МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Филиал в г. Дербенте

Кафедра естественногуманитарных, общественных

и специальных дисциплин

Э.Т.Эмирбеков

УЧЕБНО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ «ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ. ЛАБОРАТОРНЫЙ ПРАКТИКУМ В СРЕДЕ MATHCAD»

Для студентов направления подготовки бакалавров
09.03.03 Прикладная информатика

Махачкала - 2020

УДК 519.61
ББК 22.193
Э 54

Эмирбеков Э.Т.

Вычислительные методы. Лабораторный практикум в среде MathCad : учебно-методические указания / Э.Т. Эмирбеков. – Махачкала: Изд-во ДГТУ, 2020. – 160 с.

Лабораторный практикум содержит 10 работ, в которых рассматриваются: знакомство с программным пакетом MathCAD, программирование в среде MathCAD, организация простейших вычислений, методы оценки погрешностей приближенных вычислений, численные методы решения линейных и нелинейных уравнений и их систем, приближение функций, численное интегрирование и численное решение дифференциальных уравнений. В каждой лабораторной работе приведены: краткие теоретические сведения по изучаемой теме, возможности программного пакета MathCAD, вопросы для самоконтроля, индивидуальные задания для выполнения работы и литература.

Учебно-методические указания предназначены для студентов направления 09.03.03 «Прикладная информатика».

Рецензенты: к. ф.м.-н., доцент кафедры ЭиУ МФ МАДИ К. О. Курбанов, к. ф.- м.н., ст. пр. кафедры ЕГО и СД филиала ДГТУ в г. Дербенте Ганиев А.С.

Печатается по решению Ученого совета ДГТУ от______________2020 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Вычислительные методы есть раздел прикладной математики, изучающий приближенные способы решения типовых математических задач, которые либо не решаются, либо трудно решаются точными аналитическими методами. Основное содержание дисциплины «Вычислительные методы» составляют численные методы, представляющие собой итерационные процедуры, расчетные формулы, алгоритмы переработки информации с целью нахождения приближенного решения рассматриваемой задачи. Вычислительные методы носят компьютерно-ориентированный характер, т.к. решение любой задачи в конечном итоге связано с применением вычислительной техники и программирования. В настоящем пособии представлены лабораторные работы, целью проведения которых является ознакомление студентов с программным пакетом MathCAD и численными методами решения практических задач.

Лабораторные работы нацелены на выработку у студентов навыков, необходимых при решении проектных и научных задач с использованием ЭВМ. Для выполнения расчетов используется математически ориентированная система MathCAD.

Пособие предназначено студентам направлений «Прикладная информатика» по дисциплине «Вычислительные методы». Оно также будет полезно и студентам других направлений подготовки.

Общие требования к выполнению лабораторных работ

Выполненная лабораторная работа должна содержать рукописный и электронный отчёты.

Первая теоретическая часть рукописного отчёта оформляется при подготовке к лабораторной работе и должна содержать краткое изложение теории по теме работы.

Вторая практическая часть оформляется после допуска к работе. Отчет по работе должен содержать решения всех предложенных задач и оформлен в редакторе MS Word в соответствии с ГОСТом по типовым правилам составления отчетов, принятым в вузе.

Электронный отчёт должен формироваться в процессе выполнения работы и содержать: номер и тему выполняемой лабораторной работы; номера выполняемых заданий; сопроводительные тексты.

1. Лабораторная работа №1.
ЗНАКОМСТВО С ПРОГРАММОЙ MATHCAD И ОРГАНИЗАЦИЯ ПРОСТЕЙШИХ ВЫЧИСЛЕНИЙ

1.1 Ознакомление с интерфейсом MathCad

Интерфейс Mathcad по своей структуре аналогичен интерфейсу других Windows-приложений.

