Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный технический университет имени П.О. Сухого»

Кафедра «Информационные технологии»

РЕШЕНИЕ ЗАДАЧ В СИСТЕМЕ MATHCAD

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к контрольным работам по курсу «Информатика» для студентов технических специальностей заочной формы обучения

Гомель 2006

УДК 681.3.06(075.8)
ББК 32.81я73
Р47

*Рекомендовано научно-методическим советом
факультета автоматизированных информационных систем
ГГТУ им. П. О. Сухого
(протокол № 6 от 17.05.2005 г.)*

Авторы-составители: *Т. А. Трохова, Т. Л. Романькова, И. В. Стрижак*

Рецензент: канд. техн. наук, доц. каф. МИУС Бел ГУТ *М. Л. Шишаков*

Р47 **Решение** задач в системе MATHCAD : метод. указания к контрол. работам по курсу «Информатика» для студентов техн. специальностей заоч. формы обучения / авт.-сост.: Т. А. Трохова, Т. Л. Романькова, И. В. Стрижак. – Гомель : ГГТУ им. П. О. Сухого, 2006. – 54 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: http://gstu.local/lib. – Загл. с титул. экрана.

Методические указания содержат два раздела, представленных в виде тем. Каждая тема включает краткие теоретические сведения по методике решения задач данной темы. В практической части описан пошаговый алгоритм решения задач, приведены примеры в виде документов системы MATHCAD.

Для студентов технических специальностей заочной формы обучения.

**УДК 681.3.06(075.8)
ББК 32.81я73**

Раздел 1 Общая характеристика системы MathCad. Интерфейс MathCad

1.1 Общая характеристика и основные функции системы

Система MathCad - одна из самых мощных и эффективных систем математического направления, которая ориентирована на широкий круг пользователей и позволяет выполнять математические расчеты как в численном, так и в символьном виде. Причем, описание решения задач задается с помощью привычных математических формул и знаков.

Функциональный набор системы включает в себя:

- вычислительные функции (вычисление арифметических выражений, производных, интегралов, вычисление суммы и произведения, решение уравнений, неравенств и их систем, решение дифференциальных уравнений, обработка матриц, использование символьных преобразований и др.);
- графические функции (построение двухмерных графиков в различных системах координат, построение графиков поверхностей, векторных полей, трехмерных гистограмм, применение элементов анимации);
- функции программирования (создание программных модулей, состоящих из программных элементов, подобных конструкциям языков программирования);
- сервисные функции (ведение диалога с пользователем посредством меню, пиктограмм или команд, размещение на экране и редактирование математических, графических и текстовых конструкций, форматирование документа, печать документа и др.).

В настоящем практическом пособии рассматривается система MathCad 2001i Professional, но почти все приведенные здесь сведения можно успешно использовать при работе с другими версиями системы.

1.2 Интерфейс системы MathCad

После запуска системы на экране компьютера появляется заставка MathCad, а затем окно системы, показанное на рисунке 1.1.

Основное окно системы содержит следующие функциональные области:

- строка заголовка (первая строка, содержащая имя рабочего документа и стандартные кнопки управления окном);
- главное меню системы (вторая строка, включающая пункты иерархического меню, которое содержит полный набор команд работы с системой);
- панель инструментов «Стандартная» (третья строка, содержащая кнопки или пиктограммы, дублирующие наиболее важные пункты главного меню);

- панель инструментов «Форматирование» (четвертая строка, содержащая кнопки переключения вида, размера и стиля шрифтов, выравнивания текста и др.);

- наборная панель (пятая строка, содержащая набор кнопок для вывода на экран дополнительных окон с палитрами математических символов и операторов);

- окно набора и редактирования документа (основная часть окна системы).

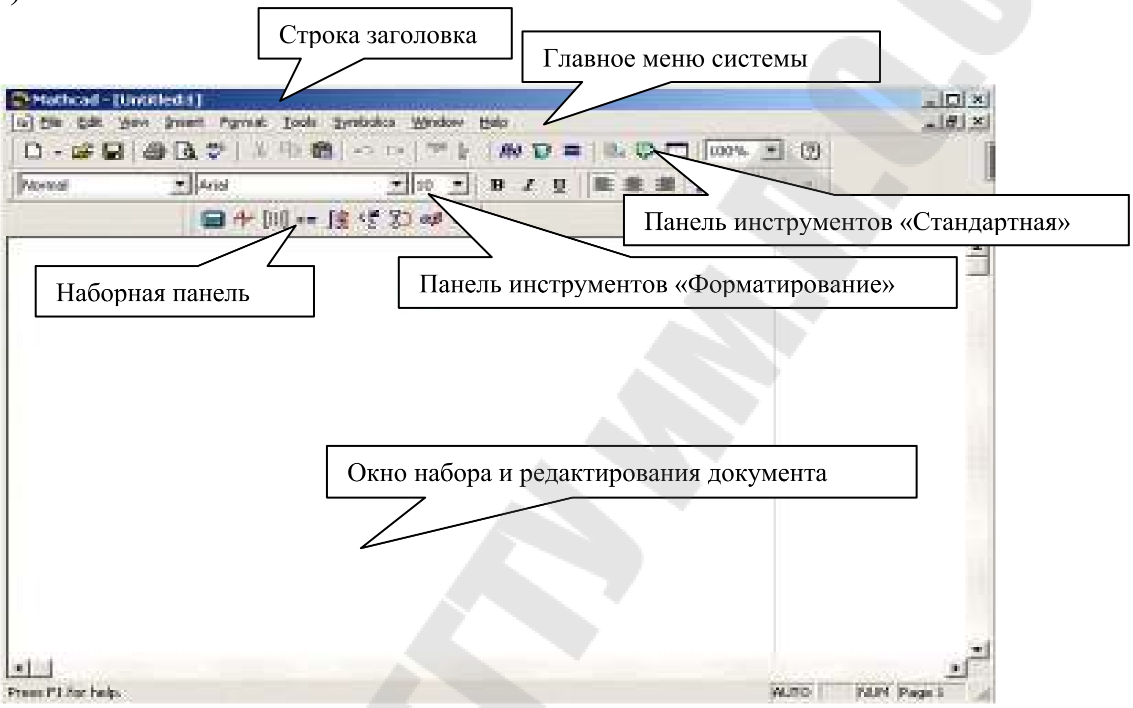


Рисунок 1.1 – Вид окна MathCad после запуска системы

Главное меню системы имеет иерархическую структуру, т.е каждый его пункт содержит дополнительное подменю. Для активизации любого пункта меню достаточно навести на него указатель мыши и щелкнуть ее левой клавишей. При этом на экран выводится ниспадающее подменю. В дальнейшем изложении команда подменю будет обозначаться следующим образом:

Команда главного меню – Команда соответствующего подменю.

Например, команда **File – New…** означает, что сначала в главном меню системы активизируется пункт **File,** после чего в появившемся подменю выбирается пункт **New…**

Работа с документами MathCad обычно не требует использования команд главного меню, так как основные из них дублируются кнопками на панелях инструментов.

Если одна из панелей инструментов отсутствует, вывести ее на экран можно с помощью команды главного меню **View – Toolbars (Просмотр – Панели инструментов)**.

Сообщение о назначении кнопок, размещенных на панелях инструментов и на наборной панели, можно получить, установив указатель мыши на соответствующую пиктограмму.

На рисунке 1.2 приведены основные палитры математических символов и операторов, позволяющие вводить общепринятые математические знаки, операторы программирования, шаблоны графиков и т.д. в месте расположения курсора.

Окна с палитрами появляются в окне редактирования документа при активизации соответствующих пиктограмм. Любую палитру можно переместить в удобное место экрана, наведя указатель мыши на заголовок окна палитры и перетаскивая палитру при нажатой левой клавиши мыши. Если палитра больше не нужна, ее можно закрыть с помощью соответствующей системной кнопки в правом верхнем углу окна палитры.

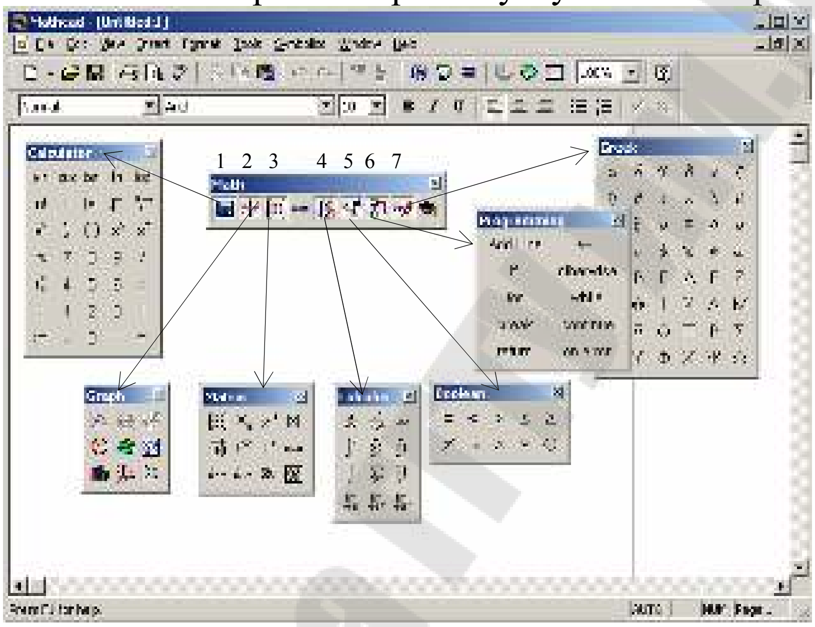


Рисунок 1.2 – Назначение кнопок наборной панели:

1 – **Calculator** (Палитра арифметических операторов);
2 – **Graph** (Палитра графиков);
3 – **Matrix** (Палитра матриц);
4 - **Calculus** (Палитра математического анализа);
5 - **Boolean** (Палитра логических операторов);
6 - **Programming** (Палитра программирования);
7 - **Greek symbol** (Палитра греческих букв).

Для вывода с помощью палитр необходимого объекта нужно поместить курсор ввода (красный крестик) в желаемое место окна редактирования и затем активизировать пиктограмму нужного объекта.

1.3 Основные приемы работы с файлами в системе MathCad

1.3.1 Создание нового документа

Сразу после запуска системы открывается окно редактирования для создания документа с именем Untitled:N, где N – порядковый номер документа (он начинается с 1) .

Если нужно создать еще один новый документ, используется команда главного меню

File – New…   (Файл – Новый…)

или кнопка на панели инструментов «Стандартная».

При этом на экран выводится текущее окно нового документа и система переходит в режим редактирования.

1.3.2 Сохранение документа в файле на диске

Для записи документа на диск с использованием его текущего имени и учетом всех произведенных изменений используется команда главного меню

File – Save   (Файл – Сохранить)

или кнопка на панели инструментов «Стандартная».

Если документ сохраняется в первый раз, на экран выводится диалоговое окно, показанное на рисунке 1.3.

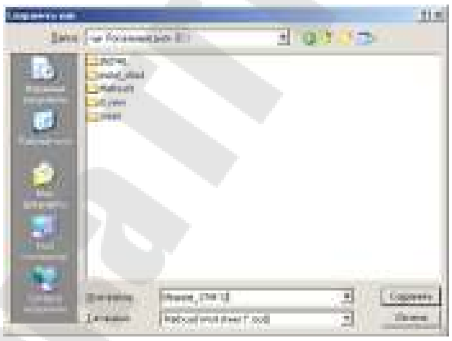


Рисунок 1.3 – Окно сохранения документа

В этом окне необходимо указать имя файла, в котором будет сохранен документ, выбрать папку для размещения этого файла и нажать кнопку **Сохранить**.

Для записи текущего документа на диск под другим именем используется команда меню

File – Save as…   (Файл – Сохранить как…)

При этом на экран выводится окно сохранения файла, изображенное на рисунке 1.3. После ввода нового имени файла и нажатия кнопки **Сохранить** отредактированный документ записывается на диск под новым именем, а файл-оригинал остается неизмененным.

1.3.3 Открытие ранее созданного документа

Для загрузки ранее сохраненного документа в окно редактирования используется команда меню

File – Open…   (Файл – Открыть…)

или кнопка на панели инструментов «Стандартная».

При исполнении этой команды появляется диалоговое окно открытия файла, в котором нужно указать требуемый файл и нажать кнопку **Открыть.** Для выбора папки, в которой размещается нужный файл, необходимо раскрыть список папок, щелкнув по кнопке со стрелкой справа от поля **Папка.**

1.3.4 Печать текущего документа

Для запуска печати текущего документа используется команда меню

File – Print…   (Файл – Печать…)

или кнопка на панели инструментов «Стандартная».

При выполнении этой команды появляется диалоговое окно, показанное на рисунке 1.4. В этом окне нужно установить все необходимые параметры печати:

- выбрать тип принтера из списка, раскрывающегося после щелчка по кнопке со стрелкой рядом с полем **Name;**

- на панели **Print Region** выбрать часть документа, которую требуется напечатать: *All* – весь документ, *Pages from N1 to N2* – часть документа со страницы N1 до страницы N2, *Selection* – выделенный фрагмент документа, *Current Page* – текущую страницу.

После нажатия кнопки **OK** будет произведена печать документа на выбранном принтере.

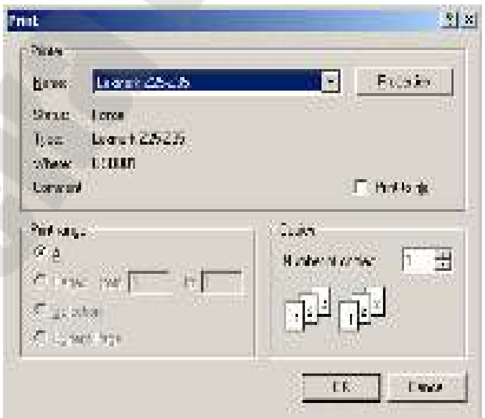


Рисунок 1.4 – Окно печати документа

Перед печатью документа рекомендуется выполнить предварительный просмотр, чтобы убедиться, что все объекты документа удачно размещены на его страницах. Это можно сделать с помощью команды меню

File – Print Preview   (Файл – Предварительный просмотр)

или с помощью кнопки на панели инструментов «Стандартная».

Раздел 2 Методические указания к выполнению контрольной работы

Тема 1. Базовые вычисления в MathCad
Краткие теоретические сведения

Документ системы MathCad строится из областей, которые делятся на вычислительные, графические, текстовые. Области создаются средствами формульного, текстового и графического редакторов системы.

Документ обрабатывается сверху вниз, а в пределах строки слева направо.

В вычислительных областях можно задавать данные, выражения, операторы и управляющие структуры.

Все данные системы можно разделить на простые и структурированные.

Простые данные представлены константами и переменными.

Структурированные данные представлены дискретными переменными, массивами и файлами.

Константы – элементы данных, хранящие некоторые значения, которые не могут быть изменены. В MathCad используются:

- целые константы (123 -56 9000)
- вещественные (4.6 -98.56 $1.2 \cdot 10^{13}$)
- комплексные (5+2.3i 9.4 – 3i)
- зарезервированные константы (e, π, %);
- строковые константы ("пример")

Переменные – поименованные объекты, имеющие некоторые значения, которые могут изменяться в процессе выполнения документа.

Имена переменных в системе MathCAD могут содержать любые латинские и греческие буквы, а также цифры, они должны начинаться только с буквы. Строчные и прописные буквы в именах различаются. Имена должны быть уникальными, т. е. они не должны совпадать с именами встроенных или определенных пользователем функций. Примеры имен переменных:

A f k21 sum γ φ5

Выражение – это совокупность данных, функций и математических объектов, связанных знаками операций. Выражения могут содержать скобки.

Операции, используемые в выражениях, можно разделить на арифметические и логические. Арифметические операции представлены в палитре арифметических операторов. Логические операции и операции отношения представлены в палитре логических операторов.

К ***базовым операторам*** системы относятся:

:= – оператор локального присваивания,

≡ – оператор глобального присваивания,

= – оператор вычисления и вывода.

Оператор локального присваивания (:=) распространяет свое действие на область документа, расположенную в строке и ниже места присваивания. Этот оператор выполняется так: данному, стоящему в левой части оператора присваивается вычисленное значение выражения, стоящего в правой части оператора.

Оператор глобального присваивания (≡) не зависит от места присвоения и распространяет свое действие на весь документ. Этот оператор выполняется точно так же, как и оператор локального присваивания.

Оператор вычисления и вывода (=) выводит вычисленное значение выражения, стоящего в его левой части, на экран.

Функция – выражение, согласно которому проводятся некоторые вычисления с ее аргументами и определяется числовое значение. Функция имеет имя и может иметь список параметров. Различают стандартные и пользовательские функции.

Система MathCAD содержит большое количество стандартных функций, которые делятся на математические, функции обработки векторов и матриц, статистические и т.д.

К основным ***математическим функциям*** относятся следующие:

- тригонометрические – ***sin(z), cos(z), tan(z), cot(z), csc(z), sec(z)****;*
- обратные тригонометрические – ***asin(z), acos(z), atan(z)****;*
- показательные и логарифмические – ***exp(z), ln(z), log(z)****.*

Аргументы тригонометрических функций задается в радианах. Обратные тригонометрические функции возвращают результат в радианах.

Для работы с пользовательскими функциями нужно сначала задать вид функции (описать ее), а затем обращаться к ней нужное количество раз для вычисления результатов.

Общий вид описания функции следующий:

ИМЯ(СФП):=выражение

где ИМЯ – имя функции; СФП – список формальных параметров функции.

При обращении к функции формальные параметры заменяются на фактические, т.е. на выражения, имеющие числовые значения. Например,

$z(m,n):=m^2 + n^2$ - описание функции,

$z(2,3) = 13$ – обращение к функции.

В MathCAD существует понятие объекта, т.е. части области рабочего документа, над которой можно произвести действия редактирования и форматирования. Визуально объект выделяется синей рамкой, которая может быть расширена с использованием клавиши «пробел».

На рисунках 2.1.1, 2.1.2 приведены примеры констант, переменных и базовых операторов MathCad.

