделов: Операторы, Символы, Программирование, Константы, Символьные операции. Вид кнопок и названия палитр инструментов приведены на рис. 1.7.

Раздел *Операторы* содержит математические операторы, которые распределены по категориям: *Алгебра*, *Векторы и матрицы*, *Математический анализ*, *Определение и вычисление*, *Проектирование*, *Сравнение*, которые показаны на рис. 1.8.

С помощью представленных знаков можно вводить в документы MathCAD практически все известные математические символы и операторы.

Раздел *Символы* содержит прописные и строчные буквы греческого алфавита, математические символы и символы денежных единиц.

Раздел *Операторы* содержит операторы (см. рис. 1.7), используемые при реализации модульного программирования.

Раздел *Символьные операции* содержит оператор символического вывода $\rightarrow$, ключевые слова, определяющие функции символических вычислений, а также модификаторы – специальные элементы, которые используются для изменения поведения ключевых слов (рис. 1.9).

Раздел *Единицы измерения* содержит полный набор единиц измерения по международной системе СИ, американской системе единиц (USCS) и системе «сантиметр-грамм-секунда» (рис. 1.10).

	←	if	else	else if
also if	while	for	break	continue
return	try			

Рис. 1.7. Операторы модульного программирования

Алгебра				
$ x $	$+$	$,$	$/$	x^y
x'	$()$	$\div$	$\cdot$	$-$
$\sqrt{x}$	$\%$			
Векторы и матрицы				
$\mathbf{x}$	$\ x\ $	$[]$	$M^{\oplus}$	$M,$
$M^{\widehat{}}$	M^T	$1 \dots n$	$1, 3 \dots n$	$\vec{V}$
Математический анализ				
$\oplus$	$*$	d/dx	$\int dx$	$\lim$
$=$	f'	Π	Σ	
Определение и вычисление				
$:=$	$=$	$\rightarrow$		
Проектирование				
$\bar{z}$	$\circ$	$\angle$	$\circ$	
Сравнение				
$\in$	$=$	$\oplus$	$>$	$\geq$
$<$	$\leq$	$\wedge$	$\neg$	$\vee$
$\neq$				

Рис. 1.8. Значки раздела *Операторы* вкладки *Математика*

1.5. Работа с документами 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 следующим способом: установите курсор в нужном месте документа; нажмите [Ctrl+T], чтобы вставить текстовое поле, или [Ctrl+Shift+T], чтобы вставить блок текста, либо на вкладке *Математика* в разделе Области щелкните Блок текста или Текстовое поле.

Вводимый текст будет заключен в прямоугольник, отображающий текстовую область. Введите или вставьте текст в поле. Текстовая область расширяется по мере ввода текста. Для ввода пустой строки нажмите клавишу [Enter], для перемещения на новую строку внутри текстового поля нажмите [Shift+Enter]. 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. Чтобы скопировать часть документа MathCAD, необходимо выделить одну или несколько областей, нажать [Ctrl+C] или [Ctrl+X]. Затем щелкнуть на пустом месте того же или другого документа и нажать [Ctrl+V], чтобы вставить области.

Кроме того, можно на вкладке *Математика* в разделе *Буфер обмена* выбрать команду Копировать или Вырезать, а затем Вставить.

Редактирование. Для редактирования текстовой области необходимо щелкнуть на ней мышью. Появится курсор в виде вертикальной черты. Тип шрифта, размер, стиль и цвет могут быть изменены с помощью вкладки *Форматирование* (рис. 1.11).

В разделе Шрифт можно выбрать шрифт (например, вид шрифта или цвет).

В разделе Параграф нажать требуемую кнопку, чтобы форматировать текст различными способами, например, используя маркированный список или отступ.

Примечание. Настройки стиля шрифта, заданные на вкладке *Форматирование*, будут применяться ко всем новым текстовым областям в документе.

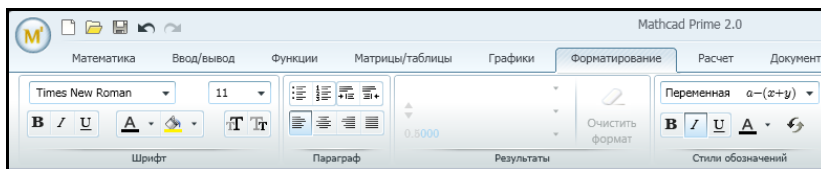


Рис. 1.11. Вкладка *Форматирование*

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 синего крестика. С этого места можно начинать набор формул.

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

Сохранение документа MathCAD. Для сохранения созданного документа MathCAD в файле необходимо использовать кнопку Mathcad и далее команду Сохранить как. В появившемся диалоговом окне выбрать папку, где будет сохранен новый файл, и назначить ему имя.

Вопросы и задания для самоконтроля по теме 1

1. Основные особенности и возможности математического пакета MathCAD Prime 2.0.
2. Что размещается в математической и текстовой областях документа MathCAD?
3. В каких форматах можно сохранить документ MathCAD Prime 2.0?
4. Какие панели инструментов присутствуют в пакете MathCAD?
5. Как можно поменять размер шрифта в документе MathCAD Prime 2.0?
6. Выполните следующие операции:
 - загрузите пакет MathCAD Prime 2.0;
 - откройте программное окно на весь экран;
 - расположите панели, как показано на рис. 1.1;
 - присвойте файлу имя и сохраните его в заданной папке.
7. Выполните следующие операции:
 - загрузите два документа MathCAD;
 - расположите их на экране монитора один над другим, один с другим бок о бок.
8. Выполните следующие операции:
 - в справочнике найдите названия команд, выполняющих функции сортировки;
 - в справочном центре найдите электронный документ с описанием применения этих команд;
 - найдите в справочном центре примеры использования команд сортировки.
9. Выполните следующие операции:
 - позиционируйте курсор на произвольном месте листа;
 - выполните действия *Математика/Текстовое поле*;
 - в текстовой области введите произвольный текст, содержащий несколько строк;
 - измените размер, цвет и жирность текста;

- 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 перекрещенных стрелок;
 - 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 при нажатой левой кнопке мыши;
 - удерживая курсор в выделенной области, нажмите клавиши [Ctrl+C], установите курсор в нужное место и нажмите клавиши [Ctrl+V];
- удалите текстовую область, для чего выделите область текста и нажмите клавишу [Delete].

Тема 2. ВЫЧИСЛЕНИЯ В ПАКЕТЕ MathCAD

В этой главе 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, –52, 0 и т.д.);
- *вещественные числовые константы*, которые могут записываться в одной из двух форм: с фиксированной точкой (например, 2.345) 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 [*]). Для ввода порядка нажать [^] – операция возведения в степень;
- *комплексные константы*, записываемые в виде $z = a + b \cdot i$, причем между величиной мнимой части b и мнимой единицей i не ставится знак умножения;
- *строковые константы* – любая последовательность символов (в том числе русские и греческие буквы), заключенные в кавычки (например, “Это строковая константа”);
- *встроенные константы* разделяются на *математические* и *физические* константы:

Математические константы

<i>Константа</i>	<i>Наименование</i>	<i>Значение по умолчанию</i>
∞	<i>Бесконечность</i>	$1 \cdot 10^{307}$
i или j	<i>Мнимые единицы</i>	<i>Квадратный корень из –1</i>
π	<i>PI</i>	3.142

Физические константы

<i>Кон- станта</i>	<i>Описание</i>	<i>Значение</i>
c	<i>Скорость света в вакууме</i>	$c = (2.998 \cdot 10^8) \frac{m}{s}$
e	<i>Элементарный за- ряд</i>	$e_e = (1.602 \cdot 10^{-19}) C$
h	<i>Постоянная План- ка</i>	$h = (6.626 \cdot 10^{-34}) \frac{kg \cdot m^2}{s}$
$\hbar$	<i>Приведенная по- стоянная Планка</i>	$\hbar = (1.055 \cdot 10^{-34}) \frac{kg \cdot m^2}{s}$
k	<i>Постоянная Больцмана</i>	$k = (1.381 \cdot 10^{-23}) \frac{kg \cdot m^2}{s^2 \cdot K}$
m_u	<i>Атомная единица массы</i>	$m_u = (1.661 \cdot 10^{-27}) kg$
N_A	<i>Число Авогадро</i>	$N_A = (6.022 \cdot 10^{23}) \frac{1}{mol}$
R	<i>Молярная газовая постоянная</i>	$R = 8.314 \frac{kg \cdot m^2}{s^2 \cdot K \cdot mol}$
R_∞	<i>Постоянная Ридберга</i>	$R_\infty = (1.097 \cdot 10^7) \frac{1}{m}$
α	<i>Постоянная тон- кой структуры</i>	$\alpha = 0.007$
γ	<i>Постоянная Эйлера – Машерони</i>	$\gamma = 0.577$

<i>Кон- станта</i>	<i>Описание</i>	<i>Значение</i>
ε_0	Магнитная проницаемость вакуума	$\varepsilon_0 = (8.854 \cdot 10^{-12}) \frac{s^4 \cdot A^2}{kg \cdot m^3}$
μ_0	Магнитная проницаемость свободного пространства	$\mu_0 = (1.257 \cdot 10^{-6}) \frac{kg \cdot m}{s^2 \cdot A^2}$
σ	Постоянная Стефана – Больцмана	$\sigma = (5.67 \cdot 10^{-8}) \frac{kg}{s^3 \cdot K^4}$
Φ_0	Квант магнитного потока	$\Phi_0 = (2.068 \cdot 10^{-15}) Wb$

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

Имя переменных в MathCAD – это набор латинских и греческих букв, цифр, символов и знаков подчеркивания. Прописные (большие) и строчные (малые) буквы определяют имена разных переменных. Начинаться имя должно только с буквы.

Использование в имени русских букв и пробелов запрещено.

Греческий символ вводится из раздела *Символы*, либо можно ввести сначала латинский символ, а затем нажать сочетание клавиш [Ctrl+G].

В MathCAD переменную не нужно предварительно описывать: ее тип определяется автоматически при задании переменной конкретного значения. Незаданные переменные выделяются на экране *красным цветом*.

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

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

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

Оператор вывода. Для вывода на экран значения переменной или выражения достаточно после выражения нажать клавишу [=]. Тогда на экране после знака [=] выведется численное значение.

Простая переменная – переменная, значение которой определяется одной величиной. Например, определение переменной x :

$$x := 2$$

Нижние индексы в именах простых переменных (например, x_C, x_A и т.д.) вводятся после нажатия кнопки *подстрочный индекс* во вкладке *Математика*.

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,

$$Name := N1 .. N2$$

где $Name$ – имя переменной, $N1$ – начальное значение, $N2$ – конечное значение. Символ многоточия «..» вводится двойным нажатием клавиши [.] или матричным оператором 1..n.

Если $N1 < N2$, шаг равен 1; если $N1 > N2$, шаг равен -1 .

Определение дискретной переменной с произвольным шагом имеет вид:

$$<Имя\ перемен> := <Нач_знач>, <След_знач> .. <Кон_знач>$$

Шаг h задается неявно величиной следующего значения и вычисляется как

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

Например,

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

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

Шаблон дискретной переменной с произвольным шагом



можно задать матричным оператором 1,3..n в категории *Векторы и матрицы* раздела *Операторы* вкладки *Математика*.

Системные переменные (встроенные переменные) – задают параметры работы MathCAD при вычислениях, при работе с файлами и массивами. К системным переменным относятся:

- TOL – определяет точность сходимости некоторых функций, таких как интеграл, производная, odesolve и root (по умолчанию $TOL=0.001$);

- $CTOL$ – определяет, насколько точно должно выполняться ограничение в блоке решения, чтобы решение считалось приемлемым при использовании функций Find, Minerr, Minimize и Maximize (по умолчанию $CTOL=0.001$);

- $ORIGIN$ – определяет индекс первого элемента массива (по умолчанию $ORIGIN=0$);

- $PRNPRECISION$ – задает число значащих цифр при записи в ASCII-файл данных с помощью функций WRITEPRN и APPENDPRN (по умолчанию $PRNPRECISION=4$);

- $PRNCOLWIDTH$ – задает ширину столбцов в ASCII-файлах, создаваемых при помощи функций WRITEPRN и APPENDPRN (по умолчанию $PRNCOLWIDTH=8$);

- CWD – текущая рабочая папка в виде строки (например: “D:\Work\Prime2\Data”).

Предустановленные значения системных переменных можно поменять в любом месте документа. Кроме того, переопределение значения переменной для всего документа производится в разделе *Параметры документа* вкладки *Расчет*.

Размерные переменные – числовые переменные и функции, обладающие размерностью и содержащие значения в системе измерения физических величин по международной системе СИ, американской системе единиц (USCS) и системе «сантиметр-грамм-секунда» (СГС). Сделано это для упрощения инженерных и физических расчетов, которые MathCAD выполняет с преобразованием единиц измерения физических величин.

Вставить единицу измерения можно с клавиатуры или из списка единиц измерений. Для ввода единицы измерения из списка, необходимо войти в соответствующую категорию единиц измерения, которая находится в разделе *ЕИ (Единицы измерения)* вкладки *Математика* и щелкнуть по требуемой единице измерения (рис. 2.1). Помеченная единица измерения появится после оператора масштаба.

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 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.

Индексированные переменные соответствуют массивам MathCAD. Они будут рассмотрены в теме 3.

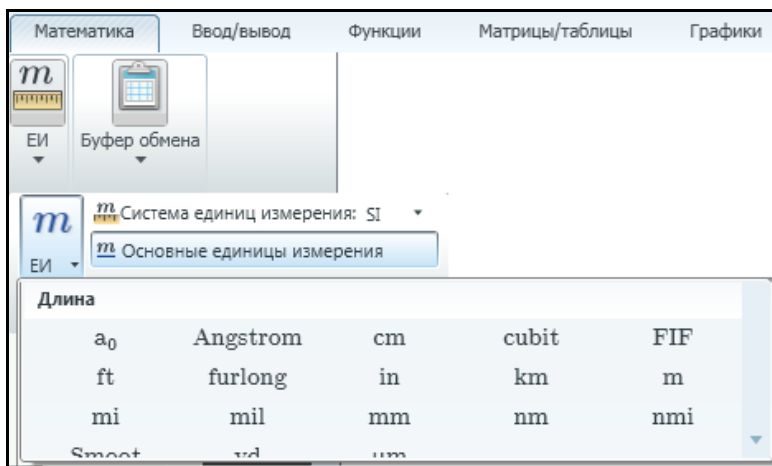


Рис. 2.1. Диалоговое окно единиц измерения

2.2. Арифметические операции MathCAD

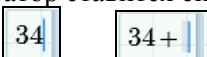
Арифметическими операциями в MathCAD являются: сложение, вычитание, умножение, деление и возведение в степень. Операции можно либо вводить с клавиатуры, либо выбрать из категории *Алгебра* раздела *Операторы*. Обозначение некоторых операций на экране отличается от обозначения клавиши, нажимаемой для ввода этой операции:

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. Начальная точка вставки помечается синим перекрестием . При вводе букв, чисел и операторов появляется область формул с курсором в виде синей вертикальной линии в точке вставки. Начальная точка вставки помечается синим перекрестием . При вводе букв, чисел и операторов появляется область формул с курсором в виде синей вертикальной линии в точке вставки. Часть математических выражений можно выделить в группу с помощью клавиши пробела, клавиш со стрелками или мышью. Математические элементы можно группировать по одному либо все одновременно. Расположение курсора с левой или правой стороны элемента может повлиять на группировку.

Если курсор справа, то следующий вводимый символ или оператор ставится справа от группы. Например:



Здесь курсор находится справа, и знак  ставится справа от выделенного элемента, а когда курсор слева, то знак  ставится слева:



Выделенная группа является операндом для вводимой операции или функции.

Например, чтобы в выражении $2 + 34 - 16$ ввести оператор деления к сумме $2 + 34$, необходимо выделить сумму элементов в группу. Если курсор будет стоять слева от группы, то при вводе оператора деления  выражение $2 + 34$ станет знаменателем дроби:



Если курсор будет стоять справа от группы, то выражение $2 + 34$ станет числителем дроби:

$$2+34-16$$

$$\frac{2+34}{-16}$$

Чтобы ввести дробную часть выражения под корень, выделим дробь так, чтобы курсор стоял слева от нее. Для этого щелкнем мышкой слева от дроби и нажмем клавишу [Пробел]:

$$\frac{2+34}{6}-16$$

$$\frac{2+34}{6}-16$$

Затем во вкладке *Математика*, разделе *Операторы* щелкнем по оператору квадратного корня:

$$\sqrt{\frac{2+34}{6}}-16$$

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.

$$a := 2.1 \quad b := 0.15 \cdot 10^{-3} \quad c := 3.5 + 4.78i$$

$$\frac{(|c|)^2 + a^2}{b} = 2.634 \cdot 10^5$$

$$y := \frac{e^{2 \cdot a} + \frac{1}{a+b}}{\sqrt[3]{a+b}}$$

$$y = 52.446 \quad z := \frac{a^{-y} + b^{-y}}{y} = 6.728 \cdot 10^{198}$$

Рис. 2.2. Программирование арифметических выражений

Пример 2.2.2. Используя оператор присваивания и размерные переменные, определить плотность стали (объем $V = 2 \text{ cm}^3$, масса $M = 15.7 \text{ gm}$). Определить удельный вес стали.

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

Дано	$V := 2 \text{ cm}^3$	объем
	$M := 15.7 \text{ gm}$	масса
Решение	$\rho := \frac{M}{V}$	плотность
	$\gamma := \rho \cdot g = (7.698 \cdot 10^4) \frac{\text{kg}}{\text{m}^2 \cdot \text{s}^2}$	удельный вес
	$\rho = (7.85 \cdot 10^3) \frac{\text{kg}}{\text{m}^3}$	

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. Весь набор встроенных функций разбит на группы (категории) в соответствии с функциональным значением.

Обозначение функции можно ввести с клавиатуры или найти нужную функцию во вкладке *Функции* и в соответствующей категории. Полный список функций и их синтаксис можно получить, щелкнув на кнопке $\boxed{f_x}$.

Чтобы найти требуемую функцию в списке, выполните следующие действия:

- щелкните значок , чтобы функции отображались в списке в алфавитном порядке;
- начните набирать слово для поиска в поле *Функции поиска* – в списке будут отображаться подходящие функции;
- чтобы увидеть группу, которой принадлежит функция, щелкните значок ;
- щелкните имя функции, чтобы вставить ее в документ.

Функция пользователя. Несмотря на широкий набор встроенных функций, очень часто возникает необходимость дополнить вычисления новыми функциями. 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. 2.4. Нижние индексы в именах простых переменных (например, x_C, x_A и т.д.) вводятся после нажатия кнопки *подстрочный индекс* во вкладке *Математика*.

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

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

Рис. 2.4. Описание функции пользователя и обращение к ней

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. 2.4).

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 из категории *Математический анализ* раздела *Операторы* (рис. 2.5).

При вводе оператора задается шаблон для последующего заполнения числовыми или символьными значениями. Приведем операторы, часто используемые при решении научно-инженерных задач.

Для ввода оператора **вычисление суммы** в документ необходимо щелкнуть на кнопке $\sum$ и заполнить поля в появившемся шаблоне (например: $\sum_{k=1}^{10} k$). Аналогичные действия необходимо

выполнить и при вводе операторов произведения Π , производной d/dx , интеграла $\int dx$ и предела $\lim$.

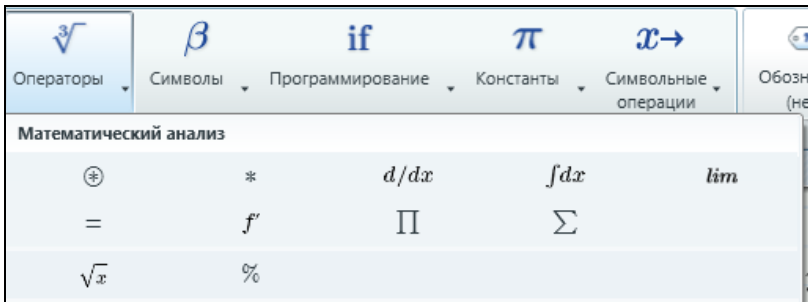


Рис. 2.5. Операторы математического анализа

Чтобы ввести двойной интеграл, надо щелкнуть по кнопке вызова оператора интеграла, установить курсор на поле шаблона, куда вводится интегрируемое выражение, и повторно вызвать оператор интеграла (рис. 2.6).



Рис. 2.6. Ввод двойного интеграла

Пример 2.4.1. Используя рассмотренные операторы математического анализа, вычислите суммы, произведения, производные и интегралы, приведенные на рис. 2.7.

Решение. Необходимые конструкции используются в документе MathCAD, который показан на рис. 2.7. Обратите внимание, что ряд вычислений оформлен в виде функций пользователя. Это дает определенные удобства при использовании операторов математического анализа.

Оператор вычисления сумм

$$\sum_{i=1}^{10} \sqrt{i} = 22.468 \quad S(n, m) := \sum_{i=1}^n \sum_{j=1}^m \frac{1}{i+j} \quad S(5, 7) = 5.899$$

Оператор вычисления произведений

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

Оператор вычисления производных

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

$$f(x, y) := \frac{x^2 + y^2}{\sqrt{|x+y|}} \quad df(x, y) := \frac{d^2}{dx^2} \frac{d^3}{dy^3} f(x, y) \quad df(3, -5) = 18.292$$

Оператор вычисления интеграла

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

Рис. 2.7. Операторы 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*.

Подробно символьные вычисления описаны в теме 5. Здесь рассматриваются символьные вычисления с помощью оператора символьного вывода $\rightarrow$, который вводится из категории *Математический анализ* раздела *Определения и вычисления*. Этот оператор ставится после записи соответствующего оператора математического анализа.

Пример 2.4.2. Используя рассмотренные операторы математического анализа, вычислите суммы, произведения, производные, приведенные на рис. 2.8, в символьном виде.

Решение. Необходимые конструкции показаны в документе MathCAD, фрагмент которого приведен на рис. 2.8.

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

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

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

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n \rightarrow e$$

Рис. 2.8. Символьные вычисления

На рис. 2.9 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. 2.9. Проверка правильности вычисления первообразной функции

Вопросы и задания для самоконтроля по теме 2

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 вкладки *Функции*, введите и вычислите приведенные ниже выражения (сравните полученные результаты с приведенными результатами):

$$\frac{23.1 \cdot \sqrt[3]{3 \cdot \pi - \frac{\ln(64)}{16}}}{4 - \pi^{3!}} = -0.05$$

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

$$e^{\sqrt{\frac{12 - 8.6}{48.4}}} - e^{\pi \cdot \left(\left(\frac{1}{3}\right)^6 - 2\right)} = 1.302$$

7. Используя оператор присваивания, задайте переменным x, y, z значения приведенных в задании 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 задания, вычислите расстояние между точками $C(2.3, 4.5, 6)$ и $D(-4.0, 3.1, -0.0)$.

11. Составьте функцию пользователя, вычисляющую факториал числа n

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

Протестируйте эту функцию для различных значений n .

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

$$\text{а) } \int_0^2 \frac{x}{x^2-1} dx \quad \text{б) } \int_1^4 \int_3^5 (2 \cdot x^4 + 13 \cdot y^5) dx dy \quad \text{в) } \sum_{i=3}^{20} e^{\frac{-i}{2}}$$

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

$$\begin{aligned} \text{а) } & \sum_{i=0}^{\infty} e^{\frac{-i}{2}} \quad \text{б) } \int_3^5 2 \cdot \pi (x^2 - 1) \cdot \sqrt{1 + \left(\frac{d}{dx} (x^2 - 1) \right)^2} dx \\ \text{в) } & \int_3^5 e^x \cdot x^2 dx \end{aligned}$$

14. Вычислите момент инерции стержня (масса $M = 5 \text{ kg}$, длина $L = 2 \text{ m}$), вращающегося относительно оси, проходящей через его конец и перпендикулярной ему:

$$I := \int_0^L x^2 \cdot \frac{M}{L} dx \quad I = 6.667 \text{ kg} \cdot \text{m}^2$$

15. Вычислите момент инерции прямоугольной пластины (масса $M = 5 \text{ kg}$, длина $L = 2 \text{ m}$, ширина $h = 10 \text{ cm}$) относительно оси, перпендикулярной ей и проходящей через ее центр:

$$I := \frac{M \cdot (h^2 + L^2)}{12} \quad I = 1.671 \text{ kg} \cdot \text{m}^2$$

16. Рассчитайте потенциальную энергию E тела массой $m = 5,67 \text{ кг}$, поднятого на высоту $h = 120 \text{ см}$ с использованием размерных переменных ($E = m \cdot g \cdot h$).

Тема 3. МАССИВЫ В ПАКЕТЕ MathCAD

В этой теме будут рассмотрены следующие вопросы: 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 задает номер строки, а второй – номер столбца, на пересечении которых находится данный элемент. Количество строк и столбцов матрицы определяет размер матрицы. Вектор можно рассматривать как матрицу, содержащую один столбец.

Для работы с векторами и матрицами система MathCAD использует вкладку *Матрицы/таблицы* строки меню, изображенную на рис. 3.1.

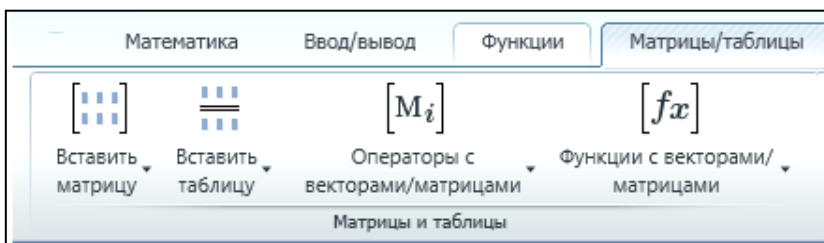


Рис. 3.1. Содержимое вкладки *Матрицы/таблицы*

Операторы работы с векторами и матрицами выбираются при нажатии на кнопку $[M_i]$ (см. рис. 3.1):

- $\vec{V}$ – оператор векторизации, поэлементно применяет операторы или функции к векторам или матрицам;
- M_i – задание индекса у элемента массива;
- $M^{(i)}$ – выделение столбца матрицы;
- $\widehat{M^i}$ – выделение строки матрицы;
- M^T – транспонирование вектора или матрицы;
- $\|x\|$ – вычисление нормы вектора или матрицы;
- $\times$ – оператор векторного произведения, вычисляет векторное произведение двух векторов или вектора и матрицы.

3.2. Создание массивов

Создание массива с использованием шаблона. Для создания массива необходимо ввести переменную – имя массива, ввести оператор присваивания и нажать кнопку *Вставить матрицу* $[::]$. В появившемся шаблоне выделить мышью требуемую конфигурацию массива (рис. 3.2).