1.1.1 Рабочее окно Mathcad

При открытии файла Mathcad.exe на экране появляется рабочее окно Mathcad с главным меню и пятью панелями: *Standard* (*Стандартная*), *Formatting (Форматирование), Math (Математическая).*

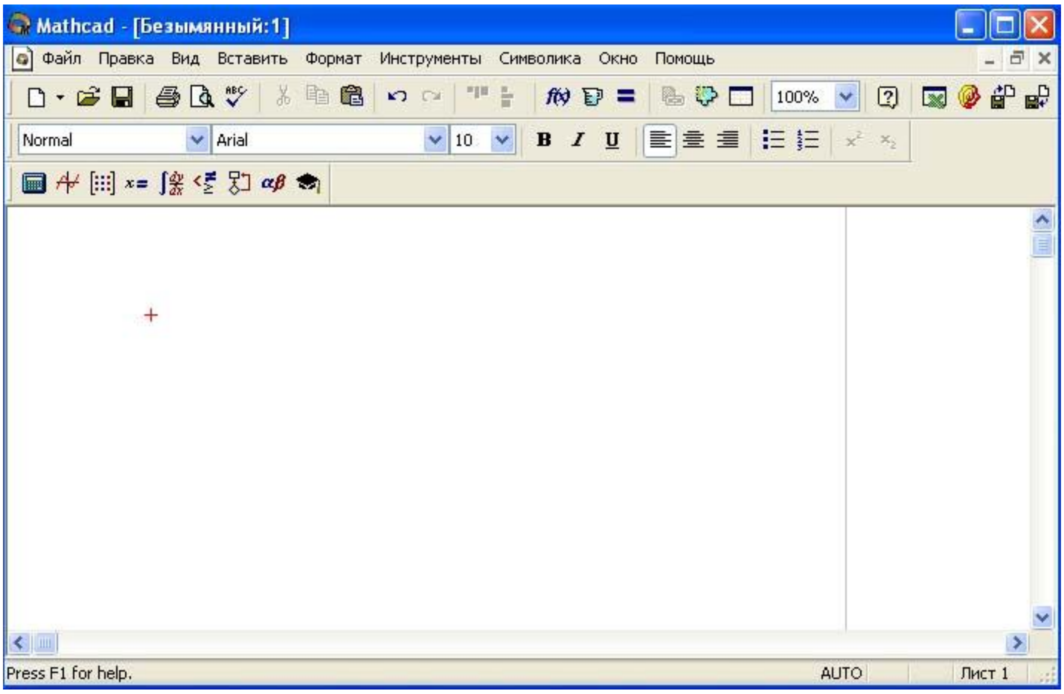


Рис. 1. Рабочее окно Mathcad

Автоматически загружается файл *Untitled:1* (*Безымянный:1*), представляющий собой шаблон *Normal* (*Обычный*) рабочего документа Mathcad, называемого *Worksheet* (*Рабочий лист*). Кроме того, автоматически загружаются окна *Tipof the day (Совет дня) и Mathcad Resource (Документация Mathcad). Перед нача*лом работы их надо закрыть. В окне *Tip of the day* следует снять флажок *Show tip at startup* (*Показывать совет при загрузке*) и щелкнут на кнопке *Close* (*Закрыть).*

1.1.2 Главное меню

Главное меню Mathcad занимает верхнюю строку рабочего окна. Все необходимые действия можно выполнить, следуя пунктам этого меню и последовательно открывающихся окон.

1.1.3 Управление рабочим окном Mathcad

1.***File*** **(*Файл*)** – команды, связанные с созданием, открытием, сохранением, пересылкой по электронной почте и печатью на принтере файлов с документами.

2. ***Edit*** **(*Правка*)** – команды, относящиеся к правке текста (копирование, вставка, удаление фрагментов и т. д.).

3. ***View*** **(*Вид*)** – команды, управляющие внешним видом документа в рабочем окне Mathcad, а также команды создания файлов анимации.

4. ***Insert*** **(*Вставка*)** – команды вставки различных объектов в документ.

5. ***Format*** **(*Формат*)** – команды форматирования текста, формул и графиков.

6. ***Tools*** **(*Инструменты*)** – команды управления вычислительным процессом.

7. ***Symbolics*** **(*Символьные вычисления*)** – команды символьных вычислений.

8.***Window*** **(*Окно*)** – команды, позволяющие управлять расположением окон с различными документами на экране.

9. ***Help*** **(*Помощь*)** – команды вызова справочной информации.

