ВВЕДЕНИЕ

Mathcad является математическим редактором, позволяющим проводить разнообразные научные и инженерные расчеты, начиная от элементарной арифметики и заканчивая сложными реализациями численных методов.

Чрезвычайная простота интерфейса Mathcad сделала его одним из самых популярных и, безусловно, самым распространенным в студенческой среде математическим пакетом. Он предоставляет пользователю обширный набор инструментов для реализации графических, аналитических и численных методов решения математических задач на компьютере. При выполнении рутинных или несущественных (в контексте изучаемого раздела) операции, пакет позволяет студенту, не владеющему в полной мере техникой математических преобразований, самостоятельно произвести громоздкие вычисления, решить содержательные задачи, приобрести устойчивые навыки решения прикладных задач [3].

Mathcad построен в соответствии с принципом WYSIWYG («What You See Is What You Get» – «что Вы видите, то и получите»). Поэтому он очень прост в использовании, в частности, из-за отсутствия необходимости сначала написать программу, реализующую те или иные математические расчеты, а потом запускать ее на исполнение.

В состав Mathcad входят несколько интегрированных между собой компонентов:

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

Сочетание этих компонентов создает удобную вычислительную среду для разнообразных математических расчетов и, одновременно, документирования результатов работы. Документы могут быть распечатаны непосредственно в Mathcad в том виде, который пользователь видит на экране компьютера, или сохранены в формате RTF для последующего редактирования в более мощных текстовых редакторах (например, Microsoft Word). Возможно полноценное сохранение документов Mathcad в формате Web-страниц (генерация вспомогательных графических файлов происходит автоматически).

В учебном пособии рассматриваются приемы работы с пакетом Mathcad 14.

1. ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ

В Mathcad интерфейс пользователя интуитивен и сходен с другими приложениями Windows (рис. 1). Отметим его основные элементы:

- верхнее меню, или строка меню (menu bar);
- панели инструментов (toolbars) **Standard** (Стандартная) и **Formatting** (Форматирование);
- панель инструментов **Math** (Математика) и доступные через нее дополнительные математические палитры символов;
- панель инструментов **Controls** (Элементы управления);
- рабочая область (**Worksheet**);
- строка состояния (**status line**, или **status bar**);
- всплывающие, или контекстные, меню (**pop-up menus**, или **context menus**);

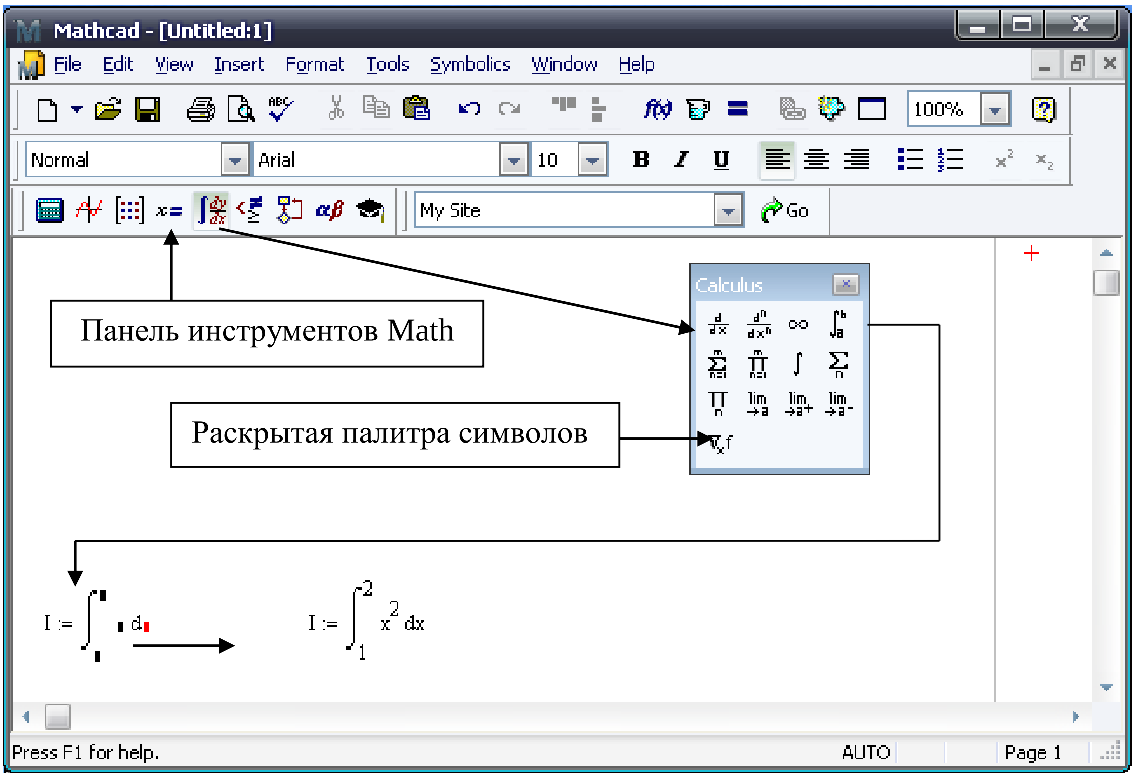


Рис. 1. Окно Mathcad

1.1. Меню

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

- **File** (Файл) – команды, связанные с созданием, открытием, сохранением пересылкой по электронной почте и распечаткой на принтере файлов с документами;
- **Edit** (Правка) – команды, относящиеся к правке текста (копирование, вставка, удаление фрагментов и т. п.);
- **View** (Вид) – команды, управляющие внешним видом документа в окне редактора Mathcad;
- **Insert** (Вставка) – команды вставки различных объектов в документы;
- **Format** (Формат) – команды форматирования текста, формул и графиков;
- **Tools** (Инструменты) – команды для создания анимации и управления вычислительным процессом;
- **Symbolics** (Символика) – команды символьных вычислений;
- **Window** (Окно) – команды управления расположением окон с различными документами на экране;
- **Help** (Справка) – команды вызова контекстно-зависимой справочной информации.

1.2. Панели инструментов

Панели инструментов служат для быстрого (в один щелчок мыши) выполнения наиболее часто применяемых команд. Все действия, которые можно выполнить с помощью панелей инструментов, доступны и через верхнее меню.

Кнопки в панелях сгруппированы по сходному действию команд:

- **Standard** (Стандартная) – служит для выполнения большинства операций, таких как действия с файлами, редакторская правка, вставка объектов и доступ к справочным системам;
- **Formatting** (Форматирование) – для форматирования (изменения типа и размера шрифта, выравнивания и т. п.) текста и формул;
- **Math** (Математика) – для вставки математических символов и операторов в документы.

• **Controls** (Элементы управления) – для создания в документе Mathcad некоторых элементов управления Windows.

Группы кнопок на панелях инструментов разграничены по смыслу вертикальными линиями – разделителями. При наведении указателя мыши на любую из кнопок рядом с кнопкой появляется всплывающая подсказка – короткий текст, поясняющий назначение кнопки.

Вызвать любую панель на экран или скрыть ее можно с помощью меню **View** (Вид)/**Toolbars** (Панели инструментов), выбирая в открывающемся подменю имя нужной панели.

Панель **Math** (Математика) предназначена для вызова на экран еще девяти панелей (*палитр символов*), с помощью которых, собственно, и происходит вставка математических операций в документы. Чтобы показать какую-либо из них, нужно нажать соответствующую кнопку на панели **Math** (рис. 1).

Перечислим назначение математических панелей (рис. 2):

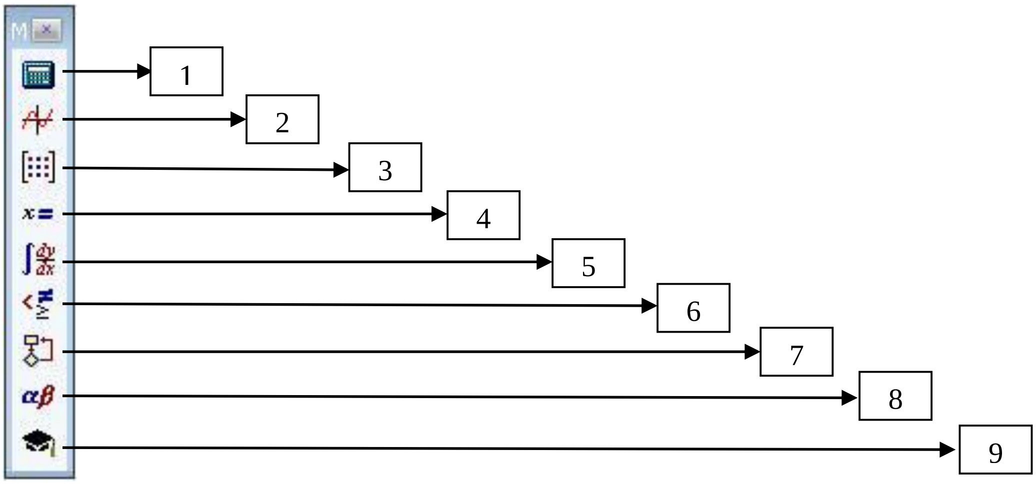


Рис. 2. Математические панели

1. **Calculator** (Калькулятор) – служит для вставки основных математических операций;
2. **Graph** (График) – для вставки графиков;
3. **Matrix** (Матрица) – для вставки матриц и матричных операторов;
4. **Evaluation** (Выражения) – для вставки операторов управления вычислениями;
5. **Calculus** (Вычисления) – для вставки операторов интегрирования, дифференцирования, суммирования;

6. **Boolean** (Булевы операторы) – для вставки логических (булевых) операторов;

7. **Programming** (Программирование) – для программирования средствами Mathcad;

8. **Greek** (Греческие символы) – для вставки греческих символов;

9. **Symbolic** (Символика) – для вставки символьных операторов.

2. РАБОТА С ДОКУМЕНТОМ MATHCAD

В Mathcad все расчеты организуются на рабочих областях, или «листах» (**Worksheets**), изначально пустых, на которые можно добавлять формулы и текст. Здесь и далее будем называть рабочий лист ***документом*** Mathcad. Документ представляет собой совокупность нескольких ***областей***, каждая из которых может быть математическим выражением, фрагментом текста или графическим изображением. Вокруг каждой области Mathcad создает ***невидимый прямоугольник***, который может быть отображен на экране по команде меню: **View/Regions** (Вид/Области).

Каждый документ представляет собой независимую серию математических расчетов и сохраняется в отдельном файле. Документ является одновременно и листингом Mathcad-программы, и результатом исполнения этой программы, получающимся при ее выполнении, и отчетом, пригодным для распечатки на принтере или публикации в Web.

2.1. Создание пустого документа

Если Mathcad запускается из главного меню Windows или с использованием пиктограммы на рабочем столе, то окно Mathcad появляется с открытым в нем новым пустым безымянным документом, условно называемым **Untitled:1**. Для того чтобы создать новый пустой документ, уже работая в Mathcad, следует выполнить одно из трех эквивалентных действий:

- нажать одновременно клавиши **<Ctrl>+<N>**;
- нажать кнопку **New** (Создать) на панели инструментов;
- выполнить команду меню **File/New** (Файл/Создать).

В последнем случае на экране появится окно диалога «**New**», содержащее список шаблонов, на основе которых может быть создан новый документ. Из представленного списка необходимо выбрать шаблон «**Normal**». В первых двух случаях шаблон «**Normal**» применяется автоматически.

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

2.2. Сохранение документа

Для сохранения документа в формате Mathcad необходимо выбрать команду меню **File/Save** (Файл/Сохранить), или нажать кнопку **Save** (Сохранить) на стандартной панели инструментов. Если созданный документ сохраняется впервые, на экран будет выведено диалоговое окно **Save as** (Сохранить как), в котором необходимо определить его имя и папку для хранения. В процессе работы документ можно сохранить и под другим именем и (или) в другой папке. Для этого необходимо использовать команду меню **File/Save As** (Файл/Сохранить как). В этом случае файл со старым названием не изменяется.

В окне **File/Save As** (Файл/Сохранить как) представлен список форматов (Тип файла) в которых может быть сохранен документ Mathcad. По умолчанию документ сохраняется в формате Mathcad XML Document **(*.xmcd)**. Если в дальнейшем документ подлежит редактированию в текстовых редакторах с целью создания отчетов, то его можно сохранить в формате Rich Text Format **(*.rtf)**. В частности, сохранив документ в RTF-файле, можно загрузить его в Microsoft Word или другой текстовый процессор, поддерживающий этот формат. Для просмотра документа в любом Интернет-браузере его необходимо сохранить в формате HTML File **(*.htm)** – формат Web-страницы.

2.3. Открытие существующего документа

Чтобы открыть существующий документ для редактирования, необходимо выполнить команду меню **File/Open** (Файл/Открыть) или нажать кнопку **Open** на стандартной панели инструментов. В диалоговом окне **Open** (Открыть) нужно выбрать необходимый файл и нажать кнопку ОК. Кроме того, открыть файл можно и в обозревателе Windows, щелкнув дважды на его имени с расширением .**xmcd**. При этом будет запущен пакет Mathcad и в нем автоматически открыт файл с именем, по которому выполнен щелчок мышью.

2.4. Построение и редактирование математических выражений

Для создания математического выражения в нужном месте документа необходимо щелчком левой клавиши мыши (ЛКМ) установить в это место курсор и ввести необходимое выражение. При этом часть символов (буквы, цифры, имена функций и др.) вводится непосредственно с клавиатуры. Для вставки в выражение сложных математических операторов (степень, суммирование, интегрирование и др.) используют комбинации соответствующих клавиш или выбирают нужный символ из соответствующей палитры символов. Для вставки в местоположение курсора некоторого символа из палитры необходимо щелчком ЛКМ раскрыть соответствующую палитру и далее щелчком ЛКМ выбрать нужный символ.

2.5. Определение переменных и функций

Чтобы определить любую переменную или функцию в документе Mathcad, необходимо:

- напечатать имя переменной или функции, которую надо определить;
- напечатать символ «Присвоить» **(«:=»)** путем ввода символа двоеточие **(«:»)**, либо выбрав его из палитры символов;
- напечатать значение переменной или выражение для определенной функции, при этом дробная часть числа отделяется от целой символом **«.»** (точка).

Например:

$$x := 3 \qquad y := x + 7 \qquad f(x,y) := x^2 + y^2$$

Если в имени переменной необходимо использовать подстрочный индекс, то для этого необходимо:

- напечатать часть имени без индекса;
- ввести символ **«.»** (точка);
- напечатать подстрочный индекс.

Например:

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

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

Замечание. Нельзя путать подстрочные нижние индексы с индексами массива (см. ниже). Буквенный или цифровой нижний индекс, созданный вводом точки, является частью имени переменной. Нижний индекс массива осуществляет ссылку на элемент массива и вводится клавишей левой скобки «[» (см. ниже).

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

Mathcad ***читает рабочий документ сверху вниз и слева направо***. Определив переменную, например, x, ее можно использовать в вычислениях везде ниже и правее равенства, в котором она определена.

2.6. Вычисление математических выражений

Для вычисления значения некоторого выражения необходимо:
- присвоить значения всем переменным, входящим в выражение;
- напечатать необходимое выражение;
- ввести символ равно «=».

После этого на экране появится результат вычислений, если установлен автоматический режим вычислений (см. ниже).

Например:

$$f(x,y) := x + y$$

$$x := 7 \qquad y := 12 \qquad z := 3$$

$$p := \sqrt{x^2 + y^2} + \frac{z}{2} \qquad p = 15.392$$

$$f(2,3) = 5 \qquad f(p,2\cdot p) = 46.177$$

2.7. Организация циклических вычислений

Для того чтобы вычислить значения некоторого выражения или функции для дискретного набора значений аргумента (циклические вычисления) необходимо определить диапазон его изменения в виде:

$$\boldsymbol{x := x1,\, x2\, ..\, xN}$$

Здесь ***x*** – имя переменной, которая изменяется дискретно, ***x*1** – начальное значение переменной ***x***, ***x*2** – следующее значение переменной ***x***, ***x*N** – конечное значение переменной ***x***, **«..»** – символ диапазона, вводится с клавиатуры набором символа **«;»** либо выбором кнопки **m..n** на панели инструментов **«Matrix»**.

Шаг изменения переменной ***x*** Mathcad определяет как:

$$\Delta x = x2 - x1$$

Если шаг изменения переменной ***x*** должен быть равным **1**, то можно написать:

$$x := x1\,..\,xN$$

Например:

$$f(x) := x^2$$

$$t := 1, 1.5\,..\,3 \qquad z := 0\,..\,3$$

$f(t) =$

1
2.25
4
6.25
9

$f(z) =$

0
1
4
9

2.8. Режимы вычислений

В пакете Mathcad имеется два режима вычислений:

- автоматический режим **(automatic mode)**;
- ручной режим **(manual mode)**.

Автоматический режим вычислений устанавливается по умолчанию для каждого нового документа. В автоматическом режиме вычисления выражений и функций выполняются сразу по мере ввода символов «=».

Выключение или включение режима автоматических вычислений выполняется командой меню **Tools/Calculate/Automatic Calculation** (Инструменты/Вычислить/Автоматические вычисления).

Ручной режим вычислений позволяет выполнять вычисления, когда режим автоматических вычислений отключен. Для этого необходимо выбрать команду меню **Tools/Calculate/Calculate Now** (Инструменты/Пересчитать/Пересчитать сейчас) или нажать клавишу **F9**. При этом пересчет документа начинается от начала и заканчивается его видимой частью.

При выборе команды меню **Tools/Calculate/Calculate Worksheet** (Инструменты/Пересчитать/Пересчитать рабочий лист) пересчитывается весь документ.

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

2.9. Типы данных

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

Перечислим основные типы данных, которые обрабатываются процессорами системы Mathcad:

- числа (в том числе, действительные, комплексные, а также встроенные константы) – Mathcad хранит все числа в формате двойной точности с плавающей точкой (не разделяя их на целые, булевы и т. д.);
- строки – любой текст, заключенный в кавычки;
- массивы – упорядоченные последовательности чисел или строк.

2.9.1. Действительные числа

Любое выражение, начинающееся с цифры, Mathcad интерпретирует как число. Несмотря на то, что Mathcad хранит все числа в одинаковом формате, вводить их можно в наиболее подходящем представлении (notation), исходя из контекста документа:

- как целое число;
- как десятичное число (decimal notation) с любым количеством десятичных цифр после точки;
- в представлении с порядком (exponential notation) – в так называемом научном формате или представлении (scientific notation), для чего после ввода числа необходимо напечатать символ умножения и ввести 10 в нужной степени;

• как число в другой системе счисления.