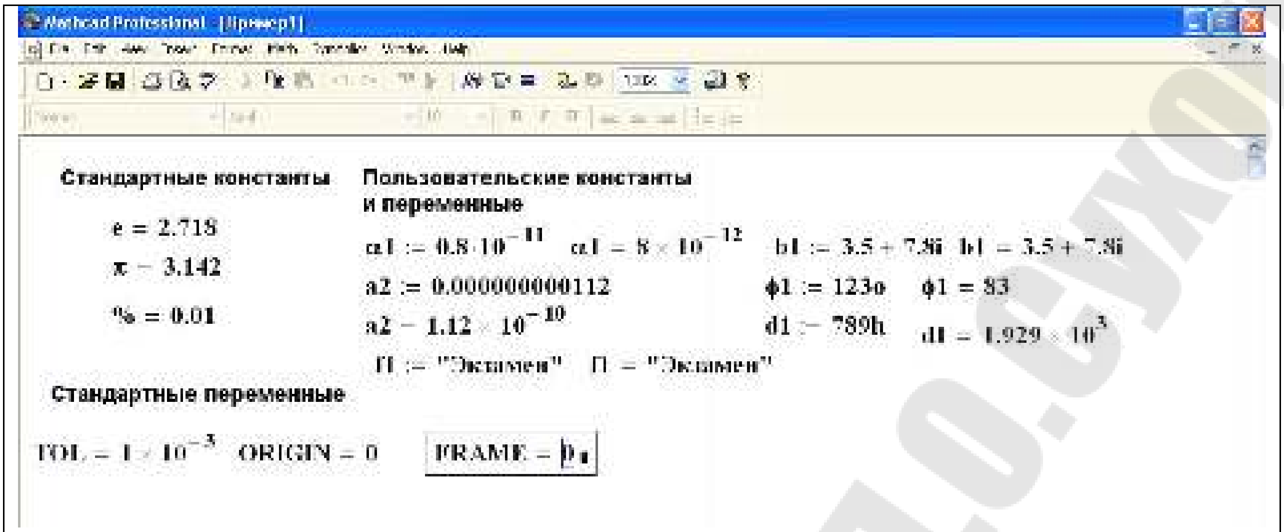


Рисунок 2.1.1 – Примеры констант и переменных

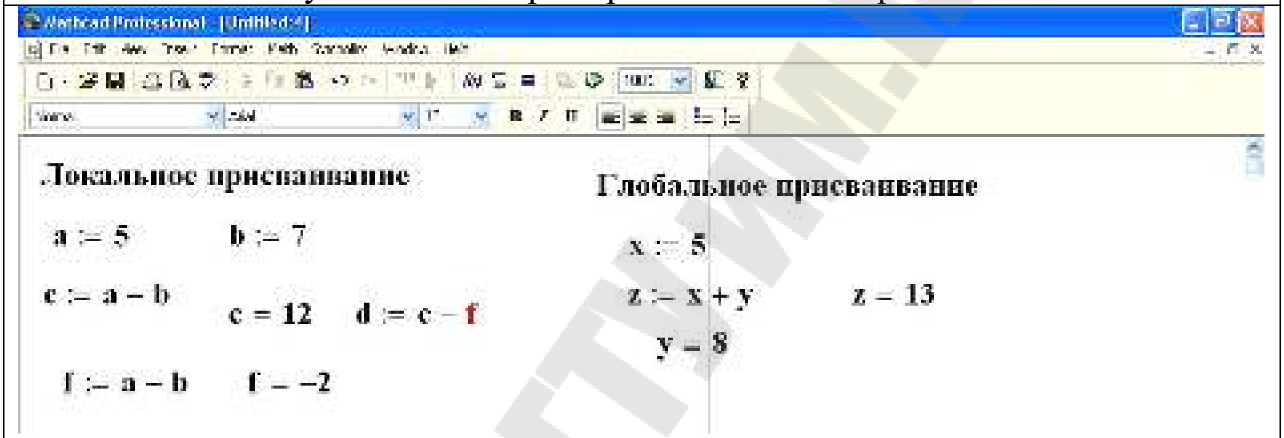


Рисунок 2.1.2 – Примеры базовых операторов

На рисунке 2.1.2 показано, что значение переменной ***d*** не может быть вычислено, т.к. значение переменной f определено в документе ниже и при вычислении переменной ***d*** считается неизвестным. Значение переменной ***y*** так же определено в документе ниже, чем оно используется при вычислении переменной ***z***, но у определено с помощью оператора глобального присваивания, который распространяет свое действие на весь документ и, следовательно, ошибки при вычислении ***z*** не возникает.

Практическая часть темы 1

1.1 Вычисление арифметических выражений

Последовательность действий для вычисления простого арифметического выражения такова:

- установить курсор в свободное место рабочего окна документа;
- открыть палитру арифметических операторов;
- используя кнопки палитры, набрать арифметическое выражение (имена переменных набираются с клавиатуры);
- при наборе воспользоваться понятием объекта (см. краткие теоретические сведения темы 1);

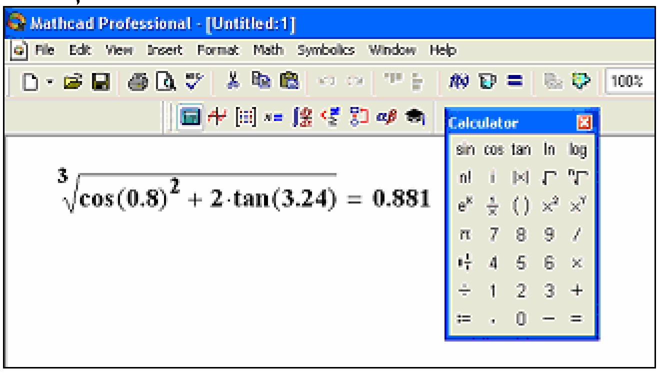
- с палитры инструментов или клавиатуры ввести оператор «=».

MathCad произведет вычисление арифметического выражения и выдаст результат в числовом виде.

Пример 1.1 Вычислить значение числового арифметического выражения

$$\sqrt[3]{\cos^2(0.8) + 2 \cdot tg(3.24)}$$

Реализация в MathCad:



1.2 Вычисление арифметических выражений с использованием оператора присваивания (линейный алгоритм)

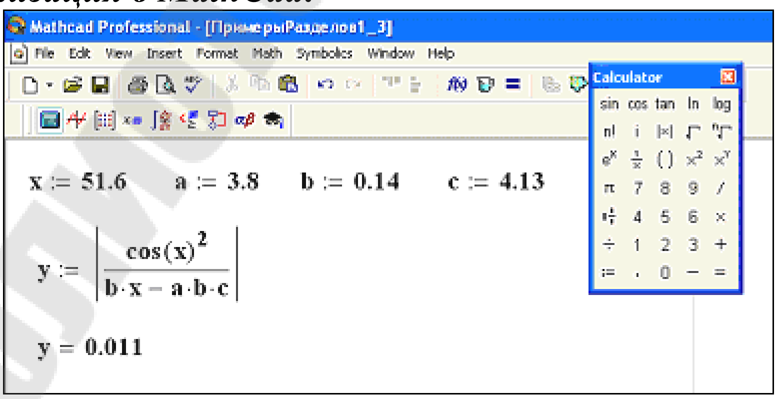
Последовательность действий для вычисления арифметического выражения с использованием оператора присваивания такова:

- установить курсор в свободное место рабочего окна документа;
- выделить в задаче исходные данные, например, a, b и поместить в эти переменные заданные числовые значения с помощью оператора «:=»;
- с помощью этого же оператора поместить значение арифметического выражение в результирующую переменную, например, в y;
- получить значение результирующей переменной в числовом виде с помощью оператора «=».

Пример 1.2 Вычислить значение функции $y = \left|\dfrac{\cos^2 x}{b \cdot x - a \cdot b \cdot c}\right|$, при

$x = 51.6;\quad a = 3.8;\quad b = 0.14\quad c = 4.13$.

Реализация в MathCad:



1.3 Создание пользовательских функций

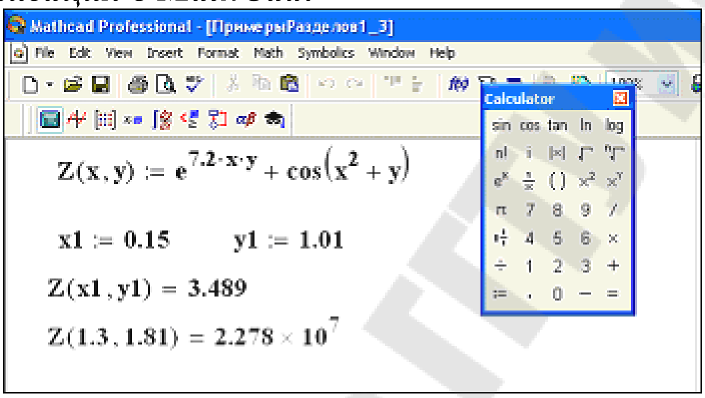
Последовательность действий для нахождения значения функции в заданных точках такова:

- установить курсор в свободное место рабочего окна документа;
- с помощью оператора присваивания описать пользовательскую функцию с параметрами, например Z(x,y);
- выделить в задаче исходные данные, например, x1, x2, y1, y2 и поместить в них заданные числовые значения с помощью оператора «:=»;
- обратиться к описанной функции, подставив вместо параметров заданные переменные с конкретными числовыми значениями и вывести результат с помощью оператора «=».

Получить значение функции можно, подставив вместо формальных параметров конкретные числа, а не заранее заданные переменные.

Пример 1.3 Создать функцию $Z(x,y)=e^{7.2\cdot x\cdot y}+\cos(x^2+y)$ и вычислить ее значение в двух заданных точках (0.15, 1.01), (1.3,1.81).

Реализация в MathCad:



В этом примере параметры функции Z(x,y) задаются через переменные x1, y1 при первом обращении к функции, и через конкретные числа – при втором обращении к функции.

<u>*Тема 2.*</u> Обработка структурированных данных в MathCad

Краткие теоретические сведения

Дискретной называется переменная, содержащая несколько значений, изменяющихся от начального до конечного на величину постоянного шага. Дискретная переменная может быть задана двумя способами:

1) a := a1, a2 .. an

2) a := a1 .. an

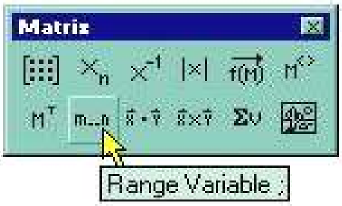
где **a** – имя дискретной переменной,

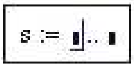
a1 – ее начальное значение,

a2 – ее второе значение,

an – ее конечное значение.

Символ «**..**» набирается либо клавишей «**;**» на клавиатуре, либо кнопкой $\boxed{\boldsymbol{m..n}}$ – в палитре матриц.





Для первого способа задания дискретной переменной шаг ее изменения равен (a2 – a1). Для второго способа задания дискретных переменных значение a2 не указывается, шаг изменения дискретной переменной равен 1, если a1<an, или -1, если a1>an.

Примеры создания дискретных переменных приведены ниже.

$x := 2..\ 7$	Создается дискретная переменная x, значения которой изменяются от 2 до 7 с шагом 1.
$y := 2, 2.3..\ 7$	Создается дискретная переменная y, значения которой изменяются от 2 до 7 с шагом 0.3.
$z := 9\ ..\ 1$	Создается дискретная переменная z, значения которой изменяются от 9 до 1 с шагом -1.
$a := 8, 7.9..\ 3$	Создается дискретная переменная a, значения которой изменяются от 8 до 3 с шагом -0.1.
$x := 0, \frac{\pi}{10} .. 2 \cdot \pi$	Создается дискретная переменная b, значения которой изменяются от 0 до 2π с шагом π/10.

Дискретные переменные могут являться аргументами функций, тогда процесс вычисления значений функции приобретает циклический характер, и для каждого значения дискретной переменной вычисляется свое значение функции по заданной аналитической зависимости.

В системе MathCAD в основном используются массивы двух типов: одномерные (векторы) и двумерные (матрицы).

Каждый элемент вектора или матрицы имеет порядковый номер в массиве. Отсчет номеров начинается с того значения, которое содержится в системной переменной ***ORIGIN***. По умолчанию эта переменная имеет значение 0, для изменения значения нужно задать, например,

ORIGIN:=1

Векторы и матрицы можно задавать различными способами: с помощью кнопки с изображением матриц на наборной панели математических инструментов; как переменную с индексами перечислением элементов массива с разделение запятой; с помощью аналитического выражения.

Массивы могут использоваться в выражениях целиком или поэлементно. Для обращения к элементам массивов нужно указать числовые значения индексов элементов в подстрочнике после имени массива. При выполнении расчетов можно обращаться к конкретной строке или столбцу матрицы с помощью верхнего индекса или нижних индексов.

На рисунке 2.2.1 приведены примеры создания массивов перечислением элементов (вектор R) и аналитически (вектор Z). Здесь же показано, как обратиться к элементу матрицы, ее столбцу или строке. Из рисунка видно, что после изменения значения переменной ORIGIN, значение элемента матрицы $M_{2,1}$ тоже изменяется.

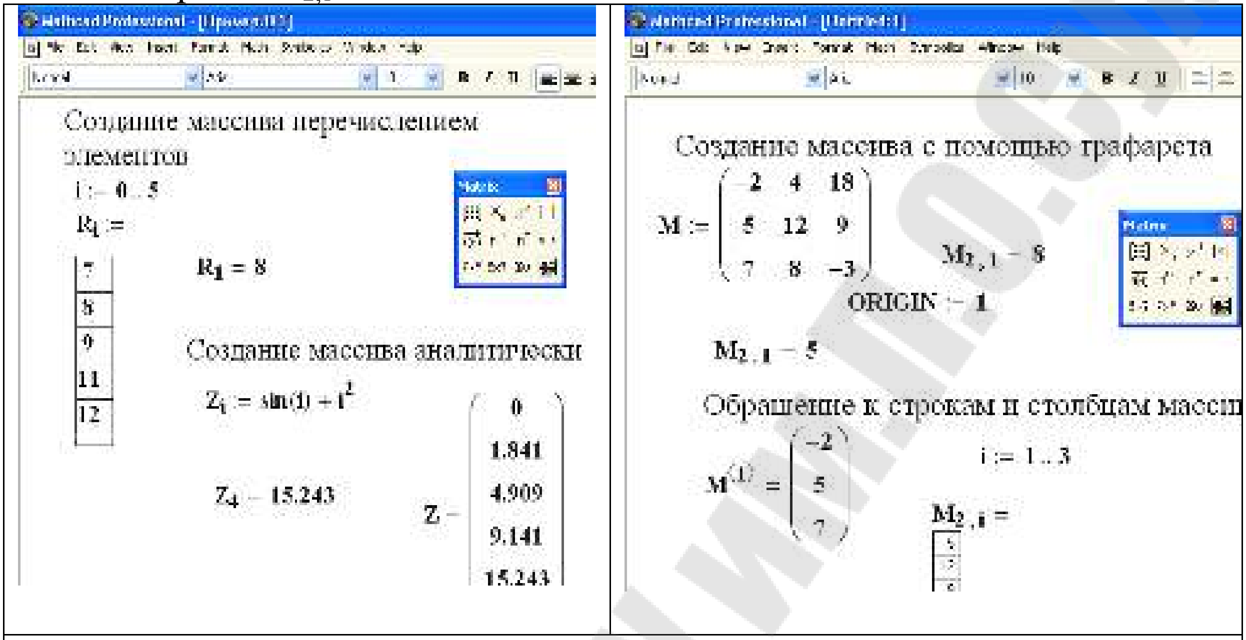
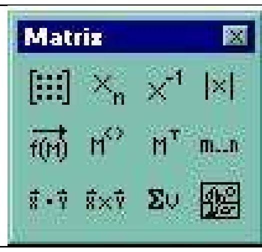


Рисунок 2.2.1 – Примеры создания массивов

Существует ряд операций над матрицами и векторами, а также встроенных векторных и матричных функций. Введем следующие обозначения: V – вектор, M – матрица. Основные операции с их назначением и правилами набора приведены в таблице 2.2.1.

Таблица 2.2.1. – Основные операции и функции для обработки массивов

Вид операции	**Назначение**	Набор
$\lvert M \rvert$	определитель матрицы	
M^{-1}	обращение матрицы	
M^{T}	транспонирование матрицы	
$M^{<>}$	выделение столбца матрицы	
$\overrightarrow{(\mathbf{V1 \cdot V2})}$	поэлементное умножение векторов	
M·V	умножение матрицы на вектор	Знак умножения набирается с палитры арифметических операторов
M1·M2	умножение двух матриц	
V1·V2	умножение двух векторов	
max(M), min(M)	максимум, минимум матрицы	Стандартные функции набираются с клавиатуры или с использованием мастера функций
cols(M), rows(M)	число столбцов и строк матрицы	

Практическая часть темы 2

2.1 Дискретные переменные, функции дискретных переменных

Определение и правила записи дискретных переменных описаны в кратких теоретических сведениях темы 2.

Последовательность действий для создания ***дискретной переменной*** такова:

<u>1-й способ</u>

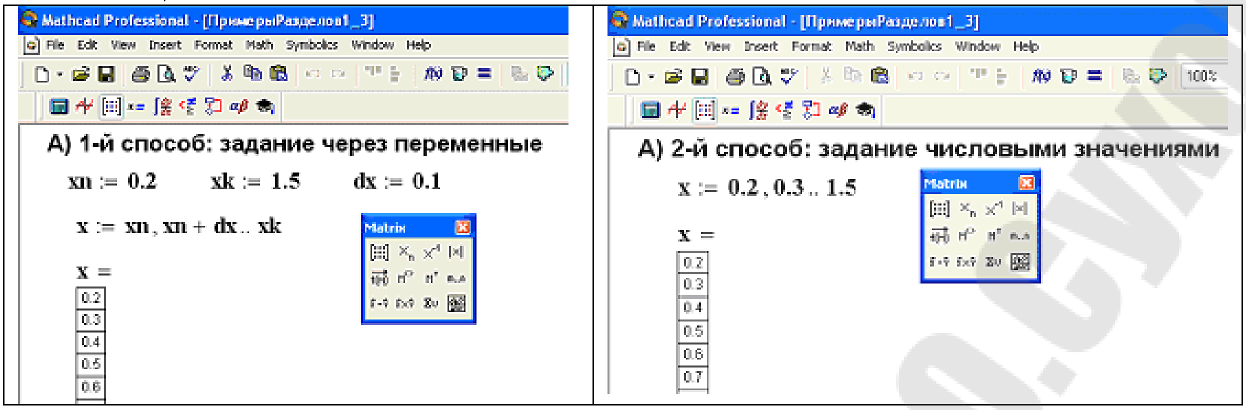
- установить курсор в свободное место рабочего окна документа;
- с помощью оператора «:=» присвоить числовые значения переменным - xn (начальное значение дискретной переменной), xk (ее конечное значение) и dx (шаг изменения дискретной переменной);
- задать имя дискретной переменной, например, ***x***; задать оператор «:=»;
- в качестве начального значения набрать xn;
- через запятую ввести второе значение дискретной переменной xn+dx;
- с помощью кнопки $\boxed{\textbf{m..n}}$ или клавиши «;» задать знак диапазона дискретной переменной;
- задать конечное значение дискретной переменной xk;
- получить значение дискретной переменной в виде таблицы с помощью оператора «=».