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

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

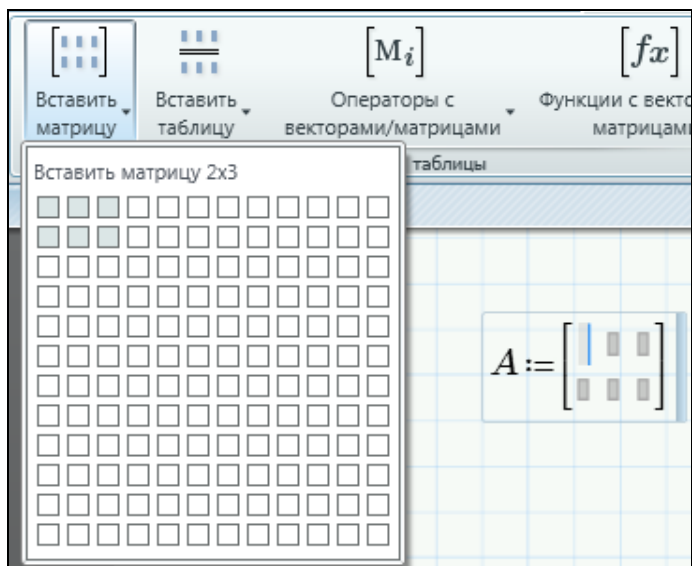


Рис. 3.2. Вставка матрицы 2×3

Создание массива вводом его элементов с клавиатуры.

Чтобы вставить и заполнить вектор, надо ввести переменную, затем оператор присваивания и «[» (левую квадратную скобку). Появится 1-элементный массив:

$$v := [\]$$

Ввод элементов *вектора* в него осуществляется последовательно, нажимая клавишу [Tab] после каждого введенного числа, чтобы вставить новую строку.

Чтобы вставить и заполнить *матрицу*, надо, как показано выше, создать 1-элементный массив. Элементы 1-й строки матрицы вводятся построчно, нажимая клавиши [Shift+Пробел] после каждого числа. Переход на 2-ю строку производится по нажатию клавиши [Shift+Enter] или установкой указателя на последний элемент строки и нажатием клавиши [Tab]. Все последующие элементы заносятся последовательным вводом элемента и нажатием клавиши [Tab].

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

Пример 3.2.1. Используя разные способы вставки векторов, создайте два вектора a и b . Вычислите их сумму, скалярное и векторное произведение.

Решение. Конструкции, осуществляющие требуемые вычисления, показаны на рис. 3.3. Для вычисления скалярного произведения двух векторов перемножим их (рис. 3.3 б).

Для получения векторного произведения двух векторов введем оператор умножения между ними, поместим курсор в левую часть области формул и дважды нажмем клавишу [Пробел], чтобы выделить область формул. На вкладке *Математика*, в разделе *Операторы*, выберем оператор *Векторизация* в списке *Векторы и матрицы*. Введем [=], чтобы вычислить выражение. Векторы будут поэлементно перемножены (рис. 3.3 в).

$$\begin{array}{lll}
 \text{a) } a := \begin{bmatrix} 2 \\ 5 \\ 8 \\ 4 \end{bmatrix} & b := \begin{bmatrix} 3 \\ 8 \\ 1 \\ 7 \end{bmatrix} & a + b = \begin{bmatrix} 5 \\ 13 \\ 9 \\ 11 \end{bmatrix} \\
 & & \text{б) } a \cdot b = 82 \\
 & & \text{в) } \overrightarrow{a \cdot b} = \begin{bmatrix} 6 \\ 40 \\ 8 \\ 28 \end{bmatrix}
 \end{array}$$

Рис. 3.3. Ввод элементов вектора

Пример 3.2.2. Создайте две матрицы размером 3×4 . Вычислите их сумму.

Решение. Создайте матрицу A с использованием шаблона, а матрицу B вводом его элементов с клавиатуры. Введите имя первой матрицы, затем оператор сложения, затем имя второй матрицы и оператор вывода [=]. Матрицы будут поэлементно просуммированы (рис. 3.4).

$$A := \begin{bmatrix} 1 & 5 & 9 & 12 \\ 5 & 7 & 2 & 9 \\ 7 & 2 & 3 & 8 \end{bmatrix} \quad B := \begin{bmatrix} 3 & 2 & 3 & 6 \\ 0 & 7 & 8 & 3 \\ 1 & 0 & 9 & 16 \end{bmatrix} \quad A + B = \begin{bmatrix} 4 & 7 & 12 & 18 \\ 5 & 14 & 10 & 12 \\ 8 & 2 & 12 & 24 \end{bmatrix}$$

Рис. 3.4. Ввод матриц и действия с ними

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 начинаются с 1.

Переопределить *ORIGIN* можно с помощью оператора присваивания ($:=$). В этом случае новое значение будет действовать правее и ниже этого оператора присваивания до следующего переопределения переменной *ORIGIN*. Этот способ иллюстрируется на рис. 3.5.

$x_1 := 10 \quad x = \begin{bmatrix} 0 \\ 10 \end{bmatrix} \quad A_{1,2} := 30 \quad A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 30 \end{bmatrix}$

Переопределим *ORIGIN*:

***ORIGIN* := 1**

$x_1 := 10 \quad x = \begin{bmatrix} 10 \\ 0 \end{bmatrix} \quad A_{1,2} := 30 \quad A = \begin{bmatrix} 0 & 30 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Рис. 3.5. Формирование массивов с разным значением *ORIGIN*

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 щелкнуть на кнопке $M^{(i)}$ в разделе *Операторы с векторами/матрицами* вкладки *Матрицы/таблицы*. Затем в появившихся угловых скобках ввести номер выделяемого столбца (с учетом значения системной переменной *ORIGIN*).

Пример 3.3.3. Из второго и четвертого столбцов матрицы B примера 3.2.2 сформировать векторы c, d .

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

$$c := B^{(2)} \quad d := B^{(4)} \quad c = \begin{bmatrix} 2 \\ 7 \\ 0 \end{bmatrix} \quad d = \begin{bmatrix} 6 \\ 3 \\ 16 \end{bmatrix}$$

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;
- облегчающих решение широкого круга задач линейной алгебры.

Вызов функций осуществляется из раздела *Функции с векторами/матрицами* вкладки *Матрицы/таблицы*.

При установке курсора на выбранную функцию выводится краткая информация о параметрах выбранной функции и ее назначении. Заметим, что обращение к функции можно ввести с клавиатуры.

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. 3.7.

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

Рис. 3.7. Вычисление характеристик массивов

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

Таблица 3.2

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

Пример 3.4.2. Формирование матриц специального типа с использованием функций из табл. 3.2 показано на рис. 3.8.

$\text{identity}(3) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	<- Единичная матрица	$M := \begin{bmatrix} 2 & 4 & 6 \\ 4 & 5 & 6 \\ 2 & 7 & 12 \end{bmatrix}$
Матрица B:	Вещественная часть:	Мнимая часть:
$B := \begin{bmatrix} 5+2i & 4 \\ 2.54 & 1+6i \\ 3+(4+8i) & 3i \end{bmatrix}$	$\text{Re}(B) = \begin{bmatrix} 5 & 4 \\ 2.54 & 1 \\ 7 & 0 \end{bmatrix}$	$\text{Im}(B) = \begin{bmatrix} 2 & 0 \\ 0 & 6 \\ 8 & 3 \end{bmatrix}$
Вектор b:		
$b := M^{(1)} \quad b = \begin{bmatrix} 4 \\ 5 \\ 7 \end{bmatrix}$	$\text{diag}(b) = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 7 \end{bmatrix}$	<- Диагональная матрица, полученная из вектора b
$f(x,y) := x^2 + (2 \cdot x \cdot y) + y^2$		
$C := \text{matrix}(3,4,f)$	$C = \begin{bmatrix} 0 & 1 & 4 & 9 \\ 1 & 4 & 9 & 16 \\ 4 & 9 & 16 & 25 \end{bmatrix}$	<- Матрица, где $C_{i,j}=f(i,j)$

Рис. 3.8. Формирование матриц специального типа

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

Формирование новых массивов из существующих. Функции MathCAD для формирования новых массивов из существующих массивов приведены в табл. 3.3.

Таблица 3.3

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

Пример 3.4.3. Используя функции табл. 3.3, сформировать новые матрицы из матриц, приведенных на рис. 3.9.

Решение. Первоначально сформируем исходные матрицы, показанные на рис. 3.9. Затем сформируем новые матрицы, приведенные на рис. 3.10.

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

Рис. 3.9. 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. 3.10. Формирование новых матриц

Таблица 3.4

$rank(M)$	Вычисляет ранг матрицы M , т.е. число линейно независимых строк или столбцов матрицы.
$tr(M)$	Вычисляет след (сумму диагональных элементов) квадратной матрицы M .
$mean(M)$	Вычисляет среднее значение элементов массива M .
$norme(M)$	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 и M , элементы которых приведены на рис. 3.11.

Решение. Вычисление характеристик показано на рис. 3.11.

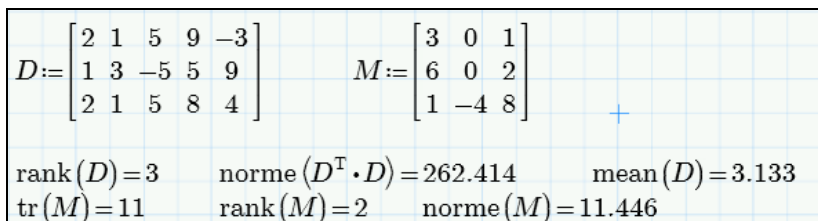


Рис. 3.11. Вычисление специальных характеристик матриц

3.5. Вычисления с массивами

Над векторами и матрицами можно производить вычисления точно 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*.

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 , для скалярных величин – z .

Таблица 3.5

Операция	Обозначение	Клавиши	Описание
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 вектора v , число столбцов в A должно быть равно числу строк в v .
Деление	$\frac{A}{z}$	/	Делит каждый элемент матрицы A на скаляр z .
Сложение векторов и матриц	$A + M$	+	Матрицы A и M должны иметь одинаковое число строк и столбцов.
Скалярная сумма	$A + z$	+	Добавляет z к каждому элементу матрицы A .

Окончание табл. 3.5

Операция	Обо- значе- ние	Клави- ши	Описание
Векторное и матричное вычитание	$A - M$	—	Вычитает соответствующие элементы матрицы M из элементов матрицы A , мат- рицы A и M должны иметь одинаковые размеры.
Скалярное вычитание	$A - z$	—	Вычитает z из каждого элемента матрицы A .
Изменение знака	$-A$	—	Умножает все элементы матрицы A на -1 .
Степень матрицы	M^n	$\wedge n$	Вычисляет n -ю степень квадратной матри- цы M , где n – целое число.
Обращение матрицы	M^{-1}	$\wedge -1$	Вычисляет матрицу, об- ратную к M .
Длина вектора	$ v $		Вычисляет величину $\sqrt{\sum_{i=1}^n v_i^2}$.
Детерми- нант	$ M $		Возвращает детерминант квадратной матрицы M .
Векторное произведе- ние	$u \times v$	[Ctrl+8]	Вычисляет векторное про- изведение, где u и v – век- торы с тремя проекциями.

Пример 3.5.1. Даны матрица M и вектор u (рис. 3.12). Вы-
полнить векторные и матричные операции над этими массивами.

Решение. На рис. 3.12 показан фрагмент документа Math-
CAD, в котором выполнены операции. Объясните:

- почему векторное произведение векторов z и u дало ну-
левой вектор;

- почему произведение матрицы M на обратную дало единичную матрицу, в которой на главной диагонали стоят 1, а все остальные элементы равны 0.

$M := \begin{bmatrix} 2 & 4 & 2 \\ 1 & 5 & -3 \\ 6 & 3 & 1 \end{bmatrix}$	$y := \begin{bmatrix} 2 \\ 0 \\ 3 \end{bmatrix}$	Исходные массивы	
$2 \cdot M = \begin{bmatrix} 4 & 8 & 4 \\ 2 & 10 & -6 \\ 12 & 6 & 2 \end{bmatrix}$	$M - 2 = \begin{bmatrix} 0 & 2 & 0 \\ -1 & 3 & -5 \\ 4 & 1 & -1 \end{bmatrix}$	$z := 3 \cdot y = \begin{bmatrix} 6 \\ 0 \\ 9 \end{bmatrix}$	
$M \cdot y = \begin{bmatrix} 10 \\ -7 \\ 15 \end{bmatrix}$	$M^2 = \begin{bmatrix} 20 & 34 & -6 \\ -11 & 20 & -16 \\ 21 & 42 & 4 \end{bmatrix}$	$M \cdot M^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	
$z - 3 = \begin{bmatrix} 3 \\ -3 \\ 6 \end{bmatrix}$	$ M = -102$	$z \cdot y = 39$	$z \times y = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

Рис. 3.12. Операции над массивами

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 K из коэффициентов системы и вектор q из значений правой части системы (рис. 3.13). Получаем систему уравнений $K \cdot d = q$, где вектор d имеет проекции x, y, z , т.е. является искомым решением системы. Используя обратную матрицу, вычисляем решение системы и делаем проверку, а именно: произведение $K \cdot d$ равно вектору q (получено тождество левой и правой части исходной системы уравнений), а разность $K \cdot d - q$ есть нулевой вектор.

$$\begin{array}{lcl}
 K := \begin{bmatrix} 0 & 1 & 2 \\ 3 & 0 & 2 \\ 5 & 1 & 3 \end{bmatrix} & q := \begin{bmatrix} 13 \\ -3 \\ 50 \end{bmatrix} & d := K^{-1} \cdot q = \begin{bmatrix} 11 \\ 49 \\ -18 \end{bmatrix} \\
 K \cdot d = \begin{bmatrix} 13 \\ -3 \\ 50 \end{bmatrix} & K \cdot d - q = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \text{проверка} \\
 & & \text{решения}
 \end{array}$$

Рис. 3.13. 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(v)</i>	Сортировка элементов векторов в порядке возрастания их значений.
<i>reverse(v)</i>	Перестановка элементов в обратном порядке.
<i>csort(M,n)</i>	Перестановка строк матрицы <i>M</i> таким образом, чтобы отсортированным оказался <i>n</i> -й столбец.
<i>rsort(M,n)</i>	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*** (рис. 3.14).

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

Рис. 3.14. Сортировка элементов массивов

Решение. На рис. 3.14 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 самоконтроля по теме 3

1. Как выглядит обращение к элементу массива MathCAD?
2. Что обозначают нижние индексы?
3. К какому типу переменных относятся нижние индексы?
4. Какую размерность массива имеет 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:

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

7. Используя операции со столбцами, получите матрицу C перестановкой 2-го и 3-го столбцов матрицы A из задания 6.

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 $2 \cdot A \cdot B - 3 \cdot C \cdot D$, 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) сформируйте матрицу R_1 , а из двух последних строк – матрицу R_2 . Выясните, 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 вычислите 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, 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 в разделе *Вставить график* вкладки *График* (рис. 4.1).

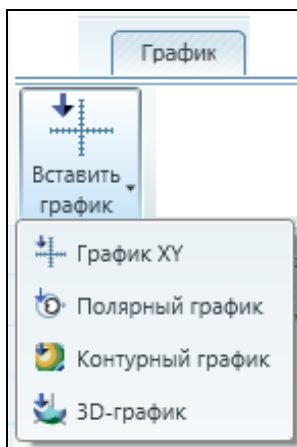


Рис. 4.1. Типы графиков

В MathCAD Prime 2.0 применяются следующие виды графиков:

График XY. Служит для построения графика функции $y = f(x)$ или для отображения точек на плоскости (x_i, y_i) .

Полярный график. Служит для построения графика функции $\rho(\varphi)$, заданной в полярных координатах, где полярный радиус ρ зависит от полярного угла φ .

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-график. Служит для точечного представления элементов матрицы или отображения значений функции $z = f(x,y)$ в заданных точках (x_i, y_j) . Эта кнопка может также использоваться для построения пространственных кривых.

4.2. Построение графиков функции одной переменной в декартовой системе координат

Построение графика одной переменной включает следующие этапы:

1. Задайте функцию или выражение, значения 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, щелкнув по кнопке  в разделе *Вставить график* вкладки *График*. На экране появится шаблон (рис. 4.2).

3. В местозаполнителе оси Y (позиция 4) введите функцию $f(x)$ или ее формулу $\sin(x)^3$. Если функция имеет размерность (метры, килограммы и т.д.), то в поле ввода единицы измерения оси Y (позиция 3) вставьте из раздела *Единицы измерения* вкладки *Математика* соответствующую единицу измерения.

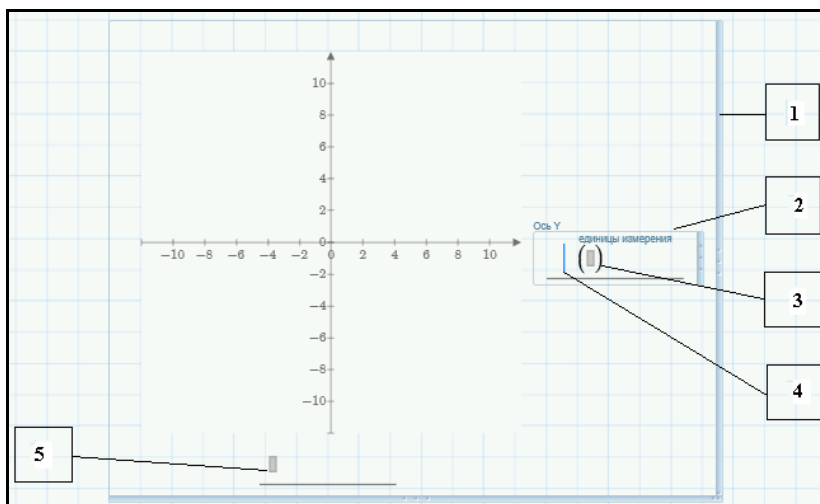
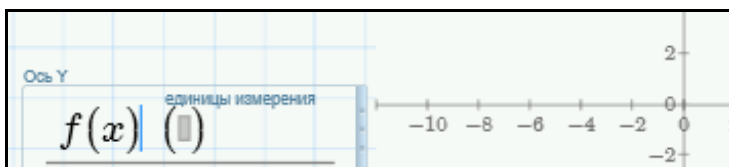


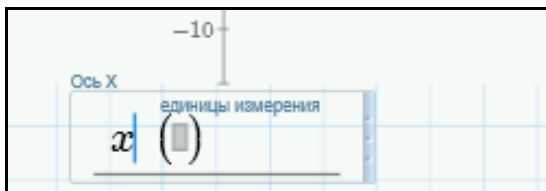
Рис. 4.2. Поля шаблона графика функции одной переменной:

- 1 – область построения графика;
- 2 – легенда;
- 3 – поле ввода единицы измерения оси Y;
- 4 – местозаполнитель оси Y;
- 5 – местозаполнитель оси X

4. Перетащите легенду оси Y в левую сторону:



5. В местозаполнителе оси X (позиция 5) введите переменную функции и, если необходимо, вставьте соответствующую единицу измерения:



6. Нажмите клавишу [Enter] или щелкните мышкой в стороне от шаблона. Появится график с линейной кривой, который будет иметь вид, показанный на рис. 4.3.

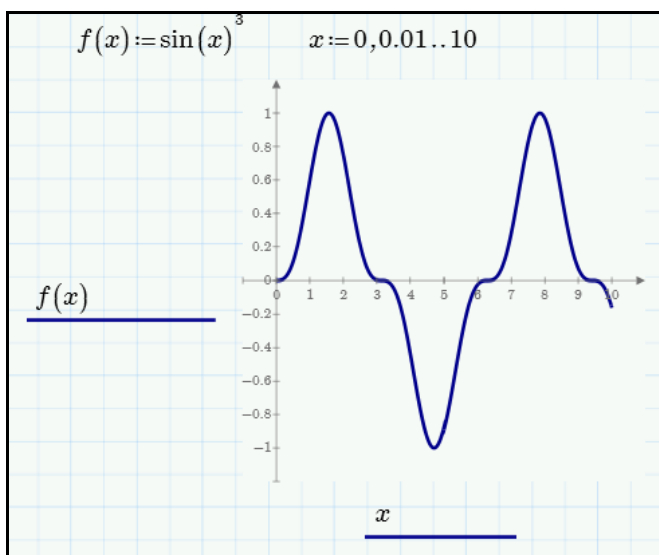


Рис. 4.3. Построение графика функции одной переменной

Изменение диапазонов по оси X и Y позволяет изменять масштаб, чтобы представить график в удобном для пользователя виде.

На оси графика, где требуется внести изменения, выделите значение редактируемого деления (первого, второго или последнего). Рядом со ставшим черным значением деления по

умолчанию появится курсор. Удалите значение деления оси и введите новое значение.

При изменении первого или последнего деления изменится видимый диапазон графика. При изменении первого или второго деления изменится интервал между делениями.

Нажмите клавишу [Enter] или щелкните мышкой в стороне от графика. Все деления оси соответственно изменятся.

Пример 4.2.1. Изменить диапазоны осей так, чтобы увеличить часть графика, выделенного окружностью (рис. 4.4 а).

Решение. Установите диапазон оси X с 2.5 до 4.0. Для этого замените на оси X крайние значения 0 и 10 на 2.5 и 4.0 соответственно. Установите диапазон оси Y с -0.25 до 0.25 . Для этого замените на оси Y крайние значения -1 и 1 на -0.25 и 0.25 соответственно. Нажмите клавишу [Enter] или щелкните мышкой в стороне от графика. Появится график с линейной кривой, который будет иметь вид, показанный на рис. 4.4 б.

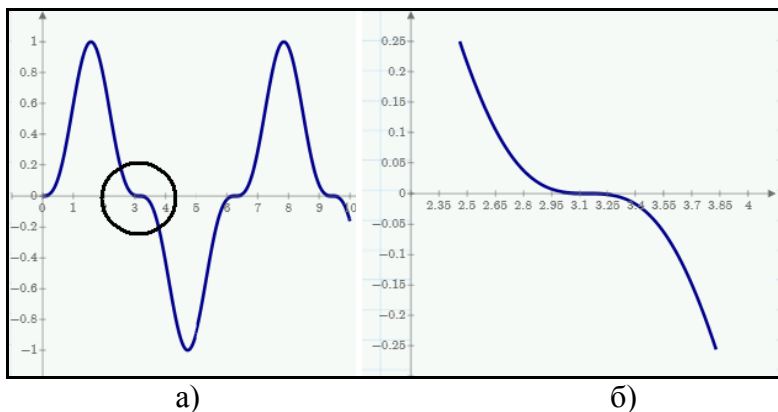


Рис. 4.4. Масштабирование графика

Форматирование графиков. Свойства графика могут быть установлены в разделе *Стили* вкладки *График*. Окно *Стили* показано на рис. 4.5.

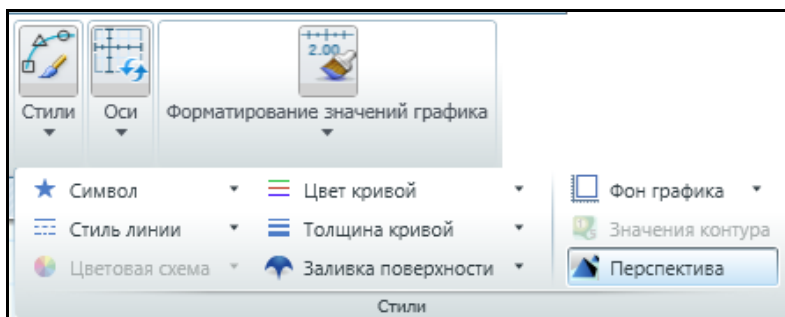


Рис. 4.5. Окно *Стили*

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

Список *Стиль линий* позволяет выбрать типы линий: непрерывная, пунктирная, штрихпунктирная и др.

Список *Цвет кривой* позволяет выбрать цвет линии или символа на графике.

Список *Толщина кривой* позволяет выбрать толщину линии или символа графика.

Список *Цветовая схема* позволяет установить изображение контурного графика в виде цветных контуров.

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

- поместить курсор рядом с именем функции оси Y; щелкнуть *Добавить кривую* на вкладке *График*; появится новый местозаполнитель оси Y под текущим местозаполнителем;

- в новый местозаполнитель оси Y ввести имя или тело функций;
- поместить курсор справа от имени переменной оси X; щелкнуть *Добавить кривую* на вкладке *График*; появится новый местозаполнитель оси X под текущим местозаполнителем; в поле местозаполнителя оси X ввести имя переменной добавляемой функции.

Примечание. Если у всех функций одинаковая переменная, то достаточно ввести ее один раз.

Когда курсор будет помещен в **крайнюю левую** точку ввода выражения оси и будет добавлено выражение оси, появится местозаполнитель оси **над** текущим местозаполнителем.

Пример 4.2.2. 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. 4.6).
2. Вызов шаблона графика, ввод в местозаполнитель оси Y тела первой функции $\sin(x)$. Установка курсора справа от имени введенной функции и выполнение действия *Добавить кривую*. Ввод в появившийся местозаполнитель имени второй функции.
3. Перенос легенды оси Y в левую сторону от шаблона.
4. Ввод в местозаполнитель оси X переменной x .
5. Щелчок указателем мыши за пределами графика.
6. С использованием раздела *Стили* вкладки *График* изменение типа, цвета и толщины кривых так, как это показано на рис. 4.6.

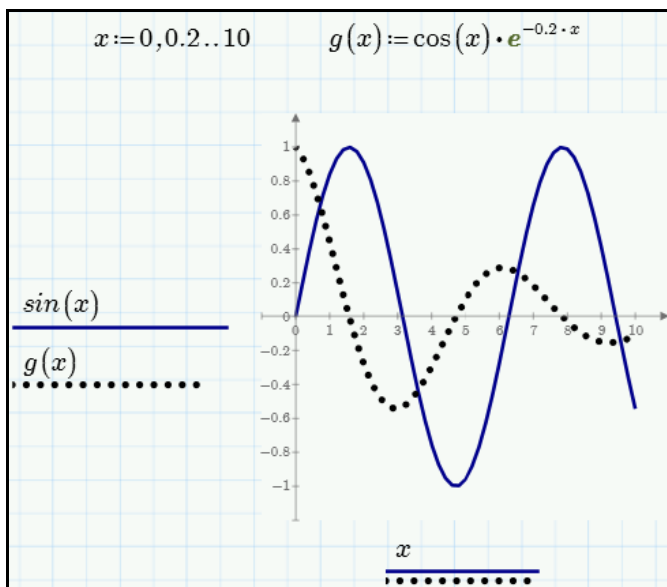


Рис. 4.6. Построение на одном рисунке двух кривых

Графическое представление элементов одномерных массивов (проекций вектора). Для этого в местозаполнитель оси X вводится переменная, являющаяся индексом y элемента одномерного массива, а в местозаполнителе оси Y стоит обращение к элементу соответствующего массива.