Ниже приведен пример ввода чисел в первых трех форматах:

$a := 100 \qquad b := 4.675 \qquad c := 2.47 \cdot 10^{-3}$

2.9.2. Комплексные числа

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

Чтобы ввести мнимое число, например 7i, необходимо:

• ввести действительный сомножитель (7);

• ввести символ «*» (умножить);

• ввести символ «мнимая единица» («i»).

Для ввода символа «мнимая единица» необходимо последовательно нажать клавиши **<1>** и **<*i*>**. Если просто ввести символ **«*i*»**, то Mathcad интерпретирует его как переменную **i**. Кроме того, мнимая единица имеет вид ***1i***, только когда соответствующая формула выделена. В противном случае мнимая единица отображается просто как ***i***.

Ниже приведен пример ввода комплексных чисел.

$z1 := 7i \qquad z2 := 2 + 3i \qquad z3 := 2 \cdot e^{5i}$

2.9.3. Встроенные константы

Некоторые имена в Mathcad зарезервированы под системные переменные, которые называются встроенными константами (built-in constants). Встроенные константы делятся на два типа:

• математические, хранящие значение некоторых общеупотребительных специальных математических символов;

• системные, определяющие работу большинства численных алгоритмов, реализованных в Mathcad.

Математические константы (math constants):

• символ бесконечности (вводится клавишами <Ctrl>+<Shift>+<z>);

• e – основание натурального логарифма (клавиша <e>);

• число π (вводится клавишами <Ctrl>+<Shift>+<p>);

• i – мнимая единица (вводится клавишами <1>, <i>).

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

Ниже приведен пример вывода значений упомянутых констант.

$$\infty = 1 \times 10^{307} \qquad e = 2.718 \qquad \pi = 3.142$$

2.9.4. Строковые выражения

Значением переменной или функции может быть не только число, но и строка, состоящая из любой последовательности символов, заключенной в кавычки.

Для присвоения переменной строкового значения необходимо:

- ввести имя переменной, например s;
- ввести символ «присвоить» «:=»;
- ввести символ «”» и набрать нужную строку.

Например.

s := "Hellow Word!"   filename := "d:\work\lab1.xmcd"

2.9.5. Создание и редактирование массивов

Массивами (arrays) называют упорядоченные последовательности чисел. Массив можно представить как таблицу, имеющую **N** строк и **M** столбцов. В дальнейшем, следуя математической терминологии, массив размером **N**×**M** (**N** строк, **M** столбцов) будем называть матрицей, а массив размером **N×1**(**N** строк, **1** столбец) вектором. Ниже приведены примеры матрицы и вектора.

$$\begin{pmatrix} 1 & 4 & 6 \\ 8 & 6 & 7 \\ 5 & 4 & 1 \end{pmatrix} \text{ - матрица} \qquad \begin{pmatrix} 7 \\ 4 \\ 9 \end{pmatrix} \text{ - вектор}$$

Для создания матрицы с нужным именем необходимо:

- установить курсор в нужное место документа;

- ввести имя матрицы и символ «присвоить»;
- выполнить команду меню Insert/Matrix (Вставка/Матрица) или нажать комбинацию клавиш <Ctrl>+<M>; либо выбрать кнопку на панели инструментов «Matrix».
- в появившемся окне диалога ввести необходимое число строк **(Rows)**, столбцов **(Columns)** и нажать кнопку **«OK»** (Создать).

В результате в нужном месте документа появится шаблон матрицы с пустыми полями для ввода значений элементов, которые необходимо заполнить нужными значениями. Для перемещения по полям шаблона удобно использовать клавишу **«Tab»**.

Если необходимо создать вектор, то в окне диалога нужно указать число столбцов равное единице (создать матрицу с одним столбцом).

Например:

$$A := \begin{pmatrix} 6 & 5 & 4 \\ 8 & 7 & 9 \\ 2 & 4 & 1 \end{pmatrix} \qquad V := \begin{pmatrix} 1 \\ 5 \\ 8 \end{pmatrix}$$

Обращение к элементам матриц и векторов производится по их индексам. При этом следует иметь ввиду, что нумерация строк и столбцов матриц, а также элементов вектора ***начинается с нуля***. Для перехода к нижнему индексу после набора имени вектора или матрицы необходимо нажать клавишу **<[>**, либо выбрать кнопку на панели инструментов **«Matrix»**.

Например, для приведенных выше матрицы A и вектора V, можно получить:

$$A_{0,0} = 6 \qquad A_{1,2} = 9 \qquad V_1 = 5$$

Для добавления в уже созданную матрицу строк или столбцов необходимо:

- установить курсор на элемент матрицы, правее и ниже которого будет осуществлена вставка столбцов и (или) строк;
- выполнить команду меню **Insert/Matrix** (Вставка/Матрица) или нажать комбинацию клавиш **<Ctrl>+<M>**;
- в появившемся окне диалога ввести необходимое число строк **(Rows)**, столбцов **(Columns)** и нажать кнопку **« Insert»** (Вставить).

Для удаления в уже созданной матрице строк или столбцов необходимо:

- установить курсор на элемент матрицы, правее и ниже которого будет осуществлено удаление столбцов и (или) строк;
- выполнить команду меню **Insert/Matrix** (Вставка/Матрица) или нажать комбинацию клавиш **<Ctrl>+<M>**;
- в появившемся окне диалога ввести необходимое число строк **(Rows)**, столбцов **(Columns)** и нажать кнопку **« Delete»** (Удалить).

В вышеописанных операциях вставки и удаления элементов матрицы допускается задание числа столбцов или строк, равных нулю.

Матрица может быть создана из предварительно подготовленного текстового файла, где ее элементы разделены пробелом. Для этой цели используется встроенная функция Mathcad (см. далее) **READPRN** (имя файла). Ниже приведен пример создания матрицы из текстового файла с именем **Matr.txt**, подготовленного в текстовом редакторе **«Блокнот»** и записанного в текущую директорию.

$$\mathrm{filename} := \text{"Matr.txt"} \qquad \mathrm{M} := \mathrm{READPRN}(\mathrm{filename})$$

$$\mathrm{M} = \begin{pmatrix} 3 & 4 & 5 & 6 \\ 7 & 5 & 3 & 2 \end{pmatrix}$$

Полученная в результате вычислений матрица может быть записана в текстовый файл с использованием встроенной функции **WRITEPRN** (имя файла).

Ниже приведен пример записи матрицы с именем P в текущую директорию.

$$\mathrm{P} := \begin{pmatrix} 1 & 5 & 7 \\ 8 & 9 & 2 \end{pmatrix}$$

$$\mathrm{filename} := \text{"Matr\_P.txt"} \qquad \mathrm{WRITEPRN}(\mathrm{filename}) := \mathrm{P}$$

2.10. Встроенные функции Mathcad

Встроенные функции – это основной набор функций, который поставляется вместе с Mathcad. Сюда относятся функции **sin(*x*)**, **cos(*x*)**, **ln(*x*)** и т. д. Их список можно просмотреть в окне **«Insert Function»** после выбора команды меню **Insert/Function** (Вставка/Функция).

Для вставки встроенной функции в математическое выражение ее можно набрать с клавиатуры либо:

• поместить курсор в нужное место документа;

• выполнить команду меню **Insert/Function** (Вставка/Функция) или нажать кнопку **«f(*x*)»** на панели инструментов;

• в появившемся окне диалога **«Insert Function»** выбрать из списка имя нужной функции и нажать клавишу **«Insert»** (Вставить).

Среди встроенных функций рассмотрим логическую функцию **if(*s*,*x*,*y*)**, где ***s*** – логическое выражение, которое может принимать значение «**истина**» – **1** или «**ложь**» – **0**.

Функция **if(*s*,*x*,*y*)** возвращает значение выражения ***x***, которое вычисляется, если значение ***s*** принимает значение **«истина»** (***s*=1**) или значение выражения ***y***, которое вычисляется, если значение ***s*** принимает значение **«ложь»** (***s*=0**).

Отметим также, что в пакете Mathcad для обозначения логической операции **«И»** служит символ « $\wedge$ » , а для обозначения логической операции «**ИЛИ**» служит символ « $\vee$ », операция «**НЕ**» обозначается символом « $\neg$ ».

Например:

$$a := 5 \qquad b := 7 \qquad c := 10$$

$$x := \text{if}(a > b, a + b, a - b) \qquad x = -2$$

$$y := \text{if}\left(a < b \wedge a > 5, a^2 + b^2, a + b\right) \qquad y = 12$$

$$z := \text{if}\left(a < b \vee a > 5, a^2 + b^2, a + b\right) \qquad z = 74$$

2.11. Текст в документе Mathcad

Текст в документе Mathcad служит для объяснения и анализа математических выражений, уравнений и графиков. Для формирования текста в документе необходимо создать специальную текстовую область.

Для создания текстовой области необходимо:

• установить курсор в нужном месте документа;

• выполнить команду меню **Insert/Text Region** (Вставка/Текстовая Область);

После ввода текста, для выхода из текстовой области, щелкнуть **ЛКМ** вне этой области.

При печати в текстовой области перенос по умолчанию производится при достижении строкой правого поля или края страницы, а для переноса в нужном месте можно использовать клавишу **<Enter>**. Часто желательно установить ширину всей текстовой области и автоматически осуществлять перенос внутри её по мере набора текста.

Для этого необходимо:

- печатать текст обычным образом, пока первая строка не достигнет нужной длины;
- нажать комбинацию клавиш **<Ctrl>+<Enter>** и продолжить печать;

Перенос строк будет выполняться автоматически по заданной ширине области.

Для изменения размеров существующей текстовой области необходимо выделить ее щелчком **ЛКМ**, заключив в выделяющий прямоугольник с маркерами, а затем изменить размер, ухватившись мышью за нужный маркер.

Текст, созданный в документе Mathcad, можно форматировать, используя команды пункта меню **«Format»** и соответствующие кнопки на панели инструментов для форматирования текста.

2.12. Изменение компоновки документа

2.12.1. Выделение областей

Если нужно изменить компоновку документа путем перемещения или копирования областей, то их необходимо предварительно выделить. Для выделения области необходимо:

- установить указатель мыши в начало выделяемой области;
- нажать **ЛКМ** и перетащить указатель мыши в конец, выделяемой области.

Выделяемая область будет заключена в пунктирный прямоугольник.

Для выделения ряда несмежных областей необходимо выполнить щелчок **ЛКМ** по каждой из них при нажатой клавише **<Shift>**.

При необходимости выделить ряд смежных областей необходимо выполнить щелчок **ЛКМ** на первой и последней из них при нажатой клавише **<Ctrl>**.

При необходимости выделить некоторое выражение или часть его внутри области его расположения надо установить курсор в начало требуемого фрагмента и протянуть его до нужного места с нажатой **ЛКМ**. При этом выражение или фрагмент его выделяются черным цветом.

2.12.2. Копирование, перемещение, вставка, удаление и выравнивание областей

Для копирования области необходимо:

- выделить область;
- выполнить команду меню **Edit/Copy** (Правка/Копировать);
- установить курсор в место вставки и выполнить команду меню **Edit/Paste** (Правка/Вставить).

Команда меню **Edit/Paste Spesial** (Правка/Специальная вставка) позволяет вставить в документ Mathcad содержимое буфера обмена в одном из допустимых форматов.

Для перемещения области необходимо:

- выделить область;
- выполнить команду меню **Edit/Cut** (Правка/Вырезать);
- установить курсор в место вставки и выполнить команду **Edit/Paste** (Правка/Вставить).

Для перемещения области с помощью мыши необходимо:

- выделить область;
- установить указатель мыши внутри выделенной области (при этом он примет вид руки);
- нажать **ЛКМ** и перетащить область на новое место.

Для удаления области необходимо:

- выделить область;
- выполнить команду меню **Edit/Cut** (Правка/Вырезать) – область помещается в буфер обмена, или команду **Edit/Delete** (Правка/Удалить) – без помещения в буфер обмена, если выделено выражение или часть его внутри области своего расположения.

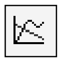
Для выравнивания областей необходимо:

- выделить области;
- выполнить команду меню **Format/Align Region/Across** (Правка/Выровнять области/По горизонтали) – выравнивание областей по верхнему краю;
- выполнить команду меню **Format/Align Region/Down** (Правка/Выровнять области/По левому краю) – выравнивание областей по левому краю.

2.13. Создание двухмерного графика

2.13.1. Построение графика функции

Для построения графика функции **f(*x*)** необходимо:

- рассчитать значения функции **f(*x*)** на необходимом интервале;
- установить курсор на нужное место в документе и ввести символ "**@**", или выбрать одномерный график **(X-Y plot)**  из палитры **«Графики»**.

На экране появляется шаблон графика, который необходимо заполнить, перемещаясь по его полям ввода используя клавишу **«Tab»**. Поля шаблона, соответствующие пределам изменения аргумента и функции, можно не заполнять. Они будут устанавливаться автоматически.

Если необходимо построить несколько кривых на одном поле графика, то все значения соответствующих функций необходимо предварительно рассчитать, а при заполнении полей шаблона графика их имена написать через запятую.

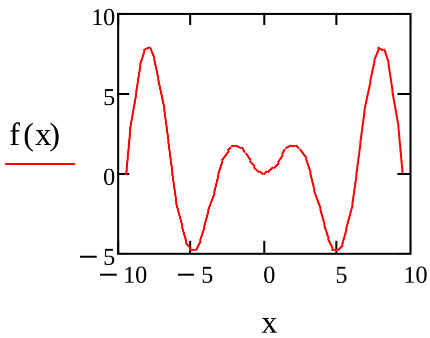
После заполнения полей шаблона необходимо щелкнуть **ЛКМ** вне области графика при работе в автоматическом режиме или клавишу **<F9>** при работе в ручном режиме. График будет отображен на экране.

Пример. Построить график функции $f(x) = x\sin(x)$ на отрезке $x \subset [-3\pi, 3\pi]$

$f(x) := x \cdot \sin(x)$ - определение функции

$x1 := -3 \cdot \pi \quad x2 := 3 \cdot \pi \quad \Delta x := \frac{\pi}{10}$ - пределы изменения и шаг аргумента

$x := x1, x1 + \Delta x \,..\, x2$ - цикл для расчета значений аргумента



- результат построения графика

Для изменения размера графика его необходимо выделить пунктирной линией и изменить размер, ухватившись мышью за соответствующий маркер (при этом курсор должен принять вид двойной стрелки).

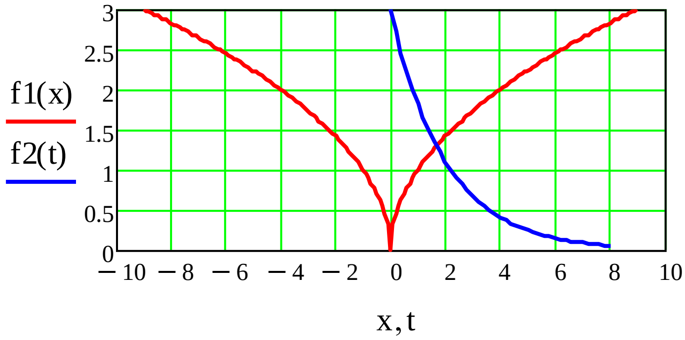
В пакете Mathcad предусмотрена возможность построения двух и более графиков функций, когда их аргументы имеют различные пределы изменения. Ниже приведен пример построения таких графиков.

$$f1(x) := \sqrt{|x|} \qquad f2(x) := 3 \cdot e^{-\left(\frac{x}{2}\right)}$$

$$x1 := -9 \quad x2 := 9 \qquad \Delta x := 0.1$$

$$t1 := 0 \qquad t2 := 8 \qquad \Delta t := 0.2$$

$$x := x1, x1 + \Delta x \,..\, x2 \qquad t := t1, t1 + \Delta t \,..\, t2$$



В Mathcad существует быстрый способ построения графика функции, когда пределы и шаг изменения аргумента для расчета ее значений устанавливаются автоматически.

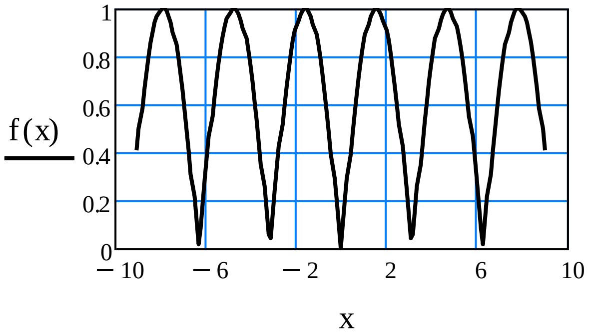
Для использования этого способа необходимо:

- определить нужную функцию;
- в нужном месте документа установить шаблон для двумерного графика;
- заполнить поля для имени функции и имени аргумента;

Пределы изменения аргумента будут автоматически установлены на отрезке $[-10, 10]$.

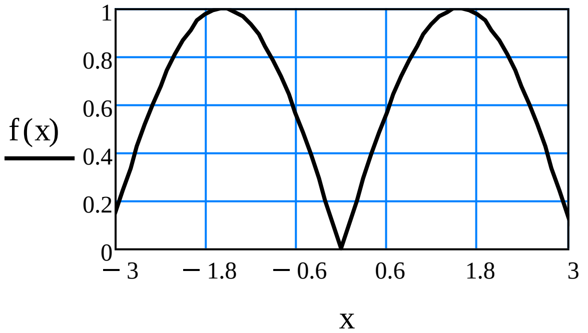
Ниже приведен пример построения графика быстрым способом.

$$f(x) := \sin(x)^2$$



Для коррекции пределов изменения аргумента необходимо выделить график щелчком ЛКМ и вручную вписать в соответствующие поля нужные значения пределов.

$$f(x) := \sin(x)^2$$



2.13.2. Форматирование графика

Для форматирования графика необходимо выполнить двойной щелчок **ЛКМ** по графику или выделить график щелчком **ЛКМ** и выполнить команду меню **Format/Graph/X-Y Plot** (Формат/График/X-Y График). После этого появится окно диалога, в котором необходимо выбрать соответствующую вкладку и сделать установки.

Диалоговое окно содержит 5 вкладок, позволяющих выполнять форматирование отдельных элементов графика:

- **X-Y Axses** (Y-X оси) – служит для форматирования осей графика (можно установить линии сетки);
- **Traces**(Линии) – служит для форматирования отдельных кривых на графике. Позволяет для каждой кривой на графике установить: **Legend label** (Имя), **Symbol** (Маркер), вид **Line** (Линии), **Color** (Цвет), **Type** (Тип кривой), **Symbol Weight** (Толщину символа) и т. д.;

• **Namber Format** – служит для форматирования числовых надписей на графике;

• **Labels** (Подписи) – позволяет установить: **Title** (Заголовок) графика; **Axis Labels** (названия осей): **X-Axis** (Ось X), **Y-Axis** (Ось Y);

• **Default** (По умолчанию) – позволяет изменить все установки на их значение по умолчанию или использовать текущие установки как установки по умолчанию.