<u>2-й способ</u>

- установить курсор в свободное место рабочего окна документа;
- задать имя дискретной переменной, например, ***x***; задать оператор «:=»;
- в качестве начального значения дискретной переменной набрать, например, 0.2;
- через запятую в качестве второго значения дискретной переменной набрать, например, 0.3;
- с помощью кнопки
- $\boxed{\textbf{m..n}}$ или клавиши «;» задать признак дискретной переменной – «..»
- задать конечное значение дискретной переменной, например, 1.5;
- получить значение дискретной переменной в виде таблицы с помощью оператора «=».

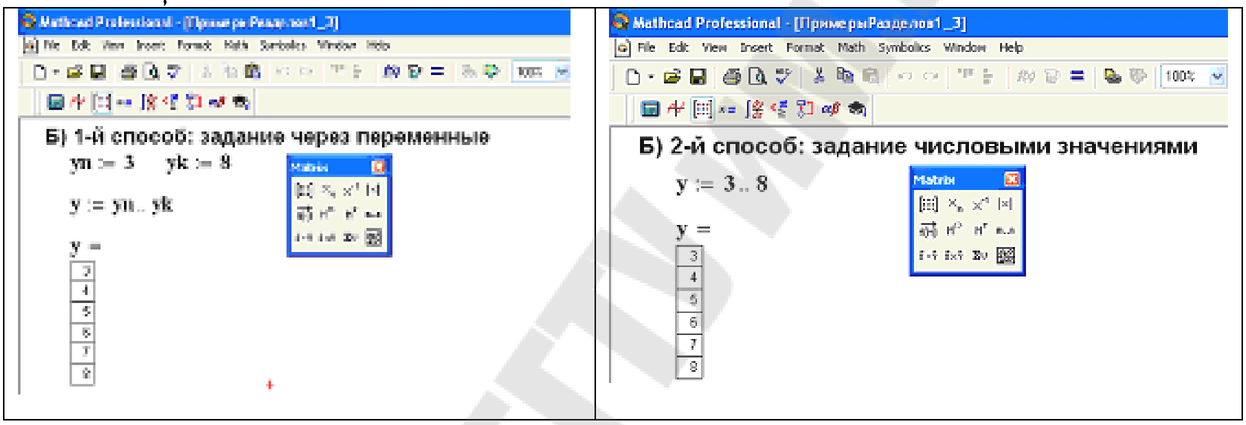
Пример 2.1. А) Определить дискретную переменную x, которая изменяется от 0.2 до 1.5 с шагом 0.1.

Реализация в MathCad:



Б) Определить дискретную переменную ***y***, которая изменяется от 3 до 8 с шагом 1.

Реализация в MathCad:



Пример 2.2. 1) Создать дискретную переменную ***x***, изменяющуюся от 1 до 1.8 и вычислить значение функции дискретной переменной $Z(x) = x^2 \cdot \sqrt{4 - x^2}$. Значение шага выбрать так, чтобы переменная имела не менее 10-15 значений.

2) Создать одномерный массив (вектор) из любых чисел, подобранных самостоятельно из диапазона изменения дискретной переменной. Массив должен содержать не менее 10 чисел. Сформировать новый одномерный массив (вектор), содержащий значения функции от элементов исходного вектора.

Для решения первой части задания необходимо:

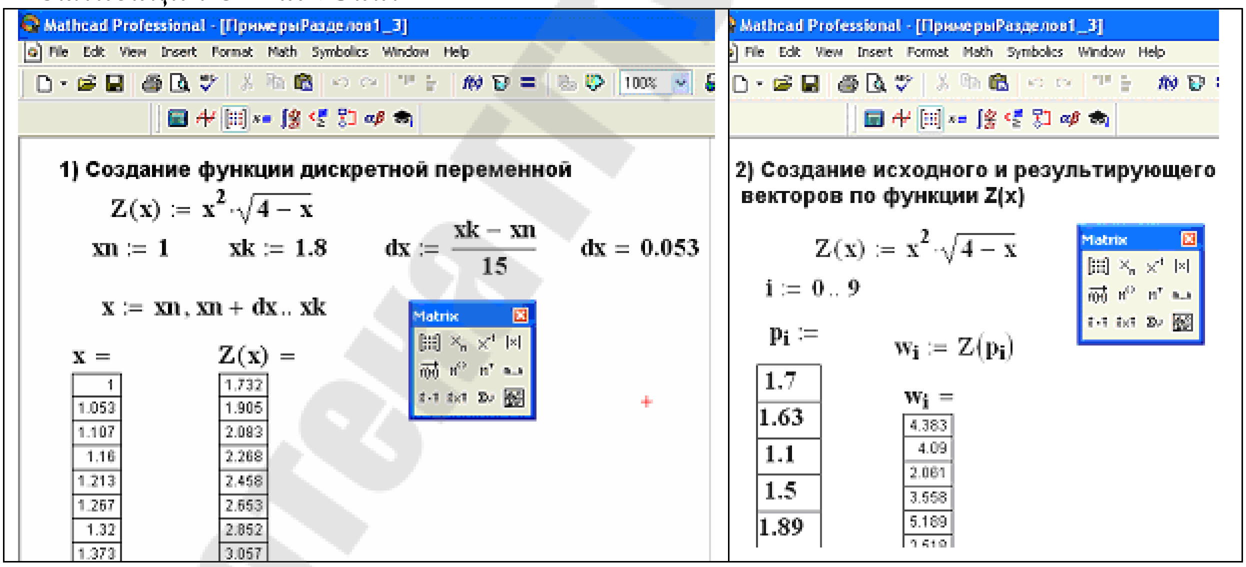
- установить курсор в свободное место рабочего окна документа;
- с помощью оператора присваивания описать пользовательскую функцию с параметром, например Z(x);
- создать дискретную переменную ***x***, как описано в примере 2.1;
- получить значение дискретной переменной в виде таблицы с помощью оператора «=»;
- получить значение функции от дискретной переменной в виде таблицы с помощью оператора «=».

Для решения второй части задания необходимо ввести следующие новые переменные для обозначения векторов: p – исходный вектор, содержащий 10 значений; w – результирующий вектор, полученный из исходного по функции Z.

Вектор ***p*** задается как переменная с индексом:

- создать дискретную переменную, которая будет управлять номером элемента в массиве, например, ***i*** (переменная должна содержать столько значений, сколько чисел должен содержать массив);
- создать переменную с индексом, например, p_i, для перехода на нижний уровень при задании индекса используется кнопка $\boxed{x_n}$ на панели матриц;
- с помощью оператора «:=» поместить в каждый элемент массива p_i числовые значения, разделяя их запятыми; после задания первого числа и запятой курсор перемещается вниз, и массив принимает вид столбца таблицы, в который заносятся новые числа;
- вывести полученный вектор ***p***;
- сформировать новый вектор ***w*** **,** присвоив переменной с индексом w_i значение функции Z с элементом p_i в качестве параметра;
- вывести вычисленные значения вектора ***w*** с помощью оператора «=».

Реализация в MathCad:



2.2 Обработка матриц и векторов

Последовательность действий для создания массивов с числовыми значениями и выполнения операций над ними такова:

- установить курсор в свободное место рабочего окна документа;
- набрать имя массива;
- после имени записать оператор «:=»;
- открыть палитру матриц;
- выбрать кнопку с изображением шаблона матрицы;

- в появившемся окне размерности матрицы ввести целочисленные значения количества строк и столбцов матрицы;
- заполнить полученный шаблон вектора или матрицы исходными данными;
- последовательно набрать вычислительные области с нужными операциями над матрицами и векторами согласно таблице 2.2.1 теоретических сведений к теме 2.

Пример 2.3. Даны две матрицы

$$A=\begin{pmatrix}2 & -1 & -3\\ 8 & -7 & -6\\ -3 & 4 & 2\end{pmatrix} \quad и \quad B=\begin{pmatrix}2 & -1 & -2\\ 3 & -5 & 4\\ 1 & 2 & 1\end{pmatrix}.$$

А) Создать вектор V1 из второго столбца матрицы А и вектор V2 из третьего столбца матрицы В.

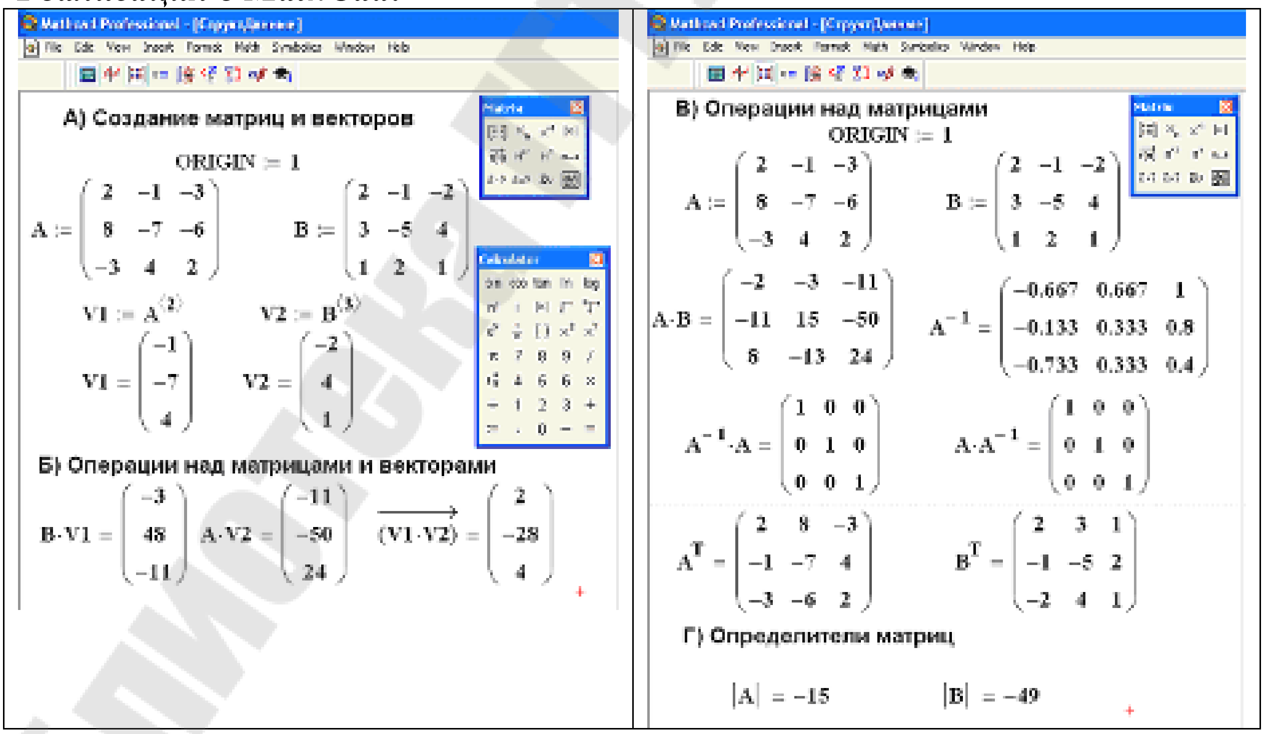
Б) Вычислить $V1\cdot V2$, $B\cdot V1$, $A\cdot V2$, $\overrightarrow{V1\cdot V2}$.

В) Вычислить $A\cdot B$, A^{-1}, $A^{-1}\cdot A$, $A\cdot A^{-1}$, A^{T}, B^{T}.

Г) Вычислить определители А и В.

Назначение переменной ORIGIN, рассматривается в кратких теоретических сведениях темы 2.

Реализация в MathCad:



<u>Тема 3.</u> **Вычисление суммы, произведения, производной и интеграла в MathCad**

Краткие теоретические сведения

Формульный редактор системы MathCad дает возможность вычислять значения сумм, произведений, производной и интегралов. Шаблоны этих операций находятся среди кнопок палитры математического анализа.

$$\sum_{\blacksquare=\blacksquare}^{\blacksquare}\blacksquare \qquad \prod_{\blacksquare=\blacksquare}^{\blacksquare}\blacksquare \qquad \frac{d}{d\blacksquare}\blacksquare \qquad \frac{d^{\blacksquare}}{d\blacksquare^{\blacksquare}}\blacksquare \qquad \int_{\blacksquare}^{\blacksquare}\blacksquare\, d\blacksquare$$

Эти операции могут быть применены как к структурированным данным (дискретным переменным, массивам), так и к функциям.

Порядок вычисления суммы и произведения осуществляется следующим образом:

а) нажатием соответствующей кнопки палитры математических инструментов выводятся шаблоны операций суммы или произведения;

б) нижнее окно шаблона заполняется начальным значением дискретной переменной;

в) верхнее окно шаблона заполняется конечным значением дискретной переменной;

г) среднее окно шаблона заполняется любым выражением, которое включает в себя дискретную переменную.

Вычисление значений m-кратных ($m \geq 1$) и частных производных функций в заданных точках, осуществляется следующим образом:

а) нажатием соответствующей кнопки палитры математических инструментов вводится шаблон операции дифференцирования;

б) нижнее окно шаблона заполняется именем переменной дифференцирования;

в) среднее окно шаблона заполняется дифференцируемой функцией;

г) для вычисления m-кратной производной вводится необходимая степень, например 2.

Система MathCad позволяет вычислять как обычные m-кратные ($m \geq 1$) определенные интегралы, так и криволинейные интегралы. Последовательность выполнения расчетов при работе с операцией интегрирования имеет следующий вид:

а) нажатием соответствующей кнопки палитры математических инструментов вводится шаблон определенного интеграла;

б) в среднем окне шаблона задается вид подынтегральной функции, после символа **d** – имя переменной интегрирования;

в) в верхних и нижних окнах возле знака интеграла задаются верхний и нижний пределы интегрирования (действительные выражения).

Подынтегральная функция может быть как действительной, так и комплексной. В качестве пределов интегрирования, а также пределов подынтегрального выражения допускается использование дискретной переменной, в этом случае результатом вычислений является таблица интегралов.

Практическая часть темы 3

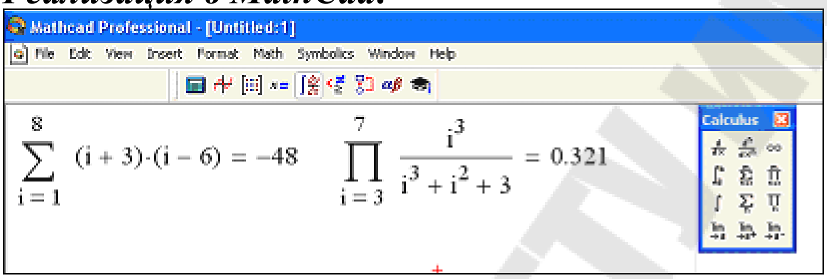
3.1. Вычисление суммы, произведения и определенного интеграла

Последовательность действий для вычисления суммы и произведения такова:

- установить курсор в свободное место рабочего окна документа;
- открыть палитру математического анализа;
- выбрать нужную операцию: суммирование или произведение;
- заполнить выделенные поля для ввода (см. краткие теоретические сведения темы 3);
- получить искомое значение в числовом виде с помощью оператора «=».

Пример 3.1. Вычислить значения суммы $\sum_{i=1}^{8}\left((i+3)\cdot(i-6)\right)$ и произведения $\prod_{i=3}^{7}\frac{i^3}{i^3+i^2+3}$.

Реализация в MathCad:



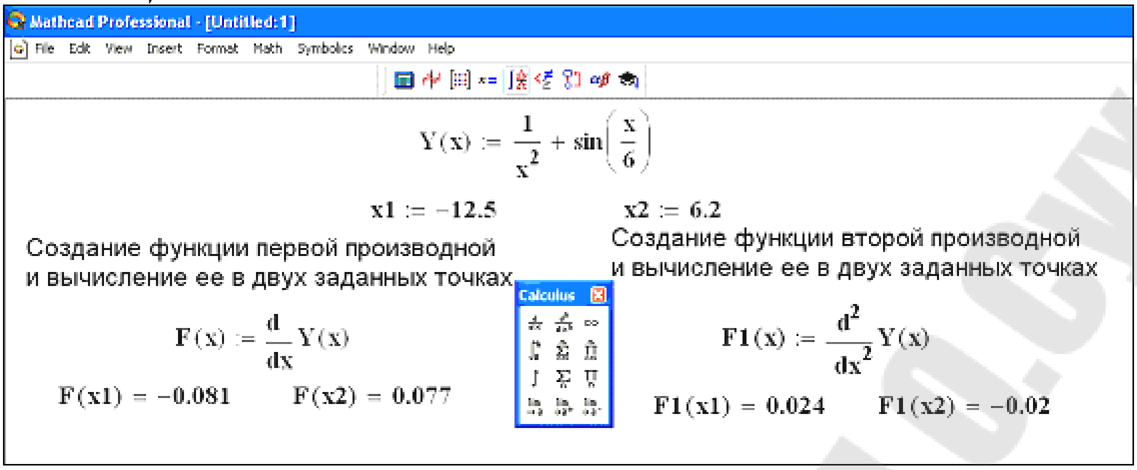
3.2 Вычисление производных в точках

Последовательность действий для вычисления производной в точке такова:

- установить курсор в свободное место рабочего окна документа;
- выделить в задаче исходные данные, например, x1 и x2, и поместить в них заданные числовые значения с помощью оператора «:=»;
- описать пользовательскую функцию с параметром, например Y(x);
- открыть палитру математического анализа;
- создать новые пользовательские функции F(x) и F1(x), в которые поместить значения первой и второй производных соответственно (для задания знаков производных использовать кнопки палитры математического анализа);
- заполнить выделенные поля для ввода (см. краткие теоретические сведения темы 3);
- получить искомые значения созданных функций производных при заданных значениях аргументов в числовом виде с помощью оператора «=».

Пример 3.2 Вычислить значения первой и второй производных функции $F(x)=\frac{1}{x^2}+\sin(\frac{x}{6})$ в двух исходных точках x1=-12.5 и x2=6.2.