Пример 4.2.3. 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 4.2.2, получаем графики, приведенные на рис. 4.7. Здесь в местозаполнитель оси X вводится переменная i – индексная переменная, а в местозаполнитель оси Y – обращение к элементам массивов x, y .

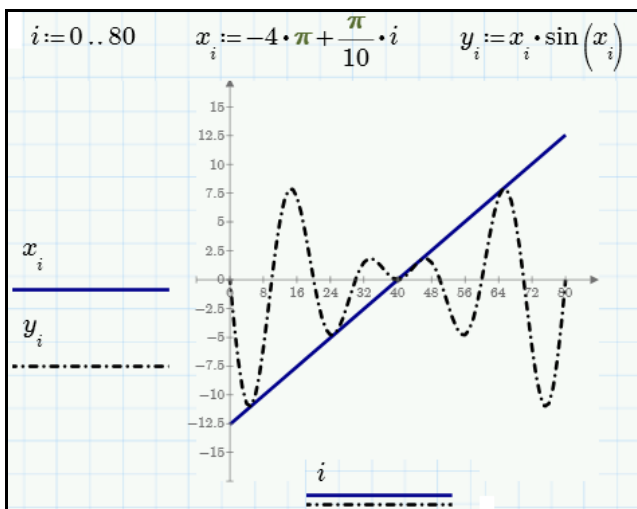


Рис. 4.7. Отображение элементов одномерных массивов

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, по оси X вводится зависимость $x(t)$, по оси Y – зависимость $y(t)$.

Пример 4.2.4. 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. 4.8. Здесь шаг изменения дискретной переменной равен 0.02, что обеспечивает «гладкость» кривой графика.

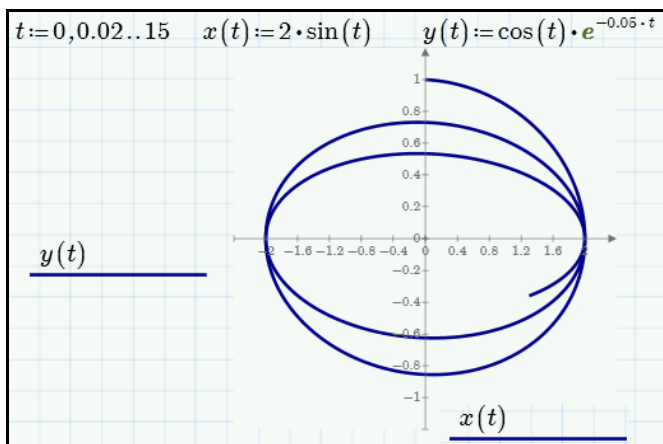


Рис. 4.8. Построение графика параметрической функции

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 (показан на рис. 4.9) вызывается кнопкой  *Полярный график* раздела *Вставить график* вкладки *График* или клавишами [Ctrl+7].

В местозаполнителе радиальной оси (справа) вводится функция или выражение, зависящее от угла.

В местозаполнителе угловой оси (внизу) вводится полярный угол.

Заметим, что можно ввести несколько зависимостей радиуса от угла, которые будут отображены соответствующими кривыми. Это иллюстрирует следующий пример.

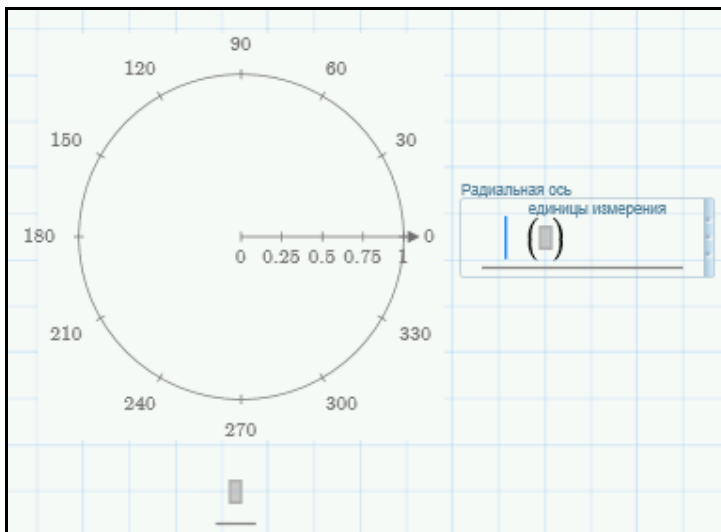


Рис. 4.9. Шаблон графика в полярной системе координат

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

$$\rho(\varphi) = 1 + \sin\left(2\varphi + 3\frac{\pi}{2}\right), \quad r(\varphi) = 1 + \frac{\sin(3\varphi + \pi)}{2} \quad \text{для } \varphi \in [0, 2\pi].$$

Решение. На рис. 4.10 приведен фрагмент документа, в котором находятся необходимые конструкции MathCAD. Заметим, что в местозаполнителе радиальной оси кроме двух функций $\rho(\varphi)$, $r(\varphi)$ введена константа 2, которая отображена на рисунке окружностью радиусом 2.

Примечание. Символ φ вводится последовательным нажатием клавиш [j] и [Ctrl+G], а символ ρ – последовательным нажатием клавиш [r] и [Ctrl+G], их можно также ввести из раздела *Операторы и символы* вкладки *Математика*.

Форматирование полярных графиков проводится так же, как и форматирование графиков в декартовой системе координат.

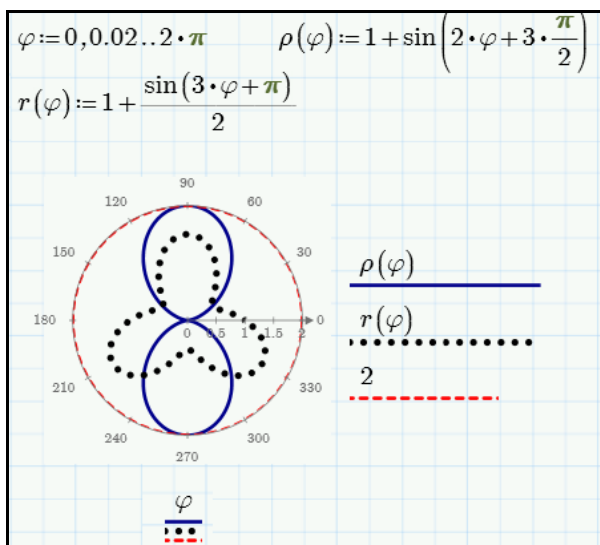


Рис. 4.10. Графики функций в полярных координатах

4.4. Построение графиков функций двух переменных

Трёхмерные графики используются для визуализации функции двух переменных – функции, принимающей значения в виде векторов, или набора 3D-данных. Входные данные для 3D-графиков могут иметь следующие типы данных:

- функция, принимающая значения в виде векторов для одного или двух параметров с тремя элементами, определяющими координаты x , y и z ;
- матрица с 3 столбцами, содержащими координаты x , y и z ;

- вектор из трех векторов, представляющих координаты x , y и z и являющихся выходными данными функции *CreateSpace*;
- вектор из трех вложенных матриц, представляющих координаты x , y и z и являющихся выходными данными функции *CreateMesh*.

Для создания графика функции двух переменных (3D-гра-

фика) необходимо щелкнуть по кнопке  раздела *Вставить график* вкладки *График* или клавишами [Ctrl+3]. 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 необходимо предварительно сформировать матрицу из значений функции в узлах прямоугольной сетки. 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. 4.11.

Заметим, что в одномерных массивах x, y находятся узлы по осям x, y , в которых вычисляются значения элементов матрицы M . На рис. 4.11 а показан 3D-график. На рис. 4.11 б показан контурный график.

Форматирование графиков производится с помощью вызова операций из раздела *Стили* (см. рис. 4.5).

Функция CreateMesh (Создать сетку). Эту функцию можно использовать для построения графика поверхности на контурном или 3D-графике. 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.

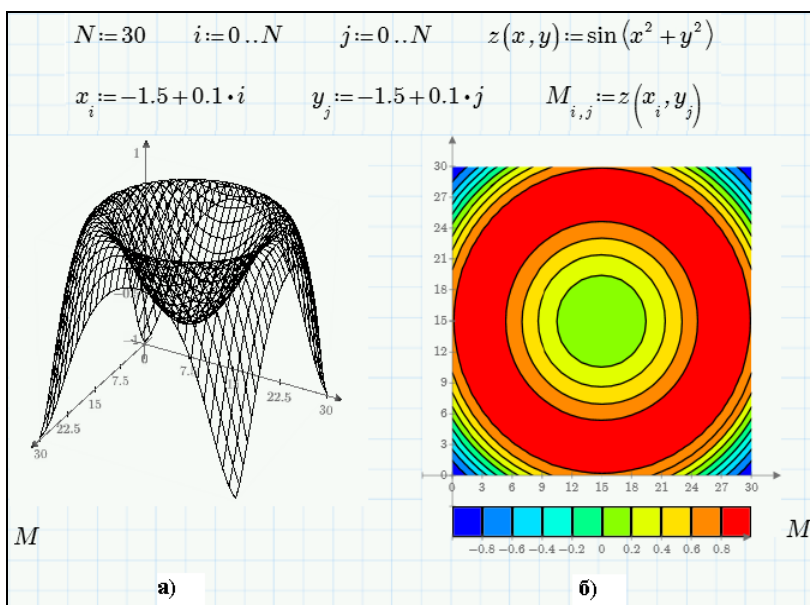


Рис. 4.11. Графики функции двух переменных

Пример 4.4.2. 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. 4.12.

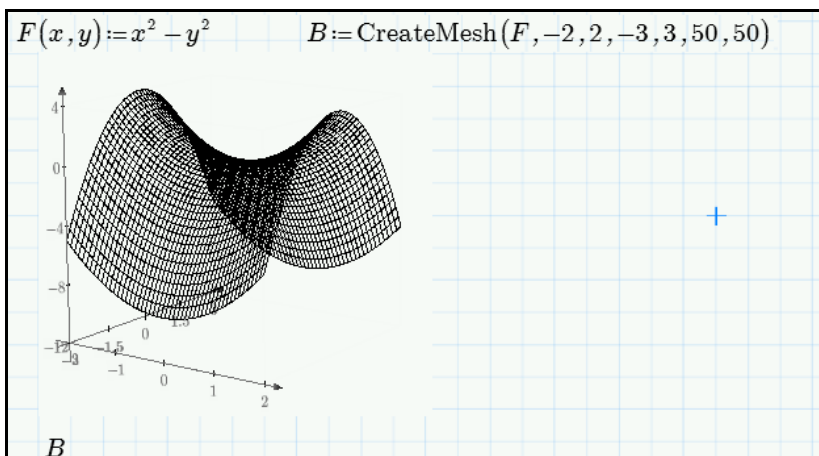


Рис. 4.12. Построение графика поверхности с помощью функции *CreateMesh*

4.5. Построение диаграмм

В MathCAD можно строить следующие диаграммы:

- диаграммы погрешностей;
- диаграммы «водопад»;
- диаграммы «ящик с усами»;
- каскадные диаграммы;
- диаграммы распределения.

Для построения диаграмм необходимо вызвать список типов из раздела *Изменить тип* вкладки *График* (рис. 4.13).

Создание диаграммы погрешностей. Диаграмма погрешностей представляет на XY-графике векторы, размер которых равен разности значений в строках матрицы. Шаги создания диаграммы:

1. Вставьте график XY в документ.
2. Введите в местозаполнитель оси Y матрицу, содержащую по два элемента в каждой строке: данные по двум «усикам» планок погрешностей.

3. В местозаполнитель оси X введите вектор-столбец, содержащий значения x .

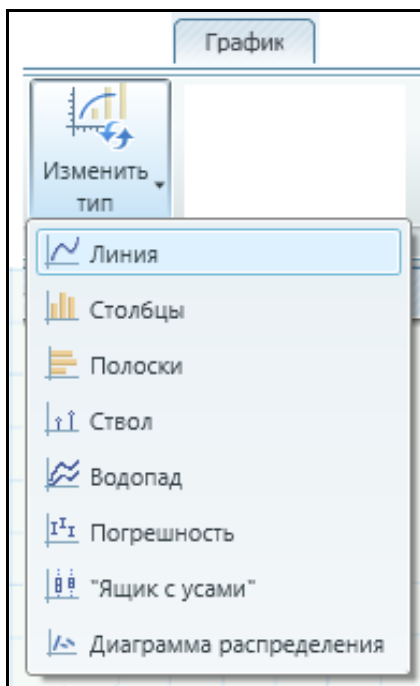


Рис. 4.13. Список типов диаграмм

4. В разделе *Изменить тип* выбрать тип *Погрешность*. Появится диаграмма погрешностей (рис. 4.14), в которой для каждой строки матриц x и y будет построена планка погрешности.

Создание диаграммы «водопад» включает следующие шаги:

1. Вставьте график XY в документ.
2. Введите в местозаполнитель оси X матрицу из значений x , а в местозаполнитель оси Y матрицу из значений y . На примере рис. 4.15 матрицами со значениями y будут векторы матрицы A .

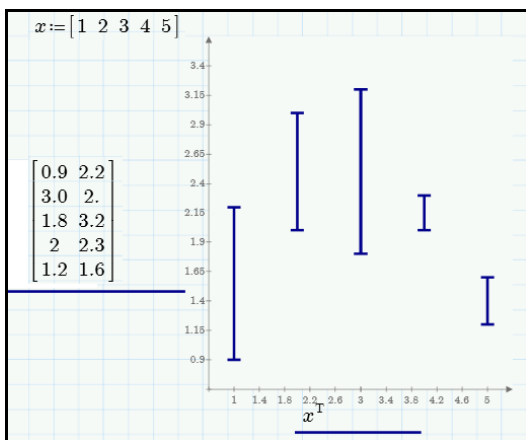


Рис. 4.14. Диаграмма погрешностей

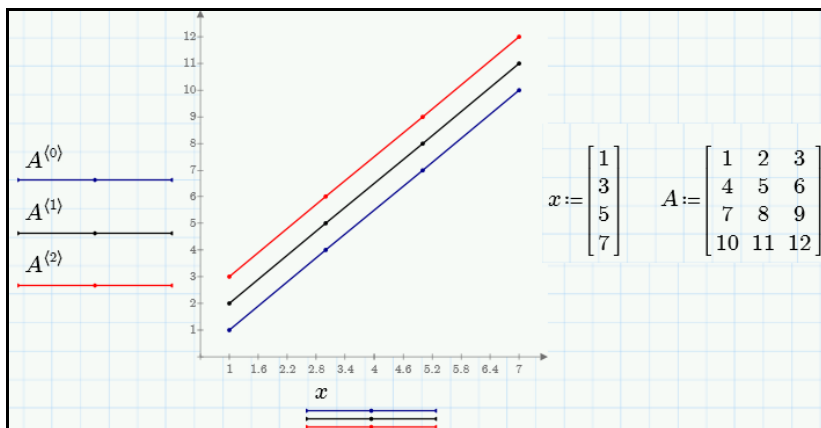


Рис. 4.15. Диаграмма «водопад»

Вопросы и задания для самоконтроля по теме 4

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 форматирование, 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 $p(x) = 5x^6 - 3$. Задайте свой цвет и стиль кривой.

7. Для индекса i , меняющегося от 0 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):

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

Выполните необходимое форматирование графиков.

9. Постройте график лемнискаты Бернулли

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

Тема 5. СИМВОЛЬНЫЕ И ЧИСЛЕННЫЕ 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 более удобны для отображения аналитического результата выражения в целом или его части, не сохраняя сам ход вычислений.

Выражения в символьных расчетах вычисляются с использованием оператора символьного вывода ($\rightarrow$) вместо оператора численного вычисления ($=$). Все стандартные операторы MathCAD и многие встроенные функции можно вычислять аналитически.

5.1. Команды символьных вычислений

Вызов символьных операций осуществляется из раздела *Операторы и символы* вкладки *Математика* (рис. 5.1).

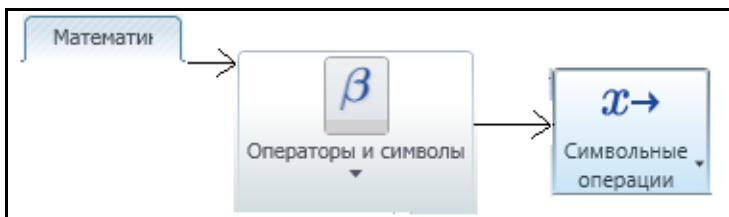


Рис. 5.1. Вызов символьных операций

В категориях символьных операций содержатся (рис. 5.2):

- оператор символьного вывода;
- ключевые слова;
- модификаторы;
- explicit.

Ключевые слова				
assume	cauchy	coeffs	collect	combine
confrac	expand	factor	float	fourier
fully	invfourier	invlaplace	invztrans	laplace
parfrac	rectangular	rewrite	series	simplify
solve	substitute	using	ztrans	
Модификаторы				
ALL	acos	acot	asin	atan
cauchy	complex	cos	cosh	cot
coth	degree	domain	even	exp
fraction	fully	gamma	integer	ln
log	matrix	max	odd	raw
real	RealRange	signum	sin	sincos
sinh	sinhcosh	tan	tanh	using
Explicit				
ALL	explicit			

Рис. 5.2. Список ключевых слов и модификаторов

Оператор символьного вывода MathCAD упрощает результаты вычислительных выражений, выполняя арифметические действия и комбинируя одинаковые переменные.

$$2 \cdot x + 5 + 4 \cdot x - 3 \rightarrow 6 \cdot x + 2.$$

Ключевые слова – это команды символьных операций. Ключевые слова используются при выполнении более сложных символьных операций, чем при использовании оператора сим-

вольного вывода. Ключевые слова отображаются над оператором символьного вывода.

$$x^2 - 2 \cdot x - 15 \xrightarrow{\text{factor}} (x+3) \cdot (x-5).$$

Описание ключевых слов приведено в таблице.

Модификаторами называются специальные элементы, которые используются для изменения поведения ключевых слов. Например, можно использовать модификатор `integer` после ключевого слова `assume`, чтобы указать, что переменная должна быть целым числом.

Explicit – вид ключевого слова, используемого для получения результата выражения с подставленными значениями переменных, но без сокращения числовых выражений.

Ключевое слово	Описание
<u>assume</u>	Делает предположение относительно области определения переменных.
<u>cauchy</u>	Возвращает главное значение Коши для интеграла.
<u>coeffs</u>	Возвращает коэффициенты полинома.
<u>collect</u>	Собирает элементы, содержащие переменную с одинаковым показателем степени.
<u>combine</u>	Комбинирует элементы выражения с использованием свойств элементарных функций.
<u>confrac</u>	Вычисляет разложение числа или функции в непрерывную дробь.
<u>expand</u>	Перемножает степени и произведения в выражении.
<u>explicit</u>	Возвращает выражения с подставленными значениями переменных, но без сокращения числовых выражений.
<u>factor</u>	Факторизует выражение.

Окончание таблицы

Ключевое слово	Описание
<u>float</u>	Возвращает результаты с доступными числовыми значениями, рассчитанными посредством вычислений с плавающей запятой с указанной точностью.
<u>fully</u>	Возвращает подробное решение уравнения.
<u>parfrac</u>	Выполняет разложение рационального выражения в сумму дробей с линейными или квадратичными знаменателями.
<u>rectangular</u>	Возвращает результаты с комплексными числовыми значениями, разделенными на действительную и мнимую части.
<u>rewrite</u>	Переписывает выражения через элементарные функции.
<u>series</u>	Выполняет разложение функции или выражения в ряд Тейлора или ряд Лорана в окрестности 0.
<u>simplify</u>	Упрощает алгебраически или вычисляет выражение.
<u>solve</u>	Решает уравнение аналитически.
<u>substitute</u>	Заменяет все вхождения одной переменной другой переменной, выражением или числом.
<u>using</u>	Заменяет созданную переменную в решении уравнением.
<u>fourier</u> , <u>laplace</u> , <u>ztrans</u> , <u>invfourier</u> , <u>invlaplace</u> , <u>invztrans</u>	Вычисляет преобразование или обратное преобразование функции.

5.2. Выполнение символьных вычислений

Вставка ключевых слов и модификаторов содержит следующие этапы:

1. Выберите выражение, которое требуется вычислить аналитически.

2. На вкладке *Математика* в разделе *Операторы и символы* щелкните *Символьные операции*, а затем щелкните стрелку оператора символьного вывода ($\rightarrow$).

Примечание. Оператор символьного вывода ($\rightarrow$) можно вставить нажатием комбинации клавиш [Ctrl+ $\rightarrow$].

3. Щелкните местозаполнитель над оператором символьного вывода и введите ключевое слово или модификатор.

Можно также на вкладке *Математика* в разделе *Операторы и символы* щелкнуть *Символьные операции* и выбрать ключевое слово или модификатор. Если курсор находится вне области формул, можно вставить ключевое слово или модификатор, а затем ввести выражение, которое требуется вычислить, в местозаполнитель слева от оператора символьного вывода.

4. Чтобы вставить несколько ключевых слов над одним оператором символьного вывода, выполните одно из следующих действий:

- введите запятую сразу после первого ключевого слова. Появится местозаполнитель. Введите или вставьте другое ключевое слово в местозаполнитель;

- введите или вставьте первое ключевое слово, нажмите клавиши [Shift+Enter], чтобы вставить новую строку, и добавьте другое ключевое слово.

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

Решение. Выполним следующую последовательность действий (рис. 5.3):

- введем выражение $\cos(2 \cdot y)$;
- щелкнем по ключевому слову *expand*;

- введем через запятую переменную, относительно которой производится вычисление;
- щелкнем мышью в стороне от выражения.

$$\cos(2 \cdot y)$$

$$\cos(2 \cdot y) \xrightarrow{\text{expand}} ?$$

$$\cos(2 \cdot y) \xrightarrow{\text{expand}, y} \cos(y)^2 - \sin(y)^2$$

Рис. 5.3. 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. Оператор *lim* вычисляет предел символьного выражения. Примеры вычисления пределов приведены на рис. 5.4.

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

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

Рис. 5.4. Символьное вычисление пределов

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

Пример 5.2.2. Выполнить символьные вычисления выражений, приведенных на рис. 5.5.

На рис. 5.5 показаны необходимые символьные вычисления с использованием операторов суммы, дифференцирования и интегрирования.

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

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

$$\int \ln(x) dx \rightarrow x \cdot (\ln(x) - 1)$$

$$\int_a^b \frac{1}{x^3} dx \xrightarrow{\text{assume, } a > 0, b > 0} \frac{1}{2 \cdot a^2} - \frac{1}{2 \cdot b^2}$$

Рис. 5.5. Примеры символьных вычислений

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

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 5.2.3. Упростить алгебраические выражения, приведенные на рис. 5.6.

Решение. Сначала введем преобразуемые выражения, а затем выполним необходимые упрощения. В первой строке для этого используется оператор символьного вывода (по умолчанию упрощение); во второй – оператор символьного вывода и команда *simplify*.

$$(3 \cdot x + 2) - (5 \cdot x - 3) \rightarrow 5 - 2 \cdot x$$

$$(3 \cdot x^2 - 2 \cdot x - 1) + (4 \cdot x^3 - 5 \cdot x^2 + 1) \xrightarrow{\text{simplify}} 2 \cdot x \cdot (x - 1) \cdot (2 \cdot x + 1)$$

Рис. 5.6. Упрощение алгебраических выражений (*simplify*)

Factor – команда, осуществляющая разложение алгебраического выражения на множители.

Пример 5.2.4. Выполнить разложение выражения, приведенного на рис. 5.7.

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

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

Рис. 5.7. 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 5.2.5. Выполнить разложение на простые дроби выражения, приведенного на рис. 5.8.

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 данного выражения (в котором только одна переменная) результаты этих двух преобразований совпадают (см. рис. 5.8).

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

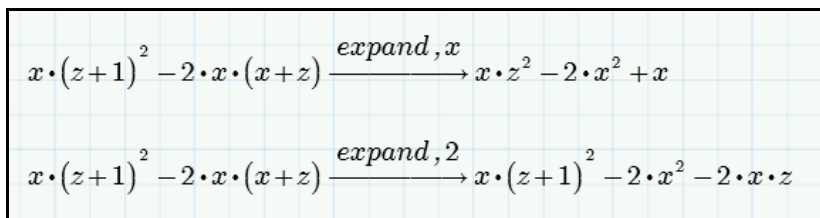
$$\frac{x^2 - 3 \cdot x + 7}{(x-1)^2 \cdot (x^2 + x + 1)} \xrightarrow{\text{parfrac}, x} \frac{6 \cdot x + 10}{3 \cdot (x^2 + x + 1)} - \frac{2}{x-1} + \frac{5}{3 \cdot (x-1)^2}$$

Рис. 5.8. Разложение на простые дроби

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

Пример 5.2.6. Выполнить раскрытие скобок и приведение подобных членов выражения (рис. 5.9).

Решение. Первоначально введем преобразуемое выражение, а затем обратимся к команде *expand*. Первая строка документа на рис. 5.9 содержит преобразования относительно переменной x . Во второй строке приведено преобразование, выполненное относительно константы 2.



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

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

Рис. 5.9. Приведение подобных членов

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 5.2.7. Вычислить коэффициенты полинома, приведенного на рис. 5.10.

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

$$\begin{array}{l}
 (w+2 \cdot u) \cdot z - z^2 \cdot (x+5 \cdot y) + z \xrightarrow{\text{coeffs}, w} \begin{bmatrix} z - x \cdot z^2 - 5 \cdot y \cdot z^2 + 2 \cdot u \cdot z \\ z \end{bmatrix} \\
 (w+2 \cdot u) \cdot z - z^2 \cdot (x+5 \cdot y) + z \xrightarrow{\text{coeffs}, z} \begin{bmatrix} 0 \\ 2 \cdot u + w + 1 \\ -x - 5 \cdot y \end{bmatrix}
 \end{array}$$

Рис. 5.10. 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. 5.11 и обозначены латинскими буквами.

$$\begin{array}{cccc}
 \text{series} \rightarrow ? & \text{series}, x = a \rightarrow ? & \text{series}, n \rightarrow ? & \text{series}, x = a, n \rightarrow ? \\
 \text{а)} & \text{б)} & \text{в)} & \text{г)}
 \end{array}$$

Рис. 5.11. Обращение к команде *series*

Поясним эти формы:

а) подразумевает $x_0 = 0$ (ряд Маклорена), число членов разложения равно 6;

б) подразумевает $x_0 = a$ (ряд Тейлора), число членов разложения равно 6;

в) подразумевает $x_0 = 0$ (ряд Маклорена), число членов разложения равно n ;

г) подразумевает $x_0 = a$ (ряд Тейлора), число членов разложения равно n .