Опишем процедуру установки линий сетки на графике.

Для этого необходимо:

• перейти в режим форматирования графика и выбрать вкладку **X-Y Axses** (Y-X оси);

• поставить метку в пункт **Grid lines** (Линии сетки) и выбрать нужный цвет для линий сетки;

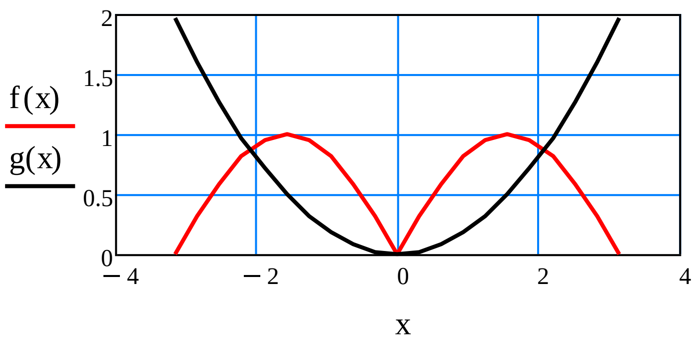
• убрать метку с пунктов **Auto grid** (Авто сетка);

• в полях ввода **Namber of grid** (Размер сетки) выбрать необходимое количество линий сетки по соответствующим осям.

Ниже приведен пример построения и форматирования графика двух функций.

$$f(x) := |\sin(x)| \qquad g(x) := \frac{x^2}{5}$$

$$x1 := -\pi \qquad x2 := \pi \qquad \Delta x := \frac{\pi}{10} \qquad x := x1, x1 + \Delta x \,..\, x2$$



2.14. Создание трехмерного графика

2.14.1. Построение графика функции двух переменных

Для построения графика функции $f(x, y)$ (трехмерного графика) необходимо:

- рассчитать матрицу значений функции $f(x, y)$ для $x \subset [x1, x2]$ с заданным шагом Δx и $y \subset [y1, y2]$ с заданным шагом Δy: $M_{i,j} = f(x_i, y_i)$, где x_i – множество значений аргумента x; y_i – множество значений аргумента y;
- нажать комбинацию клавиш **<Ctrl>+<2>**, или выполнить команду меню **Insert/Graph/Surface Plot** (Вставка/График/График поверхности) или выбрать график поверхности из палитры символов .
- в поле ввода шаблона напечатать имя матрицы, содержащей значения функции.

После заполнения поля шаблона необходимо щелкнуть **ЛКМ** вне области графика при работе в автоматическом режиме или нажать клавишу **<F9>** при работе в ручном режиме. График будет отображен на экране.

Каждый элемент матрицы представляется как точка на определенной высоте, пропорциональной значению элемента матрицы. По умолчанию ориентация поверхности такова, что первая строка матрицы простирается из дальнего левого угла сетки направо, а столбец идет из дальнего левого угла по направлению к наблюдателю. Mathcad рисует линии, чтобы соединить точки на графике. Эти линии определяют поверхность. При этом следует иметь в виду, что в рассчитанной матрице не содержится информация о том, в каких пределах изменяются аргументы x и y. Поэтому при построении графика по осям X и Y откладываются просто номера точек.

Ниже приведен пример построения графика функции двух переменных.

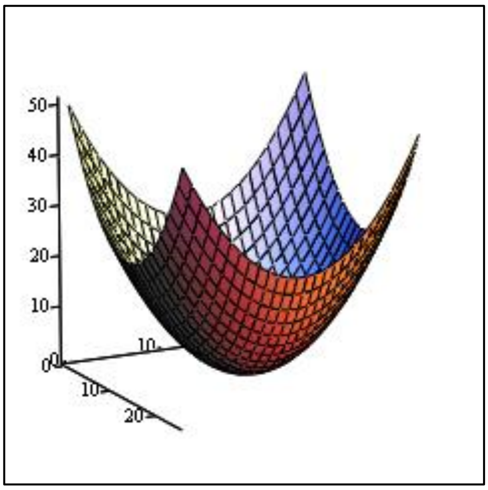
$f(x,y) := x^2 + y^2$

$x1 := -5 \quad x2 := 5 \quad \Delta x := 0.4 \quad Nx := \frac{x2 - x1}{\Delta x} \quad Nx = 25$

$y1 := -5 \quad y2 := 5 \quad \Delta y := 0.4 \quad Ny := \frac{y2 - y1}{\Delta y} \quad Ny = 25$

$i := 0 .. Nx \quad x_i := x1 + i \cdot \Delta x$

$j := 0 .. Ny \quad y_j := y1 + j \cdot \Delta y \qquad M_{i,j} := f(x_i, y_j)$



M

В Mathcad существует быстрый способ построения графика функции двух переменных, когда пределы и шаг изменения аргументов для расчета ее значений устанавливаются автоматически.

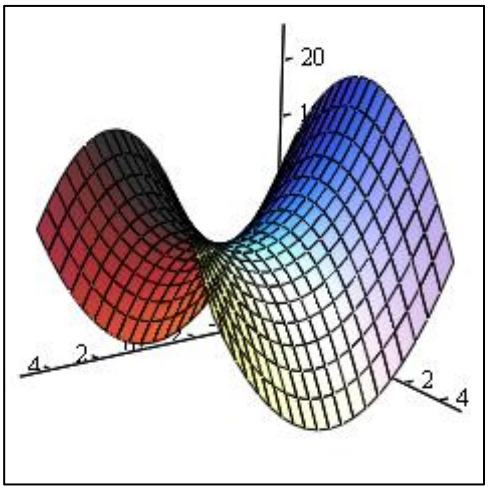
Для использования этого способа необходимо:

- определить нужную функцию;
- в нужном месте документа установить шаблон для графика функции двух переменных;
- в поле ввода шаблона написать имя функции.

Пределы изменения аргумента по каждому аргументу будут автоматически установлены на отрезке [–5, 5].

Ниже приведен пример построения графика быстрым способом.

$$f(x,y) := x^2 - y^2$$



f

Из графика видно, что при таком способе построения по осям $\boldsymbol{X}$ и $\boldsymbol{Y}$ откладываются истинные значения аргументов $\boldsymbol{x}$ и $\boldsymbol{y}$.

Для изменения пределов аргументов необходимо:

- выполнить двойной щелчок **ЛКМ** по графику;
- в появившемся окне диалога **3-D Plot Format** (Формат 3-D графика) выбрать вкладку **QuickPlot Data** (QuickPlot Данные);
- на панели **Range1** (График1) заполнить поля ввода **start** (начало) и **end** (конец) для первого аргумента;
- на панели **Range2** (График2) заполнить поля ввода **start** (начало) и **end** (конец) для второго аргумента.

После закрытия окна диалога график будет отображен для новых пределов изменения аргументов функции.

2.14.2. Форматирование графика функции двух переменных

Для форматирования трехмерного графика необходимо выполнить двойной щелчок **ЛКМ** по графику или выделить график щелчком **ЛКМ** и выполнить команду меню **Format/Graph/3D Plot** (Формат/График/Трехмерный График);

После этого появляется окно диалога, содержащее 9 вкладок, позволяющее выполнять форматирование отдельных элементов графика.

Поясним установку режима отображения трехмерного графика в цвете. Для этого необходимо:

- перейти в режим форматирования графика, выбрать вкладку **Appearance** (Отображение) и на панели **Fill Options** (Параметры заполнения) установить переключатель в положение **Fill Surface** (Заполнить поверхность);
- перейти на вкладку **Lighting** (Освещение) и на панели **Lighting** установить флажок **Enable Lighting** (Разрешить освещение);
- из списка **Lighting Scheme** (Цветовая схема) выбрать подходящую схему освещения.

После закрытия окна диалога график поверхности будет отображен в цвете.

2.14.3. Построение карты линий уровня функции 2-х переменных

Линия уровня представляет собой множество точек, определяемых уравнением $f(x,y)=h$. Другими словами – линия уровня это сечение поверхности $z = f(x,y)$ плоскостью $z=h$. Придавая h различные значения, получим некоторое множество (карту) линий уровня для заданной поверхности.

Для отображения карты линий уровня в пакете Mathcad необходимо:

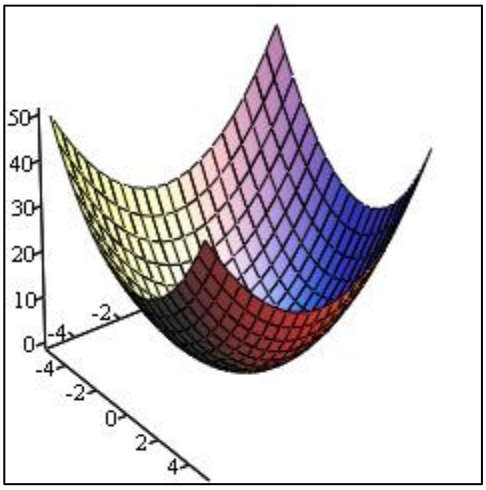
- построить график поверхности $z = f(x,y)$;
- перейти в режим форматирования графика;
- выбрать вкладку **General** (Общие) и на панели **Plot1**(если график только один) установить переключатель режимов отображения графика в положение **Contour Plot** (Контурный график);
- выбрать вкладку **Special** (Специальный) и на панели **Contour Option** (Параметры контурного графика) выбрать режим **Nambered** (Отображать значения высоты, на которой расположена линия).

Для нанесения координатной сетки на карту линий уровня необходимо:

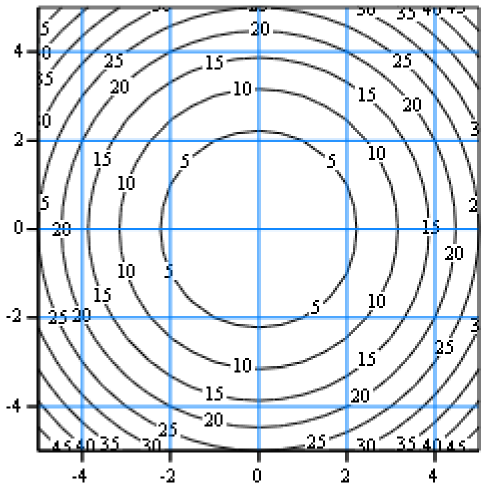
- перейти в режим форматирования графика;
- выбрать вкладку **Axes** (Оси), а на ней вкладку **X-Axis** (X-Ось), на панели **Grids** (Сетка) отметить пункт **Draw Lines** (Линии сетки), выбрать **Line Color** (Цвет линий);
- выбрать вкладку **Y-Axis** (Y-Ось) и проделать те же действия.

Ниже приведен пример графика поверхности, построенного «быстрым» способом и соответствующая карта линий уровня.

$$g(x,y) := x^2 + y^2$$



g



g

На карте линий уровня координатные оси расположены следующим образом:

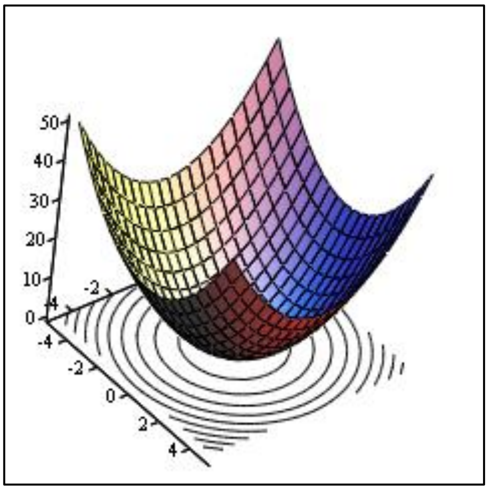
- начало координат находится в левом нижнем углу графика;
- координатная ось, соответствующая первому аргументу, направлена слева направо;
- координатная ось, соответствующая второму аргументу, направлена снизу вверх;

На графике можно одновременно отобразить поверхность и соответствующую ей карту линий уровня.

Для этого необходимо:

- построить два одинаковых графика в одной координатной системе, для чего при заполнении шаблона необходимо два раза указать имя функции через запятую;
- перейти в режим форматирования графика;
- выбрать вкладку **General** (Общие) и на панели **Plot1** (График1) или **Plot2** (График2) (так как у нас два одинаковых графика) установить переключатель режимов отображения графика в положение **Contour Plot** (Контурный график).

Ниже показан пример построения графика поверхности и карты линий уровня.



g,g

3. РЕШЕНИЕ ИНЖЕНЕРНЫХ ЗАДАЧ ЧИСЛЕННЫМИ МЕТОДАМИ

3.1. Операции с матрицами и векторами.

В настоящее время аппарат матричных вычислений широко используется для решения разнообразных научно-технических задач. Для проведения вычислений с матрицами и векторами в пакете Mathcad предусмотрен ряд встроенных функций. Ниже приведены некоторые из них.

Имя функции	**Возвращается...**
rows(**A**)	Число строк в массиве **A**. Если **A** – скаляр, возвращается 0
cols(**A**)	Число столбцов в массиве **A**. Если **A** – скаляр, возвращается 0
length (**v**)	Число элементов в векторе **v**
last(**v**)	Индекс последнего элемента в векторе **v**
max(**A**)	Максимальный элемент в массиве **A**
min(**A**)	Минимальный элемент в массиве **A**
identity(**n**)	(n×n) – единичная матрица

diag(**v**)	Диагональная матрица, содержащая на диагонали элементы вектора **v**
tr(**A**)	Сумма диагональных элементов матрицы **A** (след **A**). Матрица **A** должна быть ***квадратной***

Рассмотрим примеры некоторых операций с матрицами и векторами.

Для формирования матричных операторов удобно использовать панель инструментов **Matrix.**

Произведение матриц:

$$A := \begin{pmatrix} 1 & 2 & 4 \\ 6 & 7 & 9 \\ 4 & 7 & 8 \end{pmatrix} \quad B := \begin{pmatrix} 3 & 2 & 1 \\ 6 & 5 & 9 \\ 9 & 7 & 7 \end{pmatrix} \qquad C := A \cdot B \qquad C = \begin{pmatrix} 51 & 40 & 47 \\ 141 & 110 & 132 \\ 126 & 99 & 123 \end{pmatrix}$$

Произведение матрицы на вектор:

$$V := \begin{pmatrix} 1 \\ 6 \\ 9 \end{pmatrix} \qquad P := A \cdot V \qquad P = \begin{pmatrix} 49 \\ 129 \\ 118 \end{pmatrix}$$

Транспонирование матрицы и вектора:

$$A^{T} = \begin{pmatrix} 1 & 6 & 4 \\ 2 & 7 & 7 \\ 4 & 9 & 8 \end{pmatrix} \qquad V^{T} = \begin{pmatrix} 1 & 6 & 9 \end{pmatrix}$$

Вычисление определителя матрицы:

$$D := |A| \qquad\qquad D = 25$$

Вычисление обратной матрицы:

$$S := A^{-1} \quad S = \begin{pmatrix} -0.28 & 0.48 & -0.4 \\ -0.48 & -0.32 & 0.6 \\ 0.56 & 0.04 & -0.2 \end{pmatrix} \qquad I := A \cdot A^{-1} \quad I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Скалярное произведение векторов:

$$W := \begin{pmatrix} 2 \\ 5 \\ 7 \end{pmatrix} \qquad P := W \cdot V \quad P = 95$$

Сумма элементов вектора:

$$s := \sum V \qquad s = 16$$

Выделение из матрицы и второго столбца и второй строки:

$$Y := A^{\langle 1 \rangle} \quad Y = \begin{pmatrix} 2 \\ 7 \\ 7 \end{pmatrix} \qquad Z := \left(A^T\right)^{\langle 1 \rangle} \quad Z^T = (6 \ \ 7 \ \ 9)$$

Число строк и столбцов матрицы:

$$n := \mathrm{rows}(A) \quad n = 3 \qquad m := \mathrm{cols}(A) \quad m = 3$$

Максимальный и минимальный элемент матрицы:

$$Bmax := \max(B) \ Bmax = 9 \qquad Bmin := \min(B) \ Bmin = 1$$

Число элементов вектора:

$$k := \mathrm{length}(V) \quad k = 3$$

Номер последнего элемента вектора:

$$m := \mathrm{last}(V) \qquad m = 2$$

Создание единичной матрицы:

$$I := \mathrm{identity}(2) \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Создание диагональной матрицы из элементов вектора:

$$T := \mathrm{diag}(V) \quad T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 9 \end{pmatrix}$$

Вычисление следа (суммы диагональных элементов) квадратной матрицы:

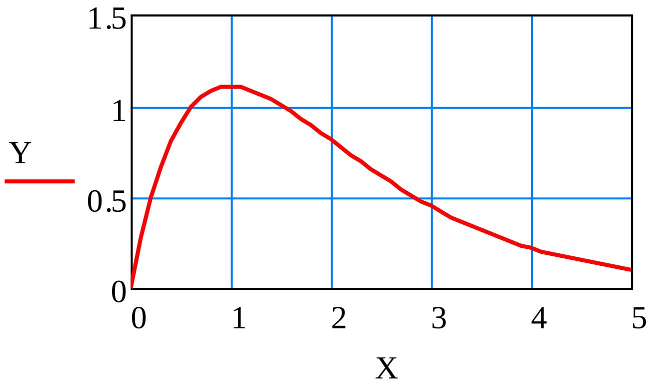
$$Sp := tr(A) \qquad Sp = 16$$

Рассмотрим построение графика функции, когда ее рассчитанные значения и значения ее аргумента предварительно записаны в массивы.

$$f(x) := 3x \cdot e^{-x}$$

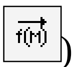
$$x1 := 0 \qquad x2 := 5 \qquad \Delta x := 0.1 \qquad N := \frac{x2 - x1}{\Delta x}$$

$$i := 0 .. N \qquad X_i := x1 + i \cdot \Delta x \quad Y_i := f\left(X_i\right)$$



Для выполнения поэлементных операций над элементами векторов или матриц в пакете Mathcad существует оператор векторизации, который предписывает Mathcad применять операторы и функции к каждому элементу массива поочередно. Так если ***V*** – вектор, то **sin(*V*)** – недопустимое выражение. Но если использовать оператор векторизации, Mathcad вычисляет синус каждого элемента вектора ***V***, а результат – новый вектор, чьи элементы – синусы элементов ***V***.

Для применения оператора векторизации к некоторому выражению необходимо:

- установить в нужном месте документа шаблон оператора векторизации (нажать комбинацию клавиш **<Ctr>**+<–> или воспользоваться палитрой символов );
- заполнить шаблон необходимым выражением.