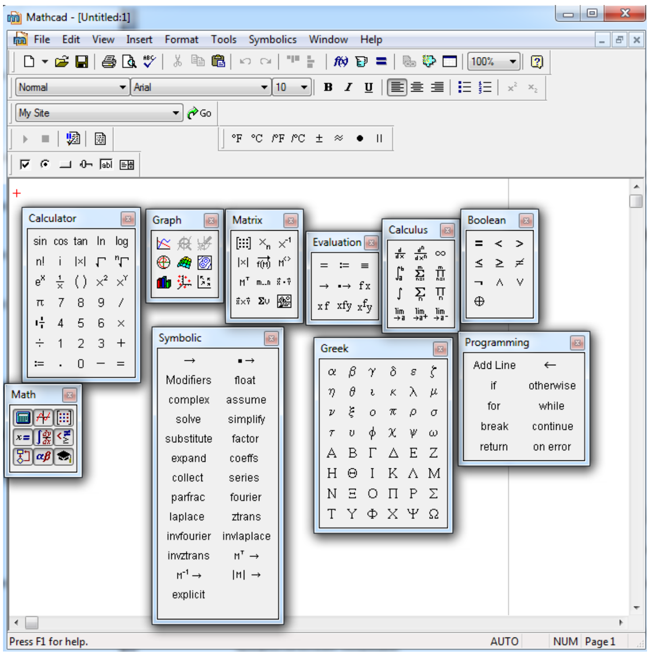


Рис. 2. Математическая панель и её подчиненные панели

В состав математической панели входят:

✓***Calculator*** (*Калькулятор*) – вставка шаблонов основных математических операций, цифр, знаков арифметических операций.

✓***Graph*** (*График*) – вставка шаблонов графиков.

✓***Matrix*** (*Вектор и матрица*) – вставка шаблонов векторов и матриц а также матричных операций.

✓***Evaluation*** (*Вычисления*) – операторы присвоения значений и вывода результатов расчета.

✓***Calculus*** (*Математический анализ*) – вставка шаблонов дифферен-цирования, интегрирования, суммирования.

✓***Boolean*** (*Булева алгебра*) – вставка логических (булевых) операторов.

✓***Programming*** (*Программирование*) – операторы, необходимые для созда-ния программных модулей.

✓***Greek*** (*Греческие символы*) – вставка греческих символов.

✓***Symbolic*** *(Символьные преобразования с ключевыми словами) – встав*ка операторов символьных вычислений

Recources *(Дополнительные ресурсы) содержит список электронных книг, включенных в оболочку Mathcad.*

Controls (*Контроль*) содержит кнопки для дополнительного контроля работы Mathcad-документа.

Debug (*Отладка*) появилась в Mathcad 13, служит для трассировки выполнения программ.

1.1.4 Ввод, редактирование и форматирование выражений

Перед началом работы курсор на экране имеет вид крестика. В момент ввода выражения курсор приобретает вид синего уголка, обрамляющего вводимое выражение. В рабочем документе нужно вводить какие-либо буквенные выражения и присваивать им численные значения. Имя выражения может состоять из латинских, русских, греческих и других букв и цифр, знаков подчеркивания [_], штриха [′], символа процента [**%**], вводимых с клавиатуры.

Имена переменных и функций не могут начинаться с цифры, знака подчеркивания, штриха, символа процента, не могут включать в себя пробелы. Символ бесконечности может быть только первым символом в имени.

Mathcad не делает различий между именами переменных и функций. Если вначале определить функцию *f(x)*, а затем – переменную *f*, окажется невозможно воспользовать *f(x)* в расчетах где-либо после определения *f*.

1.1.5 Курсоры Mathcad

Визир – + (крестообразный курсор) – используется для размещения новых выражений, графиков и текстовых областей. Визир может появляться только в свободном месте документа. При начале печати курсор принимает другую форму. Для его перемещения необходимо щелкать в свободном месте документа или используйте клавиши перемещения ([←], [→], [↑], [↓]). Для перемещения визира на четверть высоты окна вверх используются сочетания клавиш [**PageUp**] и [**PageDn**]. Чтобы перемещать визир вверх или вниз на высоту окна необходимо использовать сочетания клавиш [**Ctrl**]+[**PagegUp**] и [**Ctrl**]+[**PageDn**].

Маркер ввода – | – используется в выражениях для вставки и удаления отдельных символов, скобок и операторов. Может использоваться в области любого типа. В тексте для его перемещения можно использовать все четыре клавиши перемещения – [←],[→], [↑], [↓] – или щелчком мыши перемещаться в другую точку ввода текста. В выражении можно использовать клавиши [←] или [→] или щѐлкнуть в имени или числе.