Пример 5.2.8. Выполнить разложение функции $f(x) = e^{-x}$, используя различные формы записи команды *series*.

Решение. На рис. 5.12 приведен фрагмент документа MathCAD, в котором используются различные формы команды *series*.

$$f(x) := e^{-x}$$

$$f(x) \xrightarrow{\text{series}} 1 - x + \frac{x^2}{2} - \frac{x^3}{6} + \frac{x^4}{24} - \frac{x^5}{120}$$

$$f(x) \xrightarrow{\text{series}, 3} 1 - x + \frac{x^2}{2} \quad f(x) \xrightarrow{\text{series}, 4} 1 - x + \frac{x^2}{2} - \frac{x^3}{6}$$

Рис. 5.12. Различные формы 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*, найти решение уравнения $x - a + 5 = 0$ в символьном виде.

Решение. Обращение к команде *solve* показано в документе MathCAD, приведенном на рис. 5.13. 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.

The screenshot shows a MathCAD worksheet with a blue grid background. At the top, the expression $x - a + 5$ is entered, followed by an arrow pointing to a question mark, with the word *solve* written above the arrow. Below this, a red rectangular box contains the Russian text: "Больше одной независимой переменной для ключевого слова solve." (More than one independent variable for the keyword solve). Below the red box, the same expression $x - a + 5$ is entered, followed by an arrow pointing to the expression $a - 5$, with the text *solve, x* written above the arrow.

Рис. 5.13. Решение нелинейного уравнения примера 5.3.1

Пример 5.3.2. Используя команду *solve*, найти 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. 5.14. Заметим, что в левом поле команды *solve* записано нелинейное уравнение, в котором стоит «жирный» знак $=$.

The screenshot shows a MathCAD worksheet with a blue grid background. On the left, the equation $N = N_0 \cdot e^{-\lambda \cdot h}$ is entered, followed by an arrow pointing to the expression $\frac{\ln\left(\frac{N}{N_0}\right)}{h}$, with the word *solve* written above the arrow. On the right, the same equation $N = N_0 \cdot e^{-\lambda \cdot h}$ is entered, followed by an arrow pointing to the expression $N \cdot e^{h \cdot \lambda}$, with the text *solve, N₀* written above the arrow.

Рис. 5.14. 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. 5.15 показаны два обращения к команде *solve*. В первом обращении в левом поле команды стоит запись нелинейного уравнения, во втором – левая часть нелинейного уравнения.

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

$$x^3 - 6 \cdot x^2 + 21 \cdot x - 52 \xrightarrow{\text{solve}, x} \begin{bmatrix} 4 \\ 1 + 2i \cdot \sqrt{3} \\ 1 - 2i \cdot \sqrt{3} \end{bmatrix}$$

Рис. 5.15. 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

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

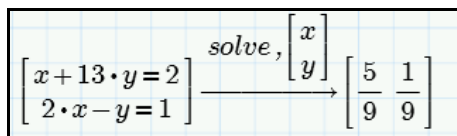
Используя команду *solve*, вычислить:

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. 5.16.

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

Решение. Фрагмент документа MathCAD, вычисляющий решение системы, приведен на рис. 5.17. Вычисленное решение системы выводится в виде строки.



$$\begin{bmatrix} x + 13 \cdot y = 2 \\ 2 \cdot x - y = 1 \end{bmatrix} \xrightarrow{\text{solve}, \begin{bmatrix} x \\ y \end{bmatrix}} \begin{bmatrix} \frac{5}{9} & \frac{1}{9} \end{bmatrix}$$

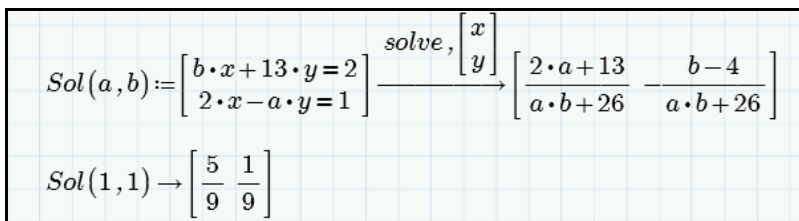
Рис. 5.17. Решение нелинейного уравнения примера 5.3.5

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. 5.18. 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 задачи 5.3.4.



$$Sol(a, b) := \begin{bmatrix} b \cdot x + 13 \cdot y = 2 \\ 2 \cdot x - a \cdot y = 1 \end{bmatrix} \xrightarrow{\text{solve}, \begin{bmatrix} x \\ y \end{bmatrix}} \begin{bmatrix} \frac{2 \cdot a + 13}{a \cdot b + 26} & \frac{b - 4}{a \cdot b + 26} \end{bmatrix}$$

$$Sol(1, 1) \rightarrow \begin{bmatrix} \frac{5}{9} & \frac{1}{9} \end{bmatrix}$$

Рис. 5.18. 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 проекций 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 (рис. 5.19) были подставлены значения, найденные в примере 5.3.3 и являющиеся решением рассматриваемой системы уравнений. Получен вектор, проекции которого содержат равенства. 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.

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. 5.20. Найдены три интервала (заданных с использованием знаков отношений) значений величины x , удовлетворяющих заданному неравенству:

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

$$\begin{array}{l} \text{substitute, } x = \frac{5}{9} \\ \text{substitute, } y = \frac{1}{9} \\ \left[\begin{array}{l} x + 13 \cdot y = 2 \\ 2 \cdot x - y = 1 \end{array} \right] \longrightarrow \left[\begin{array}{l} 2 = 2 \\ 1 = 1 \end{array} \right] \\ \\ \text{substitute, } x = \frac{5}{9} \\ \text{substitute, } y = \frac{2}{9} \\ \left[\begin{array}{l} x + 13 \cdot y = 2 \\ 2 \cdot x - y = 1 \end{array} \right] \longrightarrow \left[\begin{array}{l} \frac{31}{9} = 2 \\ \frac{8}{9} = 1 \end{array} \right] \end{array}$$

Рис. 5.19. Проверка решений нелинейного уравнения

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

Рис. 5.20. Решение неравенства примера 5.3.8

5.4. Численное решение уравнений и систем уравнений

5.4.1. Вычисление корней полиномов

Для вычисления всех корней алгебраического уравнения порядка n , имеющего вид

$$a_n x^n + \dots + a_2 x^2 + a_1 x + a_0$$

рекомендуется использовать функцию **polyroots**. При вызове функции **polyroots** надо определить вектор коэффициентов полинома (начиная со свободного члена) и затем использовать его как аргумент этой функции – **polyroots(v)**. Здесь v – вектор коэффициентов алгебраического уравнения.

Пример 5.4.1. Решить уравнение $x^3 - 3x - 1 = 0$.

С помощью вкладки *Матрицы/таблицы* зададим вектор v . Первый элемент вектора v – это точка пересечения, а последующие элементы являются коэффициентами при степенях x в порядке возрастания; на рис. 5.21 приведен документ MathCAD нахождения корней полинома.

$$v := \begin{bmatrix} -1 \\ -3 \\ 0 \\ 1 \end{bmatrix} \quad \text{polyroots}(v) = \begin{bmatrix} -1.532 \\ -0.347 \\ 1.879 \end{bmatrix}$$

Рис. 5.21. Решение уравнения

Задание 5.4.1. Решите уравнения:

а) $x^3 + (3 + 2i)x^2 + (-4 + 6i)x - 8i = 0$;

б) $2x^4 - x^2 - 10 = 0$;

в) $3x^4 + 4x^3 - 12x^2 - 5 = 0$.

5.4.2. Решение систем линейных уравнений

Для решения систем уравнений надо использовать специальный вычислительный блок – **блок решения**.

Блок решения – это специальная вычислительная область, используемая для решения систем линейных, нелинейных или дифференциальных уравнений, а также для решения задач оптимизации. В блоке решения задачи ставятся в естественной математической форме. Для выполнения решения необходимо задать начальные приближения, а также начальные или граничные условия. Блоки решения осуществляют поиск решения методом итераций, начиная с заданных значений начальных приближений.

Ввод блока решения в документ осуществляется нажатием [Ctrl+1], или на вкладке *Математика* в разделе *Области* необходимо вызвать *Блок решения*.

Функции блока решения обрабатывают задачи, определенные в блоках решения.

Функции ***find***, ***minerr*** решают системы линейных или нелинейных уравнений. Функция ***minerr*** используется для получения приближенного решения в тех случаях, когда функция ***find*** возвращает ошибку.

Функции ***minimize***, ***maximize*** выполняют соответственно минимизацию и максимизацию заданной функции.

Примечание. Функции блока решения можно использовать только в пределах блока решения.

Для выполнения решения системы линейных алгебраических уравнений (СЛАУ) щелкните внутри области блока решения. Затем задайте начальные приближения для всех неизвестных и введите уравнения системы. Нахождение корней системы уравнений осуществляется с помощью функции ***find***.

Пример 5.4.2. Дана система уравнений

$$\begin{cases} 2 \cdot x_1 + 12 \cdot x_2 - 4 = 40 \\ x_2 + 7 \cdot x_3 = 4 \\ 10 \cdot x_1 + 5 \cdot x_2 = -20. \end{cases}$$

Последовательность действий при нахождении корней системы уравнений приведена в документе MathCAD на рис. 5.22.

1. Задайте x_1, x_2, x_3

$$x_1 := 0 \quad x_2 := 0 \quad x_3 := 0$$

2. Используя логические операторы, задайте ограничения для всех неизвестных. Используйте оператор "равно".

$$2 \cdot x_1 + 12 \cdot x_2 - 4 = 40$$

$$x_2 + 7 \cdot x_3 = 4$$

$$10 \cdot x_1 + 5 \cdot x_2 = -20$$

3. Вызовите функции блока решения. Определите *sol* как результат функции *find*

$$\text{find}(x_1, x_2, x_3) = \begin{bmatrix} -4.182 \\ 4.364 \\ -0.052 \end{bmatrix}$$

Рис. 5.22. Решение СЛАУ

Задание 5.4.2. Решите системы уравнений:

$$\text{а) } \begin{cases} 100 \cdot x_1 + 6 \cdot x_2 - 2 \cdot x_3 = 200 \\ 6 \cdot x_1 + 200 \cdot x_2 - 10 \cdot x_3 = 600 \\ x_1 + 2 \cdot x_2 - 100 \cdot x_3 = 500 \end{cases}$$

$$\text{б) } \begin{cases} 10 \cdot x_1 + x_2 + x_3 = 12 \\ 2 \cdot x_1 + 10 \cdot x_2 + x_3 = 13 \\ 2 \cdot x_1 + 2 \cdot x_2 + 10 \cdot x_3 = 14 \end{cases}$$

5.4.3. Решение систем нелинейных уравнений

Для решения систем нелинейных уравнений в блоке решения используются функции **Find**, **Minerr**. Блок решения имеет следующую структуру:

<Начальные условия>

<Уравнения> или <Неравенства>

<Вызов функции Find или Minerr>

Между левой и правой частями уравнений или неравенств должны стоять знаки логических операций $=$, $>$, $<$, $\leq$, $\geq$.

Функция $Find(x1, x2, \dots)$ возвращает решение системы уравнений. Число аргументов должно быть равно числу неизвестных.

Пример 5.4.3. Решить систему нелинейных алгебраических уравнений:

$$\begin{cases} y - 3 \cdot x - 8 = 0 \\ y - x^2 = 0 \end{cases}$$

Определим начальные приближения для решения этой системы. Построим графики (рис. 5.23) указанных функций, чтобы фиксировать точки их пересечений:

$$y1(x) := x^2 \quad x := -3, -2.95..6$$

$$y2(x) := 8 + 3 \cdot x$$

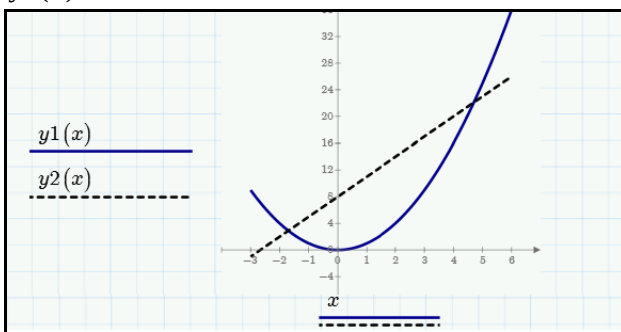


Рис. 5.23. Графики функций совместимых уравнений

1. Найдём первый корень.

- Зададим начальные приближения для нахождения первого корня:

$$x := -2 \quad y := 2$$

- Введем уравнения:

$$y - 3 \cdot x - 8 = 0$$

$$y - x^2 = 0$$

- Проекция первого решения

$$F := \text{Find}(x, y) \quad F = \blacksquare$$

2. Найдём второй корень:

$$x := 5 \quad y := 20 \quad \text{начальные приближения}$$

$$y - 3 \cdot x - 8 = 0$$

$$y - x^2 = 0$$

$$F := \text{Find}(x, y) \quad F = \blacksquare$$

Если система не имеет точного решения, то для приближенного решения используется функция $\text{Minerr}(x1, x2, \dots)$.

Пример 5.4.4. Используя функцию Minerr , решить систему несовместных уравнений:

$$\begin{cases} -3x + y + 6 = 0 \\ -x^2 + y = 0 \end{cases}$$

Чтобы показать, что уравнения несовместные, построим их графики (рис. 5.24):

$$y1(x) := x^2 \quad x := -3, -2.95..6$$

$$y2(x) := 3 \cdot x - 6$$

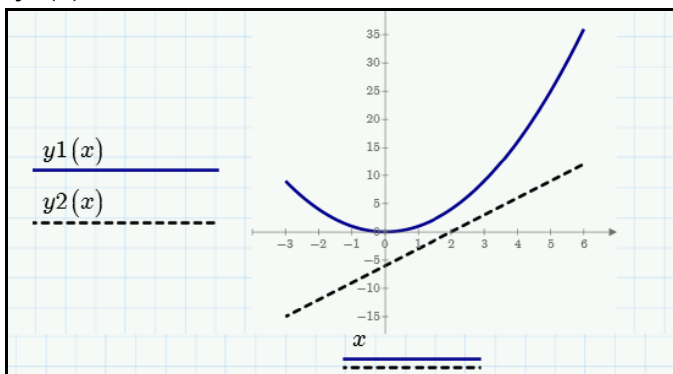


Рис. 5.24. Графики функций несовместных уравнений

Видим, что графики не имеют общих точек. При попытке найти решение с помощью функции **Find** выводится сообщение об ошибке «Решение не найдено», как показано в документе MathCAD, представленном на рис. 5.25 а.

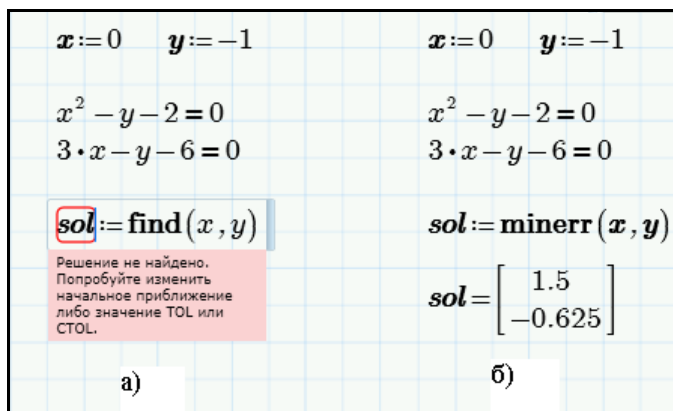


Рис. 5.25. Решение системы несовместных уравнений

При использовании функции **Minerr** получим приближенное значение решения системы нелинейных уравнений (рис. 5.25 б).

5.4.4. Поиск экстремальных значений

При поиске экстремальных значений необходимо определить условия, при которых функция принимает значения экстремума. Эти условия порождают систему уравнений (чаще всего нелинейных), которые располагаются в блоке решения вместе с ограничениями, определяющими допустимую область. Само решение ищется с помощью функций *Find*, *Minerr*.

Пример 5.4.5. В качестве тестового функционала при поиске точки минимума часто используется функционал Розенброка:

$$f(x, y) = 100(y - x^2)^2 + (1 - x^2)^2.$$

«Поверхность» этого функционала напоминает глубокий овраг, что сильно осложняет работу многих алгоритмов минимизации. Требуется вычислить точку минимума функционала при ограничениях:

$$x \geq 0; \quad y \geq 0; \quad y \leq 9 - x.$$

Документ MathCAD решения этой задачи показан на рис. 5.26.

$f(x, y) := 100 \cdot (y - x^2)^2 + (1 - x^2)^2$		Начальные приближения
$x := 2$	$y := 3$	
$\frac{d}{dx} f(x, y) = 0$	$\frac{d}{dy} f(x, y) = 0$	Условие минимума
$x \geq 0$	$y \geq 0$	Ограничения
$y \leq 9 - x$		
$\begin{bmatrix} x \\ y \end{bmatrix} := \text{Minerr}(x, y)$	$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	
$f(x, y) = 4.095 \cdot 10^{-28}$		

Рис. 5.26. Минимизация функций Розенброка

Задание 5.4.3. Используя блок решения, решить следующие системы уравнений:

$$\text{а) } \begin{cases} \sin x + 2y = 2 \\ \cos(y - 1) + x = 0.7 \end{cases} \quad \text{б) } \begin{cases} \sin y + x = -0.4 \\ 2y - \cos(x + 1) = 0 \end{cases}$$

$$\text{в) } \begin{cases} -\sin(x + 1) + y = 0.8 \\ \sin(y - 1) + x = 1.3 \end{cases} \quad \text{г) } \begin{cases} x^2 + y^2 = 36 \\ x + y = 2 \end{cases}$$

$$\text{д) } \begin{cases} x + y = 0.95 \\ (x^2 + 1)^2 + (y^2 + 1)^2 = 5.5 \end{cases}$$

Проверить найденные решения.

5.5. Решение дифференциальных уравнений

Большинство сложных физических и научно-технических задач базируются на решении систем дифференциальных уравнений. Решение осуществляется с помощью блока решения.

Дифференциальное уравнение записывается в традиционной математической форме. В выражении используется оператор логического равенства. Для определения производных используйте оператор производной или оператор штриха, например d/dx и d^2/dx^2 или $y'(x)$ и $y''(x)$.

Примечание. Для ввода символа производной «'» используется комбинация клавиш $[Ctrl + ']$; для ввода символа производной « d/dx » используется комбинация клавиш $[Ctrl + Shift + D]$.

Начальные или конечные граничные условия интегрируемой функции и ее производных вводятся в традиционной форме с использованием оператора логического равенства. Число значений должно совпадать с порядком уравнения.

Решение обыкновенного дифференциального уравнения (ОДУ) выполняет функция $Odesolve(x, xk, n)$, здесь первый аргумент x – всегда имя функции; второй аргумент xk – конечное значение независимой переменной; третий (необязательный) аргумент n – количество шагов интегрирования, за которые должно быть найдено решение. $Odesolve$ возвращает решение в виде

функции (а не в виде массива), представляющей приближенное (численное) решение дифференциального уравнения на заданном интервале времени. Данная функция может быть использована для определения значений интегрируемой функции в различных точках, а также для построения графика.

Пример 5.5.1. Решим дифференциальное уравнение $4 \cdot x''(t) + x(t) = t$ при граничных условиях $x(-4) = 4$ и $x(10) = 12$. Вычисление значения x показано на рис. 5.27:

$$\begin{aligned} 4 \cdot x''(t) + x(t) &= t \\ x(-4) &= 4 \quad x(10) = 12 \\ x &:= \text{odesolve}(x(t), 10) \end{aligned}$$

Рис. 5.27. Решение ОДУ

Построим график решения (рис. 5.28).

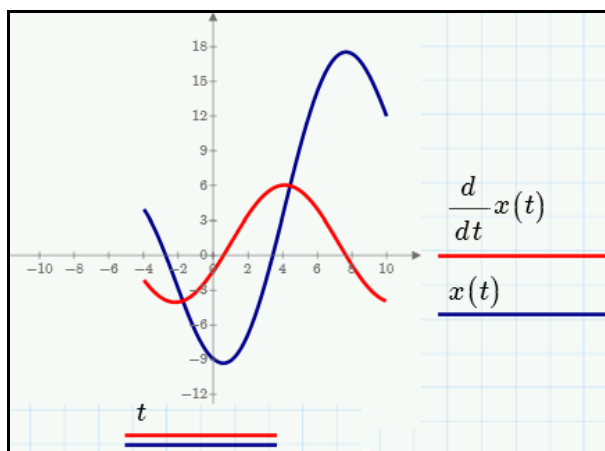


Рис. 5.28. График решения дифференциального уравнения с крайвыми условиями

Пример 5.5.2. Решить дифференциальное уравнение с начальными условиями:

$y''(x) + 4 \cdot y(x) = e^x \cdot \cos(x)$ при начальных условиях $y(0) = -10$ и $y'(0) = 4$; найти значения y и y' , построить графики решения.

Решение показано на рис. 5.29, график решения – на рис. 5.30.

$$\frac{d^2}{dx^2} y(x) + 4 \cdot y(x) = e^x \cdot \cos(x)$$

$$y(0) = -10 \quad y'(0) = 4$$

$$y := \text{odesolve}(y(x), 10)$$

Рис. 5.29. Решение дифференциального уравнения с начальными условиями

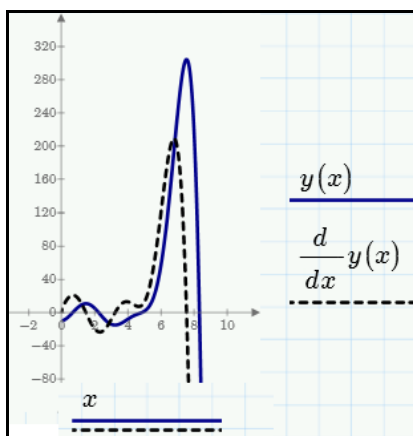


Рис. 5.30. График решения дифференциального уравнения с начальными условиями

5.6. Решение систем обыкновенных дифференциальных уравнений

Для решения систем дифференциальных уравнений используются функции *Odesolve* и *rkfixed*.

Обращение к *rkfixed* выглядит так:

$$rkfixed(y, x1, x2, npoints, D)$$

где y – вектор начальных условий размерности n ;

$x1, x2$ – граничные точки интервала, на которых ищется решение;

$npoints$ – число точек, в которых ищется решение;

$D(x, y)$ – функция, возвращающая значение в виде вектора из n элементов, содержащих первые производные неизвестных функций.

Пример 5.6.1. Решим систему дифференциальных уравнений с помощью функции *Odesolve*:

$$\begin{aligned} x'(t) + 8 \cdot x(t) - 8 \cdot y(t) &= 0, \quad y'(t) - y(t) = 30 \cdot x(t) - x(t) \cdot z(t), \\ z'(t) + 2.6 \cdot z(t) &= x(t) \cdot z(t) \end{aligned}$$

с граничными условиями: $x(0) = -1, y(0) = 0, z(0) = 1$.

Решение системы дифференциальных уравнений приведено на рис. 5.31.

1. Зададим конечную точку интервала T1:

T1 := 10	Конечная точка интервала
----------	--------------------------

2. Вызовем блок решения с помощью [Ctrl+1] и введем в него уравнения, набор начальных приближений и функцию решения *odesolve*:

$$\begin{aligned}
 x'(t) + 8 \cdot x(t) - 8 \cdot y(t) &= 0 \\
 y'(t) - y(t) &= 30 \cdot x(t) - z(t) \cdot x(t) && \text{Уравнения} \\
 z'(t) + 2.6 \cdot z(t) &= x(t) \cdot y(t) \\
 x(0) = -1 \quad y(0) = 0 \quad z(0) &= 1 && \text{Условия} \\
 \begin{bmatrix} f \\ g \\ h \end{bmatrix} &:= \text{odesolve} \left(\begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}, T1, 1 \cdot 10^3 \right)
 \end{aligned}$$

Рис. 5.31. Решение системы ОДУ с помощью *odesolve*

3. Построим график решения на одном интервале (рис. 5.32):

$$k1 := 0, \frac{T1}{1 \cdot 10^3} \dots T1$$

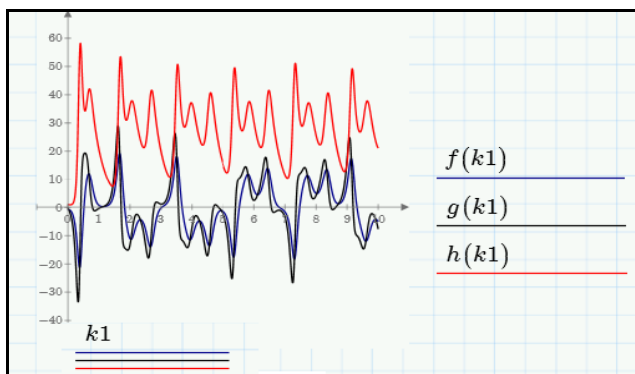


Рис. 5.32. Графики решения системы ОДУ

Пример 5.6.2. Решим систему дифференциальных уравнений с помощью функции *rkfixed*:

$$u'' = 2v, \quad v'' = 4v - 2u$$

с начальными условиями: $u'(0) = 1.5$, $u(0) = 1.5$, $v(0) = 1$, $v'(0) = 1$.

Приведем документ MathCAD с решением системы ОДУ (рис. 5.33):

$$\begin{aligned}
 u''(t) &= 2 \cdot v(t) \\
 v''(t) &= 4 \cdot v(t) \\
 y &:= \begin{bmatrix} 1.5 \\ 1.5 \\ 1 \\ 1 \end{bmatrix} \quad D(x, y) := \begin{bmatrix} y_1 \\ 2 \cdot y_2 \\ y_3 \\ 4 \cdot y_2 - 2 \cdot y_0 \end{bmatrix} \\
 Z &:= \text{rkfixed}(y, 0, 1, 100, D)
 \end{aligned}$$

Получим решение в виде матрицы значений:

$$Z = \begin{bmatrix} 0 & 1.5 & 1.5 & 1 & 1 \\ 0.01 & 1.515 & 1.52 & 1.01 & 1.01 \\ 0.02 & 1.53 & 1.54 & 1.02 & 1.02 \\ 0.03 & 1.546 & 1.561 & 1.03 & 1.03 \\ 0.04 & 1.562 & 1.582 & 1.041 & 1.041 \\ 0.05 & 1.578 & 1.603 & 1.051 & 1.051 \\ 0.06 & 1.594 & 1.624 & 1.062 & 1.062 \\ \vdots & & & & \end{bmatrix}$$

Рис. 5.33. Решение системы ОДУ с помощью *rkfixed*

Задание 5.6.1. Решить дифференциальное уравнение $y''(x) + x^2 \cdot y' + x \cdot y(x) = \cos(x)$ при граничных условиях $y'(0) = 3$, $y(0) = 8$; найти значения y и построить график решения.