Например:

$$\overrightarrow{\sin(A)} = \begin{pmatrix} 0.841 & 0.909 & -0.757 \\ -0.279 & 0.657 & 0.412 \\ -0.757 & 0.657 & 0.989 \end{pmatrix} \qquad \overrightarrow{\cos(V)} = \begin{pmatrix} 0.54 \\ 0.96 \\ -0.911 \end{pmatrix}$$

3.2. Решение уравнений и систем уравнений

3.2.1. Системы линейных уравнений

Рассмотрим систему линейных уравнений:

$$\begin{cases} \boldsymbol{a}_{11}\boldsymbol{x}_1 + \boldsymbol{a}_{12}\boldsymbol{x}_2 + \cdots + \boldsymbol{a}_{1n} = \boldsymbol{b}_1 \\ \boldsymbol{a}_{21}\boldsymbol{x}_1 + \boldsymbol{a}_{22}\boldsymbol{x}_2 + \cdots + \boldsymbol{a}_{2n} = \boldsymbol{b}_2 \\ \cdots \\ \boldsymbol{a}_{n1}\boldsymbol{x}_1 + \boldsymbol{a}_{n2}\boldsymbol{x}_2 + \cdots + \boldsymbol{a}_{nn} = \boldsymbol{b}_n \end{cases},$$

Запишем ее в матричном виде:

$$\boldsymbol{AX} = \boldsymbol{B},$$

где $A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$ – квадратная матрица размером **n×n**

$\boldsymbol{B} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$ – вектор размером **n×1**

$\boldsymbol{X} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ – вектор неизвестных размером **n×1**

Из курса линейной алгебры известно, что решение рассматриваемой системы уравнений имеет вид:

$\boldsymbol{X} = \boldsymbol{A}^{-1}\boldsymbol{B}$

Ниже приведен пример решения системы линейных уравнений:

$$\begin{cases} \mathbf{2}\boldsymbol{x} + \mathbf{3}\boldsymbol{y} - \boldsymbol{z} = \mathbf{5} \\ \boldsymbol{x} - \mathbf{2}\boldsymbol{y} + \mathbf{3}\boldsymbol{z} = \mathbf{6} \\ \mathbf{3}\boldsymbol{x} + \boldsymbol{y} - \mathbf{2}\boldsymbol{z} = \mathbf{-1} \end{cases}$$

в пакете Mathcad.

$$\mathrm{A} := \begin{pmatrix} 2 & 3 & -1 \\ 1 & -2 & 3 \\ 3 & 1 & -2 \end{pmatrix} \quad \mathrm{B} := \begin{pmatrix} 5 \\ 6 \\ -1 \end{pmatrix} \quad \begin{pmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \end{pmatrix} := \mathrm{A}^{-1} \cdot \mathrm{B}$$

$$\mathrm{X} := \begin{pmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \end{pmatrix} \qquad \mathrm{X} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \qquad \mathrm{x} = 1 \qquad \mathrm{y} = 2 \qquad \mathrm{z} = 3$$

В Mathcad имеется встроенная функция **lsolve(A,B)**, которая служит для решения системы линейных уравнений. Далее приведен пример решения рассматриваемой системы с помощью этой функции.

$$\mathrm{X} := \mathrm{lsolve}(\mathrm{A}, \mathrm{B}) \quad \mathrm{X} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

3.2.2. Одно уравнение с одним неизвестным

Для решения одного уравнения с одним неизвестным $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0}$ в пакете Mathcad используются функции **root(** $\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x}$**)** и **root(** $\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}$**)**, где $\boldsymbol{f}(\boldsymbol{x})$ – функция, определенная где-либо в рабочем документе, или выражение, которое возвращает скалярное значение, $\boldsymbol{x}$ – имя переменной, относительно которой ищется решение уравнения.

Функция **root(** $\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x}$**)** требует дополнительного задания начального значения (guess value) переменной $\boldsymbol{x}$. Для этого нужно просто предварительно присвоить $\boldsymbol{x}$ некоторое число. Поиск корня будет производиться вблизи этого числа. Таким образом, присвоение начального значения требует ***априорной информации о примерной локализации корня***.

Если функция $\boldsymbol{f(x)}$ имеет несколько корней, то найденное значение корня будет зависеть от начального приближения для переменной $\boldsymbol{x}$. Будет найдено такое значение корня, в окрестности которого задано начальное приближение.

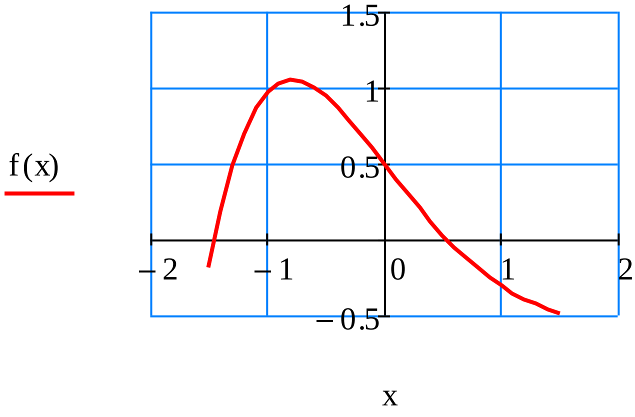
Рассмотрим пример. Пусть необходимо найти все корни уравнения $\mathbf{e}^{-x}\mathbf{cos}(x) = \mathbf{0{,}5}$ на отрезке $[-1{,}5; 1{,}5]$.

Ниже приведен пример решения этого уравнения с применением функции $\mathbf{root}(f(x), x)$. Для локализации корней используется построение графика функции на заданном интервале.

$$f(x) := e^{-x}\cdot\cos(x) - 0.5$$

$x1 := -1.5 \quad x2 := 1.5 \quad \Delta x := 0.1$

$x := x1, x1 + \Delta x \,..\, x2$



$x := -2$ - начальное приближение для первого корня

$x1 := \mathrm{root}(f(x), x) \quad x1 = -1.454$ - найден первый корень

$x := 1$ - начальное приближение для второго корня

$x2 := \mathrm{root}(f(x), x) \quad x2 = 0.54$ - найден второй корень

Иногда удобнее задавать не начальное приближение к корню, а интервал $[a, b]$, внутри которого корень заведомо находится. В этом случае следует использовать функцию **root(** $f(x), x, a, b$**)** с четырьмя аргументами, а присваивать начальное значение x не нужно, как показано ниже.

$$x1 := \mathrm{root}(f(x), x, -2, -1) \quad x1 = -1.454$$

$$x2 := \mathrm{root}(f(x), x, 0, 1) \qquad x2 = 0.54$$

При использовании функции **root(** $f(x), x, a, b$**)** необходимо помнить следующее:

- внутри интервала $[a, b]$ не должно находиться более одного корня. Иначе будет заранее неизвестно, какой именно из корней найден;
- значения функций $f(\mathrm{a})$ и $f(b)$ должны иметь разный знак, иначе будет выдано сообщение об ошибке.

Для нахождения корней полинома, т. е. для решения уравнения $a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = 0$ используется функция **polyroots** (V), где V – вектор коэффициентов полинома. $V^T = (a_0, a_1, \ldots, a_{n-1}, a_n)$. Функция возвращает вектор, элементами которого являются корни полинома.

Пример. Найти все корни уравнения $x^3 + 2x^2 - x - 2 = 0$

$V := (-2 \;\; -1 \;\; 2 \;\; 1)^T$ - вектор коэффициентов

$X := \mathrm{polyroots}(V) \quad X^T = (-2 \;\; -1 \;\; 1)$ - вектор корней

Пример. Найти все корни уравнения $x^3 - 10x + 2 = 0$

$V := (2 \;\; -10 \;\; 0 \;\; 1)^T$ - вектор коэффициентов

$X := \mathrm{polyroots}(V)$

$X^T = (-3.258 \;\; 0.201 \;\; 3.057)$ - вектор корней

3.2.3. Системы нелинейных уравнений

Пусть необходимо решить систему уравнений

$$\begin{cases} f_1(x_1, x_2, ..., x_n) = 0 \\ f_2(x_1, x_2, ..., x_n) = 0 \\ .... \\ f_n(x_1, x_2, ..., x_n) = 0 \end{cases}$$

Процедура решения в пакете Mathcad имеет следующий вид:

- задаются начальные значения для всех переменных системы:

$$x_1 := x_{10}$$
$$x_2 := x_{20}$$
$$.......... ....$$
$$x_n := x_{n0}$$

- Given – ключевое слово;
- записываются уравнения системы:

$$f_1(x_1, x_2, ..., x_n) = 0$$
$$f_2(x_1, x_2, ..., x_n) = 0$$
$$.......... ....$$
$$f_n(x_1, x_2, ..., x_n) = 0$$

- находится решение системы уравнений в виде вектора значений соответствующих переменных.

$$V := Find(x_1, x_2, ..., x_n)$$

Здесь $\boldsymbol{V} := \mathbf{Find}(\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_n})$ – функция, которая возвращает решение системы уравнений в виде вектора значений соответствующих переменных, если оно существует.

При записи системы уравнений, после ключевого слова **Given**, при необходимости можно указывать ограничения на возможные значения переменных, используя символы $\leq$, $\geq$, $<$ и $>$.

Замечание. При записи уравнений внутри блока **Given…Find** знак равенства следует вводить нажатием комбинаций клавиш **<Ctrl>+<=>** или использовать символ **Equal to** (логическое равенство) из палитры символов **Boolean** (Булево).

Если система уравнений является нелинейной, то найденное решение может быть не единственным и зависеть от начальных значений переменных.

Пример. Решить систему уравнений для $x \subset [-4, 4]$:

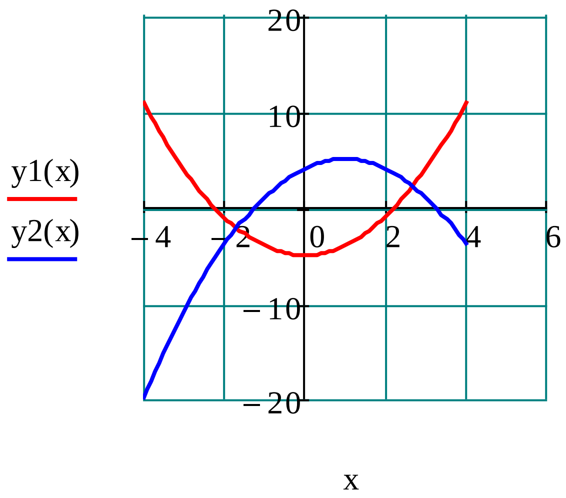
$$\begin{cases} y - x^2 + 5 = 0 \\ y + x^2 - 2x - 4 = 0 \end{cases}$$

Для поиска начальных приближений для переменных $\boldsymbol{x}$ и $\boldsymbol{y}$ выражаем из каждого уравнения $\boldsymbol{y}$ как функцию $\boldsymbol{x}$ и строим графики функций $\boldsymbol{y}1(\boldsymbol{x}) = \boldsymbol{x}^2 - 5$ и $\boldsymbol{y}2(\boldsymbol{x}) = -\boldsymbol{x}^2 + 2\boldsymbol{x} + 4$.

$$y1(x) := x^2 - 5 \qquad y2(x) := -x^2 + 2x + 4$$

$$x1 := -4 \qquad x2 := 4$$

$$x := x1, x1 + \Delta x \,..\, x2$$



Находим первое решение:

$x := -2 \qquad y := -1$

Given

$y - x^2 + 5 = 0$

$y + x^2 - 2x - 4 = 0$

$V := \mathrm{Find}(x, y) \qquad V = \begin{pmatrix} -1.679 \\ -2.179 \end{pmatrix}$

Проверка найденного решения:

$$\Delta := \left|y1\left(V_0\right) - y2\left(V_0\right)\right| \qquad \Delta = 0$$

Находим второе решение:

$x := 2 \qquad y := 1$

Given

$y - x^2 + 5 = 0$

$y + x^2 - 2x - 4 = 0$

$\begin{pmatrix} x0 \\ y0 \end{pmatrix} := \mathrm{Find}(x, y) \quad \begin{pmatrix} x0 \\ y0 \end{pmatrix} = \begin{pmatrix} 2.679 \\ 2.179 \end{pmatrix}$

Проверка найденного решения:

$$\Delta := \left|y1\,(x0) - y2\,(x0)\right| \qquad \Delta = 1.776 \times 10^{-15}$$

3.3. Дифференциальные уравнения и системы

3.3.1. Дифференциальные уравнения первого порядка

Рассмотрим обыкновенное дифференциальное уравнение (ОДУ) первого порядка

$$\frac{dy}{dx} = f\left[x, y(x)\right]$$

с начальным условием $y(x_0) = y_0$ (задача Коши).

Для решения таких уравнений в пакете Mathcad использовать вычислительный блок **Given…Odesolve**, реализующий численный метод Рунге-Кутта. Решение дифференциального уравнения находится с помощью встроенных функций Mathcad $\mathbf{Odesolve}(x, x\_end)$, или $\mathbf{Odesolve}(x, x\_end, step)$.

В этих функциях:

- x – имя переменной, относительно которой ищется решение;
- x_end – граничная точка отрезка $\lfloor x_0, x\_end \rfloor$, на котором ищется решение;
- $step$ – внутренний параметр численного метода, определяющий количество шагов, в которых метод Рунге-Кутта будет рассчитывать решение дифференциального уравнения.

Чем больше $step$, тем с лучшей точностью будет получен результат, но тем больше времени будет затрачено на его поиск. Подбором этого параметра можно заметно (в несколько раз) ускорить расчеты без существенного ухудшения их точности. Задание $step$ не является обязательным, и по умолчанию этот параметр определяется таким образом, чтобы длина шага была равна **0,1**. В случае сложных уравнений правильное определение $step$ может играть самую принципиальную роль для получения корректного решения.

Покажем на примере использование этого блока.

Пример. Выполнить интегрирование дифференциального уравнения

$$\frac{dy}{dx} + 2xy = 2x^2 \sin x$$

при начальном условии $y(0) = 1$, на отрезке $x \subset [0, 4]$.

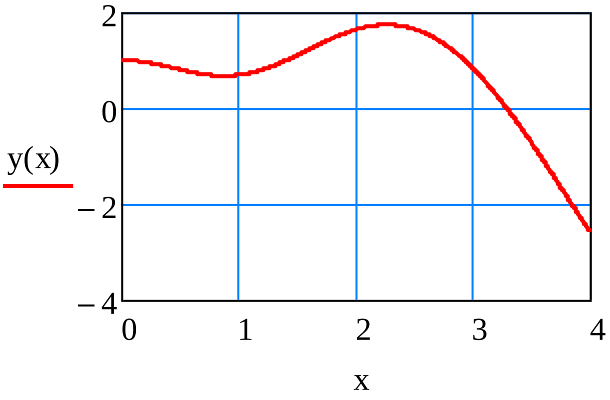
Ниже представлено решение этого уравнения в пакете Mathcad.

Given

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

$$y(0) = 1$$

$$y := \mathrm{Odesolve}(x, 4)$$



При решении дифференциального уравнения с помощью функции **Odesolve()** существует возможность определения значений искомой функции и в тех точках, которые не являются узловыми (то есть непосредственно в ходе работы численного алгоритма эти значения просчитаны не были). Достигается же это за счет задания между каждой парой точек интерполирующего полинома (точнее, кубического сплайна). Очевидно, что точность такого предсказания будет тем выше, чем меньше величина шага и чем более плавно и предсказуемо изменяется функция решения.

3.3.2. Дифференциальные уравнения высшего порядка

Рассмотрим задачу Коши для ОДУ n -го порядка

$$y^{(n)}(x) = f[x, y, y', \ldots, y^{(n-1)}]$$

с начальным условием

$$y(x_0) = y_0$$

$$y'(x_0) = y_1$$

$$\ldots$$

$$y^{(n-1)}(x_0) = y_{n-1}$$

В пакете Mathcad такая задача решается, так же как и задача Коши для ОДУ первого порядка. Ниже приведен пример решения ОДУ третьего порядка

$$y'''(x) + y''(x) + y' + y = -x\sin(x)$$

с начальными условиями $y(0)=0$, $y'(0)=1$, $y''(0)=2$ на отрезке $x \subset [0, 30]$.

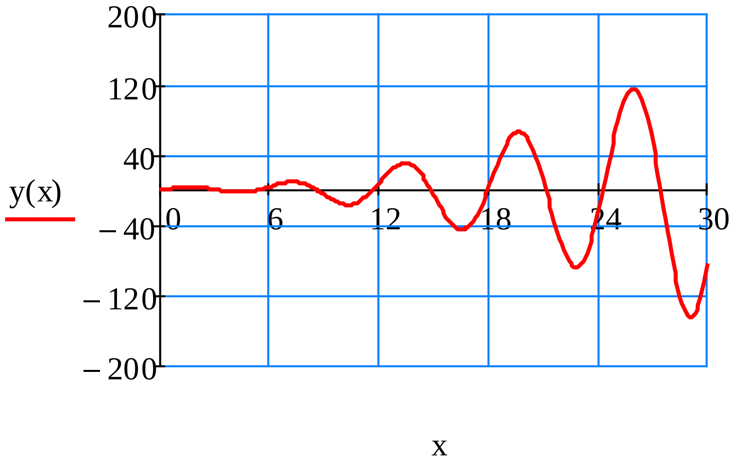
Given

$$y'''(x) + y''(x) + y'(x) + y(x) = -x \cdot \sin(x)$$

$$y(0) = 0 \qquad y'(0) = 1 \qquad y''(0) = 2$$

$$y := \mathrm{Odesolve}(x, 30)$$

$$x := 0, 0.1 .. 30$$



В приведенном примере для вставки символа производной (штрих) использована комбинация клавиш **<Ctr>+<F7>**.

3.3.3. Системы дифференциальные уравнения первого порядка

Рассмотрим систему ОДУ первого порядка

$$\begin{cases} y_1' = f_1(x, y_1, y_2, \ldots, y_n) \\ y_2' = f_2(x, y_1, y_2, \ldots, y_n) \\ \ldots \\ y_n' = f_n(x, y_1, y_2, \ldots, y_n) \end{cases}$$

и систему начальных условий

$$\begin{cases} y_1(x_0) = y_{10} \\ y_2(x_0) = y_{20} \\ \ldots \\ y_{n1}(x_0) = y_{n0} \end{cases}$$

Для поиска решения такой системы уравнений в Mathcad также можно использовать блок **Given…Odesolve**. Решение находится с помощью функции $\mathbf{Odesolve}(\boldsymbol{vector}, \boldsymbol{x}, \boldsymbol{x\_end})$.

Здесь:

$\boldsymbol{vector} = \begin{pmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{pmatrix}$ – вектор функций, относительно которых решается система дифференциальных уравнений, должен содержать имена функций без имени переменной;

- $\boldsymbol{x}$– имя переменной, относительно которой ищется решение;
- $\boldsymbol{x\_end}$ – граничная точка отрезка $\lfloor x_0, x\_end \rfloor$, на котором ищется решение.