Реализация в MathCad:



3.3 Вычисление производной в диапазоне изменения аргумента

Последовательность действий для вычисления производной в диапазоне изменения аргумента такова:

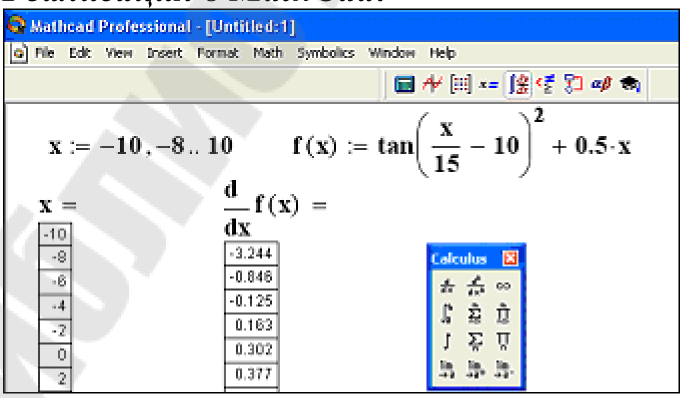
- создать дискретную переменную (см. краткие теоретические сведения темы 2);
- получить значение дискретной переменной в виде таблицы с помощью оператора «=».
- с помощью оператора присваивания описать пользовательскую функцию с параметром, например, f(x);
- в палитре математического анализа выбрать операцию вычисления первой производной и заполнить шаблон, используя в качестве аргумента функции имя заданной дискретной переменной;
- получить значения производной в числовом виде с помощью оператора «=».

Пример 3.3 Вычислить значение производной функции

$$f(x) = tg^2\left(\frac{x}{15} - 10\right) + 0.5 \cdot x$$

в дискретном интервале изменения аргумента [-10,10]. Шаг изменения аргумента выбрать так, что бы функция имела не менее 10-15 значений.

Реализация в MathCad:



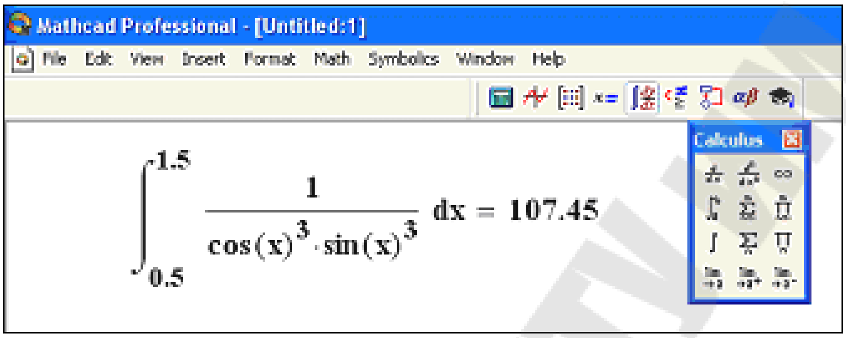
3.4. Вычисление определенного интеграла

Последовательность действий для вычисления определенного интеграла такова:

- используя кнопки палитры математического анализа выбрать операцию «определенный интеграл»;
- заполнить выделенные поля для ввода (см. краткие теоретические сведения темы 3);
- получить искомое значение в числовом виде с помощью оператора «=».

Пример 3.4. Вычислить значение определенного интеграла $\int_{0,5}^{1,5} \frac{dx}{\cos^3 x \cdot \sin^3 x}$.

Реализация в MathCad:



Тема 4. Создание программных фрагментов в MathCad

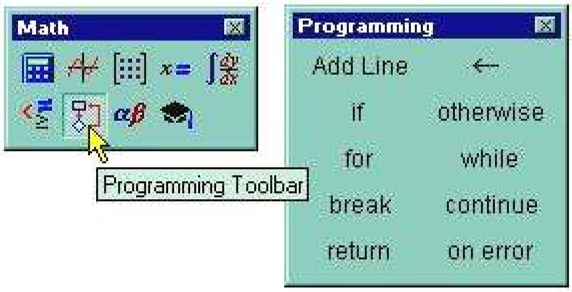
Краткие теоретические сведения

Система MathCad позволяет создавать программные фрагменты для вычисления алгоритмов, которые нельзя реализовать базовым набором средств и методов Mathcad.

Программный фрагмент можно использовать в операторе «=» или в правой части оператора «:=». Слева в операторе «:=» может находиться либо переменная, либо пользовательская функция.

Программный фрагмент состоит из строк программы, каждая из которых может содержать операторы программы.

Для создания программного фрагмента используется панель программирования, кнопки которой имеют следующее назначение:

<table>
<tr>
<td>
</td>
<td>Add Line - создание и расширение программного фрагмента;
← - оператор внутреннего локального присваивания.
if - оператор условия.
for - оператор цикла с заданным числом повторений.
while – оператор цикла с предусловием.
otherwise - оператор "иначе", обычно используется совместно с if для выполнения действий в случаи невыполнения условия.
break - оператор прерывания работы программного фрагмента.
continue - оператор продолжения работы цикла после прерывания.</td>
</tr>
</table>

Ниже приведен общий вид операторов программного фрагмента, рассмотрены правила их выполнения.

Общий вид оператора **if**:

выражение if условие

Порядок выполнения: если логическое выражение, стоящее в ***условии***, истинно, то вычисляется ***выражение***, стоящее слева от оператора if.

Общий вид ***for*** :

for Var $\in$ ***Nmin .. Nmax***.

Nmin .. Nmax – диапазон изменения переменной цикла, организуется по правилам формирования дискретных переменных.

Порядок выполнения: переменную цикла ***Var*** изменяет свое значение в пределах от ***Nmin*** до ***Nmax*** с шагом 1 или -1, при каждом новом значении переменной цикла выполняется рабочая часть цикла.

Общий вид ***while***:

while условие

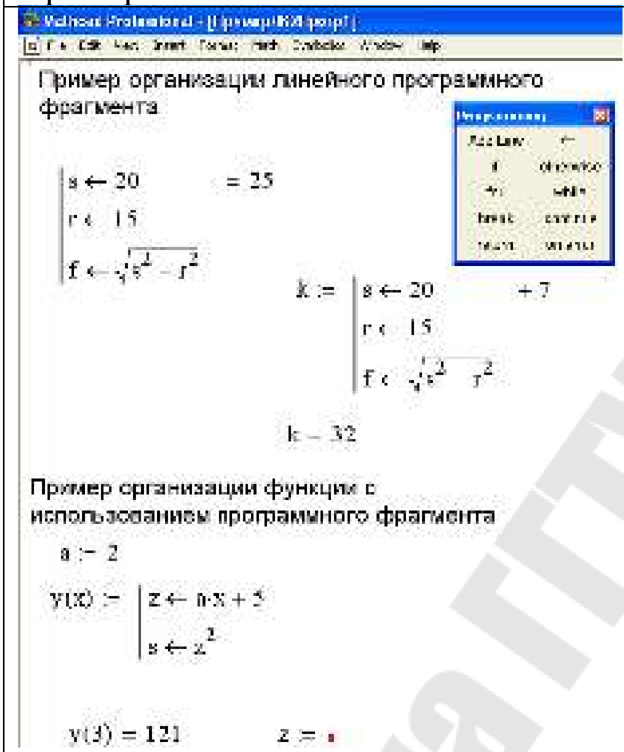
Рабочая часть цикла записывается на месте шаблона

Порядок выполнения: рабочая часть цикла выполняется до тех пор, пока логическое выражение, стоящее в условии, истинно.

Результаты работы программного фрагмента передаются следующим образом: во внешнюю вычислительную область из программного фрагмента передается значение последнего выражения, вычисленного в программном фрагменте или значение переменной, имя которой записано в последней строке программы.

На рисунке 2.4.1 приведены примеры программных фрагментов. Из примера 1 видно, что программный фрагмент, реализующий линейный алгоритм, может заканчиваться оператором «=», а может быть присвоен переменной, например, k. Программный фрагмент может участвовать при создании пользовательской функции, например, y(x), которая затем вычисляется при конкретном значении ее аргумента, например, 3. Внутренние переменные программного фрагмента являются неопределенными вне его, например, переменная z, которая использовалась в программном фрагменте, не имеет конкретного числового значения вне его.

Пример1



Пример2

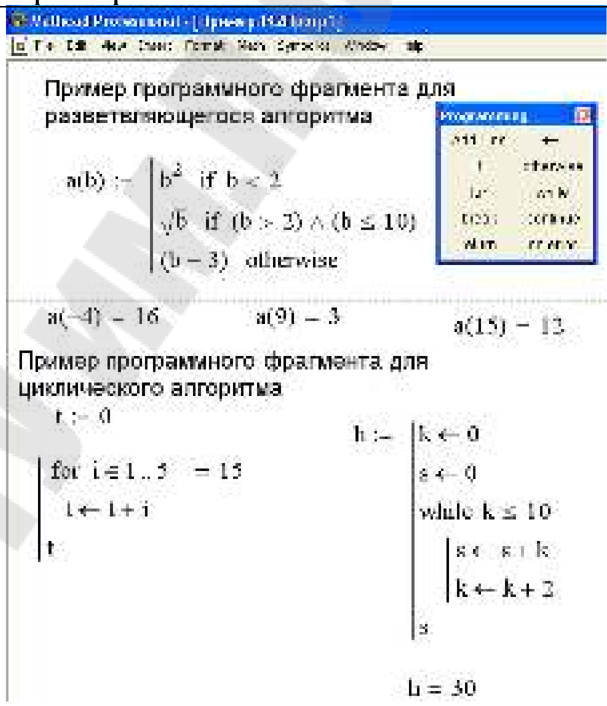


Рисунок 2.4.1 – Примеры программных фрагментов

Пример 2 демонстрирует программирование разветвляющихся и циклических алгоритмов. Для реализации разветвляющихся алгоритмов формируется пользовательская функция a(b), которая вычисляется по разным аналитическим зависимостям при различных значениях аргумента. Показана возможность вычисления этой функции в разных точках: при b равном -4, 9 и 15. Программирование циклических алгоритмов продемонстрировано на двух задачах. При решении задачи поиска суммы целых чисел от 1 до 5 применяется программный фрагмент с циклом for , результат получается с помощью оператора «=». Во второй части примера решается задача поиска суммы целых четных чисел от 0 до 10 с помощью оператора цикла while. Результат вычисления присваивается переменной h, хотя внутри цикла для накопления суммы используется переменная s.

Практическая часть темы 4

4.1 Программирование разветвляющихся алгоритмов

Последовательность действий для создания программного фрагмента вычисления значения кусочно-непрерывной функции такова:

- открыть палитру программирования и палитру логических операторов;
- набрать имя пользовательской функции, например, Y(x);
- с помощью оператора «**:=**» и кнопки «**Add line**» палитры программирования сформировать шаблон для записи операторов вычисления значения функции для различных диапазонов значения аргумента, например:

$$\mathbf{Y}(\mathbf{x}) := \left| \begin{array}{l} \blacksquare \\ \blacksquare \\ \blacksquare \end{array} \right. ;$$

- в каждую строку программного фрагмента, кроме последней, вывести шаблон оператора условия, используя кнопку «**if**» палитры программирования, например:

$$\mathbf{Y}(\mathbf{x}) := \left| \begin{array}{l} \blacksquare \ \ \mathbf{if} \ \ \blacksquare \\ \blacksquare \ \ \mathbf{if} \ \ \blacksquare \\ \blacksquare \end{array} \right. ;$$

- в каждом операторе условия слева от ***if*** набрать выражение для вычисления значения функции, а справа, используя кнопки палитры логических операторов, набрать условие, определяющее диапазон значения аргумента, например:

$$\mathbf{Y}(\mathbf{x}) := \left| \begin{array}{l} \mathbf{x}^2 \ \ \mathbf{if} \ \ \mathbf{x} \ge 0 \\ \mathbf{sin}(\mathbf{x}) \ \ \mathbf{if} \ \ -3 < \mathbf{x} \le -1 \\ \blacksquare \end{array} \right. ;$$

- в последнюю строку программного фрагмента вывести шаблон оператора «иначе», используя кнопку «**otherwis**e» палитры программирования;
- заполнить шаблон оператора «иначе» выражением для вычисления значения функции, например:

$$\mathbf{Y}(\mathbf{x}) := \left| \begin{array}{l} \mathbf{x}^2 \ \ \mathbf{if} \ \ \mathbf{x} \ge 0 \\ \mathbf{sin}(\mathbf{x}) \ \ \mathbf{if} \ \ -3 < \mathbf{x} \le -1 \\ \mathbf{cos}\left(\mathbf{x}^2\right) \ \ \mathbf{otherwise} \end{array} \right. ;$$

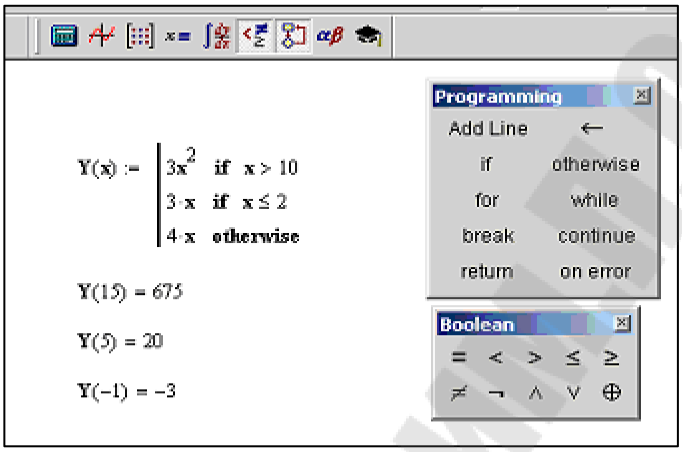
- получить значение функции Y(x) для различных значений аргумента x, (см. тему 1).

Пример 4.1. Вычислить значение кусочно-непрерывной функции

$$Y(x)=\begin{cases}3\cdot x^2, & \text{если} \quad x>10\\ 3\cdot x, & \text{если} \quad x\le 2\\ 4\cdot x, & \text{в остальных случаях}\end{cases}$$

для значений аргумента $x = 15$, $x = 5$ и $x = -1$ с использованием программного фрагмента.

Реализация в MathCad:



4.2 Программирование циклических алгоритмов

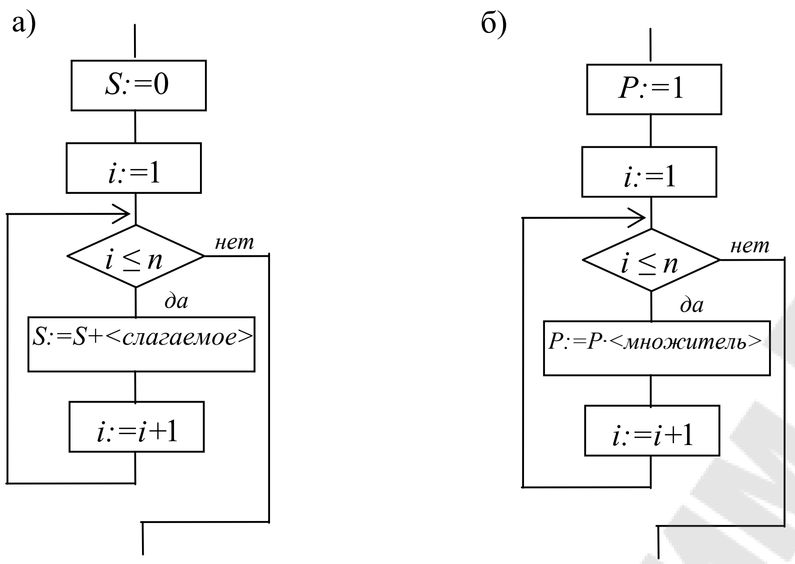
Последовательность действий для создания программного фрагмента, реализующего вычисление суммы (произведения) однотипных слагаемых (множителей), такова:

- в любой форме (например, в виде графической схемы) составить алгоритм решения задачи, например, как показано на рисунке 2.4.2;
- открыть палитру программирования;
- набрать имя переменной, в которую будет помещен результат вычислений, и знак «:=»;
- с помощью кнопки «**Add line**» палитры программирования создать программный фрагмент с необходимым количеством строк;
- в каждую строку программного фрагмента с помощью кнопок палитры программирования ввести шаблон оператора, реализующий ту или иную часть разработанного алгоритма, затем заполнить этот шаблон (например, для блока $\boxed{S:=0}$ нужно использовать оператор локального присваивания $S \leftarrow 0$, а для реализации цикла по переменной i - оператор цикла с заданным числом повторений

$$\begin{aligned}&\text{for } i \in 1..n\\ &\quad S \leftarrow S + \text{слагаемое}\end{aligned},$$

где значение n и вид слагаемого зависят от условия задачи);

- в последней строке программного фрагмента набрать имя локальной переменной, используемой для накопления суммы (произведения);
- вывести значение переменной, которой присваивается результат выполнения программного фрагмента, с помощью оператора «=».

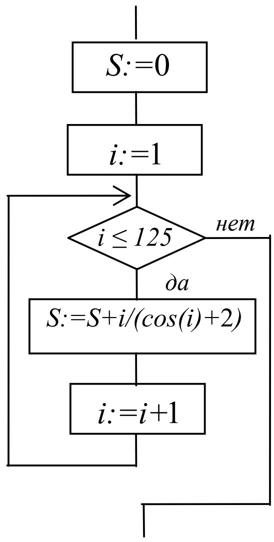


S – сумма
P – произведение
n - количество слагаемых или множителей

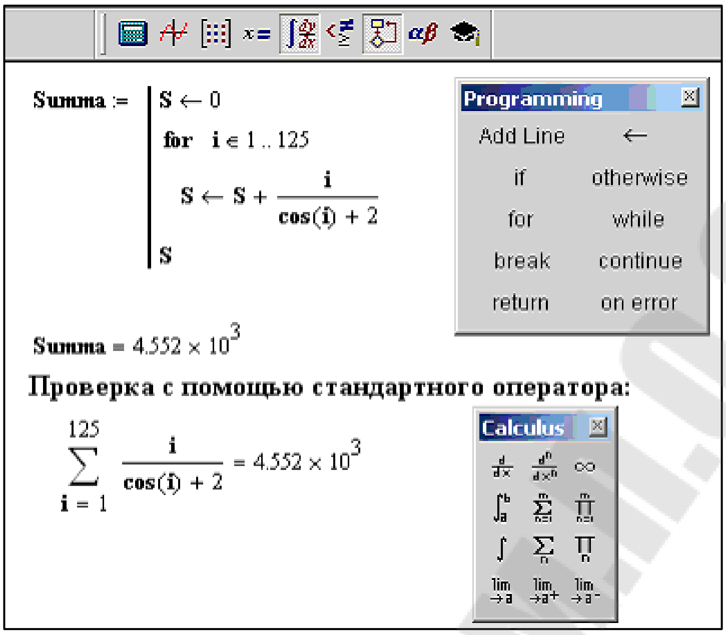
Рисунок 2.4.2 – Схемы алгоритмов вычисления
а) суммы вида $\sum_{i=1}^{n} < слагаемое >$;
б) произведения вида $\prod_{i=1}^{n} < множитель >$.