Задание 5.6.2. Решить систему дифференциальных уравнений $x''(t) + y(t) = 0$, $y'(t) - z(t) = 0$, $z'(t) - x(t) = 0$, при условиях: $x(-1) = 0$, $x(-1) = -10$, $y(-1) = -10$, $y'(-1) = 10$.

Вопросы и задания для самоконтроля по теме 5

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:

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

$$б) \frac{3 \cdot x^2 - 2 \cdot x}{6 - 7 \cdot x - 3 \cdot x^2}; \quad в) \frac{a - \frac{4 \cdot a - 4}{a}}{\frac{2}{a} - 1}.$$

5. Разложите на множители выражения:

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

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

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:

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

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

8. Определите коэффициенты следующих полиномов:

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

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

9. Вычислите решения следующих нелинейных уравнений и выполните проверку найденных решений:

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

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

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

а) косинуса; б) тангенса.

11. Решите следующие системы нелинейных уравнений:

$$\text{а) } \begin{cases} x \cdot (y + 1) = 12 \\ \frac{x}{y + 1} = 3 \end{cases} \qquad \text{б) } \begin{cases} \frac{4}{x - y} + \frac{12}{x + y} = 3 \\ \frac{8}{x - y} - \frac{18}{x + y} = -1 \end{cases}$$

Тема 6. ЧТЕНИЕ И ЗАПИСЬ ДАННЫХ

В этой теме рассматриваются различные средства MathCAD Prime 2.0 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 с помощью функций, вводимых в формулы с клавиатуры или вызываемых во вкладке *Функции*, раздел *Доступ к файлам* (рис. 6.1). Эти функции позволяют устанавливать нужный формат данных, номера считываемых строк и столбцов и выбирать разделители между данными.

Функции для чтения и записи файлов предназначены для работы с файлами различного типа. Описания сгруппированы по типу файла.

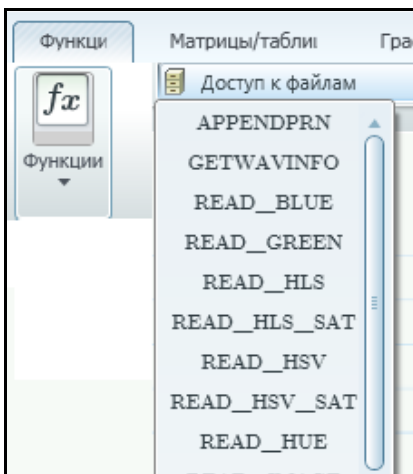


Рис. 6.1. Раздел функций *Доступ к файлам*

Файлы данных

- READFILE, WRITEFILE – читает или записывает массив значений из/в разделенные файлы данных, файлы данных фиксированной ширины или файлы данных Excel.

- READTEXT, WRITETEXT – читает или записывает массив значений из/в текстовые файлы.
- READBIN, WRITEBIN – читает или записывает массив значений из/в бинарные файлы данных.
- READCSV, WRITECSV – читает или записывает массив значений из/в файлы переменных, разделяемых запятой (CSV).
- READPRN, WRITEPRN, APPENDPRN символьного вывода читает, записывает или присоединяет массив значений из/в файлы данных ASCII.
- READEXCEL, WRITEEXCEL символьного вывода читает и записывает массив значений из/в файлы Microsoft Excel.

Файлы изображений

- READ_IMAGE – читает изображения в оттенках серого.
- READBMP, WRITEBMP – читает или записывает растровые изображения в оттенках серого.
- READRGB, WRITERGB – читает или записывает изображения RGB, сохраненные как простые упакованные матрицы.

Свойства изображений

- READ_RED, READ_GREEN, READ_BLUE – читает определенные цветовые компоненты из файлов изображений.
- READ_HLS, WRITE_HLS – читает или записывает массив значений из/в файлы изображений HLS.
- READ_HSV, WRITE_HSV – читает или записывает массив значений из/в файлы изображений HSV.
- READ_HLS_SAT, READ_HSV_SAT, READ_LIGHT, READ_VALUE, READ_HUE – читает насыщенность, светлоту, значение или компонент тона из файлов изображений HLS или HSV.

Звуковые файлы

- READWAV, WRITEWAV – читает и записывает массив, содержащий амплитуды сигнала, в файлы WAV.
- GETWAVINFO – создает вектор, содержащий информацию о файле WAV.

Примечание

- а) При задании имени файла в качестве аргумента одной из функций доступа к файлу необходимо указывать тип файла.
- б) Для всех функций записи оба аргумента *rows* и *cols* являются индексами, начинающимися с 1. Переменная ORIGIN не влияет на выполнение функций чтения.

6.2. 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 *.prn*.

Функция **WRITEPRN**(“*file*”, **M**, [**rows**, [**cols**, [“**decsymb**”]]]) записывает содержимое массива **M** в ASCII-файл с разделителями.

Для вычисления выражения в документе MathCAD оно должно быть записано в виде назначения или запроса на вычисление, например:

```
a := WRITEPRN(file, M)
```

Функция **APPENDPRN**(“*file*”, **M**) записывает содержимое массива **M** в конец ASCII-файла с разделителями. Число столбцов в массиве должно совпадать с числом столбцов в существующем файле. Если файл не существует, то функция **APPENDPRN** создаст его.

Аргументы:

“**file**” – строка, содержащая имя файла или полный путь и имя файла. Неабсолютные пути к файлам считаются заданными по отношению к текущей рабочей папке.

M – массив или (для функции **WRITEPRN**) скаляр.

rows (необязательный) – скаляр, указывающий строку матрицы **M**, с которой следует начинать запись в файл, или двумерный вектор, указывающий диапазон строк матрицы **M** (включительно), которые нужно записать. Если этот аргумент пропущен, функция **WRITEPRN** запишет все строки матрицы в файл.

cols (необязательный) – скаляр, указывающий столбец матрицы **M**, с которой следует начинать запись в файл, или двумерный вектор, указывающий диапазон столбцов матрицы **M** (включительно), которые нужно записать. Если этот аргумент пропущен, функция **WRITEPRN** запишет все столбцы матрицы в файл.

decsymb (необязательный) – используемый десятичный разделитель. Можно использовать только «.» (точку).

Примечание. rows и cols – это индексы, которые начинаются с 1. ORIGIN не влияет на поведение функций READPRN и APPENDPRN.

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. 6.2. В окне программы Блокнот хорошо видна структура сформированного файла (число строк $n = 5$, число столбцов $m = 4$).

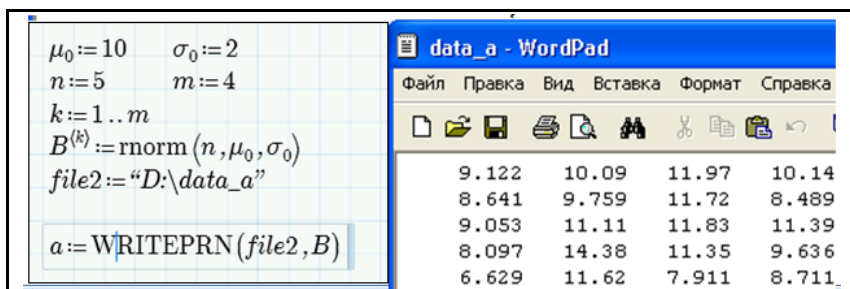


Рис. 6.2. Запись структурированного файла

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

Решение. Фрагмент документа, осуществляющий «дозапись» матрицы C в структурированный файл, приведен на рис. 6.3. 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.

$$C := \begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

$$a := \text{APPENDPRN}(file2, C)$$

$$D := \text{READPRN}("D:\text{data_a.prn}")$$

$$D = \begin{bmatrix} 9.122 & 10.09 & 11.97 & 10.14 \\ 8.641 & 9.759 & 11.72 & 8.489 \\ 9.053 & 11.11 & 11.83 & 11.39 \\ 8.097 & 14.38 & 11.35 & 9.636 \\ 6.629 & 11.62 & 7.911 & 8.711 \\ 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

Рис. 6.3. Дополнительная запись и чтение структурированного файла

6.3. Функции доступа к графическим файлам

Функции MathCAD доступа к графическим файлам, описанные выше, позволяют читать и записывать файлы изображений в различных форматах.

Признанные графические форматы – BMP, GIF, JPG, PCX и TGA.

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 $m \times n$ бок о бок матрицы (функция *augment*).

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

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

Пример 6.3.1. Считывание и обработка BMP-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. Запишем полученную матрицу в файл *MonaS.bmp*.

4. Выведем изображение, представленное матрицей S .

- На вкладке *Математика* в разделе *Области* щелкнем *Изображение*. Появится кнопка *Выбрать изображение*.
- Щелкнем *Выбрать изображение*. Появится диалоговое окно *Открыть*.
- В окне укажем файл *C:\Temp\ MonaS.bmp*. В документе появится выбранное изображение.

Результат выполненных действий показан на рис. 6.4.

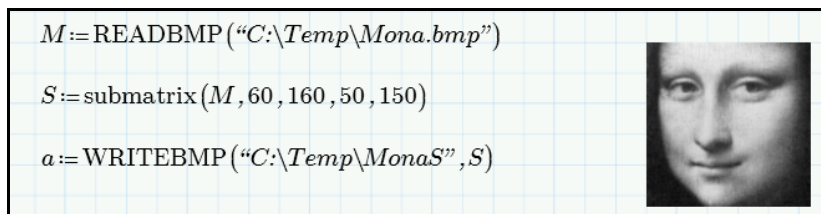


Рис. 6.4. Чтение графического файла

Пример 6.3.2. Продолжим обработку графического файла, прочитанного в примере 6.3.1.

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. 6.5.

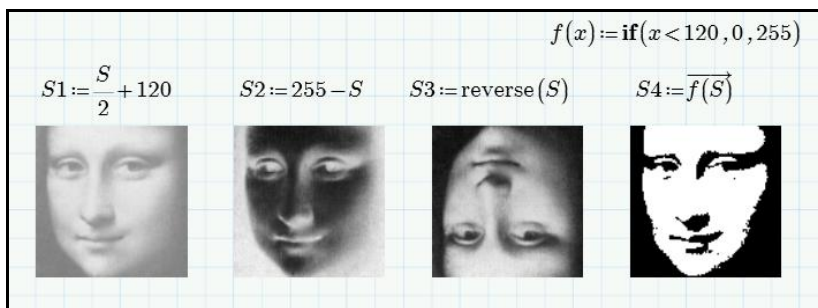


Рис. 6.5. Примеры обработки графического файла

Вопросы и задания для самоконтроля по теме 6

1. Какой файл является структурированным?
2. Какой файл называется ASCII-файлом?
3. К каким типам файлов обеспечивается доступ в MathCAD?
4. Какие форматы графических файлов поддерживают функции в MathCAD?
5. Какие значения имеют элементы массива, содержащего представление серого в растровых изображениях файлов? Что означают эти значения?

6. Составьте 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

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

7. Обработка графического файла. Выполните следующие действия:

- зайдите в MathCAD\Handbook\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). Фрагмент документа MathCAD содержит конструкции, приведенные на рис. 7.1.

$a := 2$	$b := 5$	$c := 8$	$d := \sqrt{b^2 - 4 \cdot a \cdot c} = 6.245i$
$x_1 := \frac{-b + d}{2 \cdot a} = -1.25 + 1.561i$		$x_2 := \frac{-b - d}{2 \cdot a} = -1.25 - 1.561i$	
Проверка найденных решений			
$a \cdot x_1^2 + b \cdot x_1 + c = 0$		$a \cdot x_2^2 + b \cdot x_2 + c = 0$	

Рис. 7.1. Пример программирования линейного алгоритма

Замечание 7.1.1. В этом фрагменте x_1 , x_2 являются простыми переменными, цифры 1, 2 – нижними индексами в именах (а не индексными выражениями у элементов массива). Для вставки этих индексов необходимо обратиться к пункту *Математика*, группа *Стиль*, кнопка *Подстрочный индекс*. В предыдущих версиях MathCAD нужно было нажать клавишу [.] (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 функциям 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 $y = \left(\sqrt{\frac{ax+b}{c+dx}} + \sqrt{\arctg(x)} \right) - e^{-2x}$. 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 значением переменной x . На рис. 7.2 представлена блок-схема этого алгоритма, хорошо подтверждающая название алгоритма – «разветвляющийся».

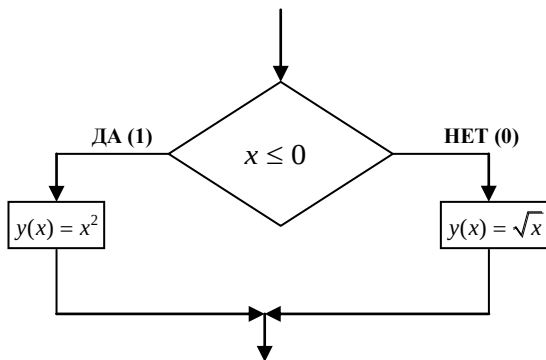


Рис. 7.2. Блок-схема разветвляющегося алгоритма (7.2.1)

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: $|<| \leq |>| \geq | \neq | = |$ (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 *Сравнение* (приведенную на рис. 7.3) или использовать клавиши, обозначения которых приведены в табл. 7.1. Для вызова палитры инструментов *Сравнение* необходимо обратиться к пункту *Математика*, группа *Операторы и символы*, кнопка *Операторы*.

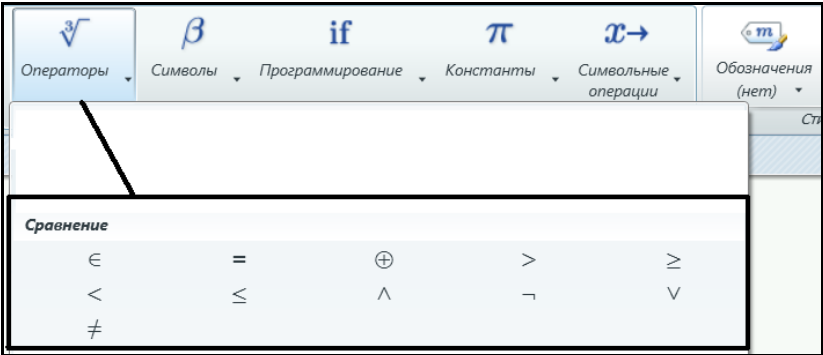


Рис. 7.3. Палитра инструментов *Сравнение*

Таблица 7.1

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

Замечание 7.2.1. Не следует путать знак операции сравнения $=$ с похожим знаком вывода значений переменных. Знак операции $=$ имеет больший размер и более жирное начертание.

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 7.2.2. На рис. 7.4 приведен фрагмент документа, в котором выполнено вычисление некоторых выражений отношений. Выражение отношений позволяет проверить двойные неравенства (см. нижнюю строку фрагмента на рис. 7.4).

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. Эти операции выполняются только над логическими значениями (при этом 1 обозначает ИСТИНА, а 0 – ЛОЖЬ).

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 *Сравнение* (см. рис. 7.3). Результат выполнения этих операций приведен в табл. 7.3.

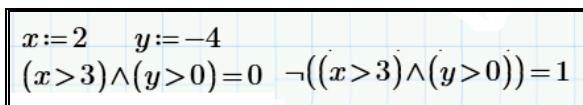
Таблица 7.3

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

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: NOT, AND, операции 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.



$$x:=2 \quad y:=-4$$

$$(x>3)\wedge(y>0)=0 \quad \neg((x>3)\wedge(y>0))=1$$

Рис. 7.5. Примеры логических выражений

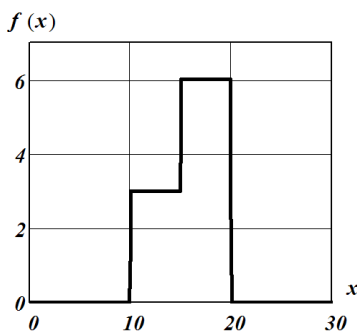
В MathCAD имеется ряд встроенных функций, которые возвращают результат, зависящий от знака или величины аргумента. Приведем некоторые из них:

- **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, значение которой определяется следующим выражением:

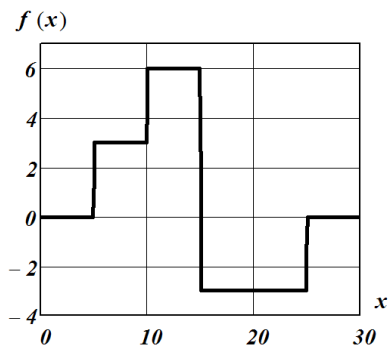
$$\Phi(x) = \begin{cases} 0, & \text{если } x < 0, \\ 0.5, & \text{если } x = 0, \\ 1, & \text{если } x > 0. \end{cases}$$

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

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



а)



б)

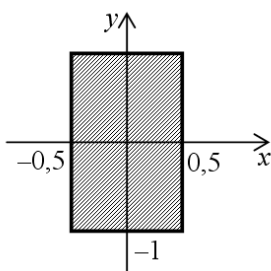
Рис. 7.6. Графики функций к примеру 7.2.3 и заданию 7.2.3

Решение показано в следующем фрагменте документа MathCAD:

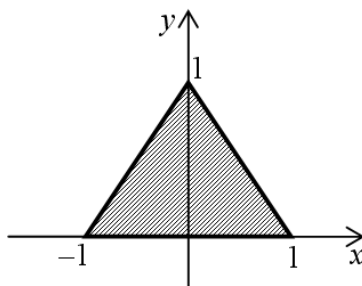
$$f(x) := 3 \cdot (\Phi(x-3) + \Phi(x-15) - 2 \cdot \Phi(x-20))$$

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

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



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:

$x := 0.2$	$y := -0.8$	$f(x, y) := (-0.5 \leq x \leq 0.5) \wedge (-1 \leq y \leq 1)$
$(-0.5 \leq x \leq 0.5) \wedge (-1 \leq y \leq 1) = 1$	$f(-1, 0.3) = 0$	$f(0.2, -0.8) = 1$

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

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.

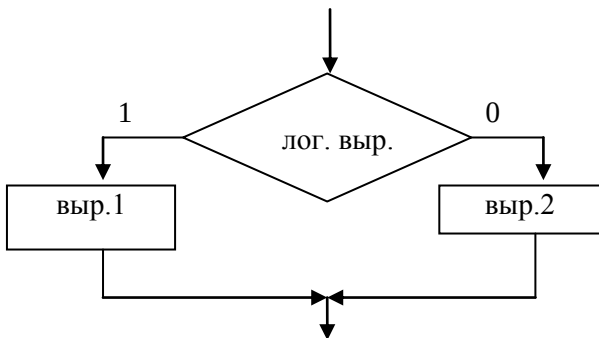


Рис. 7.8. Блок-схема функции *if*

Условная функция *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. Фрагмент документа MathCAD, реализующий этот разветвляющийся алгоритм, приведен на рис. 7.10. Здесь же выполнено тестирование запрограммированной функции.

Замечание 7.2.2. 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).

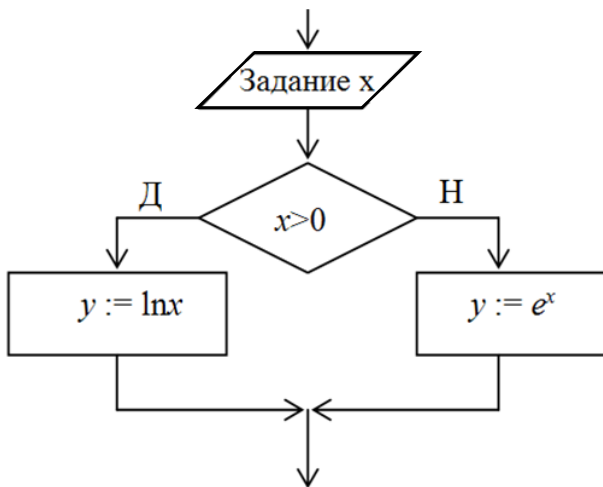


Рис. 7.9. Блок-схема разветвляющегося алгоритма примера 7.2.5:

Д – да; Н – нет

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

Рис. 7.10. Реализация разветвляющегося алгоритма (пример 7.2.5)

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.

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. 7.11. Так как алгоритм содержит три вычислительные ветви, то будем использовать вложенные условные функции, как показано в следующем фрагменте документа MathCAD, приведенном на рис. 7.12.

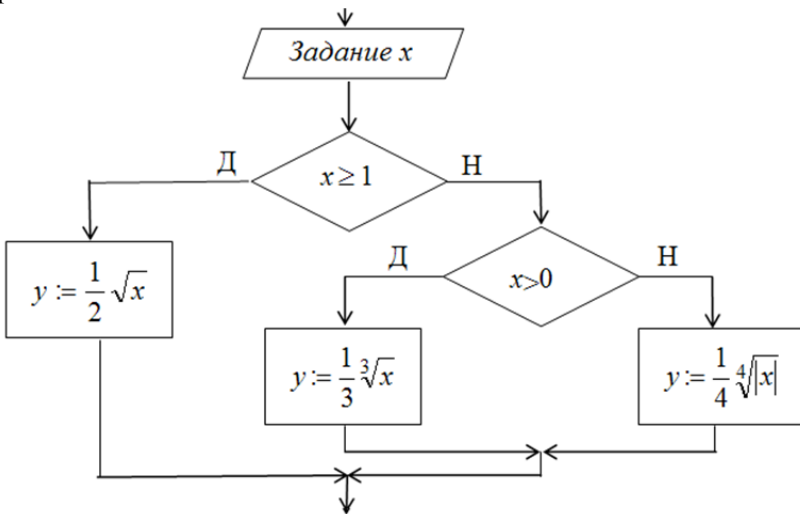


Рис. 7.11. Блок-схема разветвляющегося алгоритма примера 7.2.6

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

$$y(2) = 0.707 \quad y(1) = 0.5 \quad y(-2) = 0.21 + 0.21i$$

Рис. 7.12. Реализация 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. 7.13. Изображенный алгоритм относится к логической структуре «ЕСЛИ-ТО». В таком алгоритме выполняется только одна ветка (если заданное логическое выражение равно 1).

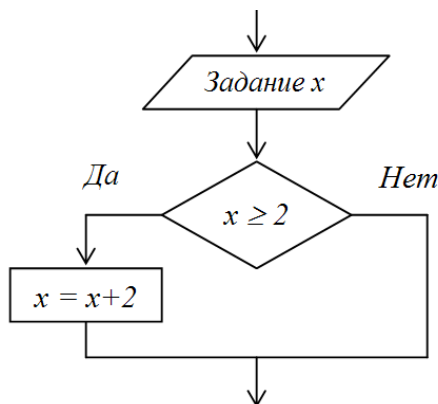


Рис. 7.13. Блок-схема структуры «ЕСЛИ-ТО» (к примеру 7.2.7)

Возникает вопрос: как запрограммировать структуру «ЕСЛИ-ТО» в документе MathCAD? Пример реализации структуры «ЕСЛИ-ТО» демонстрируется во фрагменте документа MathCAD, приведенном на рис. 7.14. Видно, что при $x < 2$ переменная x не меняет своего значения.

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

Рис. 7.14. Реализация структуры «ЕСЛИ-ТО» примера 7.2.7

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. 7.15.

Для правильной организации цикла необходимы четыре блока: блок 1 (см. рис. 7.15) – подготовка к циклу (обычно задание начального значения параметра цикла); блок 2 – тело цикла (содержит повторяющиеся вычисления); блок 3 – подготовка к повторению цикла (меняется значение параметра цикла); блок 4 – проверка условия окончания или условия продолжения цикла. 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.

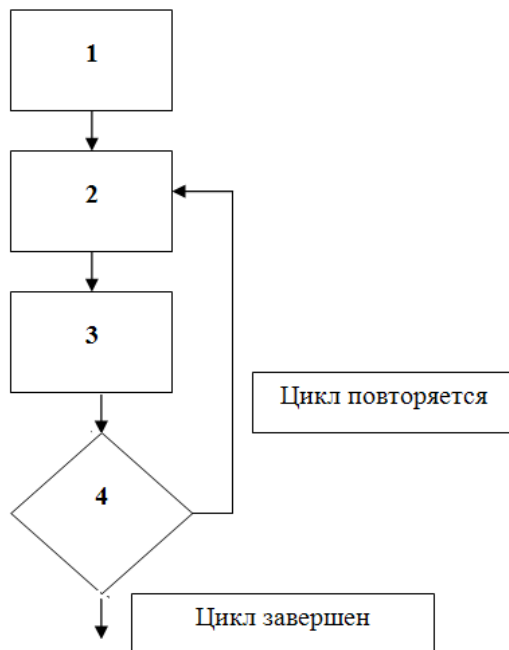


Рис. 7.15. Обобщенная блок-схема циклического алгоритма

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, поэтому невозможно априори, т.е. не выполняя вычислений, определить количество повторений тела цикла. Итерационные циклы используются в таких вычислительных алгоритмах, где вместо точного решения рассматриваемой задачи ищется приближенное решение, вычисляемое последовательно. При этом чем больше итераций (т.е. повторений тела цикла), тем с большей точностью получается 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. 7.16 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.

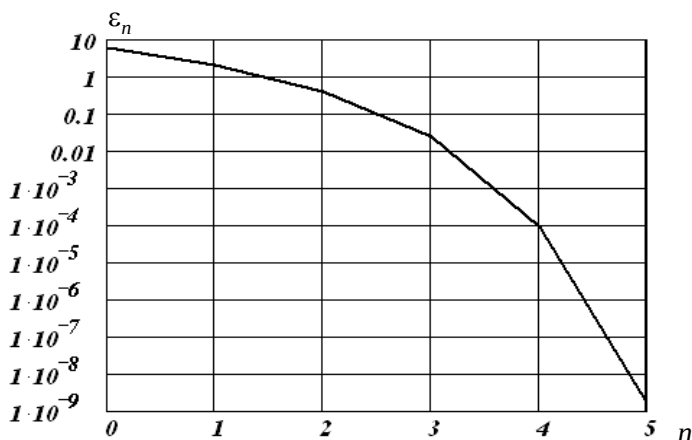


Рис. 7.16. График сходимости итерационного процесса (7.3.2)

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

$$y(x) = \frac{\ln(x) + 1}{a^2 + b^2}$$

для всех x , изменяющихся в интервале $[-0.5, 0.5]$ с шагом $\Delta x = 0.1$, где a, b – заданные вещественные числа.