Рассмотрим пример решения системы уравнений

$$\begin{cases} y'(x) = \dfrac{y^2 - yz}{x^2 - yz} \\ z'(x) = \dfrac{z(x+y)}{x^2 - yz} \end{cases}$$

для $x \subset [0, 10]$ с начальными условиями $y(0) = 1$, $z(0) = -1$

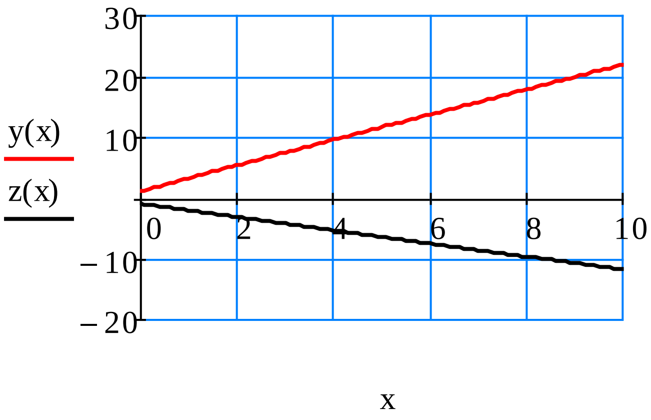
Given

$$\mathrm{y1'}(x) = \frac{\mathrm{y1}(x)^2 - \mathrm{y1}(x)\,\mathrm{z1}(x)}{x^2 - \mathrm{y1}(x)\cdot\mathrm{z1}(x)} \qquad \mathrm{y1}(0) = 1$$

$$\mathrm{z1'}(x) = \frac{\mathrm{z1}(x)\cdot(x + \mathrm{y1}(x))}{x^2 - \mathrm{y1}(x)\cdot\mathrm{z1}(x)} \qquad \mathrm{z1}(0) = -1$$

$$\begin{pmatrix} y \\ z \end{pmatrix} := \mathrm{Odesolve}\left[\begin{pmatrix} y1 \\ z1 \end{pmatrix}, x, 10\right]$$

$$x := 0, 0.1 .. 10$$



В более старых версиях Mathcad для решения ОДУ использовались специальные встроенные функции. Однако с появлением более наглядного и простого способа, связанного с применением вычислительного блока **Given…Odesolve**, они уже не представляют особой практической важности. Несмотря на это, о возможности их применения для решения ОДУ нужно иметь четкое представление, так как в некоторых случаях (например, в программировании) использовать вычислительный блок нельзя.

Рассмотрим одну из этих функций, которая служит для решения системы ОДУ первого порядка, а именно функцию **rkfixed**$(\boldsymbol{V}, \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{Np}, \boldsymbol{D})$, где:

- V – вектор начальных условий размерности $\boldsymbol{n}$ ($\boldsymbol{n}$ – порядок ОДУ или число уравнений в системе);
- $\boldsymbol{x_1}, \boldsymbol{x_2}$ – граничные точки интервала, на котором ищется решение дифференциальных уравнений (начальные условия, заданные в векторе $\boldsymbol{V}$, – это значение решения в точке $\boldsymbol{x_1}$;
- $\boldsymbol{Np}$ – число точек (не считая начальной точки), в которых ищется решение;

- $\boldsymbol{D(x,V)}$ – функция, которая возвращает значение в виде вектора из $\boldsymbol{n}$ элементов, содержащих первые производные неизвестных функций (т. е. элементами вектора являются правые части системы ОДУ первого порядка).

Функция **rkfixed** возвращает матрицу, первый столбец которой содержит точки, где ищется решение (значения аргумента), второй столбец содержит значения найденного решения в соответствующих точках для первого уравнения системы, второй столбец – для второго уравнения системы и т. д.

В качестве примера применения функции **rkfixed** рассмотрим систему ОДУ, решение которой было получено выше с использованием решающего блока **Given…Odesolve**. Для удобства перепишем ее в виде:

$$\begin{cases} y_1{}'(x) = \dfrac{y_1{}^2 - y_1 y_2}{x^2 - yz} \\ y_2{}'(x) = \dfrac{y_2(x + y_1)}{x^2 - y_1 y_2} \end{cases}$$

Будем искать решение для $x \subset [0, 10]$ ($x_1 = 0,\quad x_2 = 10$) в 100 точках ($Np = 100$) с начальными условиями $y_1(0) = 1$ ($y_1(x_1) = 1$), $y_2(0) = -1$ ($y_2(x_1) = -1$).

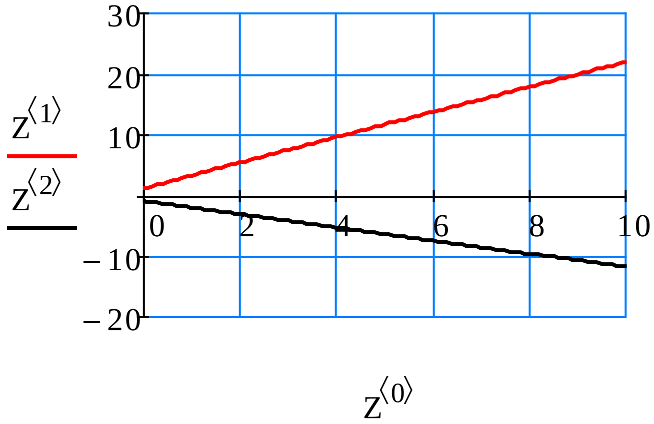
Решение в Mathcad имеет вид:

$x1 := 0 \qquad x2 := 10 \qquad Np := 100$

$y1 := 1 \qquad y2 := -1 \qquad Y := \begin{pmatrix} y1 \\ y2 \end{pmatrix}$

$$D(x, Y) := \begin{bmatrix} \dfrac{\left[(Y_0)^2 - Y_0 \cdot Y_1\right]}{x^2 - Y_0 \cdot Y_1} \\ \dfrac{Y_1 \cdot (x + Y_0)}{\left(x^2 - Y_0 \cdot Y_1\right)} \end{bmatrix}$$

$Z := \mathrm{rkfixed}(Y, x1, x2, Np, D)$



3.4. Интерполяция

Пусть некоторая функция $y = y(x)$ задана таблицей своих значений, где

x_i	x_1	x_2	…	x_n
y_i	y_1	y_2	…	yn

$$y_i = y(x_i), \qquad i = 0,1,2,\ldots,n$$

Интерполяция означает построение функции $f(x)$, аппроксимирующей зависимость $y(x)$ в промежуточных точках (между точками x_i). Поэтому интерполяцию еще по-другому называют аппроксимацией. В точках x_i значения интерполяционной функции должны совпадать с исходными данными, т. е. $f(x_i) = y(x_i)$.

Для построения интерполирующих функций в Mathcad имеются несколько встроенных функций, позволяющих «соединить» точки выборки данных (x_i, y_i) кривой разной степени гладкости.

3.4.1. Линейная интерполяция

Самый простой вид интерполяции – линейная, которая представляет искомую зависимость $f(x)$ в виде ломаной линии. Интерполирующая функция $f(x)$ состоит из отрезков прямых, соединяющих точки.

Для построения интерполирующей функции служит встроенная функция **linterp** ***(X, Y, t)*** – функция, аппроксимирующая данные векторов X и Y кусочно-линейной зависимостью.

Аргументы функции:

$\boldsymbol{X} = \left(\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}\right)^{\boldsymbol{T}}$ – вектор действительных данных аргумента;

$\boldsymbol{Y} = \left(\boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_n}\right)^{\boldsymbol{T}}$ – вектор действительных данных ($\boldsymbol{y_i} = \boldsymbol{y}(\boldsymbol{x_i})$) того же размера;

$\boldsymbol{t}$ – значение аргумента, при котором вычисляется интерполирующая функция. Элементы вектора $\boldsymbol{X}$ должны быть определены в порядке возрастания, т. е. $\boldsymbol{x_1} < \boldsymbol{x_3} < \boldsymbol{x_3} < \ldots < \boldsymbol{x_n}$.

Рассмотрим пример. Пусть в результате некоторого эксперимента получены вектора данных:

$\boldsymbol{X} = \left(\mathbf{0, 1, 2, 3, 4, 5, 6}\right)^{\boldsymbol{T}}$ и $\boldsymbol{Y} = \left(\mathbf{4.1, 2.4, 3, 4.3, 3.6, 5.2, 5.9}\right)^{\boldsymbol{T}}$.

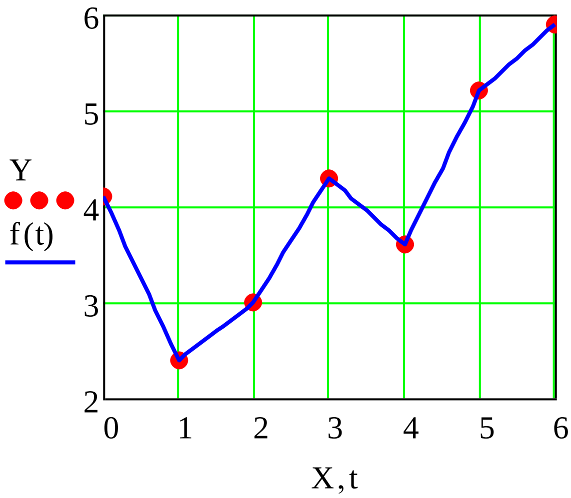
Необходимо построить линейную интерполирующую функцию для этих данных на отрезке $x \subset [0, 6]$

Ниже представлено решение поставленной задачи.

$X := (0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6)^{T} \qquad Y := (4.1\ \ 2.4\ \ 3\ \ 4.3\ \ 3.6\ \ 5.2\ \ 5.9)^{T}$

$f(t) := \mathrm{linterp}(X, Y, t)$

$\Delta x := 0.1 \qquad t := X_0, X_0 + \Delta x \,..\, X_{\mathrm{last}(X)}$



$f(2.5) = 3.65 \qquad\qquad f(1.5) = 2.7$

3.4.2. Кубическая сплайн-интерполяция

В большинстве практических приложений желательно соединить экспериментальные точки не ломаной линией, а гладкой кривой. Лучше всего для этих целей подходит интерполяция кубическими сплайнами. Смысл кубической сплайн-интерполяции заключается в том, что в промежутках между точками осуществляется аппроксимация в виде зависимости $f(t) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$. Коэффициенты a_0, a_1, a_2, a_3 рассчитываются независимо для каждого промежутка, исходя из значений y_i в соседних точках.

Для аппроксимации данных векторов X и Y кубическими сплайнами используется функция **interp(*s*, *X*, *Y*, *t*)**. Аргументы X, Y, t имеют тот же смысл что и для рассмотренной выше функции функция **linterp(*X*, *Y*, *t*)**. Аргумент s – вектор вторых производных, созданный одной из сопутствующих функций: **cspline(*X*, *Y*)** – вектор значений коэффициентов кубического сплайна; **pspline(*X*, *Y*)** – вектор значений коэффициентов квадратичного сплайна; или **lspline(*X*, *Y*)** – вектор значений коэффициентов линейного сплайна.

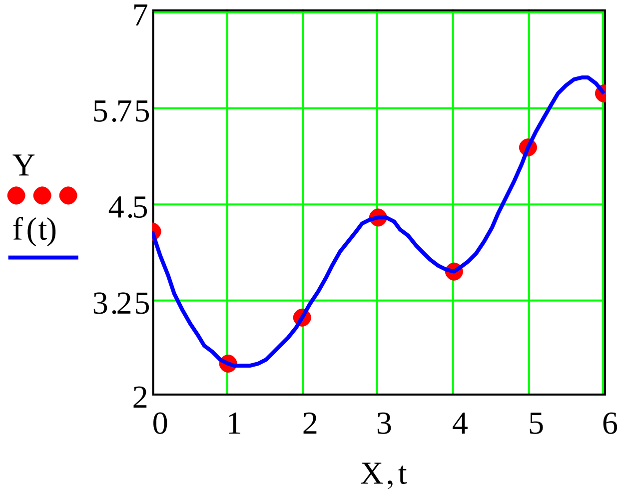
Выбор конкретной функции сплайновых коэффициентов влияет на интерполяцию вблизи конечных точек интервала интерполяции. Пример кубической сплайн-интерполяции приведен ниже.

$X := (0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6)^T \qquad Y := (4.1\ \ 2.4\ \ 3\ \ 4.3\ \ 3.6\ \ 5.2\ \ 5.9)^T$

$s := \mathrm{cspline}(X, Y)$

$f(t) := \mathrm{interp}(s, X, Y, t)$

$\Delta x := 0.1 \qquad t := X_0, X_0 + \Delta x \,..\, X_{\mathrm{last}(X)}$



f(2.5) = 3.837 f(1.5) = 2.454

3.5. Регрессия

Задачи математической регрессии имеют смысл приближения выборки данных (x_i, y_i) некоторой функцией $f(x)$, определенным образом минимизирующей совокупность ошибок $\varepsilon_i = f(x_i) - y_i$. Регрессия сводится к подбору неизвестных коэффициентов, определяющих аналитическую зависимость $f(x)$. Чаще всего неизвестные коэффициенты определяются из условия минимизации суммы квадратов ошибок $\left(\sum_{i=1}^{N} \varepsilon_i^2 \to \min\right)$. В этом случае метод подбора неизвестных коэффициентов называется методом наименьших квадратов. Заметим также, что существуют и другие методы для подбора коэффициентов [2].

3.5.1. Линейная регрессия

Самый простой и наиболее часто используемый вид регрессии – линейная. Приближение данных (x_i, y_i) осуществляется линейной функцией $y(x) = a + bx$.

Для расчета коэффициентов a и b в Mathcad имеются два дублирующих друг друга способа.

Первый способ использует функцию **line**(X, Y) – вектор из двух элементов (a, b) – коэффициентов уравнения линейной регрессии $y(x) = a + bx$.

Во втором способе для определения коэффициента a используется функция **intercept** (X, Y), а для вычисления коэффициента b функция **slope**(X, Y).

Аргументами этих функций являются:

- $X = (x_1, x_2, \ldots, x_n)^T$ – вектор действительных данных аргумента;
- $Y = (y_1, y_2, \ldots, y_n)^T$ – вектор действительных данных ($y_i = y(x_i)$) того же размера;

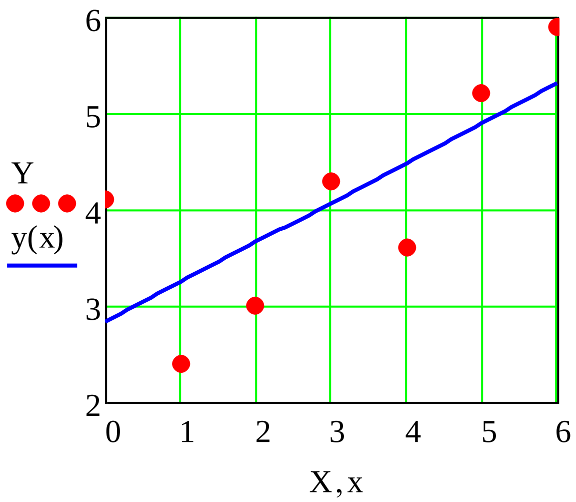
Ниже приводится пример применения этих функций для построения линейной регрессии.

$X := (0 \;\; 1 \;\; 2 \;\; 3 \;\; 4 \;\; 5 \;\; 6)^T \qquad Y := (4.1 \;\; 2.4 \;\; 3 \;\; 4.3 \;\; 3.6 \;\; 5.2 \;\; 5.9)^T$

$\begin{pmatrix} a \\ b \end{pmatrix} := \mathrm{line}(X, Y) \qquad \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 2.829 \\ 0.414 \end{pmatrix}$

$y(x) := a + b \cdot x$

$\Delta x := 0.1 \qquad x := X_0, X_0 + \Delta x \,..\, X_{last(X)}$



Коэффициенты $\boldsymbol{a}$ и $\boldsymbol{b}$ можно вычислить и так:

$a := \mathrm{intercept}(X, Y) \qquad a = 2.829$

$b := \mathrm{slope}(X, Y) \qquad b = 0.414$

3.5.2. Регрессия общего вида

К сожалению, линейная функция далеко не во всех случаях подходит для описания зависимости данных. При более сложных зависимостях между данными приходится использовать уравнение регрессии в виде линейной комбинации известных функций.

$$y(x) = a_0 f_0(x) + a_1 f_1(x) + ... + a_n f_n(x),$$

где коэффициенты $a_0, a_1, ..., a_n$ подлежат определению.

Неизвестные коэффициенты можно определить использованием встроенной функции **linfit**(X, Y, F).

Аргументы функции:

- $X = (x_1, x_2, \ldots, x_n)^T$ – вектор действительных данных аргумента, элементы которого расположены в порядке возрастания;
- $Y = (y_1, y_2, \ldots, y_n)^T$ – вектор действительных данных ($y_i = f(x_i)$) того же размера;
- F – функция, представляющая собой вектор, элементами которого являются функции, которые нужно объединить в виде линейной комбинации, т. е. $F = \left[f_0(x), f_1(x), \ldots, f_n(x) \right]^T$.

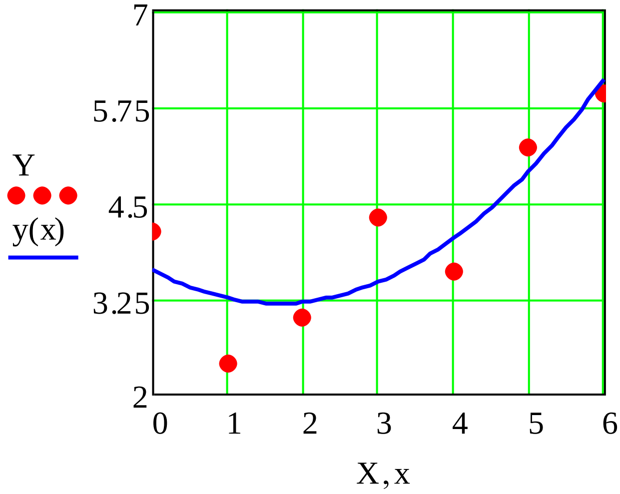
Ниже приведен пример построения уравнения регрессии вида $y(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$ по набору данных, записанных в вектора X и Y.

$X := (0 \;\; 1 \;\; 2 \;\; 3 \;\; 4 \;\; 5 \;\; 6)^T \qquad Y := (4.1 \;\; 2.4 \;\; 3 \;\; 4.3 \;\; 3.6 \;\; 5.2 \;\; 5.9)^T$

$F(x) := \left(1 \;\; x \;\; x^2\right)^T \qquad A := \mathrm{linfit}(X, Y, F)$

$A^T = (3.602 \;\; -0.514 \;\; 0.155) \qquad y(x) := A \cdot F(x)$

$\Delta x := 0.1 \qquad x := X_0, X_0 + \Delta x \,..\, X_{last(X)}$



3.6. Математическая статистика

Пакет Mathcad имеет развитый аппарат работы с задачами математической статистики. С одной стороны, имеется большое количество встроенных специальных функций, позволяющих рассчитывать плотности вероятности и другие основные характеристики основных законов распределения случайных величин. Наряду с этим, в Mathcad запрограммировано соответствующее количество генераторов псевдослучайных чисел для каждого закона распределения, что позволяет эффективно проводить моделирование методами Монте-Карло. Имеется возможность строить гистограммы и рассчитывать статистические характеристики выборок случайных чисел и случайных процессов, таких как средние, дисперсии, корреляции и т. п. При этом случайные последовательности могут как создаваться генераторами случайных чисел, так и вводиться пользователем из файлов.