Пример 4.2. С использованием программного фрагмента вычислить сумму $\sum_{i=1}^{125} \frac{i}{\cos(i)+2}$.

Схема алгоритма решения:



Реализация в MathCad:

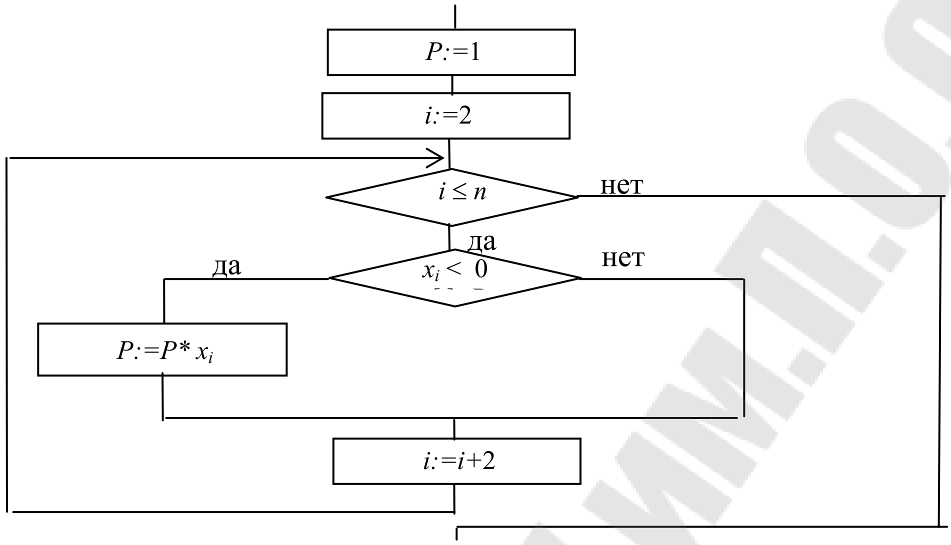


4.3 Программирование алгоритмов работы с массивами

Последовательность действий для создания программного фрагмента, предназначенного для обработки массива, такова:

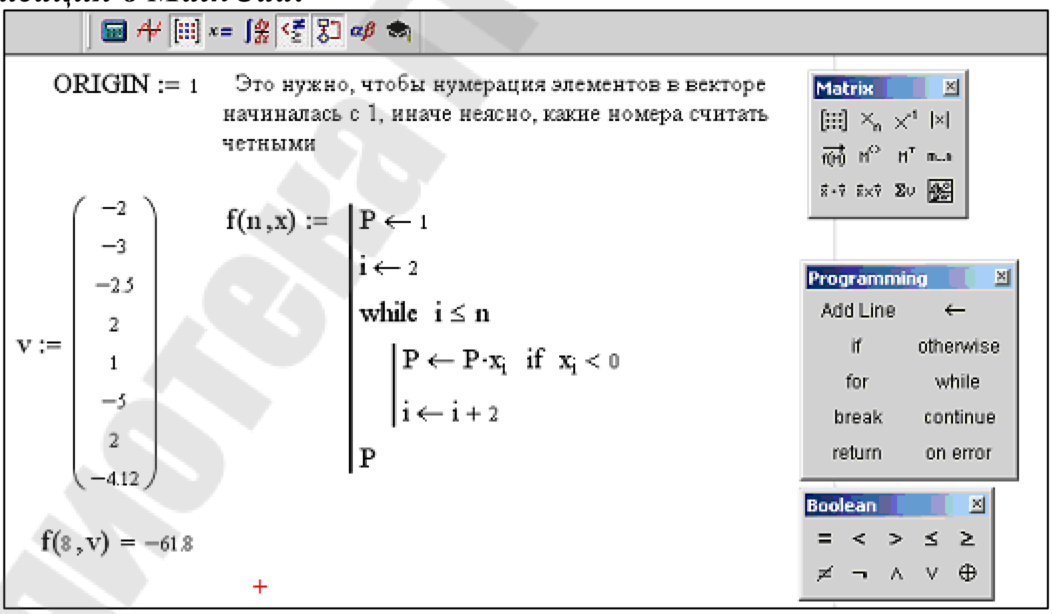
- в любой форме (например, в виде графической схемы) составить алгоритм решения задачи;
- открыть все необходимые палитры;
- задать исходный вектор чисел;
- определить функцию для решения поставленной задачи в виде программного фрагмента следующим образом:
  - набрать имя функции с двумя формальными параметрами: первый
  - размерность массива, второй – имя массива, а затем оператор «**:=**»;
  - с помощью кнопки «**Add line**» палитры программирования создать программный фрагмент с необходимым количеством строк;
- в каждую строку программного фрагмента с помощью кнопок палитры
  программирования ввести шаблон оператора, реализующий ту или иную часть разработанного алгоритма, затем заполнить этот шаблон (для реализации цикла по номеру элемента массива *i* нужно использовать оператор цикла с предусловием **while**, если шаг изменения номера *i* не равен 1);
- в последней строке программного фрагмента набрать имя локальной переменной, содержащей результат вычисления;
- вывести значение функции для заданных исходных данных, подставив в качестве фактических параметров количество элементов массива и имя определенного выше массива, с помощью оператора «=».

Пример 4.3. Дан вектор чисел произвольной длины. Используя программный фрагмент, вычислить произведение отрицательных элементов с четными номерами.
Схема алгоритма имеет вид:



Здесь x – исходный вектор, n – количество элементов этого вектора, i – номер текущего элемента вектора, P - произведение отрицательных элементов с четными номерами.

Реализация в MathCad:



<u>Тема 5.</u> Построение графиков

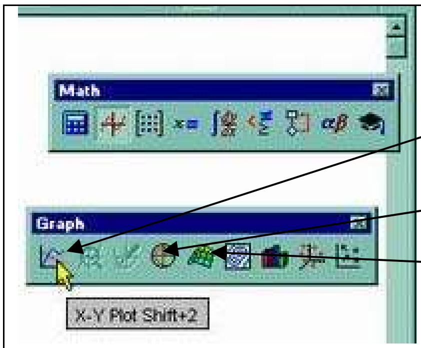
Краткие теоретические сведения

MathCAD позволяет обрабатывать различные виды графической информации. Возможности системы по работе с графикой таковы:

- построение двумерных графиков в декартовой и полярной системах координат,
- построение трехмерных поверхностных графиков,
- внесение рисунков, созданных другими компьютерными системами;
- создание анимационных клипов.

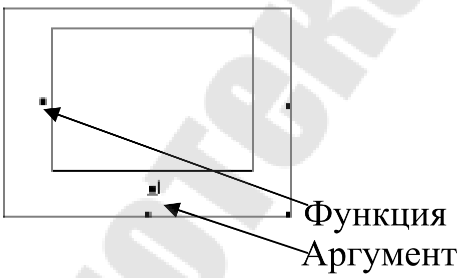
Соответственно, графические области делятся на четыре основных типа – область двумерных графиков, трехмерных графиков, область внешних графических объектов и область анимации.

Для построения графиков используется палитра графиков, вид которой приведен ниже. Перечень возможных типов графиков приведен в основном меню ***Insert*** – ***Graph,*** например:



- ***X-Y Plot*** – график в декартовой системе координат;
- ***Polar Plot***– график в полярной системе координат;
- ***Surface Plot*** – трехмерный график (график поверхности);

При построении двумерных графиков после нажатия соответствующей кнопки на панели графических инструментов появляется шаблон вида:



В шаблоне графика по вертикали задаются через запятую функции, а по горизонтали – аргументы. График строится по точкам соединяющихся между собой разнообразными линиями (сплошной, пунктирной и т. д.). Исходные (узловые) точки могут быть показаны в виде маркеров (квадратов, ромбов, окружностей и т. д.). Крайние шаблоны данных служат для указания предельных значений абсцисс и ординат, т. е. они задают масштабы графика. Если оставить эти шаблоны незаполненными, то масштабы по осям графика будут устанавливаться автоматически.

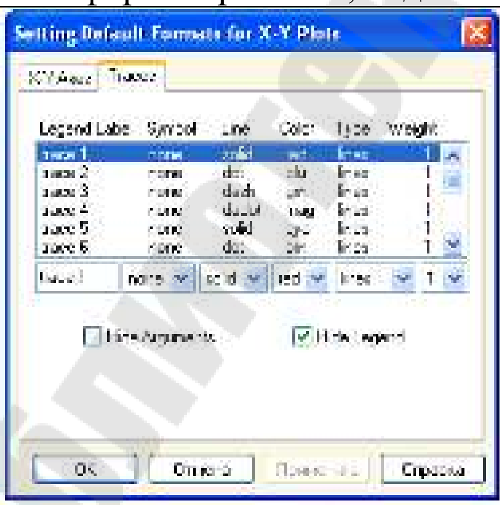
В полярной системе координат каждая точка задается углом ***a***, радиусом и длиной радиус–вектора ***R(a).*** График функции обычно строится при изменении угла ***a*** в пределах от 0 до 2π.

После построения график может быть отформатирован по следующим направлениям: форматирование осей графика, форматирование линий графика, форматирование надписей на графике.

Для вызова окна форматирования графика нужно либо дважды щелкнуть левой кнопкой мыши на поле графика, либо, выделив график, выбрать из основного меню команду ***Format – Graph.*** Вид окна форматирования осей графика приведен ниже.

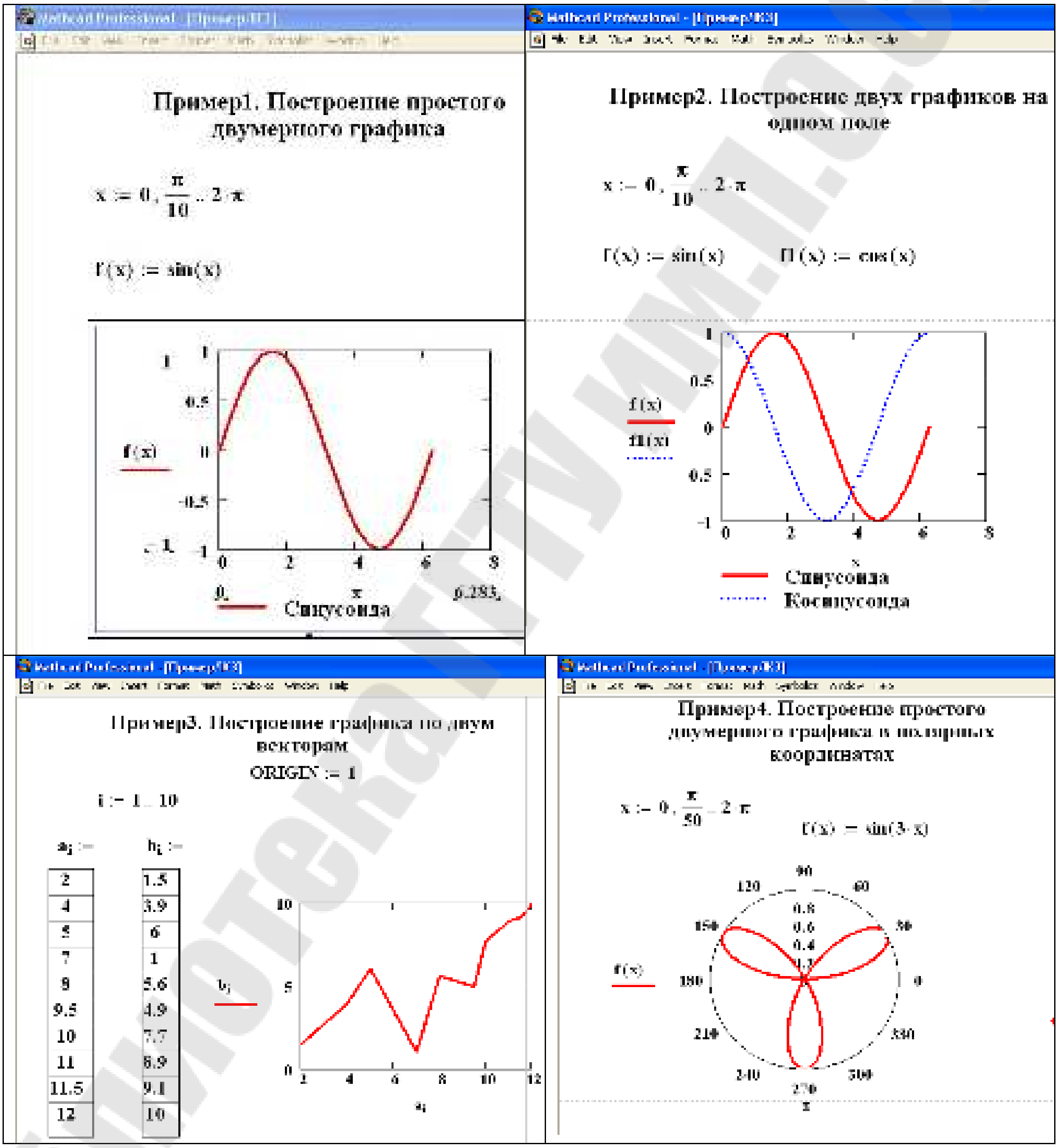
Log Scale (Лог. масштаб) — установка логарифмического масштаба;
Crid Lines (Линии сетки) — установка линий масштабной сетки;
Numbered (Пронумеровать) — установка цифровых данных по осям;
Autoscale (Автомасштаб) — автоматическое масштабирование графика;
Show Markers (Нанести риски) — установка фоновых линий по осям;
Auto Grid (Автосетка) — автоматическая установка масштабных линий;
Nunber of Grids (Число интервалов) — установка заданного числа масштабных линий.
Boxed (Рамка) — оси в виде прямоугольника;
Crossed (Репер) — оси в виде креста;
None (Ничего) — отсутствие осей;

При форматировании линий графика выбирается закладка **Traces** в окне форматирования, вид окна форматирования линий приведен ниже.



Legend Label (Имя кривой) — указание названий линий в легенде графика;
Symbol (Маркер) — установка символа отметки базовых точек графика;
Line (Линия) — установка типа линий;
Color (Цвет) — установка цвета линии и базовых точек;
Type (Тип) — тип графиков;
Weight (Толщина) — толщина линий.
Hide Argument — скрываются обозначения математических выражений по осям графика;
Hide Legend — скрываются обозначения имен кривых графика.

Ниже приведено несколько примеров построения двумерных графиков различных типов. В примере 1 аргумент задан в виде дискретной переменной x, сформирована функция дискретной переменной f(x). Построен график, изменен цвет линии графика, нанесена легенда. Во втором примере в окно оси ординат трафарета графика вносится имена первой и второй функций, разделенные запятыми, аргумент функций – одна и та же дискретная переменная x. График форматируется, наносится легенда.



В примере 3 создается график, у которого аргумент и функция являются одномерными массивами, заданными, как переменные с индексами. Элементы этих массивов задаются перечислением, разделяются запятыми. В примере 4 построен двумерный график в полярной системе координат.

Практическая часть темы 5

5.1 Построение двумерных графиков

Последовательность действий для построения графиков функции и ее первой и второй производных, а также графического определения точек экстремума, такова:

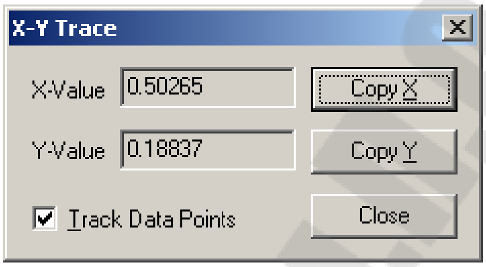
- описать исходную функцию (см. теоретические сведения темы 1), например,
  $\mathbf{f(x) := \sin(3 \cdot x)}$;
- описать функции первой и второй производных с использованием соответствующих кнопок палитры математического анализа (см. теоретические сведения темы 3), например,
  $\mathbf{f1(x) := \frac{d}{dx} f(x)} \qquad \mathbf{f2(x) := \frac{d^2}{dx^2} f(x)}$;
- определить аргумент в виде дискретной переменной (см. краткие теоретические сведения темы 2), например, $\mathbf{x := 0, \frac{\pi}{50} .. \pi}$;
- открыть палитру графиков;
- нажать кнопку ***X-Y Plot*** палитры для построения графика в декартовой системе координат, после чего в рабочий документ по местоположению курсора будет вставлен шаблон графика;
- в поле ввода ■ по оси абсцисс поместить имя аргумента, а по оси ординат ввести имена функций через запятую (например, **f(x), f1(x), f2(x)**);
- двойным щелчком мыши по области графика открыть окно форматирования;
- произвести форматирование осей графика (см. краткие теоретические сведения темы 5), рекомендуется установить оси в виде креста;
- для вывода легенды в окне форматирования графика выбрать закладку **Traces** и задать названия для каждой функции в поле ***Legend Label***:
  - выбрать в списке соответствующую линию, например, **trace1**;
  - в поле ввода под списком набрать название линии,

  например,

  для **f(x)** вместо **trace1** набрать **функция f(x)**,

  для **f1(x)** вместо **trace2** набрать **первая производная функции f(x)**;
- щелчком мыши убрать флажок переключателя ***Hide Legend***;

- для нанесения фоновых линий выбрать закладку **X-Y Axes** окна форматирования и щелчком мыши установить флажок переключателя ***Show Markers*** для оси абсцисс (**X-Axis**);
- закрыть окно форматирования графика нажатием кнопки **ОК**, при этом под осью абсцисс появится легенда и два пустых поля ■ для определения точек нанесения фоновых линий;
- для определения точек экстремума функции выполнить команду меню **Format – Graph – Trace…**, после чего будет открыто окно **X-Y**

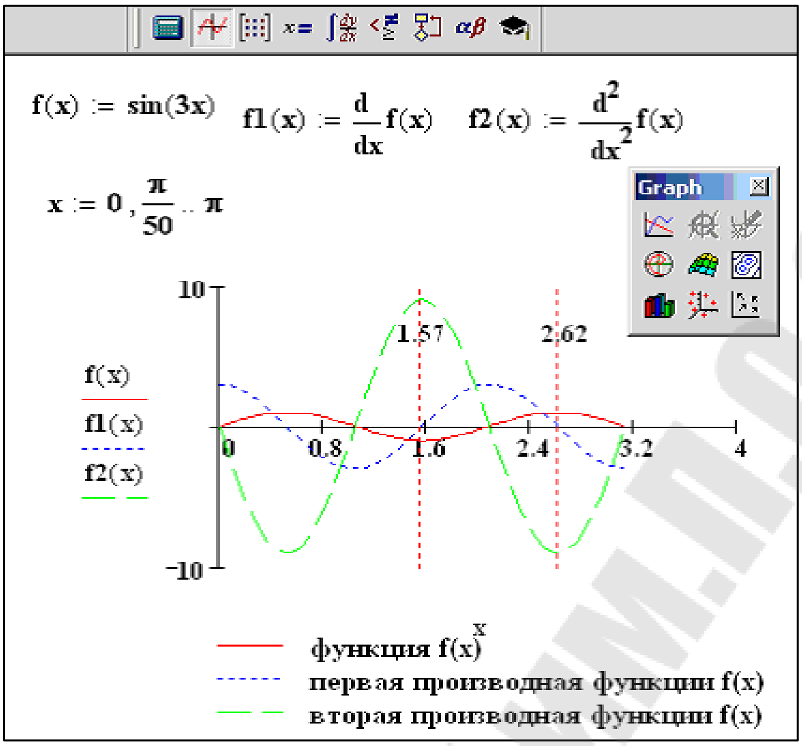


Trace

(_Замечание_: в точках экстремума первая производная функции равна нулю);
- щелчком мыши убрать флажок переключателя ***Track Data Points***;
- при нажатой левой кнопке мыши перемещать указатель вдоль линии графика первой производной до точки пересечения ее с осью абсцисс, после чего отпустить кнопку мыши, при этом в поле ***X-Value*** окна **X-Y Trace** появится значение аргумента, при котором первая производная равна нулю;
- заполнить одно из пустых полей ■ под осью абсцисс, набрав значение из поля ***X-Value***;
- для визуализации графика щелкнуть левой кнопкой мыши или нажать клавишу **Enter**;
- закрыть окно **X-Y Trace.**

Пример 5.1 Построить график функции $f(x) = \sin(3 \cdot x)$ и графики ее первой и второй производной. Нанести фоновые линии в точках экстремума функции $\left(\frac{d}{dx} f(x) = 0\right)$. Функция должна рассчитываться не менее чем в n=50 точках, при $x_{нач} = 0$ и $x_{кон} = \pi$.