Решение. В данной задаче переменная x является управляющей переменной цикла. На рис. 7.17 а приведена блок-схема изменения значений этой переменной. В программной реализации (см. рис. 7.17 б) параметр цикла задается дискретной переменной, которая объединяет функции блоков 1, 3, 4 (см. рис. 7.15). Тело цикла начинается после задания дискретной переменной и включает вычисление очередного значения функции $y(x)$. Описание

функции пользователя $y(x)$ и задание переменных a, b размещаются в документе MathCAD до начала цикла.

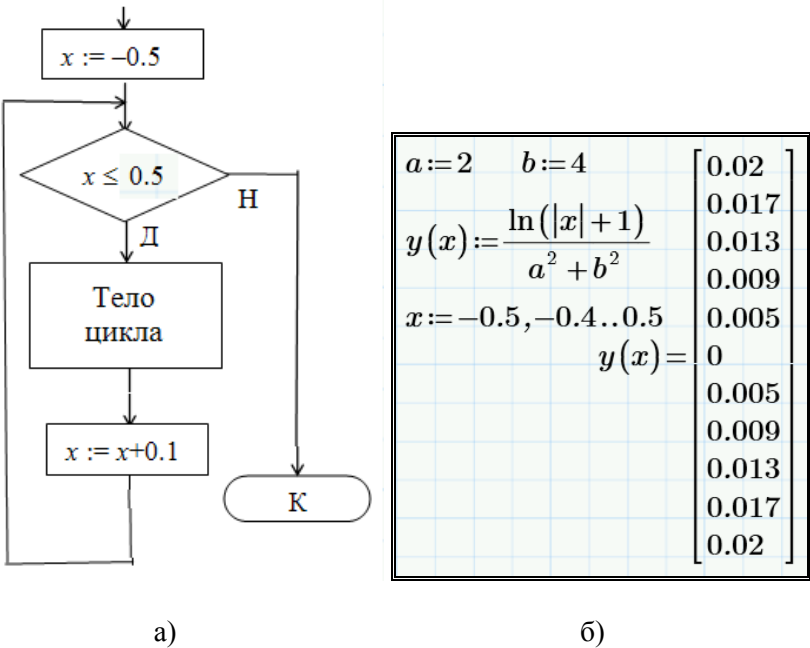


Рис. 7.17. К примеру 7.3.3:
Д – да; Н – нет; К – конец цикла

Задание 7.3.1. Определите количество вычисленных значений функции $y(x)$ в примере 7.3.3, используя формулу (7.3.1).

Фрагмент документа, в котором реализован алгоритм решения задачи примера 7.3.1, приведен на рис. 7.18. Здесь параметром цикла является переменная i , которая задается дискретной переменной.

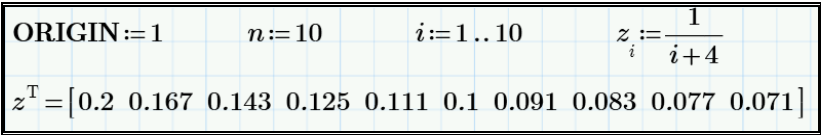
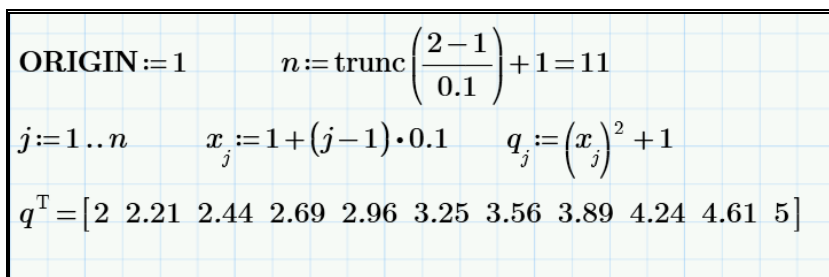


Рис. 7.18. Формирование вектора z примера 7.3.1

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

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 документа, реализующий рассмотренный алгоритм формирования вектора q , приведен на рис. 7.19.



The image shows a screenshot of a MathCAD document with a light blue grid background. The code is as follows:

$$\text{ORIGIN} := 1 \qquad n := \text{trunc} \left(\frac{2 - 1}{0.1} \right) + 1 = 11$$

$$j := 1 \dots n \qquad x_j := 1 + (j - 1) \cdot 0.1 \qquad q_j := (x_j)^2 + 1$$

$$q^T = [2 \quad 2.21 \quad 2.44 \quad 2.69 \quad 2.96 \quad 3.25 \quad 3.56 \quad 3.89 \quad 4.24 \quad 4.61 \quad 5]$$

Рис. 7.19. Формирование вектора в примере 7.3.4

Замечание 7.3.1. 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) B по следующему правилу:

$$B_{i,j} = \frac{1}{i+j+1}; \quad i = 1, \dots, 3; \quad j = 1, \dots, 6.$$

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 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. 7.20 приведен фрагмент документа, формирующий матрицу B в соответствии с принятыми параметрами цикла. В этом фрагменте для каждого значения параметра внешнего цикла i (в документе он задается раньше) параметр внутреннего цикла принимает все значения от 1 до 6, т.е. формирование матрицы выполняется построчно.

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

Рис. 7.20. Формирование матрицы B примера 7.3.5

Заметим, что в приведенных на рис. 7.15, 7.16, 7.17 фрагментах системная переменная $ORIGIN=1$.

Пример 7.3.6. Сформировать матрицу (двумерный массив)

D следующей структуры: $D = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 4 \end{bmatrix}$. Видно, что эле-

менты, лежащие левее главной диагонали, равны 0, а остальные равны номеру строки. Эту закономерность можно выразить следующим алгоритмом:

$$D_{i,j} = \begin{cases} 0, & \text{если } j < i, \\ i, & \text{если } j \geq i. \end{cases}$$

Решение. Видно, что приведенный алгоритм формирования содержит двойной цикл, и в теле внутреннего цикла находится разветвляющийся алгоритм вычисления элемента $D_{i,j}$. Фрагмент документа, реализующий описанный алгоритм формирования, показан на рис. 7.21.

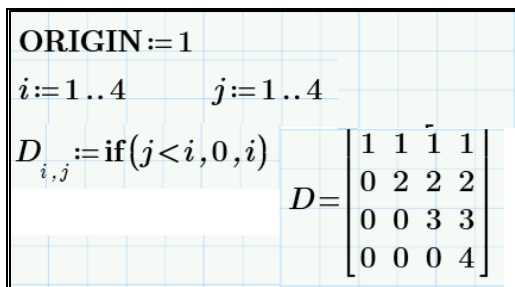


Рис. 7.21. Формирование матрицы D примера 7.3.6

Программирование итерационных циклов. Для программирования итерационных циклов необходимо использовать подпрограмму-функцию с оператором цикла *while*.

В заключение заметим, что рассмотренные способы безмодульного программирования можно рекомендовать для реализации достаточно простых алгоритмов. Более сложные алгоритмы целесообразно реализовывать в виде подпрограммы-

функции, которая может содержать конструкции, присутствующие в алгоритмических языках высокого уровня (условные операторы, операторы цикла и т.д.). В следующих темах будут рассмотрены конструкции пакета MathCAD Prime 2.0, позволяющие реализовывать различные алгоритмы в подпрограмме-функции.

Вопросы и задания для самоконтроля по теме 7

1. Назовите характерную черту безмодульного программирования.
2. Назовите характерную особенность линейного алгоритма.
3. Составьте блок-схему и реализуйте 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. Назовите характерную особенность разветвляющегося алгоритма.
5. Условная функция *if* и результат ее выполнения.
6. Составьте блок-схему и реализуйте 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. Составьте блок-схему и реализуйте 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. Составьте блок-схему и реализуйте в документе MathCAD алгоритм формирования вектора, состоящего из $n = 10$ элементов по правилу:

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

Выведите сформированный вектор.

10. Составьте блок-схему и реализуйте в документе MathCAD алгоритм формирования вектора z , состоящего из $n = 6$ элементов по правилу:

$$z_i = \begin{cases} x_i, & \text{если } x_i \leq 0, \\ \ln(x_i), & \text{если } x_i > 0, \end{cases} i = 1, 2, \dots, n.$$

Выведите сформированный вектор.

11. В чем заключается характерная особенность двойного цикла?

12. Какая закономерность существует в изменении значений параметров внутреннего и внешнего циклов?

13. Составьте блок-схему и реализуйте в документе MathCAD алгоритм формирования матрицы M размером 4×4 следующей структуры:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}.$$

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 реализации 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 опре-

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

- Модульную программу легче сопровождать и модифицировать. В модуль можно внести 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 описание.

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. К этой палитре можно обратиться через пункт *Математика*, группа *Операторы и символы*, кнопка *Программирование*.

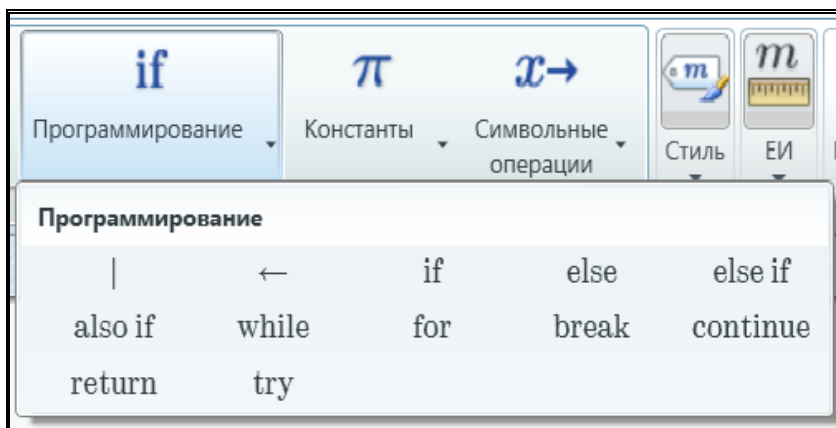


Рис. 8.1. Палитра инструментов *Программирование*

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 внутрь 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 8.2.1. 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:

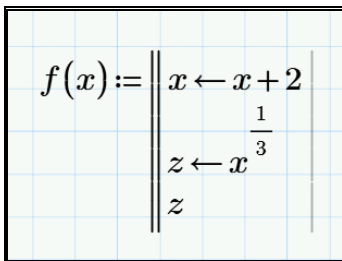
- ввести имя П-Ф и список формальных параметров, заключенный в круглые скобки (см. замечание 8.2.1);
- ввести символ “:” — на экране отображается как “:=”;
- открыть палитру *Программирование* и щелкнуть кноп-

кой  с изображением вертикальной черты (см. рис. 8.1). На экране появится вертикальная черта и одна строка (поле) для ввода первого оператора тела П-Ф;

- для ввода других строк (полей) в тело П-Ф необходимо нажать клавишу [Enter]). При этом новая строка добавляется внизу от выделенной к этому моменту строки;
- в последней строке тела П-Ф размещается конструкция, определяющая результат работы П-Ф, и этот результат передается через имя П-Ф в программу, из которой была вызвана П-Ф;

- для удаления того или иного оператора или поля ввода из тела П-Ф нужно выделить исключающий элемент и нажать клавишу [Delete].

На рис. 8.2 приведен пример описания П-Ф, в которой формальным параметром является простая переменная x , тело П-Ф включает два локальных оператора присваивания (см. п. 8.3) и значение переменной z , которая определяет возвращаемый (через имя П-Ф) результат выполнения П-Ф. Эта переменная находится в последней строке П-Ф.



$$f(x) := \begin{array}{|l} x \leftarrow x + 2 \\ z \leftarrow x^{\frac{1}{3}} \\ z \end{array}$$

Рис. 8.2. Пример описания подпрограммы-функции

Замечание 8.2.2. 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:

$$< \text{имя переменной} > \leftarrow < \text{выражение} >$$

Символ $\leftarrow$ вводится с палитры *Программирование*.

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.

$\langle \text{имя П-Ф} \rangle (\langle \text{список фактических параметров} \rangle)$

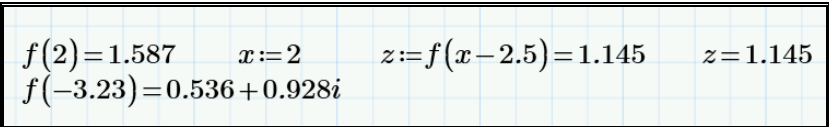
Фактические параметры указывают, при каких конкретных значениях осуществляются вычисления в теле П-Ф. Фактические параметры отделяются друг от друга запятой.

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 8.3.1. Обращение к П-Ф должно находиться после ее описания, и *к моменту обращения фактические параметры должны быть определены*.

Пример 8.3.1. Обращения к программе $f(x)$, приведенной на рис. 8.2, могут иметь следующий вид:



The image shows a screenshot of a MathCAD worksheet with a light blue grid background. It contains two lines of text. The first line shows three calculations: $f(2) = 1.587$, $x := 2$, and $z := f(x - 2.5) = 1.145$, followed by $z = 1.145$. The second line shows $f(-3.23) = 0.536 + 0.928i$.

$$f(2) = 1.587 \quad x := 2 \quad z := f(x - 2.5) = 1.145 \quad z = 1.145$$
$$f(-3.23) = 0.536 + 0.928i$$

Рис. 8.3. Обращения к подпрограмме-функции

Заметим, что переменная z никак не связана с локальной переменной z , используемой внутри тела П-Ф.

Замечание 8.3.2. Передать данные внутрь П-Ф можно используя внутри подпрограммы переменные, определенные до описания П-Ф (см. пример на рис. 8.4).

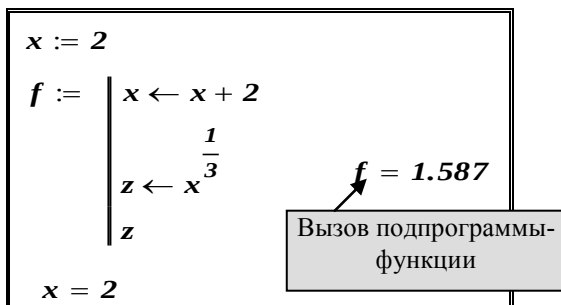


Рис. 8.4. Подпрограмма-функция без формальных параметров

Хотя значение переменной x изменилось внутри П-Ф, вне описания П-Ф эта переменная сохранила свое прежнее значение.

Замечание 8.3.3. 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. 8.5. Эта 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 имена этих переменных имеют нижние индексы. Видно, что приведенная П-Ф реализует **линейный алгоритм** – все операторы выполняются всегда строго последовательно.

The image shows a MathCAD worksheet with a blue grid background. It contains the following code:

$$\text{root_poly2}(a, b, c) := \left[\begin{array}{l} d \leftarrow \sqrt{b^2 - 4 \cdot a \cdot c} \\ x_1 \leftarrow \frac{-b + d}{2 \cdot a} \\ x_2 \leftarrow \frac{-b - d}{2 \cdot a} \\ \left[\begin{array}{l} x_1 \\ x_2 \end{array} \right] \end{array} \right]$$

$$z := \text{root_poly2}(12, 7, 10) = \left[\begin{array}{l} -0.292 + 0.865i \\ -0.292 - 0.865i \end{array} \right]$$

Рис. 8.5. Программирование линейного алгоритма
(к примеру 8.4.1)

Задание 8.4.1. Вставьте в документ MathCAD описание П-Ф *root_poly2* и подберите коэффициенты квадратного уравнения, чтобы корни уравнения были вещественными.

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.

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 появится конструкция с двумя полями ввода, изображенная на рис. 8.6.

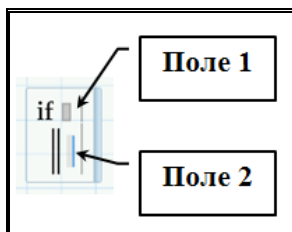


Рис. 8.6. Структура условного оператора *if*

В поле 1 вводится логическое выражение УСЛ (в простейшем случае это выражение отношений). В поле 2 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 *else*, вводимый с палитры *Программирование*. Для ввода этого оператора достаточно установить курсор в поле 1 или в поле 2 оператора *if*, а затем щелкнуть на кнопке **else** палитры *Программирование*. Получаем конструкцию, показанную на рис. 8.7 а. В поле 3 вводятся конструкции, которые будут выполняться, если логическое выражение УСЛ в поле 1 будет равно 0 (т.е. FALSE).

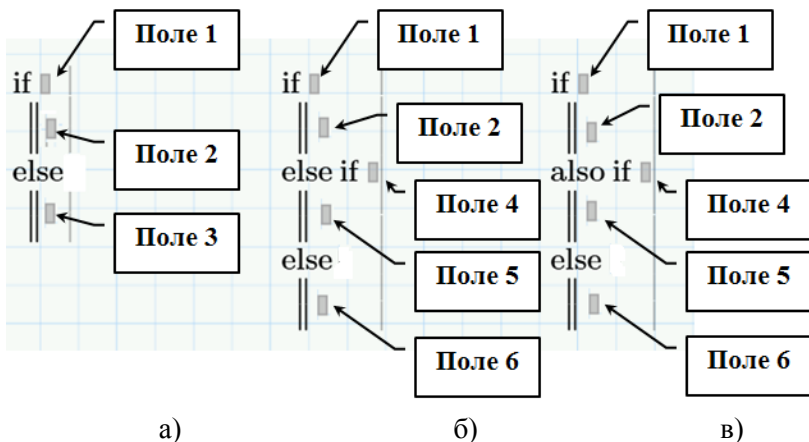


Рис. 8.7. Различные структуры условных операторов

Пример 8.5.1. Составим описание П-Ф, вычисляющей функцию $y(x)$, заданную в примере 7.2.1.

Решение. Описание и вызов П-Ф приведены на рис. 8.8. Заметим, что тело П-Ф содержит только один условный оператор, и результат его выполнения определяет значение П-Ф.

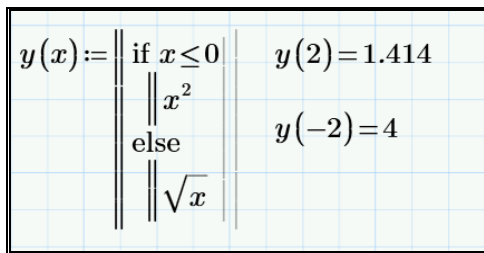


Рис. 8.8. Реализация разветвляющегося алгоритма примера 8.5.1

Замечание 8.5.1. Конструкции, стоящие в поле 3 оператора `else`, будут выполняться только тогда, когда не выполнены конструкции, стоящие в других полях оператора `if`. Поэтому этот оператор стоит, как правило, в конце оператора `if` (см. формы на рис. 8.7).

Для реализации более сложных разветвляющихся алгоритмов используются формы оператора *if*, показанные на рис. 8.7 б и 8.7 в, образованные с помощью операторов *elseif* и *alsoif*. Эти операторы вводятся в оператор *if* точно так же, как и в оператор *else*.

Оператор *elseif* работает следующим образом: конструкции, стоящие в поле 5 (см. рис. 8.7 б), будут выполняться только в том случае, если логическое выражение в поле 4 равно 1 (т.е. TRUE) и не были выполнены конструкции, стоящие выше этого оператора *elseif*.

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. 8.9. Здесь имена формального параметра (переменная t) и фактического параметра (значения дискретной переменной x) не совпадают.

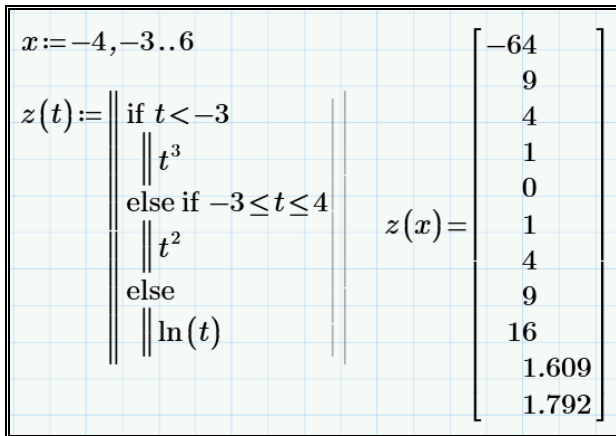


Рис. 8.9. Реализация разветвляющегося алгоритма примера 8.5.2

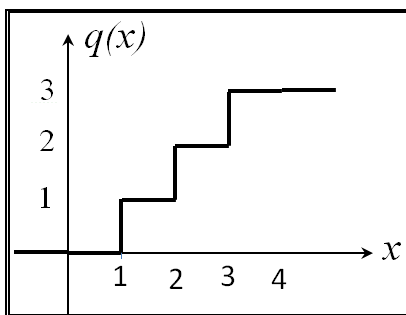
Оператор *alsoif* работает следующим образом: конструкции, стоящие в поле 5 (см. рис. 8.7 в), будут выполняться только в том случае, если логическое выражение в поле 4 равно 1 (т.е. TRUE) и не зависит от того, были ли выполнены конструкции, стоящие выше этого оператора *alsoif*.

Пример 8.5.3. Используя оператор *alsoif*, запрограммировать в виде П-Ф алгоритм вычисления функции, график которой приведен на рис. 8.10 а.

Решение. Фрагмент документа с описанием П-Ф приведен на рис. 8.10 б.

Замечание 8.5.2. Операторы *elseif* и *alsoif* используются для реализации разветвляющегося алгоритма с тремя и более ветвями вычислений путем реализации вложенных условных конструкций. Однако такой способ реализации имеет определенные недостатки: он не соответствует аксиоматике структурного программирования; затруднителен при анализе программной реализации. Поэтому приведем следующие рекомендации по программированию разветвляющихся алгоритмов в П-Ф MathCAD:

- если разветвляющийся алгоритм содержит две ветви, то следует использовать структуру «**ЕСЛИ-ТО-ИНАЧЕ**» (см. пример 8.5.1);
- если разветвляющийся алгоритм содержит три ветви и более, то следует использовать структуры «**ЕСЛИ-ТО**», при этом количество структур равно (или меньше на одну) числу ветвей алгоритма (см. пример 8.5.3).



а)

```

q(x) := || if x > 1
        || qq ← 1
        || also if x > 2
        || qq ← 2
        || also if x > 3
        || qq ← 3
        || else
        || qq ← 0

```

б)

Рис. 8.10. К примеру 8.5.3

Пример 8.5.4. Составим описание П-Ф для вычисления переменной $z(t)$, определенной в примере 8.5.2, используя структуры «ЕСЛИ-ТО».

Решение. Описание П-Ф и ее вызов приведены на рис. 8.11. Сравнивая реализацию разветвляющегося алгоритма в этом примере и в примере 8.5.2, можно сделать вывод, что реализация этого примера обеспечивает более легкое чтение и понимание вычислительного алгоритма.

$x := -4, -3..6$		-64
$z(t) :=$	if $t < -3$	9
	$y \leftarrow t^3$	4
	if $-3 \leq t \leq 4$	1
	$y \leftarrow t^2$	0
	if $t > 4$	1
	$y \leftarrow \ln(t)$	4
	y	9
		16
		1.609
		1.792

$z(x) =$

Рис. 8.11. Реализация разветвляющегося алгоритма примера 8.5.4

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 $y = |x|$

Задание 8.5.2. На рис. 8.12 приведены два варианта П-Ф. Определите, будут ли эти варианты (и если да, то какие) правильно реализовывать алгоритм вычисления функции $z(t)$ примера 8.5.2.

Замечание 8.5.3. Если в полях 2, 3, 5, 6 (см. рис. 8.7) необходимо разместить несколько конструкций, то для ввода дополнительных полей достаточно поставить курсор в нужное поле и нажать клавишу [Enter].

$z(t) := \begin{cases} y \leftarrow t^3 \\ \text{if } -3 \leq t \leq 4 \\ \quad y \leftarrow t^2 \\ \text{if } t > 4 \\ \quad y \leftarrow \ln(t) \\ y \end{cases}$	$z(t) := \begin{cases} \text{if } t < -3 \\ \quad y \leftarrow t^3 \\ \text{if } -3 \leq t \leq 4 \\ \quad y \leftarrow t^2 \\ y \leftarrow \ln(t) \\ y \end{cases}$
а)	б)

Рис. 8.12. Варианты реализации алгоритма вычисления $z(t)$

Пример 8.5.5. 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 и обращение к ней при разных значениях фактических параметров приведено на рис. 8.13. Обратите внимание, что коэффициенты полиномов передаются через массивы a , b .

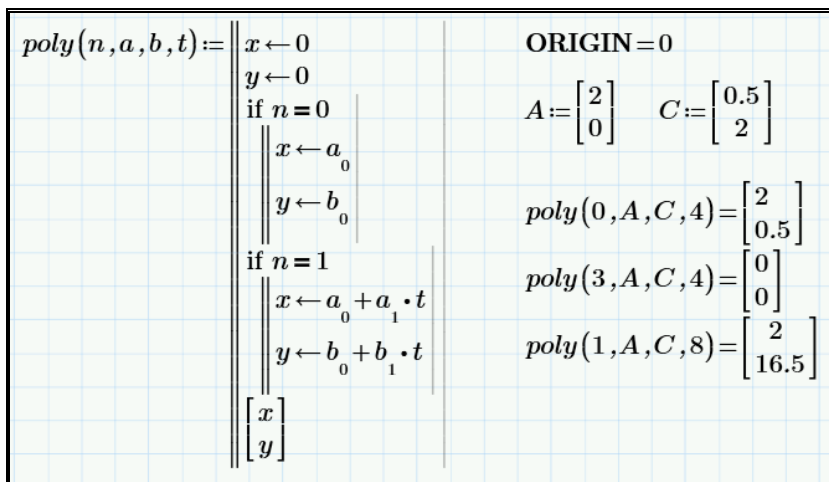


Рис. 8.13. Реализация алгоритма примера 8.5.5

Пример 8.5.6. Даны два числа x , y . Составить описание П-Ф, которая формирует вектор, первая проекция которого равна $\max(x, y)$, а вторая проекция – $\min(x, y)$.

Решение. Описание П-Ф и обращение к ней приведено на рис. 8.14. Выходным параметром является массив v .

$arrangement(x, y) :=$	if $x > y$	ORIGIN = 0
	$v_0 \leftarrow x$	$arrangement(13, 10) = \begin{bmatrix} 13 \\ 10 \end{bmatrix}$
	$v_1 \leftarrow y$	
	else	$arrangement(10, 13) = \begin{bmatrix} 13 \\ 10 \end{bmatrix}$
	$v_0 \leftarrow y$	
	$v_1 \leftarrow x$	$arrangement(13, 13) = \begin{bmatrix} 13 \\ 13 \end{bmatrix}$
	v	

Рис. 8.14. Реализация алгоритма примера 8.4.4

Задание 8.5.3. 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. Выполнить тестирование П-Ф.