3.6.1. Случайные величины

Для моделирования различных физических, экономических и прочих эффектов широко распространены методы, называемые методами Монте-Карло. Их основная идея состоит в создании определенной последовательности случайных чисел, моделирующей тот или иной эффект, например шум в физическом эксперименте, случайную динамику биржевых индексов и т. п. Для этих целей в Mathcad встроен ряд генераторов псевдослучайных чисел. Такие генераторы реализованы в виде встроенных функций, создающих выборку псевдослучайных данных с соответствующим законом распределения.

Рассмотрим некоторые из этих функций:

- **rnorm**(N, m, σ) – возвращает вектор N случайных чисел, имеющих нормальное распределение с математическим ожиданием m и среднеквадратичным отклонением σ (квадратный корень из дисперсии).
- **runif**(N, a, b) – возвращает вектор N случайных чисел, имеющих равномерное распределение на отрезке $[a, b]$, $a < b$.
- **rnd**(x) – возвращает равномерно распределенное случайное число на отрезке $[0, x]$. Эквивалент **runif**$(1, 0, x)$.

3.6.2. Числовые характеристики случайных величин

В Mathcad имеется ряд встроенных функций для расчетов числовых статистических характеристик случайных данных.

Пусть $X = \left(x_1, x_2, \ldots, x_n\right)^T$ – вектор действительных случайных данных (выборка). Тогда для него могут быть вычислены следующие числовые характеристики:

- **mean**(X) – выборочное среднее значение, $\mathbf{mean}(X) = \frac{1}{n}\sum_{i=1}^{n} x_i$;
- **median**(X) – выборочная медиана (**median**) – значение аргумента, которое делит гистограмму (см. ниже) плотности вероятностей на две равные части;
- **var**(X) – выборочная дисперсия (**variance**),
- $\mathbf{var}(X) = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \mathbf{mean}(X)\right)^2$;
- **stdev**(X) – среднеквадратичное (или «стандартное») отклонение (standarddeviation), $\mathbf{stdev}(X) = \sqrt{\mathbf{var}(X)}$;
- **max**(X), **min**(X) – максимальное и минимальное значения выборки;
- **mode**(X) – наиболее часто встречающееся значение выборки;

3.6.3. Построение гистограммы

Гистограммой называется график, аппроксимирующий по случайным данным $\boldsymbol{X} = \left(x_1, x_2, \ldots, x_n\right)^{\mathrm{T}}$ плотность их распределения. При построении гистограммы область значений случайной величины $[\boldsymbol{a}, \boldsymbol{b}]$ разбивается на некоторое количество $\boldsymbol{N}$ сегментов, а затем подсчитывается процент попадания данных в каждый сегмент. Для построения гистограмм в Mathcad имеется несколько встроенных функций. Рассмотрим наиболее удобную из них, функцию **histogram(** $\boldsymbol{N}, \boldsymbol{X}$**)**. Функция возвращает матрицу, состоящую из двух столбцов. Первый столбец содержит сегменты разбиения данных, а второй – число попадающих в них данных. Рассматриваемая функция рассчитывает гистограмму для сегментов равной длины.

Приведем пример, демонстрирующий применение рассмотренных функций в задаче статистического анализа выборки случайных данных.

Сформулируем следующую задачу:

- получить выборку из **100** случайных чисел, имеющих нормальный закон распределения вероятностей со средним значением **5** и дисперсией **2**;
- определить числовые характеристики выборки;
- построить гистограмму.

Ниже приведен пример решения поставленной задачи в пакете Mathcad.

$N := 100 \quad m := 5 \quad D := 2 \qquad \sigma := \sqrt{D}$

$X := \mathrm{rnorm}(N, m, \sigma)$

$m\_X := \mathrm{mean}(X) \qquad m\_X = 4.787$

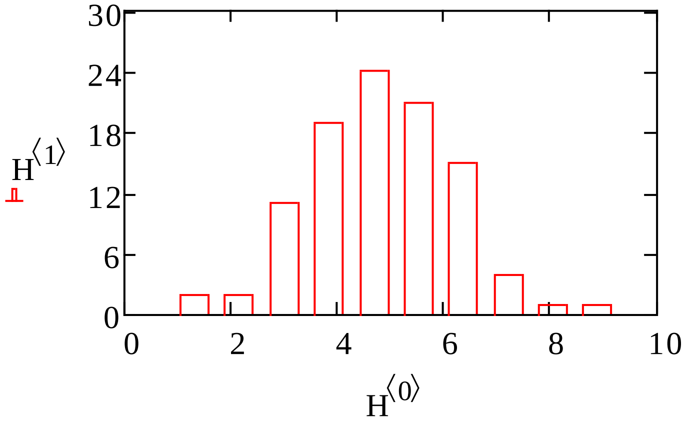
$var\_X := \mathrm{var}(X) \qquad var\_X = 1.91$

$\sigma\_X := \mathrm{stdev}(X) \qquad \sigma\_X = 1.382$

$max\_X := \max(X) \qquad max\_X = 9.311$

$min\_X := \min(X) \qquad min\_X = 0.898$

$median\_X := \mathrm{median}(X) \qquad median\_X = 4.901$

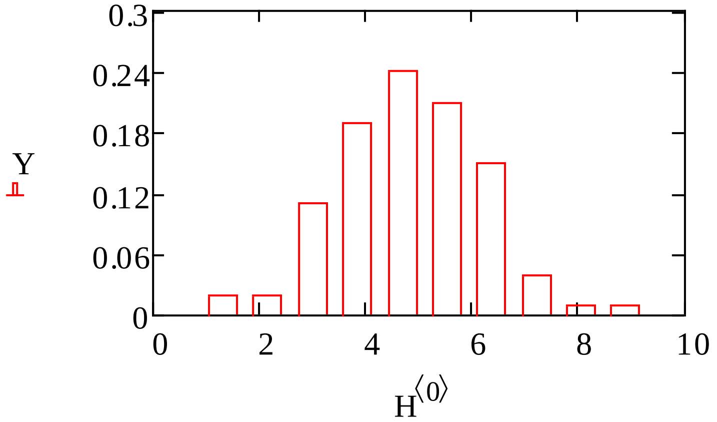


Для отображения графика в виде гистограммы необходимо:

- построить двухмерный график;
- перейти в режим форматирования и выбрать вкладку **Traces** (Линии);
- раскрыть список **Type** (Тип) и выбрать тип диаграммы **bar**.

Если на гистограмме необходимо отображать не число, а частоту попаданий данных в соответствующие интервалы, то второй столбец матрицы ***H*** необходимо пронумеровать, как показано ниже.

$$Y := \frac{H^{\langle 1 \rangle}}{N}$$



3.7. Поиск экстремума функции

Задачи поиска экстремума функции означают нахождение ее максимума (наибольшего значения) или минимума (наименьшего значения) в некоторой области определения ее аргументов. Ограничения значений аргументов, задающих эту область, как и прочие дополнительные условия, должны быть определены в виде системы неравенств и (или) уравнений. В таком случае говорят о задаче на условный экстремум. Для решения задач поиска максимума и минимума в Mathcad имеются встроенные функции **Minimize** и **Maximize**. С помощью этих функций решается задача поиска ***локальных*** экстремумов функции.

3.7.1. Экстремум функции одной переменной

Для поиска экстремума функции одной переменной рассматриваемые встроенные функции имеют следующий синтаксис: **Minimize(** $\boldsymbol{f}, \boldsymbol{x}$**)** и **Maximize(** $\boldsymbol{f}, \boldsymbol{x}$**)**. Здесь $\boldsymbol{f}$ – имя функции, для которой находится экстремум, а $\boldsymbol{x}$ – имя переменной, по которой выполняется поиск.

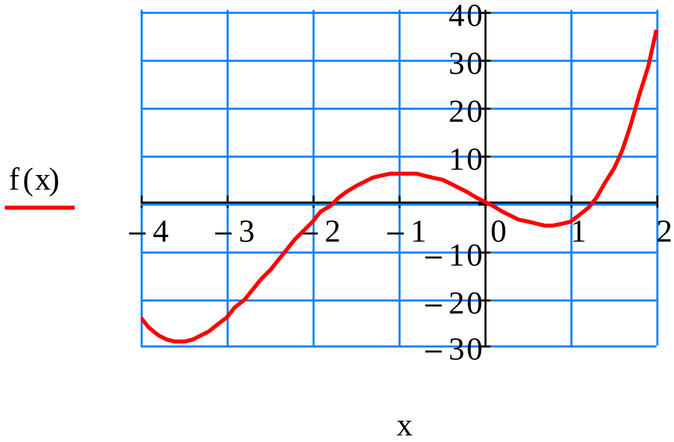
Рассмотрим следующий пример. Пусть требуется найти все локальные экстремумы функции $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{x}^4 + 5\boldsymbol{x}^3 - \boldsymbol{x} - 10$ на отрезке $\boldsymbol{x} \subset [-5, 5]$.

Построим на указанном отрезке график функции $\boldsymbol{f}(\boldsymbol{x})$

$$f(x) := x^4 + 5x^3 - 10x$$

$$x1 := -4 \qquad x2 := 2 \qquad \Delta x := 0.1$$

$$x := x1, x1 + \Delta x\, ..\, x2$$



Из графика видно, что на указанном отрезке функция имеет три локальных экстремума: два локальных минимума и один локальный максимум. Для поиска первого локального минимума укажем, исходя из графика функции, для переменной x некоторое начальное приближение к точке $x1\_min$ и воспользуемся функцией **Minimize** как показано ниже. Затем вычислим значение функции в найденной точке $x1\_min$.

$$x := -4 \quad x1\_min := Minimize(f, x) \qquad x1\_min = -3.552$$

$$y1\_min := f(x1\_min) \qquad y1\_min = -29.371$$

Аналогичным образом определяем x_max, $x2\_min$ и значения функции в этих точках. Результат показан ниже.

$$x := -4 \quad x1\_min := Minimize(f, x) \qquad x1\_min = -3.552$$

$$y1\_min := f(x1\_min) \qquad y1\_min = -29.371$$

$$x := -1.5 \qquad x\_max := Maximize(f, x) \qquad x\_max = -0.944$$

$$y\_max := f(x\_max) \qquad y\_max = 6.028$$

$$x := 1 \qquad x2\_min := Minimize(f, x) \qquad x2\_min = 0.746$$

$$y2\_min := f(x2\_min) \qquad y2\_min = -5.074$$

3.7.2. Экстремум функции многих переменных

Для поиска экстремума функции многих переменных

$$y = f(x_1, x_2, \ldots, x_n)$$

рассматриваемые встроенные функции имеют следующий синтаксис:

$$\textbf{Minimize}(f, x_1, x_2, \ldots, x_n) \text{ и } \textbf{Maximize}(f, x_1, x_2, \ldots, x_n).$$

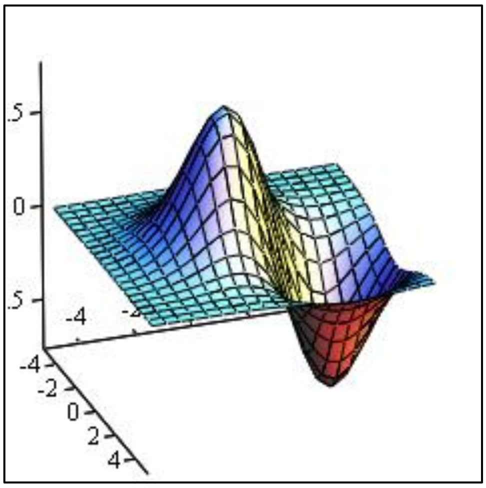
Здесь f – имя функции, для которой находится экстремум, а $x_1, x_2, \ldots, x_n$ – имена переменных по которым выполняется поиск. При этом необходимо указать начальное приближение к искомой точке экстремума в виде: $x_1 = x_1^0,\ x_2 = x_2^0, \ldots,\ x_n = x_n^0$. Указанное требование в общем случае может оказаться весьма непростой задачей для функции многих переменных.

Отдельно рассмотрим вопрос о поиске экстремума функции двух переменных $z = f(x, y)$. В рассматриваемом случае для поиска начального приближения можно воспользоваться картой линий уровня. Рассмотрим следующий пример. Пусть требуется найти все экстремумы функции

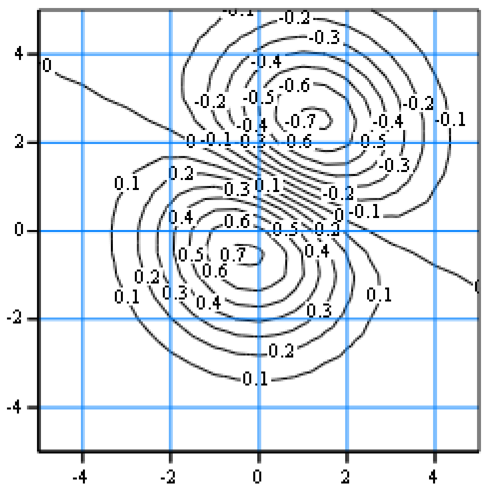
$$f(x, y) = e^{-\frac{x^2+y^2}{5}} - e^{-\frac{(x-1)^2+(y-2)^2}{5}} \quad \text{для } x \subset [-5, 5] \text{ и } y \subset [-5, 5]$$

Построим график функции и карту линий уровня.

$$f(x,y) := e^{-\left(\frac{x^2+y^2}{5}\right)} - e^{-\left[\frac{(x-1)^2+(y-2)^2}{5}\right]}$$



f



f

Используя карту линий уровня, выбираем начальное приближение $(\boldsymbol{x} = -1, \quad \boldsymbol{y} = -2)$ для поиска локального максимума и находим его.

$$x := -1 \qquad y := -2$$

$$\begin{pmatrix} x\_max \\ y\_max \end{pmatrix} := \text{Maximize}(f, x, y)$$

$$x\_max = -0.272 \qquad y\_max = -0.543$$

$$f(x\_max, y\_max) = 0.73$$

Аналогично определяем начальное приближение для поиска локального минимума $(\boldsymbol{x} = 3, \quad \boldsymbol{y} = 4)$ и находим его.

$$x := 3 \qquad y := 4$$

$$\begin{pmatrix} x\_min \\ y\_min \end{pmatrix} := \text{Minimize}(f, x, y) \qquad x\_min = 1.272$$

$$y\_min = 2.543 \qquad f(x\_min, y\_min) = -0.73$$

Рассмотрим еще один пример на условный экстремум. Пусть необходимо найти минимум функции $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^2 + \boldsymbol{y}^2 + 2$ при условии, что $\boldsymbol{x} + \boldsymbol{y} = 1$.

Поскольку данная функция является квадратичной, то она имеет единственный минимум. Это означает, что в качестве начального приближения для поиска минимума можно взять любое число, например $\boldsymbol{x} = 0$ и $\boldsymbol{y} = 0$.

Для поиска условного экстремума функции необходимо использовать блок **Given**, как показано ниже.

$x := 0 \qquad y := 0$

Given

$x + y = 1$

$$\begin{pmatrix} x\_min \\ y\_min \end{pmatrix} := \mathrm{Minimize}(f, x, y)$$

$x\_min = 0.5 \qquad y\_min = 0.5$

$f(x\_min, y\_min) = 2.5$

4. РЕШЕНИЕ ИНЖЕНЕРНЫХ ЗАДАЧ СИМВОЛЬНЫМИ МЕТОДАМИ

В данной главе рассматриваются возможности символьного процессора Mathcad. Он позволяет решить многие задачи математики аналитически, без применения численных методов и, соответственно, без погрешностей вычислений. Mathcad позволяет проводить широкий спектр аналитических преобразований, таких как алгебраические и матричные операции, основные действия математического анализа и интегральные преобразования функций.

Для использовании символьных вычислений следует установить режим автоматических вычислений, команда меню **Tools/Calculate/ Automatic Calculation** (Инструменты/Вычислить/Автоматические вычисления).

4.1. Способы символьных вычислений

Символьные вычисления в Mathcad можно осуществлять в двух различных вариантах:

- с помощью команд меню;
- с помощью оператора символьного вывода «→» (символьного знака равенства), ключевых слов символьного процессора и обычных формул.

Первый способ более удобен, когда требуется быстро получить какой-либо аналитический результат для однократного использования, не сохраняя сам ход вычислений. Второй способ более нагляден, т. к. позволяет записывать выражения в традиционной математической форме и сохранять символьные вычисления в документах Mathcad.

Кроме того, аналитические преобразования, проводимые через меню, касаются только одного, выделенного в данный момент, выражения. Соответственно, на них не влияют формулы, находящиеся в документе Mathcad выше этого выделенного выражения (например, операторы присваивания значений каким-либо переменным). Оператор символьного вывода, напротив, учитывает все предыдущее содержимое документа и выдает результат с его учетом.

Для символьных вычислений при помощи команд предназначено главное меню **Symbolics** (Символика), объединяющее математические операции, которые Mathcad умеет выполнять аналитически (рис. 3).

Для реализации второго способа применяются все средства Mathcad, пригодные для численных вычислений (например, панели **Calculator**, **Evaluation** и т. д.), и специальная математическая панель инструментов (рис. 4), которую можно вызвать на экран нажатием кнопки **Symbolic Keyword Toolbar** (Панель символики) на панели **Math** (Математика). На панели **Symbolic** (Символика) находятся кнопки, соответствующие специфическим командам символьных преобразований.