Реализация в MathCad:



5.2 Построение графиков кусочно-непрерывных функций

Последовательность действий для построения графика кусочно-непрерывной функции такова:

- с помощью программного фрагмента задать кусочно-непрерывную функцию (см. тему 4), например,

$$y(x) := \begin{cases} 3\sin(x) & \text{if } x > 25 \\ \dfrac{x^2}{15} & \text{if } x < 12 \\ (x-20) & \text{otherwise} \end{cases};$$

- определить аргумент в виде дискретной переменной таким образом, чтобы функция вычислялась по каждой из формул (см. краткие теоретические сведения темы 2), например $x := 1, 1.5 .. 35$;
- построить график заданной функции (см. краткие теоретические сведения темы 5);
- двойным щелчком мыши по области графика открыть окно форматирования;
- для нанесения вспомогательных линий щелчком мыши установить флажок переключателя ***Crid Lines*** для каждой оси;

- для ввода названий осей и самого графика в окне форматирования выбрать закладку **Labels**, ввести заголовок графика в поле ***Title*** и установить флажок переключателя ***Show Title***;
- ввести названия осей в полях ввода панели ***Axis Labels***;
- для изменения толщины линии выбрать закладку **Traces** и задать толщину линии графика, используя список поля ***Weight*** :

;

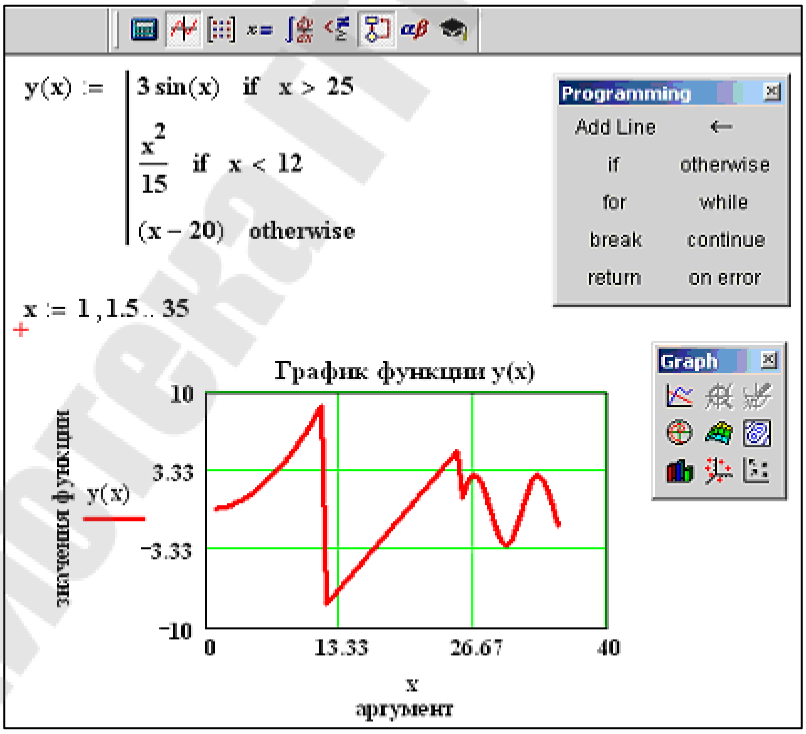
- закрыть окно форматирования графика нажатием кнопки **ОК**.

Пример 5.2 Построить график кусочно-непрерывной функции

$$y = \begin{cases} 3\sin(x)\,, & \textit{если} \quad x > 25 \\ \dfrac{x^2}{15}\,, & \textit{если} \quad x < 12 \\ x - 20\,, & \textit{в} \quad \textit{остальных} \quad \textit{случаях} \end{cases} .$$

Пределы изменения аргумента подобрать так, чтобы перекрывались все три диапазона. При задании вида функции использовать программный фрагмент.

Реализация в MathCad:



<u>Тема 6.</u> **Решение уравнений и систем**

Краткие теоретические сведения

Для алгебраических уравнений вида ***f(x)=0*** решение в MathCad находится с помощью функции ***root.***

Общий вид функции следующий:

root(f(x), x), где

f(x) – функция, описывающая левую часть выражения вида ***f(x)=0,***

x – имя переменной, относительно которой решается уравнение.

Функция ***root*** реализует алгоритм поиска корня численным методом и требует предварительного задания начального приближения искомой переменной ***x***. Поиск корня будет производиться вблизи этого числа. Таким образом, присвоение начального значения требует предварительной информации о примерной локализации корня.

Функция позволяет найти как вещественные корни, так и комплексные. В случае комплексного корня начальное приближение нужно задать в виде комплексного числа.

Если после многих итераций Mathcad не находит подходящего приближения, то появится сообщение «отсутствует сходимость».

Эта ошибка может быть вызвана следующими причинами:

- уравнение не имеет корней;
- корни уравнения расположены далеко от начального приближения;
- выражение ***f(x)*** имеет разрывы между начальным приближением и корнем;
- выражение имеет комплексный корень, но начальное приближение было вещественным и наоборот.

Для изменения точности, с которой функция ***root*** ищет корень, нужно изменить значение системной переменной ***TOL***. Например, просле задания в документе оператора ***TOL:=0.00001*** точность вычисления корня станет равной 0.00001.

Для нахождения корней полиномиального уравнения вида

$$v_n \cdot x^n + v_{n-1} \cdot x^{n-1} + \ldots + v_1 \cdot x + v_0 = 0$$

используется функция ***polyroots.***

В отличие от функции ***root, polyroots*** не требует начального приближения и вычисляет сразу все корни, как вещественные, так и комплексные.

Общий вид:

polyroots(v),

где ***v*** – вектор коэффициентов полинома длины ***n+1, n*** – степень полинома. Вектор ***v*** формируется следующим образом: в первый его элемент заносится значение коэффициента полинома при x^0, т.е. $\boldsymbol{v_0}$, во второй элемент - значение коэффициента полинома при x^1, т.е. v_1 и т.д. Таким образом, вектор заполняется коэффициентами перед степенями полинома справа налево.

Функция вычисляет вектор длины n, состоящий из корней полинома.

На рисунке 2.6.1 приведены примеры вычисления корней уравнений с помощью функций ***root*** и ***polyroots***.

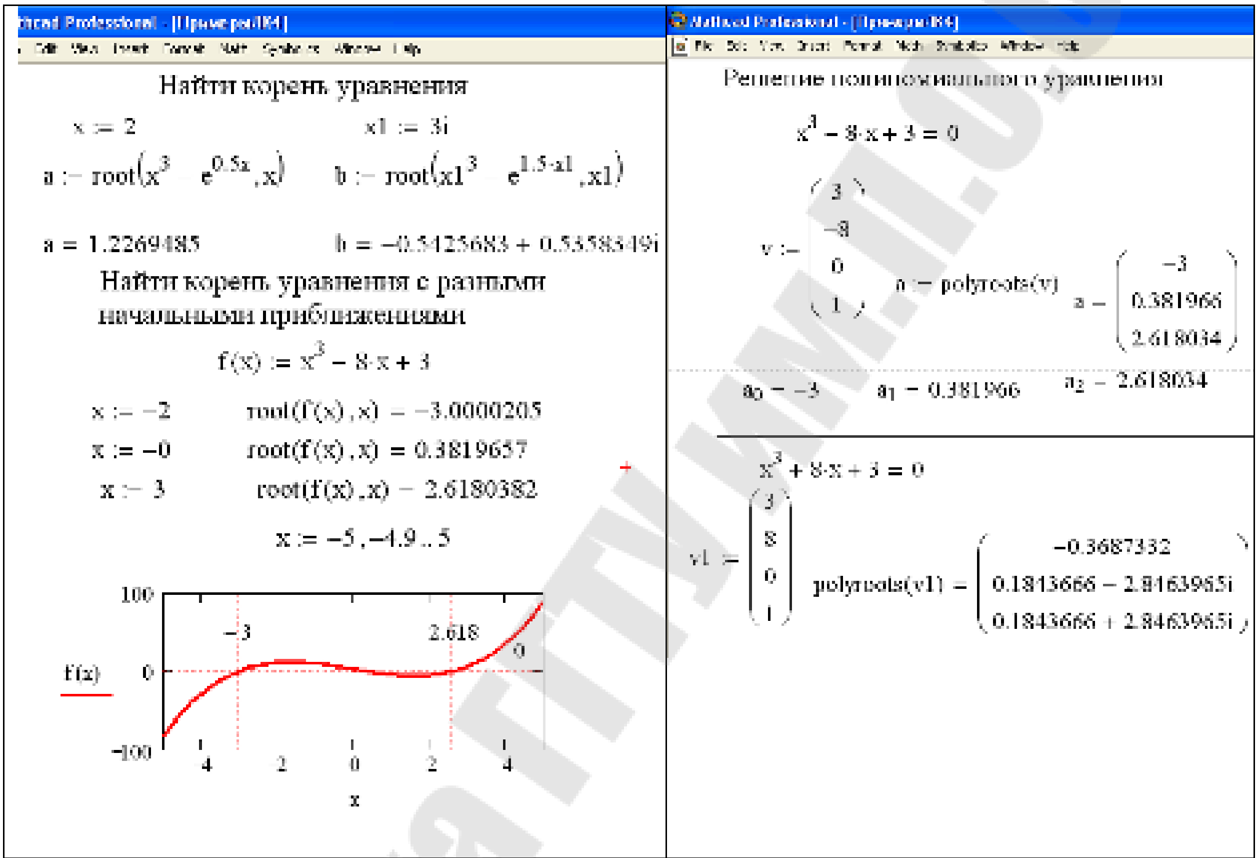


Рисунок 2.6.1 – Примеры решения уравнений

MathCAD дает возможность решать системы уравнений и неравенств.

Наиболее распространенным методом решения уравнений в Mathcad является блочный метод. Для решения системы этим методом необходимо выполнить следующее:

а) задать начальное приближение для всех неизвестных, входящих в систему уравнений;

б) задать ключевое слово ***Given***, которое указывает, что далее следует система уравнений;

в) ввести уравнения и неравенства в любом порядке (использовать кнопку логического равенства на панели знаков логических операций  для набора знака «=» в уравнении);

г) ввести любое выражение, которое включает функцию ***Find***.

Решающим блоком называется часть документа, расположенная между ключевыми словами ***Given*** и ***Find.***

После набора решающего блока Mathcad возвращает точное решение уравнения или системы уравнений.

Обратиться к функции ***Find*** можно несколькими способами:

Find(x1, x2,…) = - корень или корни уравнения вычисляются и выводятся в окно документа.

x := Find(x1, x2,…) – формируется переменная или вектор, содержащий вычисленные значения корней.

Сообщение об ошибке «*Решение не найдено*» появляется тогда, когда система не имеет решения или для уравнения, которое не имеет вещественных корней, в качестве начального приближения взято вещественное число и наоборот.

Приближенное решение уравнения или системы можно получить с помощью функции ***Minerr.***

Если в результате поиска не может быть получено дальнейшее уточнение текущего приближения к решению, ***Minerr*** возвращает это приближение. Функция ***Find*** в этом случае возвращает сообщение об ошибке. Правила использования функции ***Minerr*** такие же, как и для функции ***Find***. Часть документа, расположенная между ключевыми словами ***Given*** и ***Minerr*** так же носит название решающего блока.

Примеры решения систем уравнений с помощью решающего блока приведены на рисунке 2.6.2.

Для решения систем линейных уравнений можно использовать общепринятые математические методы: метод Крамера, матричный метод и т.д.

Матричный метод решения системы линейных уравнений реализован в функции ***lsolve.*** Общий вид функции:

lsolve(a, b)

где ***a*** – матрица коэффициентов перед неизвестными, ***b*** – вектор свободных членов.

Матричный метод можно реализовать и с помощью обратной матрицы. Примеры решения систем линейных уравнений с помощью матричного метода приведены на рисунке 2.6.2.

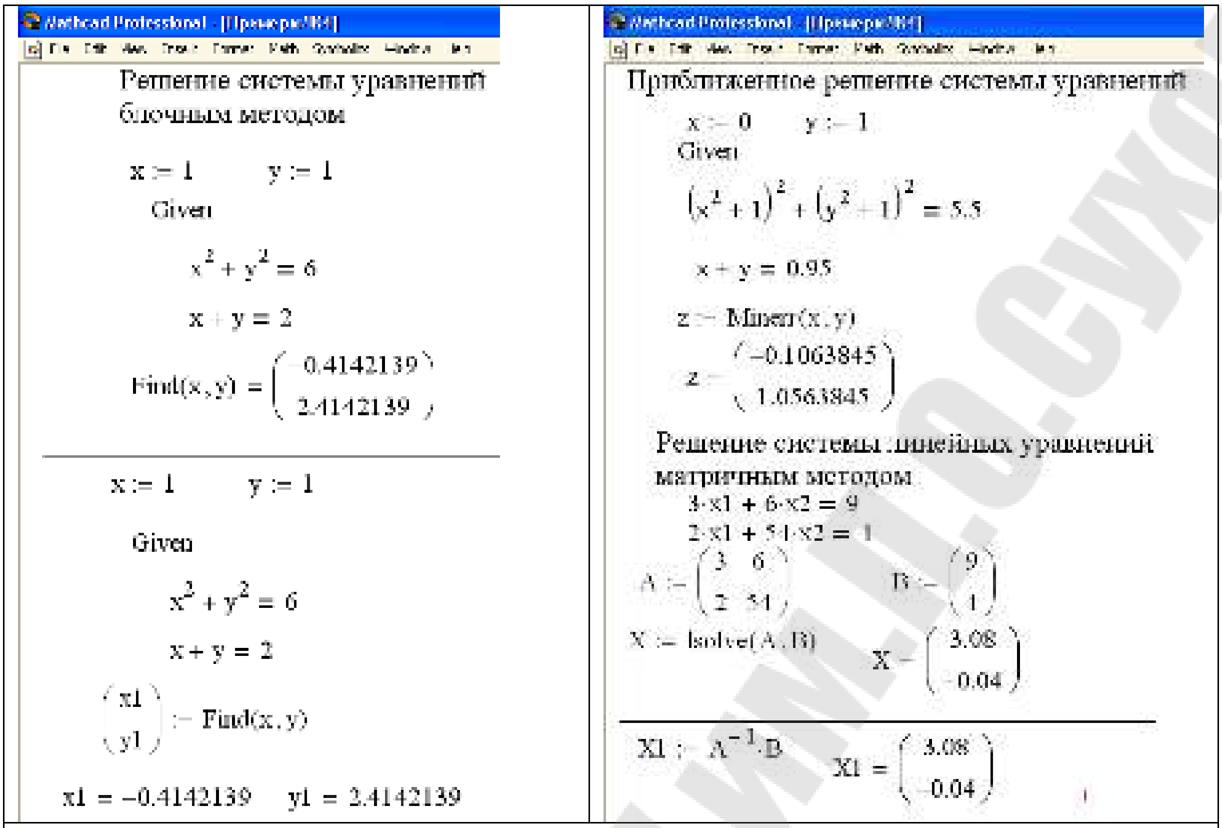


Рисунок 2.6.2 – Примеры решения систем уравнений

Из рисунка 6.2 видно, что при решении системы уравнений блочным методом можно получить численные значения корней системы уравнений, без присваивания и с присваиванием их в переменные x1 и x2. При решении системы уравнений матричным методом продемонстрированы два варианта: с использованием стандартной функции ***lsolve*** и обратной матрицы.