Задание 8.5.4. Координаты точки на плоскости задаются двумя числами x, y . Составить П-Ф, вычисляющую номер четверти на плоскости, в которую попала точка. Выполнить тестирование П-Ф.

Задание 8.5.5. 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. Выполнить тестирование П-Ф.

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:

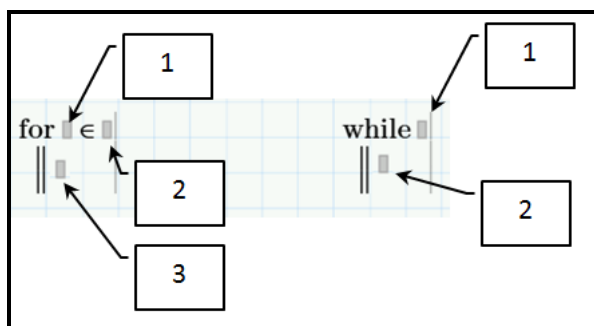
- *циклы типа арифметической прогрессии;*
- *итерационные циклы.*

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. 8.15 а;



а) б)
Рис. 8.15. Структуры операторов циклов

- 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.

Пример 8.6.1. Составить описание П-Ф, реализующей алгоритм примера 7.3.1.

Решение. Описание П-Ф и ее вызов приведены на рис. 8.16. Заметим, что значение системной переменной *ORIGIN* (начальное значение индексного выражения) задается равным 1.

ORIGIN := 1

$$Form_Vec(n) := \left\| \begin{array}{l} \text{for } i \in 1..n \\ \left\| z_i \leftarrow \frac{1}{i^2 + 1} \right\| \\ z \end{array} \right\| \quad Form_Vec(5) = \begin{bmatrix} 0.5 \\ 0.2 \\ 0.1 \\ 0.059 \\ 0.038 \end{bmatrix}$$

Рис. 8.16. Подпрограмма-функция формирования вектора

Пример 8.6.2. Для x , меняющегося от -2 до 2 с шагом 0.5 , вычислить значение $f(x) = e^{-x} \cdot \cos(2x)$ 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. 8.17. Видно, что в теле цикла выполняется два оператора. Первый оператор формирует элемент массива y , а второй изменяет на 1 значение индекса.

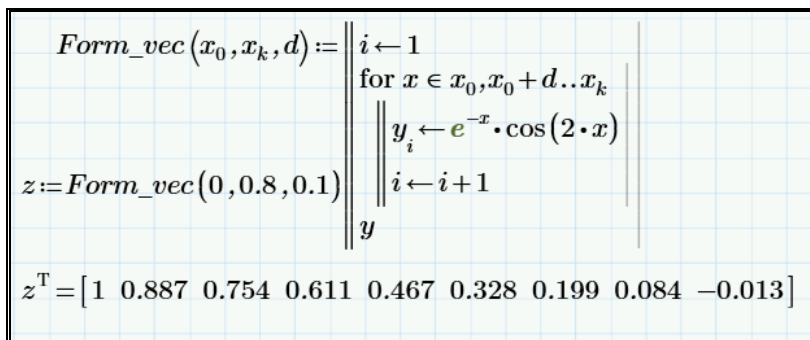


Рис. 8.17. Формирование вектора примера 8.6.2

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 8.6.2.

Вторая особенность заключается в том, что значения переменной цикла могут задаваться вектором или матрицей, а также последовательностью чисел. Это иллюстрируется примерами 8.6.3 и 8.6.4.

Третья особенность состоит в том, что переменная цикла может принимать значения различных типов – численное, текстовое и т.д. Эту особенность иллюстрирует пример 8.6.5.

Пример 8.6.3. Составить описание П-Ф, вычисляющей сумму вектора с проекциями $|2, 5, 7|$.

Решение. На рис. 8.18 приведено описание П-Ф и обращение к ней, в которой слагаемые задаются значениями параметра цикла.

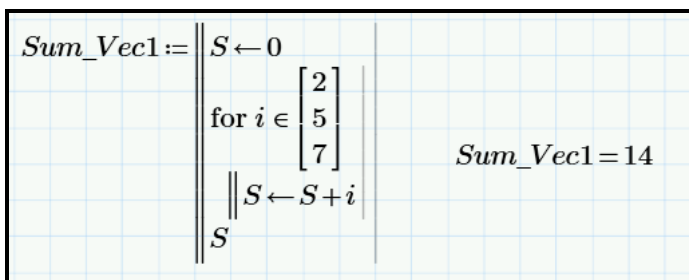


Рис. 8.18. Описание П-Ф примера 8.6.3

Пример 8.6.4. Запрограммировать П-Ф, которая формирует двумерный массив из объектов разных типов.

Решение. На рис. 8.19 приведено описание П-Ф и обращение к ней. Видно, что переменная цикла i_M принимает значения различных типов: вещественное, комплексное и строковое.

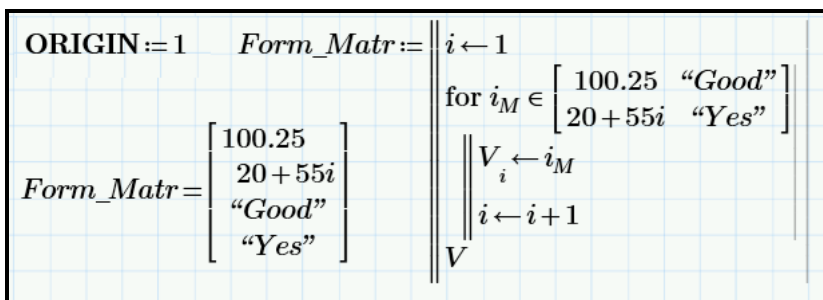


Рис. 8.19. Описание П-Ф примера 8.6.4

Задание 8.6.1. Составьте описание П-Ф формирования вектора y примера 8.6.2, приняв в качестве параметра цикла переменную i . Выполните тестирование этой П-Ф.

Задание 8.6.2. Составьте описание П-Ф формирования вектора q примера 7.3.4. Выполните тестирование этой П-Ф.

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 появятся элементы, показанные на рис. 8.15 б;
- в поле 1 ввести условие выполнения цикла;
- в поле 2 ввести операторы тела цикла. В теле цикла должны присутствовать операторы, которые *должны менять 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 8.6.5. Составим П-Ф, реализующую итерационную процедуру приближенного вычисления корня квадратного, описанную в примере 7.3.2.

Решение. Описание П-Ф приведено на рис. 8.20. Как видно из текста П-Ф, нет необходимости хранить в памяти все приближенные решения $x_0, x_1, x_2, \dots$. Достаточно хранить предыдущее (старое) значение (обозначим его как x_c) и последующее (новое) значение x_n . В теле цикла меняется значение x_n , которое входит в условие продолжения цикла.

$$sqroot(a, \varepsilon) := \left\| \begin{array}{l} xn \leftarrow a \\ \text{while } |xn^2 - a| > \varepsilon \\ \quad \left\| \begin{array}{l} xc \leftarrow xn \\ \quad \quad \quad xn \leftarrow \frac{xc + \frac{a}{xc}}{2} \end{array} \right. \end{array} \right\|$$

$$sqroot(9, 10^{-3}) = 3.0000916$$

$$sqroot(9, 10^{-6}) = 3$$

$$sqroot(2, 10^{-6}) = 1.414$$

Рис. 8.20. Реализация итерационного цикла примера 8.6.5

Оператор *break* (средство от заикливания). К сожалению, организация итерационного цикла с помощью оператора *while* без дополнительных средств контроля может привести к заикливанню, 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 условия.

Для ввода оператора *break* необходимо щелкнуть на кнопке *break* панели *Программирование* (нельзя вводить этот оператор с клавиатуры по символам). Оператор *break* размещается в поле ввода 2 условного оператора *if*, а в поле 1 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 заикливания с использованием оператора *break*.

Пример 8.6.6. Составим П-Ф, реализующую итерационную процедуру вычисления корня квадратного без заикливания.

Решение. Описание П-Ф приведено на рис. 8.21. 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.

столбца, на пересечении которых находится указанный элемент. Например, запись $A_{2,4}$ означает обращение к элементу, стоящему на пересечении второй строки и четвертого столбца.

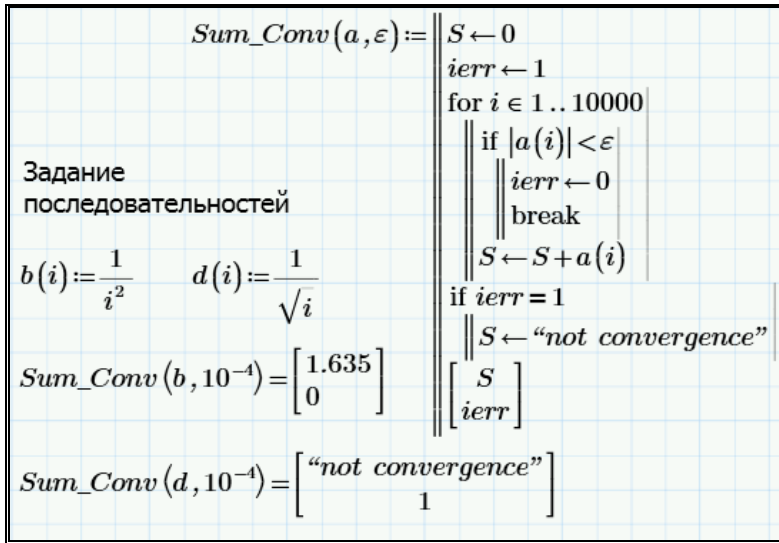


Рис. 8.22. Реализация итерационного цикла примера 8.6.7

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. 8.23).

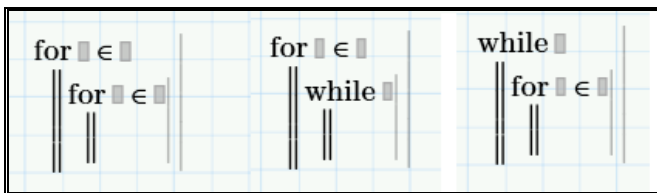


Рис. 8.23. Варианты вложений операторов цикла

Пример 8.6.8. 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. 8.24. Системная переменная $ORIGIN = 1$. В этой П-Ф параметром внешнего цикла является переменная i , а параметром внутреннего – переменная j .

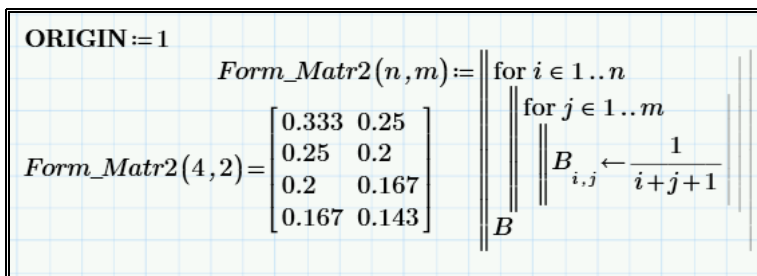


Рис. 8.24. Реализация двойного цикла примера 8.6.8

Пример 8.6.9. 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. 8.25. Обратите внимание, что в качестве начального значения переменной min задано заведомо большое значение модуля разности

между элементами матрицы и заданным числом c . Напомним, что системная переменная $ORIGIN = 1$. Заметим, что в последней строке тела П-Ф стоит вектор, состоящий из искомым числовых характеристик.

Дополнительные операторы, используемые при программировании циклов. Рассмотрим несколько операторов, которые могут быть полезны при программировании циклов.

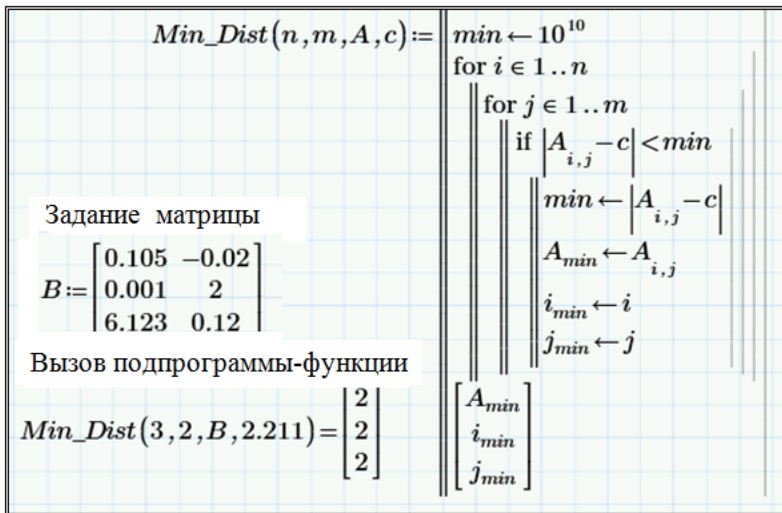


Рис. 8.25. Реализация двойного цикла примера 8.6.9

Оператор *continue*. Обычно используется для продолжения выполнения цикла путем возврата в начало тела цикла с соответствующим изменением параметра цикла оператора *for*. В сочетании с условным оператором *if* позволяет пропустить выполнение оставшегося тела цикла и перейти к следующему повторению тела цикла, начиная с его первого оператора. Следующий пример поясняет работу этого оператора.

Пример 8.6.10. Составить описание П-Ф, формирующей новый вектор из положительных проекций исходного вектора.

Решение. Описание П-Ф приведено на рис. 8.26. 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 показывает правильность работы запрограммированного алгоритма.

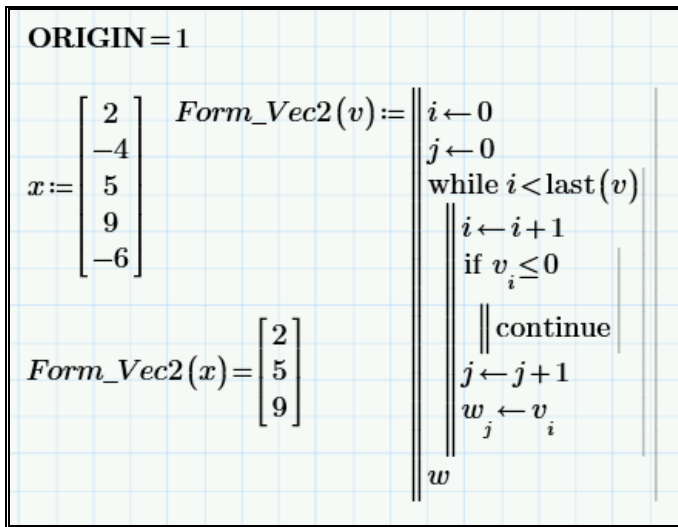


Рис. 8.26. Реализация алгоритма примера 8.6.10

Пример 8.6.11. Составить описание П-Ф, вычисляющей сумму нечетных чисел, находящихся в интервале от 0 до n .

Решение. Запрограммируем алгоритм вычисления суммы с использованием оператора *continue*. Описание П-Ф и обращение к ней приведено на рис. 8.27. Если значение k четное, то это значение при суммировании пропускается.

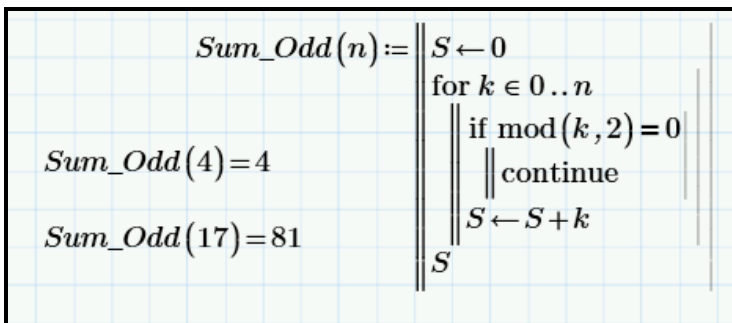


Рис. 8.27. Реализация алгоритма примера 8.6.11

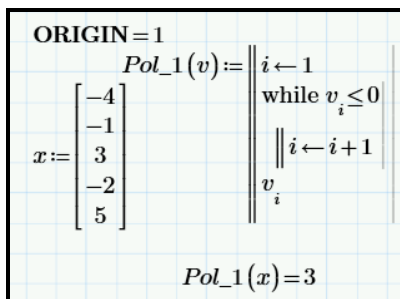
Оператор *return*. Прерывает выполнение П-Ф и возвращает значение операнда, стоящего в поле оператора *return*.

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. 8.28). Вариант *Б* представляется более простым и «элегантным».

Вариант *А*



Вариант *Б*

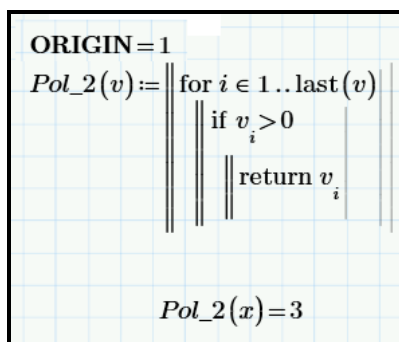


Рис. 8.28. Два варианта реализации алгоритма примера 8.6.12

Оператор *try* (вместе с оператором *on error*). Этот оператор является обработчиком возникающих при выполнении тех или иных ошибок вычисления. Для ввода оператора *try* щелкнуть на кнопке *try* палитры *Программирование*. Появится конструкция, показанная на рис. 8.29 (оператор *on error* добавляется автоматически).

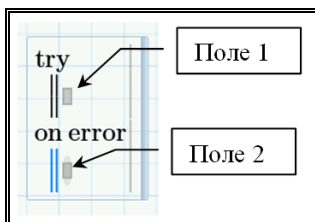


Рис. 8.29. Структура оператора *try*

Оператор выполняется следующим образом. В поле 1 находится одна или несколько конструкций, которые выполняются при отсутствии какой-либо ошибки выполнения (например, деление на ноль). Если при выполнении конструкций поля 1 возникает ошибка, то выполняется конструкция, размещенная в поле 2.

Пример 8.6.13. Используя оператор *try*, необходимо предотвратить появления ошибки «деление на ноль» при вычислении функции $Otn(x,y)$.

Решение. На рис. 8.30 представлены два варианта этой функции. Первый вариант (первая строка документа) приводит к аварийному останову программы при $x = 0$, и выполнение программы прекращается. При использовании функции *try* в ситуации «деление на 0» выводится запрограммированное диагностическое сообщение и продолжается выполнение программы.

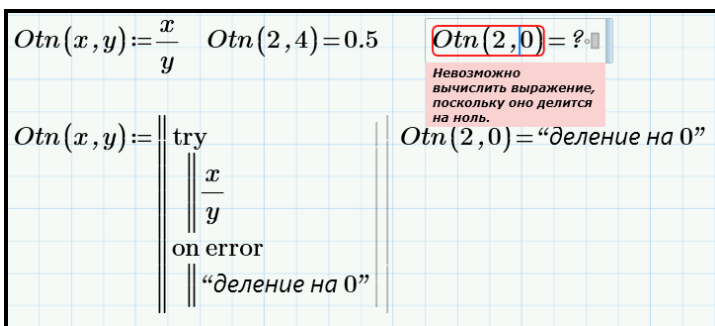


Рис. 8.30. Использование оператора try – on error

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 поле 2 условного оператора *if*, как показано в следующем примере.

Пример 8.6.14. Запрограммировать в П-Ф вывод диагностического сообщения при попытке спроецировать вектор v на нулевой вектор w . Описание П-Ф и ее вызовы приведены на рис. 8.31.

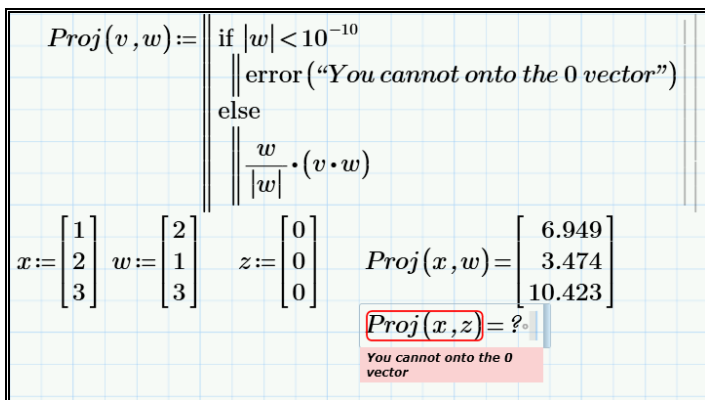


Рис. 8.31. Использование в П-Ф функции error

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. 8.31).

Вопросы и задания для самоконтроля по теме 8

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. 8.32? Объясните, почему вывелось такое значение.

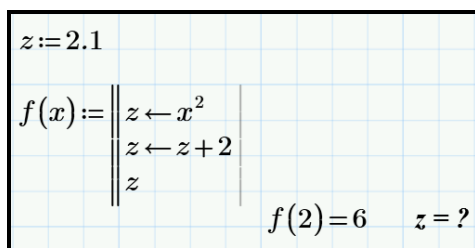


Рис. 8.32. Фрагмент документа к вопросу 8

9. Запрограммируйте П-Ф, вычисляющую значение функции $z(x)$, определяемую выражением $z(x) = a \cdot y^{5+x} + b \cdot \cos(|y|)$, где $y = b \cdot \sin^2(x) + b^2 \cdot \sin^4(x^2)$. Протестируйте составленную П-Ф.

10. Составьте 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. Какие операторы используются 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**»?

14. Какая конструкция используется для реализации в MathCAD логической структуры «**ЕСЛИ-ТО-ИНАЧЕ**»?

15. Как работает и когда используется оператор **else if**?

16. Составьте П-Ф для вычисления переменной y как минимальное значение из трех различных чисел, т.е. $y = \min(a, b, c)$. Протестируйте составленную П-Ф.

17. Составьте П-Ф, вычисляющую значение y в соответствии с формулой

$$y = \begin{cases} x, & \text{если } n = 1, \\ 2\sqrt{|x|}, & \text{если } n = 2 \text{ или } n = 3, \\ e^x, & \text{если } n = 4. \end{cases}$$

18. Составьте 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».

19. Какая конструкция используется для программирования цикла типа арифметической прогрессии в П-Ф?

20. Какое значение будет выведено в конце фрагмента (оператор вывода со знаком ?), приведенного на рис. 8.33?

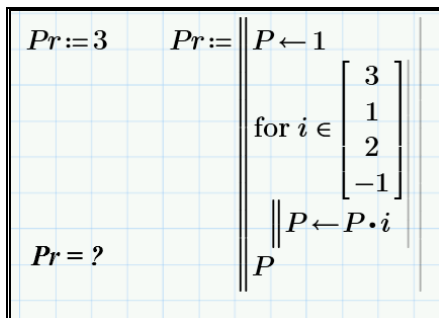


Рис. 8.33. Фрагмент документа к заданию 20

21. Дан вектор x , состоящий из n проекций. Составьте П-Ф, вычисляющую сумму положительных проекций, имеющих четные номера. *Рекомендация:* для проверки четности используйте функцию MathCAD $\text{mod}(x, d)$, которая определяет остаток от деления величины x на величину d . Величины x , d могут быть как целыми, так и вещественными. Примеры использования функции mod приведены на рис. 8.34.

$\text{mod}(5, 2) = 1$	$\text{mod}(8, 2) = 0$	$\text{mod}(4.5, 2) = 0.5$
$\text{mod}(4, 2.5) = 1.5$	$\text{mod}(4.5, 2.5) = 2$	

Рис. 8.34. Примеры использования функции mod

22. 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 $j2$.

Внимание! Значение системной переменной **ORIGIN** = 0.

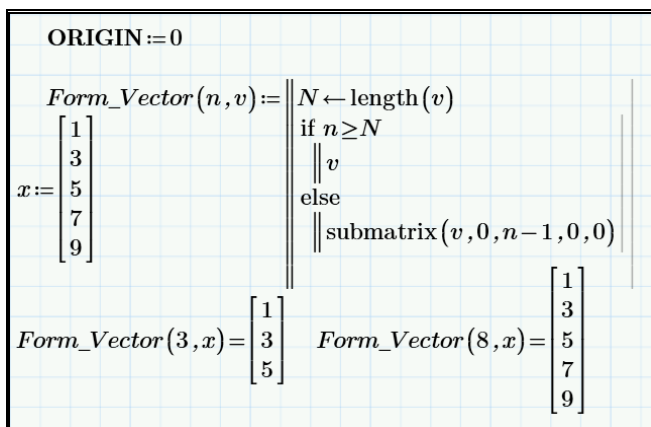


Рис. 9.1. Реализация алгоритма примера 9.1.1

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. 9.1.2. 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 .

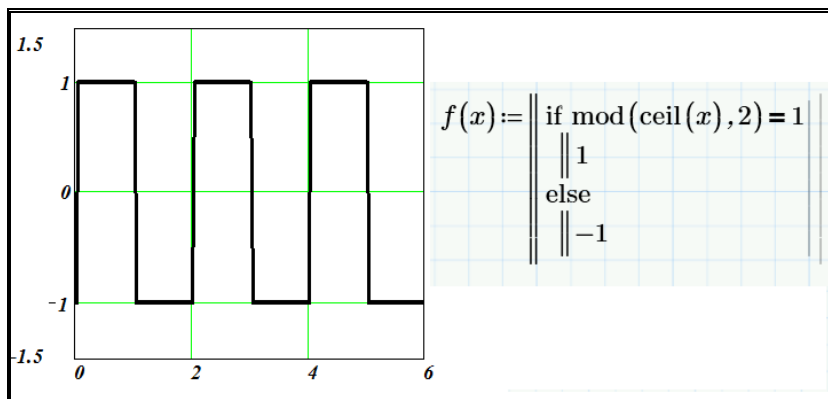


Рис. 9.2. Реализация алгоритма примера 9.1.2

Задание 9.1.1. Составить описание П-Ф, вычисляющей прямоугольный сигнал амплитудой A на интервале $[a, b]$. Вне этого интервала значение равно B . Построить график функции при конкретных значениях параметров.

Задание 9.1.2. Составить описание П-Ф, вычисляющей значения прямоугольных сигналов такой же формы, как на рис. 9.2, но с периодом: а) равным 1; б) равным 8. Построить графики этих сигналов (см. пример 9.1.2).