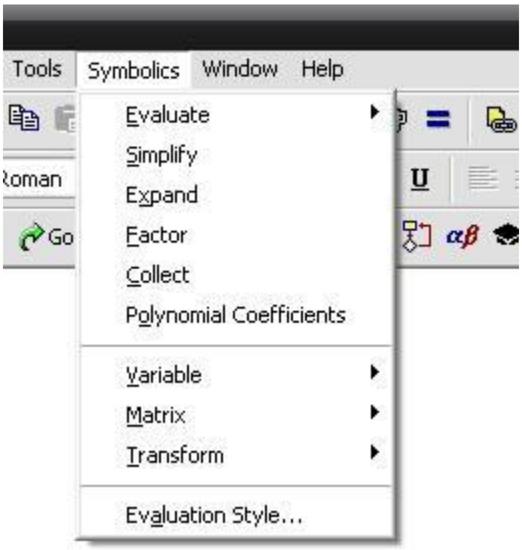


Рис. 3. Меню Symbolics (Символика)

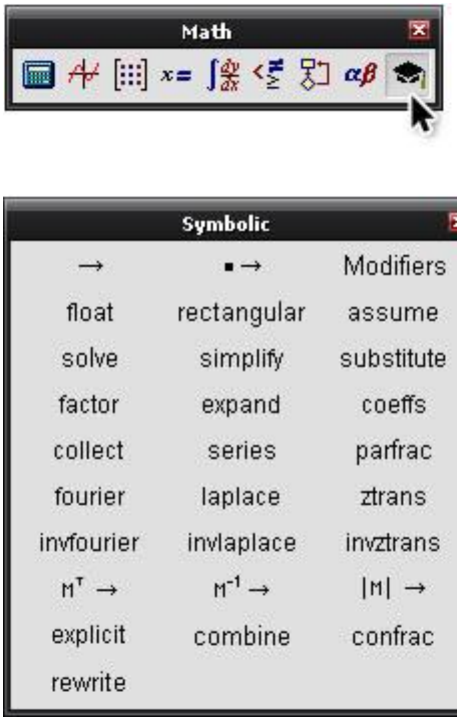


Рис. 4. Панель Symbolic (Символика)

Например, таким как разложение выражения на множители, расчет преобразования Лапласа и другим операциям, которые в Mathcad нельзя проводить численно, и для которых, соответственно, не предусмотрены встроенные функции.

Рассмотрим способ символьных вычислений с использованием команд меню **Symbolics** (Символика) на простом примере разложения на сомножители выражения $\sin(2x)+\cos(2x)$.

При использовании этого способа (с помощью меню) необходимо:

- ввести выражение $\sin(2x)+\cos(2x)$;
- выделить его цветом целиком;
- выбрать в главном меню пункты **Symbolics/Expand** (Символика/ Разложить).

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

$\sin(2\cdot x) + \cos(2\cdot x)$ – исходное выражение

$\cos(x)^2 + 2\cdot\cos(x)\cdot\sin(x) - \sin(x)^2$ – результат

Символьные операции с помощью меню возможны лишь над каким-либо объектом (выражением, его частью или отдельной переменной). Для того чтобы правильно осуществить желаемое аналитическое преобразование, предварительно необходимо выделить цветом тот объект, к которому оно будет относиться. В данном случае преобразование было применено ко всему выражению $\sin(2\boldsymbol{x})+\cos(2\boldsymbol{x})$. Если же выделить часть формулы, например $\sin(2\boldsymbol{x})$, то соответствующее преобразование будет отнесено к выделенной части.

$\sin(2\cdot x) + \cos(2\cdot x)$ – исходное выражение

$2\cdot\cos(x)\cdot\sin(x) + \cos(2\cdot x)$ – результат

Рассмотрим способы символьных вычислений с использованием панели инструментов **Symbolic Keyword Toolbar**.

На символьной панели инструментов **(Symbolic Keyword Toolbar)** есть два символьных знака равенства: с одним местом ввода и с двумя местами ввода. Одно место ввода предназначено для ввода выражения или функции (встроенной или пользователя), второе для ввода ключевого слова, означающего выполняемое действие.

Напомним, что при использовании символьного знака равенства Mathcad учитывает все предыдущие присвоения значений константам и переменным и подставляет их в результат символьных вычислений.

Если значение не задано, Mathcad окрашивает данный символ в красный цвет, но символьное выражение вычисляет правильно. Поэтому желательно использовать величины, которым не присвоены численные значения.

Если это не желательно, то надо использовать меню **Symbolics**(Символика), которое игнорирует все ранее присвоенные численные значения.

По умолчанию символьный знак равенства упрощает выражение слева от знака аналогично команде **Evaluate/Sybolically** (Преобразовать символьное) из меню **Symbolics**(Символика). Если преобразование невозможно, то он возвращает исходное выражение.

Рассмотрим сначала способ символьных вычислений с использованием одноместного символьного знака равенства (с помощью оператора «→») на примере вычисления суммы $\sum_{\mathbf{n=0}}^{\infty}\frac{\boldsymbol{x}^{\mathbf{n}}}{\mathbf{n!}}$.

При использовании этого способа необходимо:

• ввести выражение $\sum_{\mathbf{n=0}}^{\infty}\frac{\boldsymbol{x}^{\mathbf{n}}}{\mathbf{n!}}$ или выделить его щелчком **ЛКМ**, если оно было введено ранее;

• ввести оператор символьного вывода «→»;

• нажать клавишу <**Enter**> либо просто щелкнуть мышью за пределами выражения – появится ответ.

$$\sum_{n=0}^{\infty}\frac{x^n}{n!} \qquad \sum_{n=0}^{\infty}\frac{x^n}{n!}\rightarrow \qquad \sum_{n=0}^{\infty}\frac{x^n}{n!}\rightarrow e^x$$

Оператор символьного вывода можно ввести в редакторе Mathcad нажатием кнопки «→» на панели **Symbolic** (Символика) либо сочетания клавиш <**Ctrl**>+<**.**>.

Ниже приведены некоторые примеры символьных вычислений с помощью одноместного оператора символьного вывода.

$$\int (x+1)^2\,dx \rightarrow \frac{(x+1)^3}{3} \quad \left[\int_a^b (x+1)^2\,dx\right] \rightarrow \frac{b^3}{3} - a^2 - a - \frac{a^3}{3} + b^2 + b$$

$$\lim_{x \to 0} \frac{\sin(x)}{x} \rightarrow 1 \qquad \frac{d}{dx}\left(x^2 + x + 1\right) \rightarrow 2\cdot x + 1$$

Пусть в документе Mathcad определена некоторая функция пользователя, например $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{x}^2 \cdot \sin(\boldsymbol{x})$.

Ниже приведены примеры символьных вычислений с этой функцией.

$$f(x) := x^2\cdot\sin(x) \qquad \int f(x)\,dx \rightarrow 2\cdot\cos(x) - x^2\cdot\cos(x) + 2\cdot x\cdot\sin(x)$$

$$(f(x))^2 \rightarrow x^4\cdot\sin(x)^2 \qquad \frac{d}{dx}f(x) \rightarrow x^2\cdot\cos(x) + 2\cdot x\cdot\sin(x)$$

Рассмотрим применение одноместного символьного знака равенства для решения уравнений и систем уравнений.

Для этих целей можно использовать функции **root(** $\boldsymbol{f(x), x, a, b}$**)** и **Find(** $\boldsymbol{x, y}$**)**.

При символьном решении эти функции не требуют начальных приближений.

Ниже приведен фрагмент документа Mathcad, иллюстрирующий применение вышеупомянутых функций для решения уравнений и систем уравнений.

$$f(z) := \frac{a\cdot z + 1}{z - b} - e^{-a}$$ – левая часть уравнения $\boldsymbol{f(z) = 0}$

$$\mathrm{root}(f(x), x, -\infty, \infty) \rightarrow -\frac{b\cdot e^{-a} + 1}{a - e^{-a}}$$ – поиск корней на всей числовой оси

$$ff(a,b) := (\mathrm{root}(f(x), x, -\infty, \infty)) \rightarrow -\frac{b\cdot e^{-a} + 1}{a - e^{-a}}$$ – функция от параметров решения

$$ff(a1,0) \rightarrow -\frac{1}{a1 - e^{-a1}} \qquad ff(0,0) = 1$$

$$f(x) := x^2 - 4\cdot x - 12$$

$$\mathrm{root}(f(x), x, -\infty, \infty) \rightarrow \begin{pmatrix} -2 \\ 6 \end{pmatrix}$$ – поиск корней на всей числовой оси

$$V := \mathrm{root}(f(x), x, -\infty, \infty) \rightarrow \begin{pmatrix} -2 \\ 6 \end{pmatrix} \qquad V = \begin{pmatrix} -2 \\ 6 \end{pmatrix}$$ – запись корней в матрицу

$$\mathrm{root}(f(x), x, 0, 10) \rightarrow 6$$ – поиск корней на отрезке [0,10]

Найдем в символах решение системы линейных уравнений

$$\begin{cases} x + 2y = a \\ 4x + y = b \end{cases}$$

Результат приведен ниже.

Given

$$x + 2\cdot y = a$$

$$4\cdot x + y = b$$

$$\mathrm{Find}(x, y) \rightarrow \begin{pmatrix} \frac{2\cdot b}{7} - \frac{a}{7} \\ \frac{4\cdot a}{7} - \frac{b}{7} \end{pmatrix}$$

Рассмотрим теперь применение команд символьного процессора для выполнения символьных преобразований.

На символьной панели инструментов (рис. 2) расположены кнопки с названием символьных операций. Все эти операции есть и в меню **Simbolics** (Символика).

При нажатии на каждую кнопку на экране появляется шаблон символьной операции с указанием ключевого слова. Назначение символьных операций приведено в табл.

Назначение символьных операций

Шаблон	Назначение команды
▪ float , ▪ →	Вычисление с плавающей точкой
▪ expand , ▪ →	Разложение степеней и произведений, сумм нескольких переменных

■ solve , ■ →	Решение уравнений и систем уравнений для указанной после запятой переменной
■ simplify , ■ →	Упрощение выражений
■ substitute , ■ = ■ →	Вычисление выражений с подстановкой определенных значений или с заменой переменных
■ collect , ■ →	Упрощение выражений представлением их суммой простых полиномов
■ assume , ■ →	Вычисления с присвоением определенных значений переменных
■ parfrac , ■ →	Разложение выражения на простые дроби
■ coeffs , ■ →	Определение вектора коэффициентов полинома
■ factor , ■ →	Разложение выражения на произведения
■ laplace , ■ →	Прямое преобразование Лапласа
■ ztrans , ■ →	Прямое z-преобразование
■ invfourier , ■ →	Обратное преобразование Фурье

Окончание табл.

Шаблон	Назначение команды
■ invlaplace , ■ →	Обратное преобразование Лапласа
■ invztrans , ■ →	Обратное z-преобразование
$M^{T} \rightarrow$	Транспонирование матрицы
$M^{-1} \rightarrow$	Обращение матрицы
$\|M\| \rightarrow$	Вычисление определителя матрицы
■ combine , ■ →	Преобразование выражений с использованием определенных функций, имена которых указываются в шаблоне справа
■ rewrite , ■ →	Представление выражений в терминах определенных функций, имена которых указываются в шаблоне справа
■ explicit , ■ →	Представление выражений с заменой переменных на их заранее определенные значения
■ confrac , ■ →	Представление выражений в виде цепной дроби
■ collect , ■ →	Представление выражений в виде полинома по степеням переменной, имя которой указано в шаблоне справа
Modifiers	Вывод набора модифицированных команд

Следует заметить, что знакоместо, которое расположено в шаблоне справа от ключевого слова, служит для уточнения параметров выполняемой операции. Поскольку все операции имеют некоторые параметры по умолчанию, то при выборе операции правое знакоместо в шаблоне в большинстве случаев не отображается. Если же необходимо изменить принятый по умолчанию параметр операции, то после ключевого слова следует поставить запятую. В результате появится правое знакоместо, в которое следует записать нужный параметр.

Таким образом, для использования команд символьной панели инструментов необходимо:

- ввести выражение, подлежащее преобразованию или выделить его щелчком **ЛКМ**, если оно было введено ранее;
- ввести шаблон необходимой операции символьного процессора;
- при необходимости уточнить параметр операции (см. выше);
- нажать клавишу **<Enter>**, либо просто щелкнуть мышью за пределами выражения – появится ответ.

Ниже приведены примеры символьных преобразований с использованием команд символьной панели инструментов.

Использование команды **«expand»** без уточняющих параметров:

$$\sin(2\cdot x)+\cos(2\cdot y)\ \text{expand}\ \rightarrow \cos(y)^2-\sin(y)^2+2\cdot\cos(x)\cdot\sin(x)$$

Без разложения слагаемого $\sin(2x)$:

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

Без разложения слагаемого $\cos(2y)$:

$$\sin(2\cdot x)+\cos(2\cdot y)\ \text{expand}, \cos(2\cdot y)\ \rightarrow \cos(2\cdot y)+2\cdot\cos(x)\cdot\sin(x)$$

Разложение функции $\sin(x)$ в ряд:

$$\sin(x)\ \text{series}\ \rightarrow x-\frac{x^3}{6}+\frac{x^5}{120}$$

Разложение функции $\sin(x)$ в ряд с требованием, чтобы максимальная степень разложения была равной семи:

$$\sin(x)\ \text{series}, 7\ \rightarrow x-\frac{x^3}{6}+\frac{x^5}{120}-\frac{x^7}{5040}$$

Разложение на множители полинома:

$$p(x) := x^3-2\cdot x^2-5\cdot x+6 \qquad p(x)\ \text{factor}\ \rightarrow (x-1)\cdot(x-3)\cdot(x+2)$$

Для решения уравнений и неравенств необходимо:

- на символьной панели инструментов выбрать команду **«solve»**, появится шаблон с двумя местами ввода;
- слева ввести уравнение с использованием жирного (логического) знака равенства;
- справа ввести имя переменной, относительно которой надо решить уравнение.

$$g(x) := x^2 - 2\cdot x - 15 \qquad g(x) = 0 \text{ solve}, x \rightarrow \begin{pmatrix} -3 \\ 5 \end{pmatrix}$$

$$g(x) < 0 \text{ solve}, x \rightarrow -3 < x < 5 \qquad g(x) > 0 \text{ solve}, x \rightarrow x < -3 \vee 5 < x$$

$$q(x,a,b) := a\cdot x^2 + b\cdot x + 1 \quad q(x,a,b) = 0 \text{ solve}, x \rightarrow \begin{pmatrix} -\dfrac{\dfrac{b}{2} + \dfrac{\sqrt{b^2 - 4\cdot a}}{2}}{a} \\ -\dfrac{\dfrac{b}{2} - \dfrac{\sqrt{b^2 - 4\cdot a}}{2}}{a} \end{pmatrix}$$

5. ПРОГРАММИРОВАНИЕ В MATHCAD

Mathcad – это система, ориентированная на пользователя, который не обязан абсолютно ничего знать о программировании. Создатели Mathcad изначально поставили перед собой такую задачу, чтобы дать возможность профессионалам-математикам, физикам и инженерам самостоятельно проводить сложные расчеты, не обращаясь за помощью к программистам. Несмотря на блестящее воплощение этих замыслов, выяснилось, что вовсе без программирования Mathcad серьезно теряет в своей силе, в основном, из-за недовольства пользователей, знакомых с техникой создания программ и желающих осуществить свои расчеты в привычном для себя программистском стиле. Последние версии Mathcad имеют не очень мощный, но весьма элегантный собственный язык. С одной стороны, он дает возможность программисту эффективно применять программный код в документах Mathcad, с другой, простота и интуитивность языка программирования позволяет быстро ему обучиться. Наконец, программные модули внутри документа Mathcad сочетают в себе и обособленность (поэтому их легко отличить от остальных формул), и простоту смыслового восприятия. Несмотря на небольшое число операторов, язык программирования Mathcad позволяет решать самые различные, в том числе и довольно сложные, задачи и является серьезным подспорьем для научно - технических расчетов.

Для создания программ в Mathcad используется панель инструментов **Programming (**Программирование**)** (рис. 5).

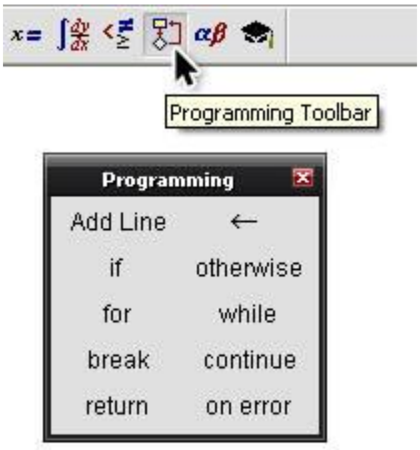


Рис. 5. Панель инструментов Programming (Программирование)

5.1. Общие сведения о программе-функции

• Известно, что реализовать тот или иной алгоритм вычисления в пакете Mathcad можно двумя способами:

• вставляя соответствующие операторы или функции в текст документа Mathcad. Такой способ называется **программированием в тексте документа**;

• используя так называемые программы-функции (или просто функции), которые содержат конструкции, во многом подобные конструкциям таких языков, как Visual Basic или Pascal: операторы присваивания, операторы циклов, условные операторы и т. д. Написание программ-функций в Mathcad позволяет решить задачи, которые невозможно решить, используя только операторы и функции Mathcad. Такой способ будем называть **программированием в программе-функции**. Такое программирование включает два этапа:

- описание программы-функции;
- вызов программы-функции.

Перед тем как использовать программу-функцию, нужно ее создать, т. е. выполнить описание. Описание функции размещается в рабочем документе перед ее вызовом и включает в себя имя функции, список формальных параметров и тело функции.

Каждая программа-функция Mathcad имеет оригинальное имя, при помощи которого осуществляется обращение к этой функции. Через это же имя (и только через это имя) функция возвращает в рабочий документ результат своей работы.

После имени функции располагается список формальных параметров, заключенный в круглые скобки. Через формальные параметры внутрь программы-функции передаются данные, необходимые для выполнения вычислений внутри программы. В качестве формальных параметров могут использоваться имена ***простых переменных***, ***массивов и функций***. Формальные параметры отделяются друг от друга запятой.

Тело программы-функции включает любое число локальных операторов присваивания, условных операторов и операторов цикла, а также вызов функций пользователя и встроенных функций Mathcad.

Для вызова функции в документе к ней необходимо обратиться по имени с указанием списка фактических параметров, т. е.

имя_переменной:=<имя_функции>(список фактических параметров)

или

имя_переменной←<имя_функции>(список фактических параметров),

если вызов функции выполняется внутри другой программы-функции.

Фактические параметры указывают, при каких конкретных значениях осуществляются вычисления в теле программы. 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 параметра 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, то и фактическим параметром должен являться тот же объект.

Обращение к функции должно находиться после ее описания и к моменту обращения фактические параметры должны быть определены.

Значение результата, возвращаемого функцией, присваивается переменной, стоящей в левой части оператора «присвоить».

5.2. Программирование линейных алгоритмов

Рассмотрим процесс создания пользовательской программы-функции на примере программирования простого линейного алгоритма (алгоритма без ветвлений и повторений).

Пусть необходимо создать функцию для вычисления значения синуса некоторого угла φ, когда он задан в градусах:

$$\boldsymbol{Sing}(\varphi) = \mathbf{sin}\left(\frac{\varphi \cdot \boldsymbol{\pi}}{\mathbf{180}}\right)$$

Для создания функции необходимо:

1) ввести имя программы и оператор «присвоить» **«:=»**;
2) нажать кнопку <**Add Line**> (Добавить линию) на панели **Programming** (Программирование) или нажать клавишу <**]**> (рис. 6) столько раз, сколько строк должна содержать программа;
3) в появившиеся места ввода ввести нужные операторы;
4) удалить лишние места ввода.