Практическая часть темы 6

6.1 Поиск корней уравнения, графическая интерпретация

Последовательность действий для нахождения корня уравнения с применением функции **root** такова:

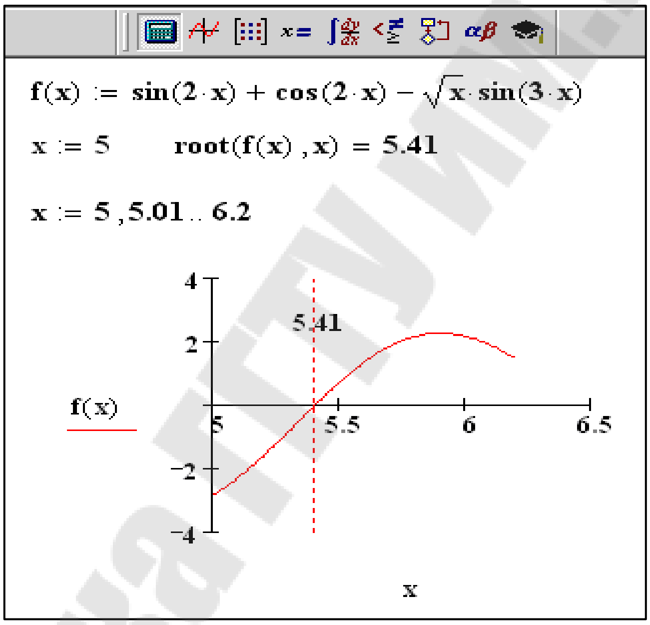
- привести уравнение к виду ***f(x)=0***, если это необходимо, например, уравнение $\sin(2x)+\cos(2x)=\sqrt{x}\sin(3x)$ преобразуется в $\sin(2x)+\cos(2x)-\sqrt{x}\sin(3x)=0$;
- установить курсор в свободное место рабочего окна документа MathCad;
- определить функцию, содержащуюся в левой части уравнения вида ***f(x)=0***, например, $f(x):=\sin(2x)+\cos(2x)-\sqrt{x}\sin(3x)$;
- задать начальное приближение искомой переменной ***x***;
- подставляя в качестве параметров функции ***root*** имя функции левой части уравнения и переменную, содержащую начальное приближение, вывести значение корня уравнения при заданном начальном

приближении с помощью оператора «=»(см. краткие теоретические сведения темы 6);

- задать аргумент функции ***f(x)*** в виде дискретной переменной, при этом диапазон изменения аргумента должен включать найденный корень уравнения (см. тему 2);
- построить график функции ***f(x)*** (см. тему 5);
- установить оси графика в виде креста (см. тему 5);
- нанести фоновую линию в точке, совпадающей с найденным корнем уравнения (см. тему 5).

Пример 6.1. Найти корень уравнения $\sin(2x)+\cos(2x)=\sqrt{x}\sin(3x)$ при заданном начальном значении $x\approx 5$. Выполнить графическую интерпретацию результата.

Реализация в MathCad:



6.2 Поиск корней полиномиального уравнения, графическая интерпретация

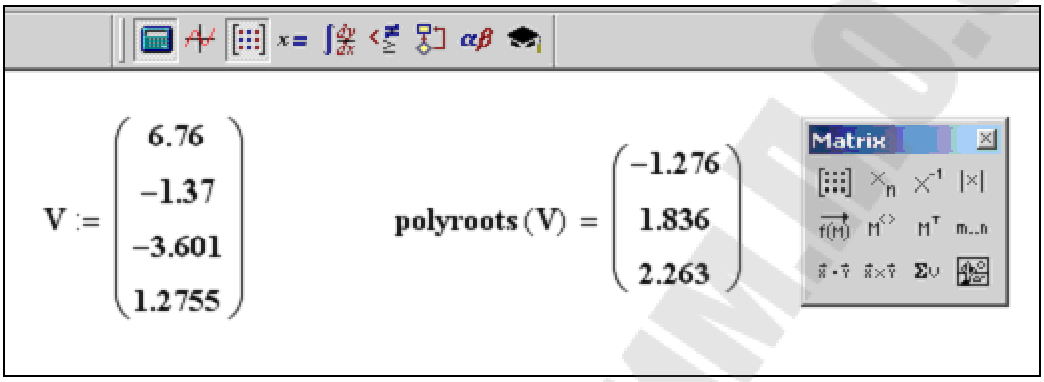
Последовательность действий для нахождения корней полиномиального уравнения вида $v_n \cdot x^n + v_{n-1} \cdot x^{n-1} + \ldots + v_1 \cdot x + v_0 = 0$ с использованием функции ***polyroots*** такова:

- создать вектор длиной (***n***+**1**), состоящий из коэффициентов полинома, расположенного в левой части уравнения (первый элемент - значение коэффициента полинома при x^0, второй элемент - значение коэффициента полинома при x^1 и т.д. и присвоить его какой-либо переменной (см. тему 2 и краткие теоретические сведения темы 6);

- используя имя вектора в качестве аргумента функции ***polyroots***, получить числовые значения корней полиномиального уравнения с помощью оператора «=».

Пример 6.2. Вычислить множество корней полиномиального уравнения $1.2755x^3 - 3.601 \cdot x^2 - 1.37 \cdot x + 6.76 = 0$ с использованием функции ***polyroots***.

Реализация в MathCad:



6.3 Решение системы линейных уравнений

А) Последовательность действий для решения системы линейных уравнений методом Крамера такова:

- создать матрицу коэффициентов системы линейных уравнений, например, $A := \begin{pmatrix} 2.48 & 1.75 & -3.24 \\ 4.78 & 1.2 & 2.2 \\ -3.24 & 2.19 & -1.86 \end{pmatrix}$ (см. краткие теоретические сведения темы 2);
- создать вектор свободных членов, например $B := \begin{pmatrix} 1.23 \\ 3.55 \\ -0.15 \end{pmatrix}$;
- с помощью оператора «**:=**» создать матрицу, равную матрице коэффициентов, например, $A1 := A$;
- заменить в созданной матрице первый столбец вектором свободных членов, используя операцию выделения столбца матрицы, например, $A1^{<1>} := B$ или $A1^{<0>} := B$ (в зависимости от значения переменной ***ORIGIN***);
- аналогично из матрицы коэффициентов создать матрицу, в которой второй столбец заменен вектором свободных членов, затем матрицу, в которой третий столбец заменен вектором свободных членов, и т.д. (количество таких матриц определяется количеством неизвестных в системе уравнений);

- найти первый корень, разделив определитель матрицы с замененным первым столбцом на определитель матрицы коэффициентов, например: $\frac{|A1|}{|A|}=$;
- найти остальные корни системы уравнений аналогично.

Б) Последовательность действий для решения системы линейных уравнений матричным методом такова:

- создать матрицу коэффициентов системы линейных уравнений, например, A (см. краткие теоретические сведения темы 2);
- создать вектор свободных членов системы линейных уравнений, например, B;
- получить решение системы с помощью функции ***lsolve***, параметрами которой являются матрица коэффициентов и вектор свободных членов, например: $X := lsolve(A, B)$

  (решение также можно получить, умножив матрицу, обратную к матрице коэффициентов, на вектор свободных членов: $X := A^{-1} \cdot B$);
- вывести полученный вектор, содержащий корни системы, с помощью оператора «=».

В) Последовательность действий для решения системы линейных уравнений блочным методом такова:

- задать начальные приближения для всех неизвестных, входящих в систему уравнений;
- набрать ключевое слово ***Given***;
- ниже слова ***Given*** набрать уравнения, отделяя правую и левую части символом логического равенства «**=**» (см. краткие теоретические сведения темы 6);
- набрать функцию ***Find***, подставляя в качестве аргументов имена неизвестных системы;
- вывести вектор, содержащий вычисленные значения корней, с помощью оператора «=», например ***Find(x1,x2,x3)=***.

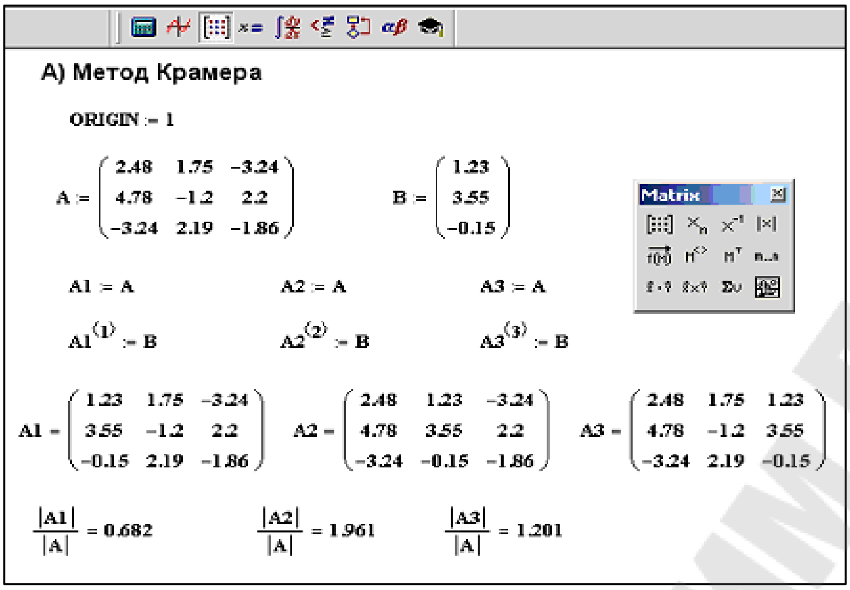
Замечание. Корни системы уравнений, полученные разными способами, должны совпасть.

Пример 6.3. Решить систему линейных уравнений

$$\begin{cases} 2.48 \cdot x_1 + 1.75 \cdot x_2 - 3.24 \cdot x_3 = 1.23 \\ 4.78 \cdot x_1 - 1.2 \cdot x_2 + 2.2 \cdot x_3 = 3.55 \\ -3.24 \cdot x_1 + 2.19 \cdot x_2 - 1.86 \cdot x_3 = -0.15 \end{cases}$$

методом Крамера, матричным и блочным методами. Сравнить полученные результаты. Начальные значения корней при использовании блочного метода принять равными 1.

Реализация в MathCad:



А) Метод Крамера

ORIGIN := 1

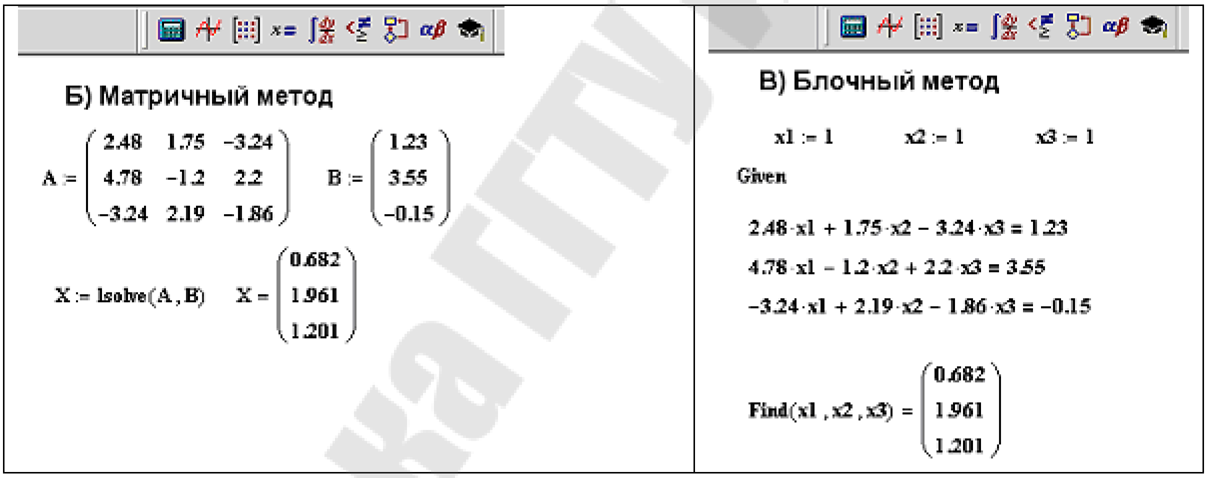
$$A := \begin{pmatrix} 2.48 & 1.75 & -3.24 \\ 4.78 & -1.2 & 2.2 \\ -3.24 & 2.19 & -1.86 \end{pmatrix} \qquad B := \begin{pmatrix} 1.23 \\ 3.55 \\ -0.15 \end{pmatrix}$$

$A1 := A \qquad A2 := A \qquad A3 := A$

$A1^{\langle 1\rangle} := B \qquad A2^{\langle 2\rangle} := B \qquad A3^{\langle 3\rangle} := B$

$$A1 = \begin{pmatrix} 1.23 & 1.75 & -3.24 \\ 3.55 & -1.2 & 2.2 \\ -0.15 & 2.19 & -1.86 \end{pmatrix} \quad A2 = \begin{pmatrix} 2.48 & 1.23 & -3.24 \\ 4.78 & 3.55 & 2.2 \\ -3.24 & -0.15 & -1.86 \end{pmatrix} \quad A3 = \begin{pmatrix} 2.48 & 1.75 & 1.23 \\ 4.78 & -1.2 & 3.55 \\ -3.24 & 2.19 & -0.15 \end{pmatrix}$$

$$\frac{|A1|}{|A|} = 0.682 \qquad \frac{|A2|}{|A|} = 1.961 \qquad \frac{|A3|}{|A|} = 1.201$$



Б) Матричный метод

$$A := \begin{pmatrix} 2.48 & 1.75 & -3.24 \\ 4.78 & -1.2 & 2.2 \\ -3.24 & 2.19 & -1.86 \end{pmatrix} \qquad B := \begin{pmatrix} 1.23 \\ 3.55 \\ -0.15 \end{pmatrix}$$

$$X := \mathrm{lsolve}(A, B) \qquad X = \begin{pmatrix} 0.682 \\ 1.961 \\ 1.201 \end{pmatrix}$$

В) Блочный метод

$x1 := 1 \qquad x2 := 1 \qquad x3 := 1$

Given

$2.48 \cdot x1 + 1.75 \cdot x2 - 3.24 \cdot x3 = 1.23$

$4.78 \cdot x1 - 1.2 \cdot x2 + 2.2 \cdot x3 = 3.55$

$-3.24 \cdot x1 + 2.19 \cdot x2 - 1.86 \cdot x3 = -0.15$

$$\mathrm{Find}(x1, x2, x3) = \begin{pmatrix} 0.682 \\ 1.961 \\ 1.201 \end{pmatrix}$$

6. 4 Решение систем нелинейных уравнений

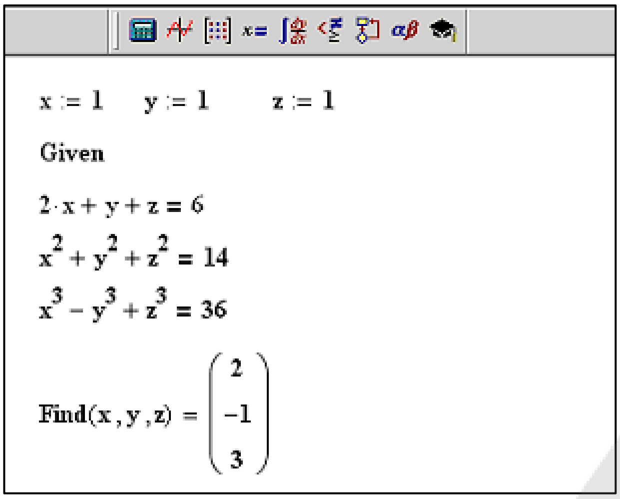
Последовательность действий для решения системы нелинейных уравнений блочным методом такая же, как и для решения систем линейных уравнений, изложенная в пункте 6.3В.

Пример 6.4. Решить систему нелинейных уравнений

$$\begin{cases} 2 \cdot x + y + z = 6 \\ x^2 + y^2 + z^2 = 14 \\ x^3 - y^3 + z^3 = 36 \end{cases}$$

блочным методом. Начальные значения корней принять равными 1.

Реализация в MathCad:



<u>Тема 7.</u> Решение дифференциальных уравнений и систем в MathCad

Краткие теоретические сведения

Для решения дифференциальных уравнений с начальными условиями система ***Mathcad*** имеет ряд встроенных функций:

rkfixed – функция для решения ОДУ и систем ОДУ методом Рунге–Кутта четвертого порядка с постоянным шагом;

Rkadapt – функция решения ОДУ и систем ОДУ методом Рунге–Кутта с переменным шагом;

Odesolve – функция, решающая ОДУ блочным методом.

Ниже приведено описание стандартной функции ***rkfixed*** с указанием параметров функции.

rkfixed(y, x1, x2, p, D)

Аргументы функции:

y – вектор начальных условий из ***k*** элементов (k – количество уравнений в системе);

x1 и ***x2*** – левая и правая границы интервала, на котором ищется решение ОДУ или системы ОДУ;

p – число точек внутри интервала ***(x1, x2)***, в которых ищется решение;

D – вектор, состоящий из ***k***-элементов, который содержит первую производную искомой функции или первые производные искомых функций, если речь идет о решении системы.

Результатом работы функции является матрица из p+1 строк, первый столбец которой содержит точки, в которых получено решение, а остальные столбцы – сами решения.

На рисунке 2.7.1 приведены конкретные примеры решения различных дифференциальных уравнений и систем ОДУ в MathCAD.

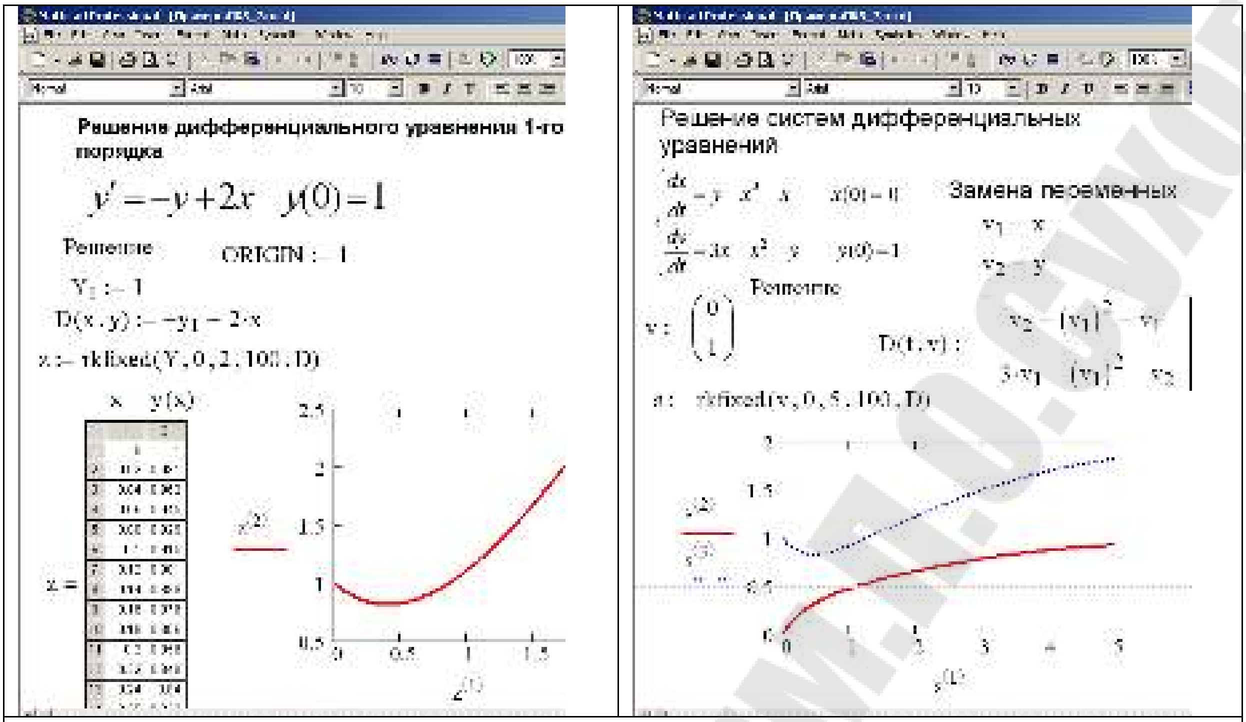


Рисунок 2.7.1 – Примеры решения дифференциальных уравнений и систем

При решении дифференциального уравнения первого порядка нужно создать вектор начальных условий из одного элемента Y_1, который затем используется при формировании вектора-функции правой части дифференциального уравнения. При обращении к функции ***rkfixed*** указывается имя вектора ***Y,*** границы интервала, на котором ищется решение уравнения, например, (0 ; 2), количество точек, в которых ищется решение – 100, вектор-функция, описывающая правую часть дифференциального уравнения – ***D***. В результате получается матрица ***z,*** в первом столбце которой содержатся значения аргумента искомой функции, во втором – значения самой результирующей функции. При построении графика функции первый столбец полученной матрицы указывается как аргумент, второй столбец – как функция.