Пример 9.1.3. Составить фрагмент документа MathCAD для вычисления значения $z = x_{\max} + y_{\text{пол}}$, 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. 9.3. В этом фрагменте вызывается П-Ф вычисления корней квадратного уравнения, описание которой приведено на рис. 8.1 (см. пример 8.1.1).

$$\begin{aligned} X &:= \text{root_poly2}(1, 5.45, -8.12) = \begin{bmatrix} 1.218 \\ -6.668 \end{bmatrix} \\ Y &:= \text{root_poly2}(1, 25.3, -9.5) = \begin{bmatrix} 0.37 \\ -25.67 \end{bmatrix} \\ x_{max} &:= \text{if}(|X_0| > |X_1|, X_0, X_1) = -6.668 \\ y_{pol} &:= \text{if}(Y_0 > 0, Y_0, Y_1) = 0.37 \quad z := x_{max} + y_{pol} = -6.298 \end{aligned}$$

Рис. 9.3. Реализация алгоритма примера 9.1.3

Задание 9.1.3. Составить описание П-Ф, реализующей вычислительный алгоритм, приведенный на рис. 9.4 а. Составить фрагмент для проверки правильности работы П-Ф.

Задание 9.1.4. Составить описание П-Ф, вычисляющей площадь S геометрической фигуры по ее номеру $n = 1, 2, 3$. Расчетные соотношения приведены на рис. 9.4 б, 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 (к заданиям 9.1.3 и 9.1.4)

9.2. Программирование циклов типа арифметической прогрессии

Циклические вычислительные процессы при обработке массивов в большинстве случаев используют следующие базовые алгоритмы:

- *формирование массива по заданному правилу;*
- *подсчет числа элементов массива, удовлетворяющих заданным условиям (организация счетчика);*
- *накопление в цикле суммы или произведения;*
- *поиск минимального или максимального элемента массива.*

Эти базовые алгоритмы используются как при обработке одномерных массивов, так и при обработке двумерных массивов. Однако при обработке матриц базовые алгоритмы можно разделить на два класса:

- вычисление характеристик для всей матрицы (например, подсчитать сумму положительных элементов матрицы);
- вычисление характеристик для каждой строки или каждого столбца матрицы (например, подсчитать сумму положительных элементов для каждой строки матрицы).

Ниже будет рассмотрено решение перечисленных типовых задач как для одномерного, так и для двумерного массивов.

Тип 1 (формирование элементов массива, удовлетворяющих заданным условиям)

Пример 9.2.1. Составить П-Ф, формирующую матрицу A по следующему правилу:

$$a_{i,j} = \begin{cases} \sin(i - j), & \text{если } i = j, \\ \sin i - \cos j, & \text{если } i > j, \\ \sin j - \cos i, & \text{если } i < j. \end{cases}$$

Выполнить тестирование составленной П-Ф.

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.

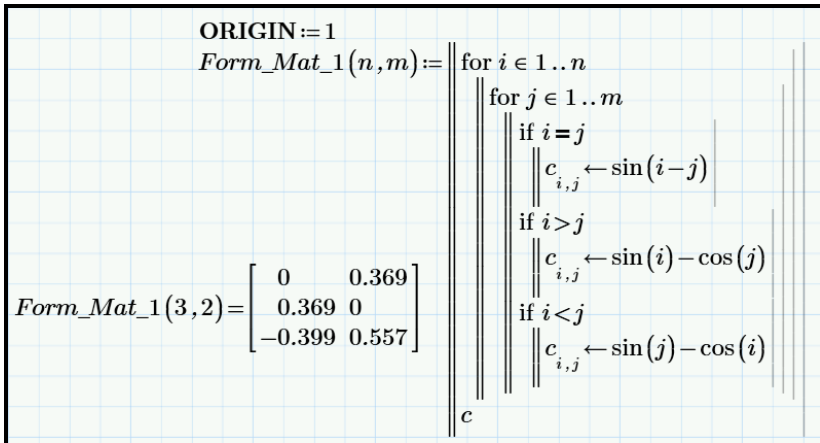


Рис. 9.5. Реализация алгоритма примера 9.2.1

Задание 9.2.1. 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 радиусом r с центром в начале координат, и равен 0 в противном случае.

Задание 9.2.2. 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 9.2.3. 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 одномерный массив y , состоящий из n элементов. Составить П-Ф, определяющую:

- число элементов $k1$, удовлетворяющих условию $a \leq |y_i| \leq b$;
- число положительных элементов $k2$.

Решение. Описание П-Ф приведено на рис. 9.6. До начала цикла задаются нулевые начальные значения переменных $k1, k2$.

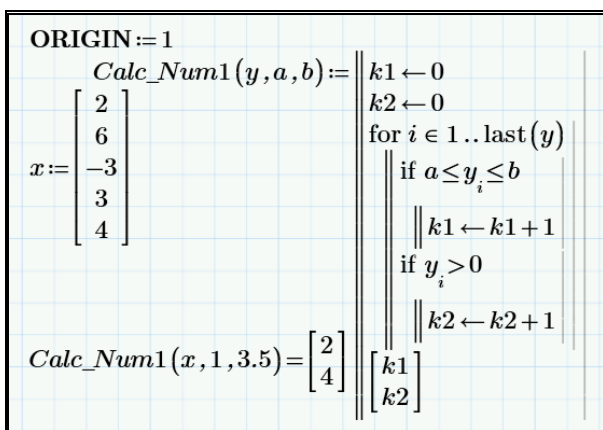


Рис. 9.6. Реализация алгоритма примера 9.2.2

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.

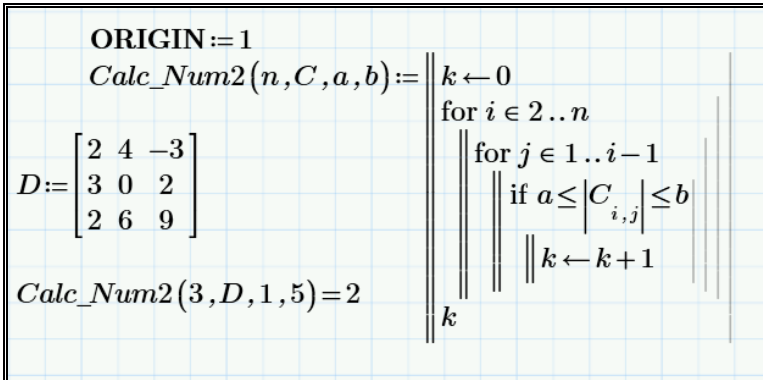


Рис. 9.7. Реализация алгоритма примера 9.2.3

Пример 9.2.4. Дана последовательность

$$a_i = \cos\left(\frac{i}{10} + i\right), \text{ где } i = 1, 2, \dots, n; n = 10.$$

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 не обязательно 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.

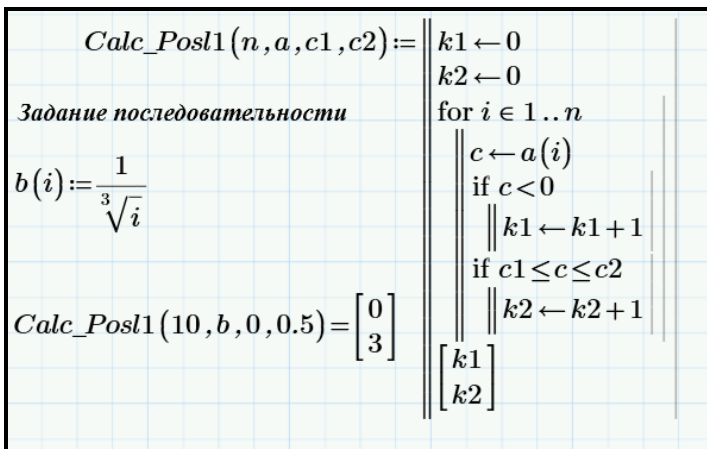


Рис. 9.8. Реализация алгоритма примера 9.2.4

Задание 9.2.4. Даны две матрицы A, B размерности $n \times m$. Составить П-Ф, подсчитывающую число случаев $A_{i,j} \geq B_{i,j}$.

Замечание 9.2.1. В предыдущих примерах искомые характеристики вычислялись для всего заданного массива, т.е. используемые алгоритмы относились к первому классу алгоритмов обработки массивов. Поэтому начальные значения искомых характеристик задавались до начала цикла (в случае матрицы – до начала внешнего цикла). В задачах второго класса 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 искомая характеристика (количество положительных элементов) вычисляется для каждой строки матрицы. Поэтому нулевое на-

начальное значение для переменной k задается в теле внешнего цикла до начала внутреннего цикла. Размеры матрицы не передаются через формальные параметры, а вычисляются внутри тела П-Ф с использованием функций $rows$ (число строк) $cols$ (число столбцов).

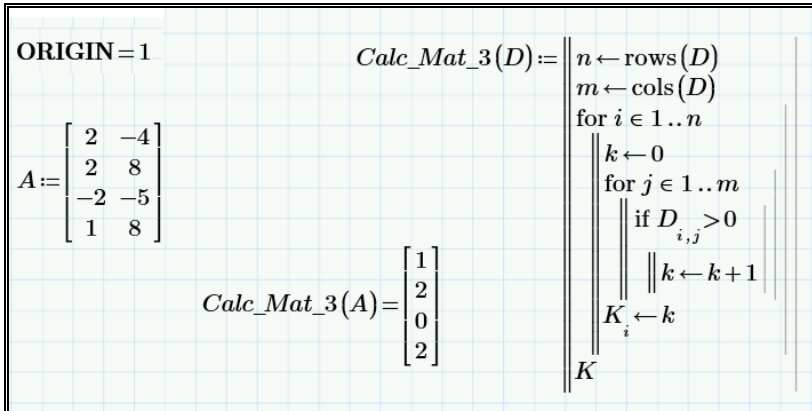


Рис. 9.9. Реализация алгоритма примера 9.2.5

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 также внимание на задание

переменной pr начального значения, равного 1. Почему оно не равно нулю, как это сделано для суммы?

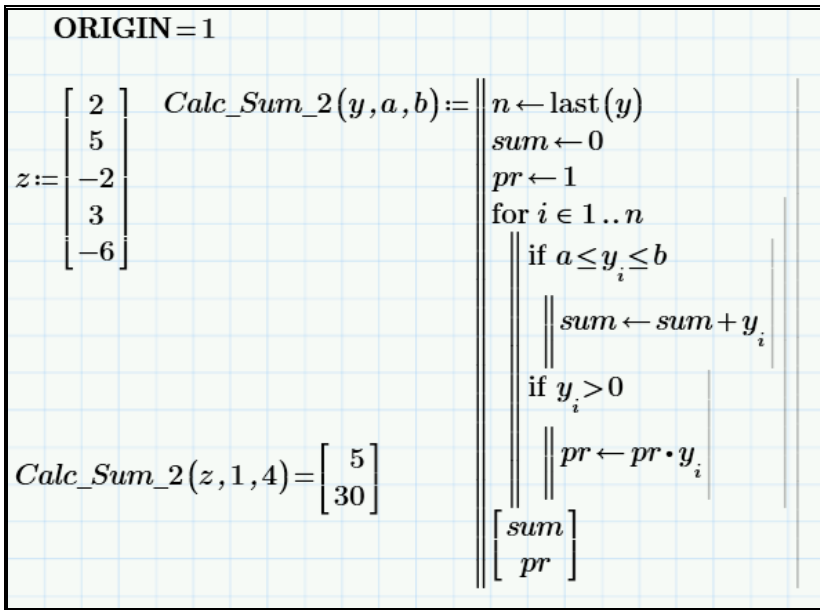


Рис. 9.10. Реализация алгоритма примера 9.2.6

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. Эта задача относится ко второму классу, так как сумма вычисляется для каждой строки матрицы. Поэтому начальное нулевое значение для суммы задается во внешнем цикле до внутреннего, и после окончания внутреннего цикла вычисленное значение суммы записывается в элемент $A_{i,1}$.

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.

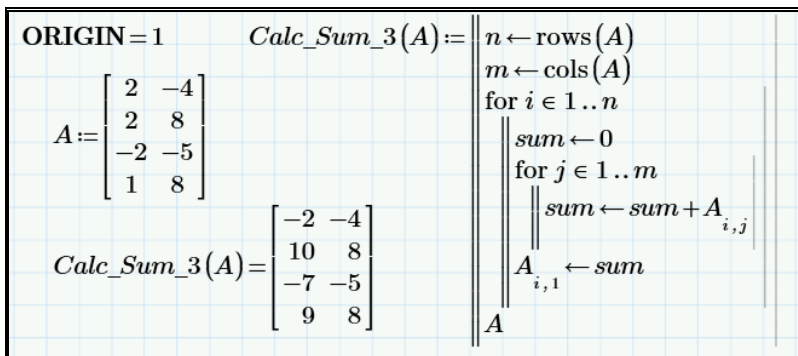


Рис. 9.11. Реализация алгоритма примера 9.2.7

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.

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 массива 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.

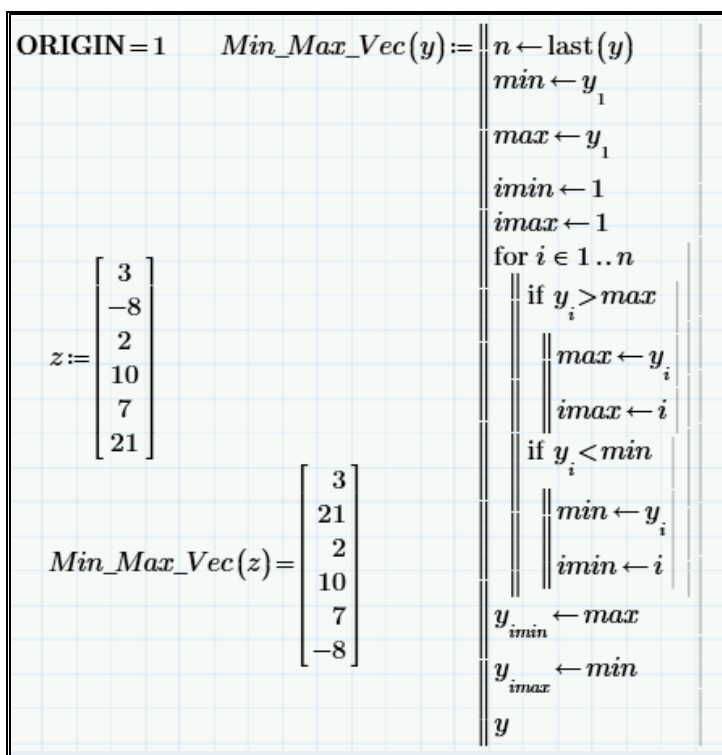


Рис. 9.12. Реализация алгоритма примера 9.2.8

Замечание 9.2.2. 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 второго 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.

The diagram illustrates the implementation of the algorithm for finding the maximum value of a function on a vector. It is divided into three main sections:

- Function Definition:** $\psi(x) := e^{\frac{-(x-3.5)^2}{3}}$
- Vector Definition:** $y := \begin{bmatrix} 3.5 \\ 6.1 \\ -4.23 \\ 0.22 \\ 12 \end{bmatrix}$
- Procedure Definition and Call:**
 - Procedure Definition:**

```

ORIGIN = 1    Max_Fun( $x, f$ ) :=
   $n \leftarrow \text{last}(x)$ 
   $fmax \leftarrow -10^{20}$ 
  for  $i \in 1..n$ 
    if  $f(x_i) > fmax$ 
       $fmax \leftarrow f(x_i)$ 
       $imax \leftarrow i$ 
       $xmax \leftarrow x_i$ 
   $\begin{bmatrix} fmax \\ xmax \\ imax \end{bmatrix}$ 

```
 - Procedure Call:** $\text{Max_Fun}(y, \psi) = \begin{bmatrix} 1 \\ 3.5 \\ 1 \end{bmatrix}$

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 которого равен номеру столбца, в котором на i -й строке матрицы стоит минимальный элемент.

Решение. Описание П-Ф приведено на рис. 9.14. Сформулированная задача относится ко второму классу, поэтому начальное значение минимального элемента задается для каждого столбца матрицы.

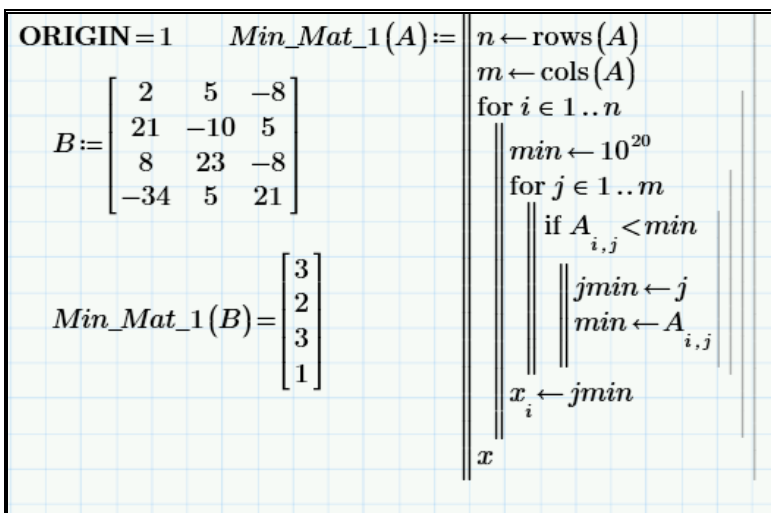


Рис. 9.14. Реализация алгоритма примера 9.2.10

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.

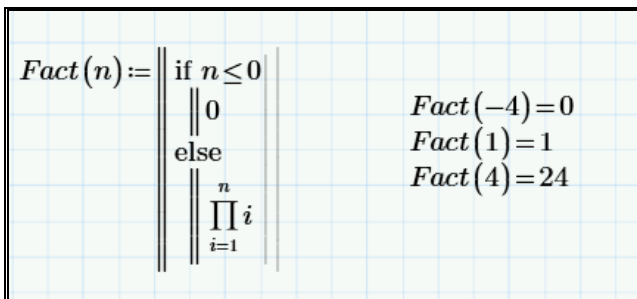
Задание 9.2.15. Имеется экзаменационная ведомость студенческой группы из $n = 20$ человек по $m = 5$ дисциплинам. Оценки из этой ведомости занесены в матрицу размерности $n \times m$. Составить П-Ф, вычисляющую номер дисциплины (т.е. номер столбца), имеющей максимальный средний балл.

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

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$$

Решение. Описание П-Ф с использованием оператора произведения приведено на рис. 9.15.



$$Fact(n) := \begin{cases} 0 & \text{if } n \leq 0 \\ \prod_{i=1}^n i & \text{else} \end{cases}$$

$$\begin{aligned} Fact(-4) &= 0 \\ Fact(1) &= 1 \\ Fact(4) &= 24 \end{aligned}$$

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. 8.6).

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 заикливание – 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.

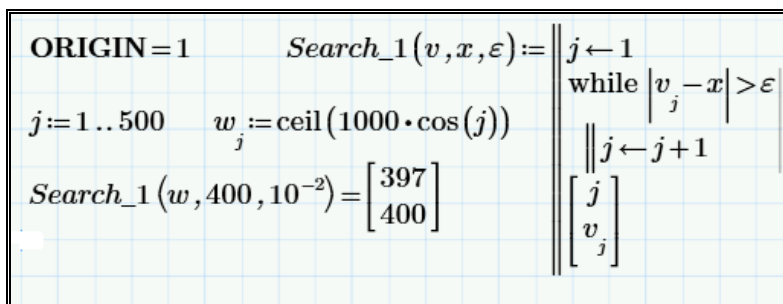


Рис. 9.16. Реализация алгоритма примера 9.3.1

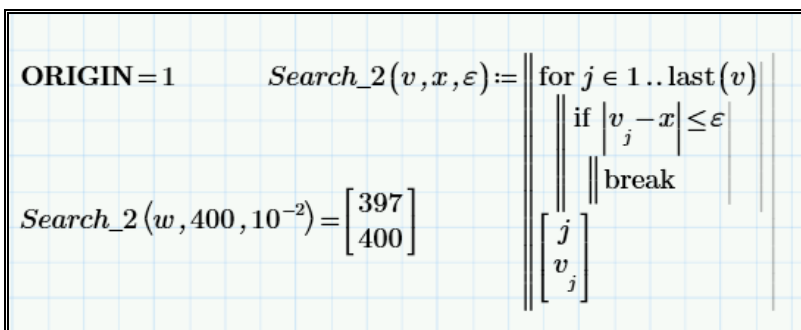


Рис. 9.17. Реализация алгоритма примера 9.3.1

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 приближений.

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), 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

$$|x_n - x^*| \leq \varepsilon, \quad (9.3.7)$$

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. $\varphi(x) = -e^{\frac{x}{2}}$. Можно показать (сделайте это самостоятельно), что на интервале $[-4, 0]$ условие (9.3.4) выполняется и в качестве величины q можно принять значение 0.5. Также выполняется условие (9.3.5), т.е. 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):

$$\eta = \frac{10^{-6} \cdot (1-0.5)}{0.5} = 10^{-6}.$$

Описание П-Ф, реализующей метод последовательных приближений, приведено на рис. 9.17. Здесь же показаны вычисленное приближенное значение корня и его проверка. Имя функции $\varphi(x)$ является формальным параметром. Для предотвращения закливания итерационной процедуры (9.3.3) используется оператор цикла *for*. Заметим, что переменные x_0, x_n , используемые в теле П-Ф, являются простыми переменными.

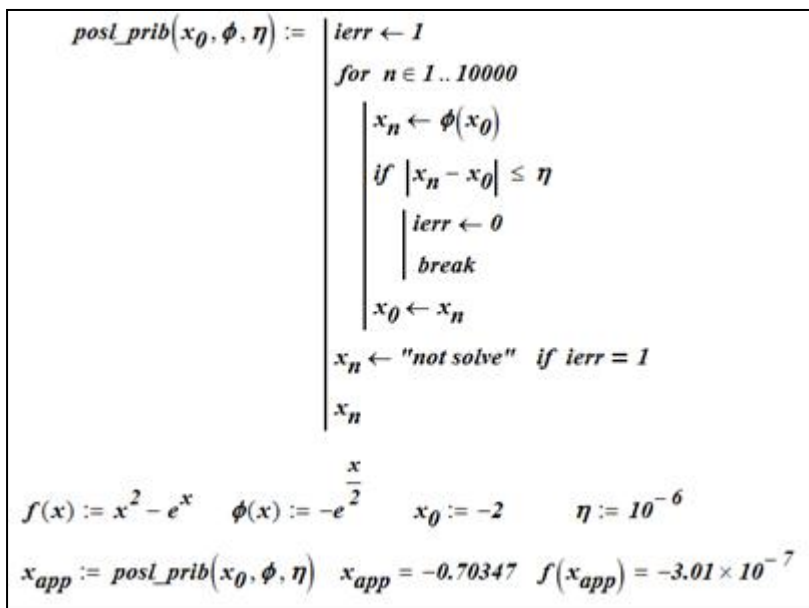


Рис. 9.17. Программная реализация метода последовательных приближений

Вопросы и задания для самоконтроля по теме 9

1. Когда используется оператор цикла *while*? Нарисуйте 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? Приведите пример такой задачи.

5. Матрица A размером $n \times k$ содержит экзаменационные оценки n студентов по k учебным дисциплинам. Запрограммируйте П-Ф, вычисляющую номер студента, имеющего максимальный средний балл, и величину максимального среднего балла. Выполните тестирование составленной П-Ф.

6. В каких задачах обработки матриц начальные значения искомых характеристик матрицы задаются в теле внешнего цикла до начала внутреннего цикла? Приведите пример такой задачи.

7. Дана матрица A размерности $n \times m$. Составьте П-Ф, которая вместо последнего элемента i -й строки записывает сумму всех элементов этой строки матрицы.

8. Задана матрица C размерностью 3×4 . Составьте П-Ф, вычисляющую номер строки, сумма элементов которой максимальна. Выполните тестирование П-Ф.

9. В матрице S размерностью 20×4 приведены оценки 20 студентов за 4 экзамена последней сессии. Составьте П-Ф, формирующий вектор, содержащий средние баллы каждого студента. Выполните тестирование П-Ф.

10. Составьте П-Ф, реализующую итерационный цикл суммирования 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 $|a_{n+1}| \leq \varepsilon$. Определите число итераций, необходимых для вычисления суммы с заданной точностью. Осуществите тестирование составленной П-Ф.

Рекомендация: для прекращения итерационного цикла используйте команду *break*.

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 [5, 8–10];
- решение задач вычислительной математики в пакете MathCAD [3, 11, 12];
- решение задач теории вероятностей и математической статистики в пакете MathCAD [5, 13];
- решение задач регрессионного анализа экспериментальных данных и моделирования в пакете MathCAD [4, 14, 15];
- решение других задач в пакете MathCAD [3, 8–10].

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 (Сибстрин). – Новосибирск : НГАСУ (Сибстрин), 2006. – 116 с. (электронная версия – <ftp://ftp.sibstrin.ru/prikl/archives/mathcad2006.zip>).
3. Основы вычислений и программирования работы 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.
4. 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, 2002. – 136 с. (электронная версия – <ftp://ftp.sibstrin.ru/prikl/archives/math03.zip>).
5. Кирьянов Д. В. Самоучитель Mathcad 12 / Д. В. Кирьянов. – СПб. : БХВ-Петербург, 2004. – 576 с.
6. Очков В. Ф. Советы пользователям Mathcad / В. Ф. Очков. – М. : Изд-во МЭИ, 2006. – 196 с.
7. 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>).
8. Гурский Д. Вычисления в Mathcad 12 / Д. Гурский. – СПб. : Питер, 2006. – 544 с.
9. Дьяконов В. П. MathCAD 8 Professional в математике, физике и Internet / В. П. Дьяконов, И. В. Абраменкова. – М. : Нолидж, 1999. – 512 с.

10. Черняк В. А. Математика для экономистов на базе Mathcad / В. А. Черняк, В. А. Новиков. – СПб. : БХВ-Петербург, 2003. – 496 с.
11. Поршнев С. В. Численные методы на базе Mathcad / С. В. Поршнев. – СПб. : БХВ-Петербург, 2005. – 464 с.
12. 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 (Сибстрин). – Новосибирск : НГАСУ (Сибстрин), 2005. – 96 с. (электронная версия – <ftp://ftp.sibstrin.ru/prikl/archives/mathcad2005.zip>).
13. Ивановский Р. И. Теория вероятностей и математическая статистика с примерами и задачами в среде Mathcad / Р. И. Ивановский. – СПб. : БХВ-Петербург, 2008. – 528 с.
14. Воскобойников Ю. Е. Регрессионный анализ данных в пакете Mathcad / Ю. Е. Воскобойников. – СПб. : Лань, 2011. – 286 с.
15. Воскобойников Ю. Е. Построение регрессионных моделей в пакете Mathcad : учеб. пособие / Ю. Е. Воскобойников ; Новосиб. гос. архитектур.-строит. ун-т (Сибстрин). – Новосибирск : НГАСУ (Сибстрин), 2009. – 220 с.