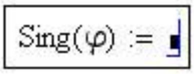
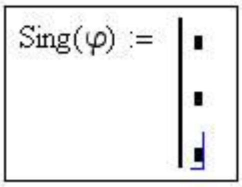
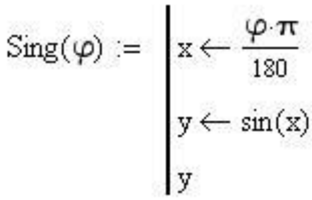
1	2	3
		

Рис. 6

Оператор «присвоить» обозначается в программе-функции символом «⟵» и вводится нажатием соответствующей кнопки на панели **Programming** (Программирование) или нажатием клавиши <**{**> на клавиатуре.

Программа-функция возвращает значение, которое записано в переменной, являющейся последней в списке операторов тела функции.

В рассматриваемом примере такой переменной является переменная y (рис. 6).

Созданная нами функция имеет имя $\boldsymbol{Sing}$, формальный параметр φ и тело, содержащее два оператора «присвоить» и переменную y для вывода результата своей работы.

Ниже показаны примеры вызова функции в документе Mathcad.

$x := Sing(30) \qquad x =$

$p := 30 \qquad z := Sing(p) \qquad z =$

$a := 15 \qquad b := 15 \qquad s := Sing(a + b) \qquad s =$

В процессе создания программы-функции можно добавлять недостающие места ввода или удалять лишние.

Для создания недостающего места ввода необходимо клавишей «Пробел» выделить всю строку в программе и нажать кнопку **<Add Line>** (Добавить линию) на панели **Programming** (Программирование). При этом возможны два варианта:

- если синий угол курсора находится в начале строки, то после нажатия кнопки **<Add Line>** место ввода появится выше этой строки;
- если синий угол курсора находится в конце строки, то после нажатия кнопки **<Add Line>** место ввода появится ниже этой строки.

Для удаления лишнего места ввода необходимо установить курсор в нужное поле и нажать клавишу **<Del>**.

Для ввода в программу комментария необходимо:

- установить курсор в нужное место ввода и нажать клавишу **<”>** (двойная кавычка), появляется курсор ввода, заключенный в двойные кавычки **(“ | ”)**;
- ввести в двойные кавычки текст комментария.

Ниже приведен текст разработанной нами функции с введенными комментариями.

Sing(φ) := | "Ïåðåâîäèì ãðàäóñû â ðàäèàíû"

$x \leftarrow \frac{\varphi \cdot \pi}{180}$

"Âû÷èñëÿåì ñèíóñ îò óãëà â ðàäèàíàõ"

$y \leftarrow \sin(x)$

"Âîçâðàùàåì ðåçóëüòàò"

y

5.3. Программирование алгоритмов с условием

Для реализации вычислений с условием на панели **Programming** (Программирование) есть два оператора: **if** (если) и **otherwise** (иначе).

Рассмотрим их применение на конкретных примерах.

Пример 1. Вычислить $y = \begin{cases} \sqrt{x}, & \text{åñëè } x \ge 0 \\ \text{íè÷åãî íå äåëàòü, åñëè} & x < 0 \end{cases}$.

Блок схема алгоритма приведена на рис. 7.

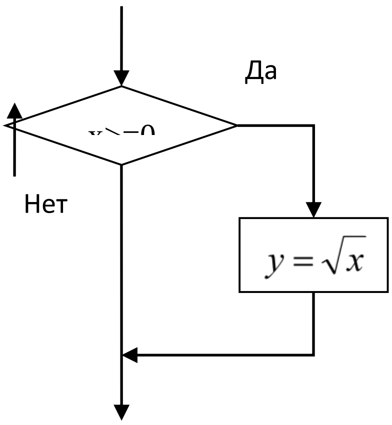


Рис. 7. Блок-схема

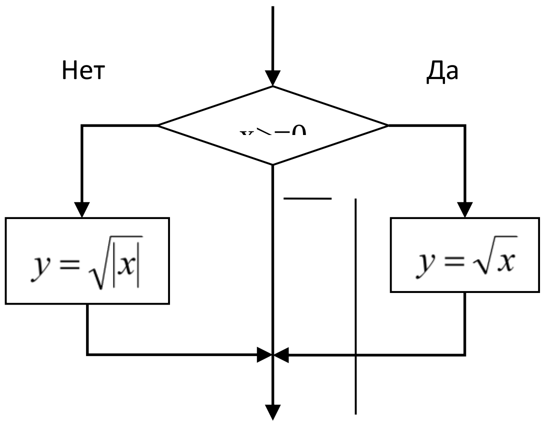


Рис. 8. Блок-схема

Ниже приведен пример программы-функции и вызов ее из документа Mathcad.

Шаблон	Шаблон с **if**	Функция
Y(x) := ▪ ▪	Y(x) := ▪ if ▪ ▪	Y(x) := $y \leftarrow \sqrt{x}$ if $x \geq 0$ y

$Y(4) = 2$ $\qquad Y(-2) = 0$ $\qquad Y(-10) = 0$

Видно, что при отрицательных значениях аргумента, когда значение переменной ***y*** внутри функции не определено, ей присваивается значение «ноль» и этот результат возвращается функцией.

Заметим, что рассматриваемый пример носит чисто иллюстративный характер по применению оператора **if**. Следует избегать ситуации, когда результат работы программы-функции неопределен.

Пример 2. Вычислить $y = \begin{cases} \sqrt{x}, & \text{åñëè } x \geq 0 \\ \sqrt{|x|}, & \text{åñëè } x < 0 \end{cases}$

Блок схема алгоритма приведена на рис. 8.

Пример программы-функции и вызов ее из документа Mathcad приводится ниже.

Шаблон с **if**	Шаблон с **if** и **otherwise**	Функция
Y(x) := ▪ i ▪ ▪	Y(x) := ▪ if ▪ ▪ other ▪	Y(x) := $y \leftarrow \sqrt{x}$ if x ≥ $y \leftarrow \sqrt{\lvert x \rvert}$ othe y

$Y(9) = 3$ $\qquad Y(-9) = 3$

Изменим теперь предыдущую функцию так, чтобы результат работы функции был определен и при неправильном значении аргумента.

Y(x) := $y \leftarrow \sqrt{x}$ if $x \geq 0$
$y \leftarrow$ "Îøèáêà, îòðèöàòåëüíûé àðãóìåíò!" otherwise
y

Y(−2) = "Îøèáêà, îòðèöàòåëüíûé àðãóìåíò!"

5.4. Программирование циклических алгоритмов

Циклические алгоритмы (циклы) – это алгоритмы, которые содержат повторяющиеся вычисления, зависящие от некоторой переменной. Такая переменная называется переменной цикла, а сами повторяющиеся вычисления составляют тело цикла.

Для реализации циклических вычислений на панели **Programming** (Программирование) предусмотрено два оператора: **for** и **while**.

Рассмотрим сначала создание оператора цикла **for**.

Для создания оператора цикла **for** необходимо:

- установить курсор в нужном месте документа и на панели программирования нажать кнопку **for**, появится шаблон с тремя местами ввода (рис. 9);
- справа от слова **for** (Поле 1) ввести имя переменной цикла, справа от знака ∈ (Поле 2) ввести диапазон изменения переменной цикла. Переменной цикла может быть: ***ряд чисел***, ***вектор***, ***список скаляров или диапазон векторов***, ***разделенных запятой***;

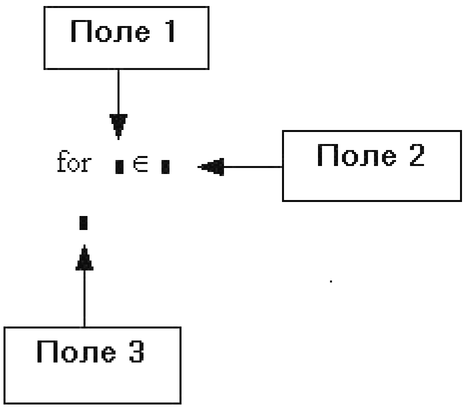


Рис. 9. Шаблон оператора цикла **for**

- в оставшееся поле ввода (внизу под словом **for**) (Поле 3) ввести выражение, которое вычисляется в цикле;
- если в цикле необходимо вычислять несколько выражений, то вначале нужно установить курсор на место ввода и нажать кнопку **<Add Line>** столько раз, сколько строк будет содержать цикл. Затем заполнить места ввода нужными выражениями и удалить лишние места ввода.

Пример 3. Создать программу-функцию для вычисления массива значений произвольной функции при изменении ее аргумента на отрезке $[\boldsymbol{a}, \boldsymbol{b}]$ с шагом $\Delta\boldsymbol{x}$.

Вид функции и вызов ее в документе Mathcad приведен ниже.

```
MasF(F,a,b,dx) := | "F - èìÿ ôóíêöèè"
                  | "a - ëåâàÿ ãðàíèöà äëÿ àðãóìåíòà"
                  | "b - ïðàâàÿ ãðàíèöà äëÿ àðãóìåíòà"
                  | "dx - øàã èçìåíåíèÿ àðãóìåíòà"
                  | N ← (b − a)/dx + 1
                  | for  i ∈ 0 .. N − 1
                  |    | x ← a + i·dx
                  |    | V_i ← F(x)
                  | V
```

$$p(x) := x^2 \quad x1 := -2 \qquad x2 := 2 \qquad \Delta x := 0.5$$

$$V := \mathrm{MasF}(p, x1, x2, \Delta x)$$

$$V^T = (\,4 \quad 2.25 \quad 1 \quad 0.25 \quad 0 \quad 0.25 \quad 1 \quad 2.25 \quad 4\,)$$

Пример 4. Создать программу-функцию для вычисления значения выражения $s = \sqrt{\sum_{i=1}^{N} V_i^2}$, где $V = (V_1, V_2, \ldots, V_N)^T$

Ниже приводится вид функции и вызов ее в документе Mathcad.

```
Func(V) := | s ← 0
           | for  x ∈ V
           |    s ← s + x^2
           | s ← √s
           | s
```

$$V := (1 \quad 2 \quad 3)^T \qquad s := Func(V) \qquad s = 3.742$$

В рассматриваемом примере переменная цикла ***x*** последовательно принимает все значения элементов вектора ***V***.

Перейдем к рассмотрению оператора цикла **while**.

Для создания оператора цикла **while** необходимо:

• установить курсор в нужном месте документа и на панели программирования нажать кнопку **while**, появится шаблон с двумя местами ввода (рис. 10);

• справа от слова **while** (Поле 1) ввести условие выполнения цикла, обычно это логическое выражение;

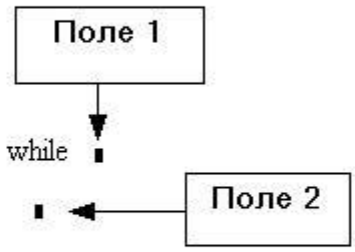


Рис. 10. Шаблон оператора цикла **while**

• в оставшееся поле ввода (внизу под словом while) (Поле 2) ввести выражение, которое вычисляется в цикле;

• если в цикле необходимо вычислять несколько выражений, то вначале нужно установить курсор на место ввода и нажать кнопку **<Add Line>** столько раз, сколько строк будет содержать цикл. Затем заполнить места ввода нужными выражениями и удалить лишние места ввода.

Рассмотрим пример функции с использованием оператора **while**.

Создадим программу-функцию по условию примера «Пример 1», но с использованием оператора **while**.

Ниже приводится вид функции и ее вызов в документе Mathcad.

```
MasF1(F,a,b,dx) := | "F - èìÿ ôóíêöèè"
                   | "a - ëåâàÿ ãðàíèöà äëÿ àðãóìåíòà"
                   | "b - ïðàâàÿ ãðàíèöà äëÿ àðãóìåíòà"
                   | "dx - øàã èçìåíåíèÿ àðãóìåíòà"
                   | i ← 0
                   | x ← a
                   | while  x ≤ b
                   |     | V_i ← F(x)
                   |     | i ← i + 1
                   |     | x ← x + dx
                   | V
```

$q(x) := x^3 \quad x1 := -2 \qquad x2 := 2 \qquad \Delta x := 0.5$

$R := MasF1(q, x1, x2, \Delta x)$

$R^T = (\,-8 \quad -3.375 \quad -1 \quad -0.125 \quad 0 \quad 0.125 \quad 1 \quad 3.375 \quad 8\,)$

5.5. Операторы управления вычислительным процессом

На панели **Programming** (Программирование) расположены кнопки для создания еще трех операторов, которые используются в программах-функциях для управления вычислительным процессом. Это операторы **return**, **break** и **continue**.

Их назначение заключается в следующем:

- оператор **return** обеспечивает досрочный выход и любого места программы-функции с возвращением результата;
- оператор **break** обеспечивает выход из цикла по некоторому условию;
- оператор **continue** обеспечивает переход к началу цикла до полного прохода его тела по некоторому условию.

Операторы **return, break** и **continue** нельзя набирать на клавиатуре, их необходимо вводить только с панели **Programming (**Программирование**)**.

Рассмотрим некоторые примеры создания программ-функций с использованием этих операторов.

Пример 5. Создать функцию для вычисления выражения

$$\boldsymbol{f(x)} = \begin{cases} \sqrt{|x|}, \text{ при } x < -5 \\ x^2, \text{ при } |x| \le 5 \\ \sqrt{x}, \text{ при } x > 5 \end{cases}$$

Ниже приводится вид функции и ее вызов в документе Mathcad.

```
F(x) :=  | return √|x|  if  x < 5
         | return √x  if  x > 5
         | return x²  otherwise
```

$F(-9) = 3 \qquad F(-1) = 1 \qquad F(16) = 4$

Пример 6. Создать функцию, которая суммирует случайные числа из отрезка $[0,1]$ и возвращает число слагаемых в сумме. Суммирование прекращается, когда текущая сумма превысит заданное значение.

Ниже приводится вид функции и ее вызов в документе Mathcad.

```
SumRnd(s) :=  | "s - çàäàííîå çíà÷åíèå ñóììû"
              | N ← 0
              | sum ← 0
              | while  1
              |     | sum ← sum + rnd(1)
              |     | N ← N + 1
              |     | break  if  sum > s
              | N
```

N := SumRnd(10)   N = 24   N := SumRnd(10)   N = 17

Пример 7. Создать функцию, которая из одномерного массива (вектора) формирует другой массив, содержащий только положительные значения первого массива. Если первый массив не содержит положительных чисел, то функция должна возвратить значение –1.

Ниже приводится вид функции и ее вызов в документе Mathcad.

$$
\mathrm{SubVec}(V) := \left|\begin{array}{l}
k \leftarrow -1 \\
\text{for } \ x \in V \\
\quad \left|\begin{array}{l}
\text{continue} \quad \text{if } \ x < 0 \\
k \leftarrow k + 1 \\
W_k \leftarrow x
\end{array}\right. \\
\text{return } \ k \ \text{ if } \ k = -1 \\
\text{return } \ W \ \text{ otherwise}
\end{array}\right.
$$

$$V := (-1 \quad 2 \quad -3 \quad -5 \quad 4)^T \qquad S := (-1 \quad -2 \quad -3 \quad -5 \quad -4)^T$$

$$W := \mathrm{SubVec}(V) \qquad W^T = (2 \quad 4)$$

$$R := \mathrm{SubVec}(S) \qquad R = -1$$

5.6. Операторы для диагностики ошибок

Оператор on error. Этот оператор является обработчиком возникающих при выполнении тех или иных вычислений ошибок и записывается в виде:

< выражение 1 > **on error** < выражение 2 >

Если при выполнении < выражение 2 > возникает ошибка, то выполняется < выражение 1>. Если ошибка не возникает, то выполняется <выражение 2>.

Ниже приводятся примеры использования этого оператора

$$q(x) := \text{"Äåëåíèå íà íîëü!"} \qquad \text{on error } \frac{1}{x}$$

$$q(2) = 0.5 \qquad\qquad q(0) = \text{"Äåëåíèå íà íîëü!"}$$

$$G(x) := \left| \begin{array}{l} y \leftarrow 2 \cdot x - 1 \\ y \leftarrow \text{"Îøèáêà: äåëåíèå íà íîëü!"} \quad \text{on error } y \leftarrow \dfrac{1}{y} \\ y \end{array} \right.$$

$G(2) = 0.333$ $\quad G(0.5) = $ "Îøèáêà: äåëåíèå íà íîëü!"

Функция error. Используется для вывода диагностических сообщений при возникновении в вычислениях ошибки и записывается в виде:

error ("< диагностическое сообщение пользователя >")

Функция **error** набирается на клавиатуре и используется в левом поле условного оператора **if**, как показано в следующем примере, использующем функцию из примера «Пример 1».

$$Y(x) := \left| \begin{array}{l} \text{error("Îøèáêà: îòðèöàòåëüíûé àðãóìåíò!")} \quad \text{if } x < 0 \\ y \leftarrow \sqrt{x} \\ y \end{array} \right.$$

$y := Y(4)$ $\quad y = 2$

$\boxed{z := Y(-4)}$ $\quad z = ▮$

Ошибка: отрицательный аргумент!

ЛИТЕРАТУРА

1. Гурский, Д. А. Вычисления в Mathcad 12 / Д. А. Гурский, Е. С. Турбина. – СПб.: Питер, 2006. 544 с.

2. Кирьянов, Д. В. Самоучитель Mathcad 11 / Д. В. Кирьянов. – СПб.: БХВ-Петербург, 2003. – 560 с.

3. Плис, А. И. Mathcad: математический практикум для экономистов и инженеров: учеб. пособие / А. И. Плис, Н. А. Сливина. – М: Финансы и статистика, 1999. – 656с.

4. Дятко, А. А. Математический пакет Mathcad 6.0 Plus: учеб. пособие / А. А. Дятко, Т. В. Кишкурно. – Минск: БГТУ, 1999. – 97 с.

5. Поршнев, С. В. Численные методы на базе Mathcad / С. В. Поршнев. – СПб.: БХВ-Петербург, 2005. – 464 с.: ил.

6.Васильев, А. Н. Mathcad 13 на примерах / А. Н. Васильев. -– СПб.: БХВ-Петербург, 2006. – 528 с.: ил.

7. Бертяев, В. Д. Теоретическая механика на базе Mathcad / В. Д. Бертяев. -– СПб.: БХВ-Петербург, 2005. – 752 с.: ил.

8. Охорзин, В. А. Прикладная математика в системе Mathcad: учеб. пособие. 2-е изд., испр. и доп. / В. А. Охорзин. -– СПб.: Издательство «Лань», 2008. – 352 с.: ил.

СОДЕРЖАНИЕ