При решении системы дифференциальных уравнений нужно создать вектор начальных условий из двух элементов, например, вектор ***v***, который затем используется при формировании вектора-функции правой части дифференциального уравнения. При обращении к функции ***rkfixed*** указывается имя вектора ***v,*** и границы интервала, на котором ищется решение уравнения, например, (0 ; 5), количество точек, в которых ищется решение – 100, вектор-функция, описывающая правую часть дифференциального уравнения – ***D***. В результате получается матрица ***s,*** в первом столбце которой содержатся значения аргумента искомых функций, во втором и третьем столбцах – значения самих функций при соответствующем значении аргумента. При построении графика можно воспользоваться первым столбцом полученной матрицы как аргументом, а вторым и третьим столбцами – как функциями.

На рисунке 2.7.2 приведен пример решения дифференциального уравнения второго порядка с использованием функции ***rkfixed***. Необходимо решить дифференциальное уравнение второго порядка с заданными начальными условиями вида:

$$y'' = -y' + 2y \quad y(0) = 1 \quad y'(0) = 3$$

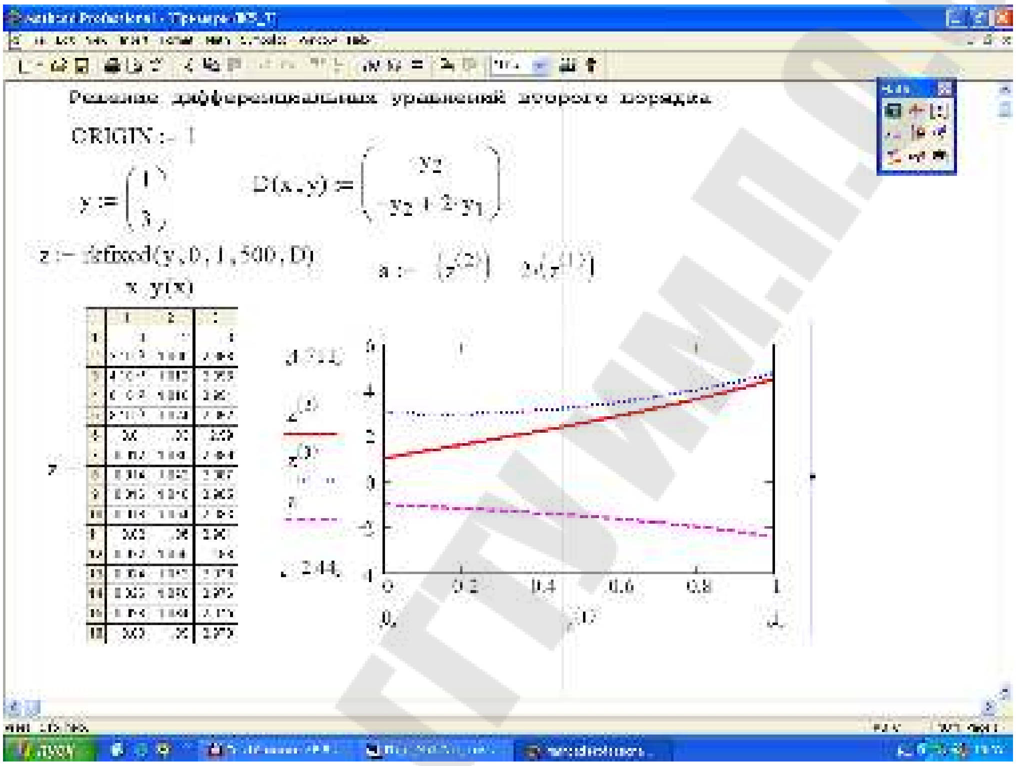


Рисунок 2.7.2 – Пример решения дифференциальных уравнений второго порядка с помощью ***rkfixed***

Для решения уравнения с помощью функции ***rkfixed*** нужно выполнить замену переменных и привести дифференциальное уравнение второго порядка к двум дифференциальным уравнениям первого порядка. Вид этих уравнений приведен ниже.

$$y - y_1 \quad \frac{d}{dx}y_1 = y_2$$

$$\frac{d}{dx}y_2 = -y_2 + 2 \cdot y_1$$

Документ формируется точно так же, как и при решении системы ОДУ.

На рисунке 2.7.2 показана возможность вычисления вектора второй производной найденной функции – вектора ***a***, построены графики исходной функции, функций первой и второй производных.

Практическая часть темы 7

7.1 Решение дифференциальных уравнений первого порядка

Последовательность действий для решения дифференциального уравнения первого порядка такова:

- сформировать вектор начальных условий из одного элемента, присвоив начальное значение искомой функции переменной с индексом, например: $Y_1 :=$ или $Y_0 :=$ (в зависимости от значения переменной ***ORIGIN***);
- определить вектор-функцию из одного элемента, которая содержит первую производную неизвестной функции:
  - набрать имя функции с двумя параметрами: первый параметр – аргумент искомой функции (независимая переменная), второй – имя вектора, содержащего искомую функцию (можно использовать имя вектора начальных условий), например, $D(x,Y)$;
  - набрать оператор «**:=**» и выражение для первой производной (выразить из дифференциального уравнения), в котором вместо имени искомой функции подставлен первый элемент вектора-параметра, например, для уравнения $y'(3 \cdot x - y) = y$ вектор-функция будет определятся следующим образом: $D(x,Y) := \dfrac{Y_1}{3 \cdot x - Y_1}$ (если ***ORIGIN*=0**, подставлять Y_0);
- присвоить некоторой переменной значение функции ***rkfixed***, указав в скобках следующие параметры:
  - первый – имя вектора начальных условий,
  - второй – левая граница интервала, на котором ищется решение, в виде числовой константы,
  - третий – правая граница интервала, на котором ищется решение, в виде числовой константы,
  - четвертый – количество точек, в которых ищется решение,
  - пятый – имя вектора-функции, описывающего первую производную, без параметров;

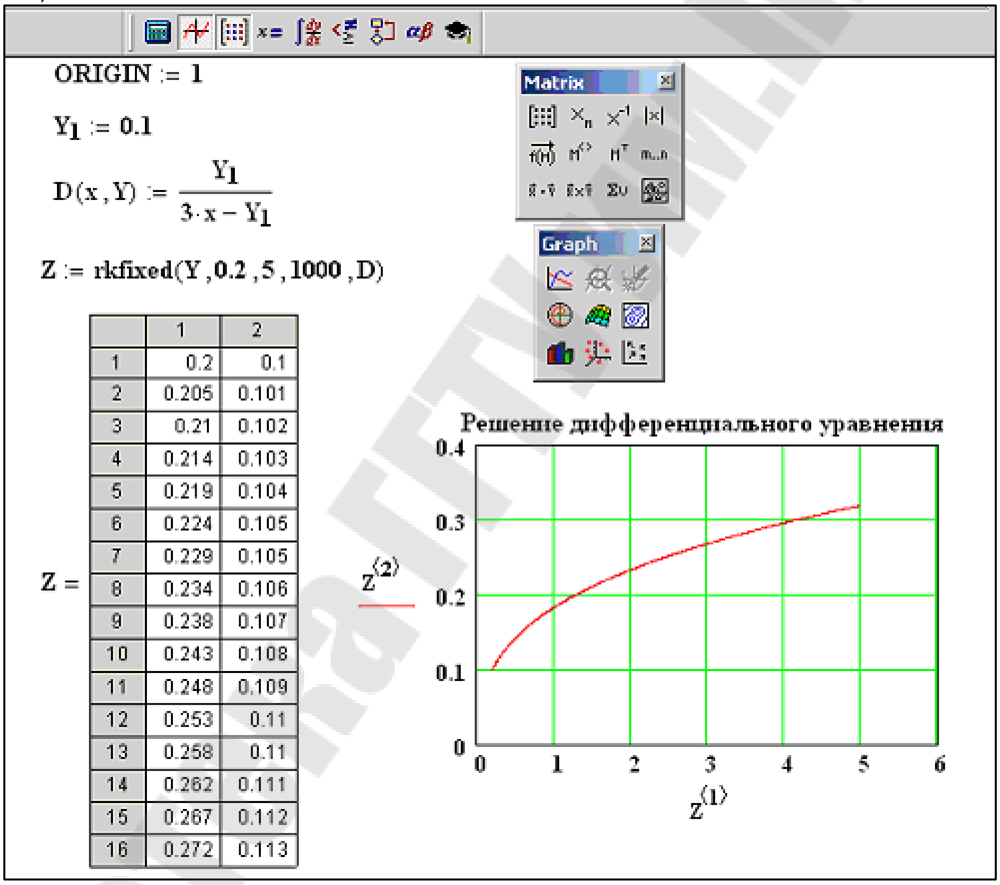
  например: $Z := rkfixed(Y, 0.2, 5, 1000, D)$,

  (в результате получится матрица *Z*, в первом столбце которой содержатся значения аргумента искомой функции, во втором – значения самой функции);

- вывести матрицу, содержащую решение ДУ с помощь оператора «=», например: $Z=$;
- построить график найденной функции (см. тему 5), указав в качестве аргумента по оси абсцисс столбец $Z^{<1>}$, а в качестве значения функции по оси ординат – столбец $Z^{<2>}$ (если ***ORIGIN*=0**, набирать соответственно $Z^{<0>}$ и $Z^{<1>}$).

Пример 7.1 Найти численное решение дифференциального уравнения первого порядка $y'(3\cdot x-y)=y$ на интервале от 0.2 до 5 в 1000 точках, при начальном условии y(0)=0.1.
Выполнить графическую интерпретацию результатов.

Реализация в MathCad:



	1	2
1	0.2	0.1
2	0.205	0.101
3	0.21	0.102
4	0.214	0.103
5	0.219	0.104
6	0.224	0.105
7	0.229	0.105
8	0.234	0.106
9	0.238	0.107
10	0.243	0.108
11	0.248	0.109
12	0.253	0.11
13	0.258	0.11
14	0.262	0.111
15	0.267	0.112
16	0.272	0.113

7.2 Решение систем дифференциальных уравнений

Последовательность действий для решения системы дифференциальных уравнений первого порядка такова (описана для значения ***ORIGIN*=0**):

- перейти в исходной системе уравнений к однотипным обозначениям функций и выразить первые производные, например, систему $\begin{cases} x' + 2 \cdot y = 3 \cdot x \\ y't = 2 \cdot x + 8 \cdot y \end{cases}$ можно преобразовать в

$$\begin{cases} V_1' = 3 \cdot V_1 - 2 \cdot V_2 \\ V_2' = \dfrac{2 \cdot V_1 + 8 \cdot V_2}{t} \end{cases};$$

- в документе MathCad сформировать вектор начальных условий, количество элементов которого равно количеству уравнений системы, присвоив его некоторой переменной (см. тему 2);

  например, $V := \begin{pmatrix} 0.1 \\ 1 \end{pmatrix}$;

- определить вектор-функцию, которая содержит первые производные искомых функций:
  - набрать имя функции с двумя параметрами: первый параметр – аргумент искомых функций (независимая переменная), второй – имя вектора, содержащего искомые функции (можно использовать имя вектора начальных условий), например, $D(t, V)$;

  (<u>*Замечание*</u>: *если независимая переменная явно не присутствует в системе, то в качестве ее имени можно выбрать любую переменную*)
  - набрать оператор «**:=**» и вставить шаблон вектора, количество элементов которого равно количеству уравнений системы (см. тему 2)
  - набрать в качестве элементов вектора правые части системы уравнений, в которых искомые функции представлены соответствующими элементами вектора-параметра, например,

    $$D(t,V) := \begin{pmatrix} 3 \cdot V_1 - 2 \cdot V_2 \\ \dfrac{2 \cdot V_1 + 8 \cdot V_2}{t} \end{pmatrix};$$

- присвоить некоторой переменной значение функции ***rkfixed***, указав в скобках следующие параметры:
  - первый – имя вектора начальных условий,
  - второй – левая граница интервала, на котором ищется решение, в виде числовой константы,
  - третий – правая граница интервала, на котором ищется решение, в виде числовой константы,
  - четвертый – количество точек, в которых ищется решение,
  - пятый – имя вектора-функции, описывающего первые производные, без параметров;

  например: $Z := rkfixed(V, 0, 0.5, 1000, D)$, (в результате получится матрица Z, в первом столбце которой содержатся значения аргумента искомых функций, во втором – значения первой функции, в третьем – значения второй функции и т.д.);

- вывести матрицу, содержащую решение системы ДУ с помощь оператора «=», например: $Z =$;
- построить графики найденных функций (см. тему 5), указав в качестве аргумента по оси абсцисс первый столбец матрицы решений, например, $Z^{<1>}$, а в качестве значений функций по оси ординат – остальные столбцы матрицы через запятую, например, $Z^{<2>}$, $Z^{<3>}$ и т.д.

Пример 7.2 Найти решение системы дифференциальных уравнений

$$\begin{cases} x' = 3\cdot x - 2\cdot y \\ y' = 2\cdot x + 8\cdot y \end{cases}$$

на интервале от 0 до 0.5 в 1000 точках, при следующих начальных условиях: x(0)=0.1 и y(0)=1.

Выполнить графическую интерпретацию результатов.

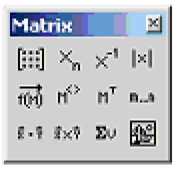
Реализация в MathCad:

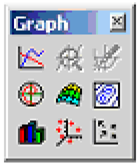
ORIGIN := 1

$$V := \begin{pmatrix} 0.1 \\ 1 \end{pmatrix}$$

$$D(t, V) := \begin{pmatrix} 3\cdot V_1 - 2\cdot V_2 \\ 2\cdot V_1 + 8\cdot V_2 \end{pmatrix}$$

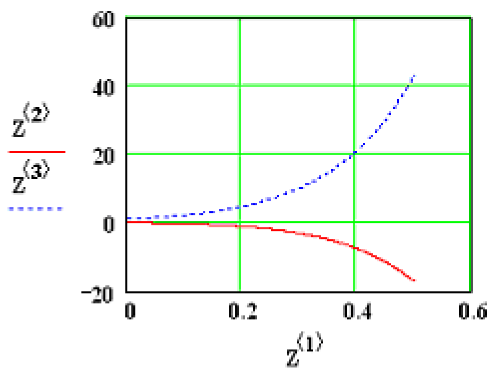
Z := rkfixed(V, 0, 0.5, 1000, D)





Z =

	1	2	3
1	0	0.1	1
2	$5\cdot10^{-4}$	0.099	1.004
3	$1\cdot10^{-3}$	0.098	1.008
4	$1.5\cdot10^{-3}$	0.097	1.012
5	$2\cdot10^{-3}$	0.097	1.017
6	$2.5\cdot10^{-3}$	0.096	1.021
7	$3\cdot10^{-3}$	0.095	1.025
8	$3.5\cdot10^{-3}$	0.094	1.029
9	$4\cdot10^{-3}$	0.093	1.033
10	$4.5\cdot10^{-3}$	0.092	1.038
11	$5\cdot10^{-3}$	0.091	1.042
12	$5.5\cdot10^{-3}$	0.09	1.046
13	$6\cdot10^{-3}$	0.089	1.05
14	$6.5\cdot10^{-3}$	0.088	1.055
15	$7\cdot10^{-3}$	0.088	1.059



СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Трохова Т.А. Практическое пособие по теме «Основные приемы работы в системе MathCad, версии 6.0» курса «ВТ и программирование» для студентов всех специальностей дневного и заочного отделений. – Гомель: ГГТУ, 1998. – 42с.

2 Новиков А.А. Практическое пособие к лабораторным и контрольным работам по теме «Решение инженерно-экономических задач в среде MathCad for Windows» курса «Информатика» для студентов заочного отделения. – Гомель: ГГТУ им.П.О.Сухого, 2000. – 46с.

3 Грудецкий Г.А., Мурашко И.А. Графические средства пакета MathCad: Практическое пособие для студентов всех специальностей дневного и заочного отделений. – Гомель: ГГТУ им.П.О.Сухого, 2001. – 36с.

4 Математический пакет MathCad: Практикум по курсу «Информатика» к лабораторным работам для студентов всех специальностей заочного отделения / Грудецкий Г.А., Коробейникова Е.В., Самовендюк Н.В., Трохова Т.А., Токочаков В.И. – Гомель: ГГТУ им П.О.Сухого,2003. – 49с.

5 Токочаков В.И. Практическое пособие по теме «Решение систем алгебраических и дифференциальных уравнений в среде MathCAD Windows» для студентов всех специальностей дневного и заочного отделений. – Гомель: ГГТУ им П.О.Сухого,2000. – 26с.

6 Дьяконов В.П. Справочник по MathCad Plus 7.0 PRO. – М.: СК Пресс, 1997. – 352с.

СОДЕРЖАНИЕ

РЕШЕНИЕ ЗАДАЧ В СИСТЕМЕ MATHCAD

Методические указания к контрольным работам по курсу «Информатика» для студентов технических специальностей заочной формы обучения

Авторы-составители: **Трохова** Татьяна Анатольевна
Романькова Татьяна Леонидовна
Стрижак Инесса Владимировна

Подписано в печать 18.12.06.
Формат 60x84/16. Бумага офсетная. Гарнитура Таймс.
Ризогрфия. Усл. печ. л. 3,14. Уч.-изд. л. 3,56.
Изд. № 111.
E-mail: ic@gstu.gomel.by
http://www.gstu.gomel.by

Отпечатано с макета оригинала авторского
для внутреннего использования.
Учреждение образования «Гомельский государственный технический
университет имени П. О. Сухого».
246746, г. Гомель, пр. Октября, 48, т. 47-71-64.