Выделяющая рамка –  – используется в выражениях для вставки и удаления операторов, чисел, а также имен функций и переменных. Может использоваться только в математической и графической областях. Угол указывает направление вставки или удаления. Для изменения направления необходимо нажать клавишу **[Ins].**

1.1.6 Константы и переменные

Константами называются поименованные объекты, хранящие некоторые значения, которые не могут быть изменены.

В MathCAD применяются десятичные, восьмеричные и шестнадцатеричные числовые константы. Десятичные константы могут быть целочисленными, вещественными, заданными с фиксированной точкой, и вещественными, заданными в виде мантиссы и порядка.

В MathCAD содержится особый вид констант - размерные. Помимо своего числового значения они характеризуются еще и указанием на то, к какой физической величине они относятся. Для этого указания используется символ умножения. В системе MathCAD заданы следующие основные типы физических величин: time (время), length (длина), mass (масса) и charge (заряд). При необходимости их можно изменить на другие.

Переменные являются поименованными объектами, которым присвоено некоторое значение, которое может изменяться по ходу выполнения программы. Тип переменной определяется ее значением; переменные могут быть числовыми, строковыми, символьными и т. д. Имена констант, переменных и иных объектов называют идентификаторами.

Имя переменной называется идентификатором. MathCAD различает в идентификаторах символы верхнего и нижнего регистров. Например: ABC и AbC имена разных переменных.

Идентификаторы MathCAD должны начинаться с буквы и могут содержать следующие символы:

-латинские буквы любого регистра;

-арабские цифры от 0 до 9;

-символ процент (%) .

1.1.7 Определение переменных

Переменные должны быть предварительно определены пользователем, т. е. им необходимо хотя бы однажды присвоить значение. В качестве оператора присваивания используется знак :=, тогда как знак = отведен для вывода значения константы или переменной. Попытка использовать неопределенную переменную ведет к выводу сообщения об ошибке.

В MathCAD различают: локальные и глобальные переменные.

Локальные переменные вводятся:

Имя_переменной **:=** выражение

Глобальные переменные вводятся:

Имя_переменной ≡ выражение

Если переменной присваивается начальное значение с помощью оператора **:=**, такое присваивание называется локальным. До этого присваивания переменная не определена и ее нельзя использовать. MathCAD читает рабочий документ слева направо и сверху вниз, поэтому определив переменную, ее можно использовать в вычислениях везде правее и ниже равенства, в котором она определена. Однако с помощью знака ≡ (три горизонтальные черточки) можно обеспечить глобальное присваивание, т. е. оно может производиться в любом месте документа. К примеру, если переменной присвоено таким образом значение в самом конце документа, то она будет иметь это же значение и в начале документа.

Например:

local := 137 локальное определение переменной local;

global ≡ 987.23 глобальное определение переменной global.

1.1.8 Дискретные переменные

Выражение «дискретная переменная» – диапазон изменения переменной: вместо непрерывной переменной используется ряд чисел, выстроенных в порядке возрастания или убывания. Дискретная переменная задает ряд значений переменной, для которых вычисляется функция пользователя. Этот ряд значений функции можно вывести в виде графика или таблицы.

Определение дискретной переменной имеет вид x:= 0..5, что означает задание ряда значений x = 0, 1, 2, 3, 4, 5. Дискретная переменная может задавать как целые, так и дробные значения переменной, но **обязательно равноотстоящие** друг от друга, например: x:= 0..5 – ряд целых чисел от 0 до 5; x:= 1,1.1..5 – ряд дробных чисел, где 1 – первое число, 1.1 – второе число, 5 – последнее число. Интервал между числами 1.1-1=0.1; *x* := min, min .. max – ряд чисел, где min – первое, шаг (интервал), max – последнее число (min, и max должны быть заданы заранее). Такая форма записи удобна, когда рассматриваются разные варианты одного расчета и изменение констант позволяет мгновенно пересчитать результаты и перестроить графики.

Если константы, входящие в правую часть функции пользователя, не задаются непосредственно перед использованием функции, Mathcad берет их значения, использовавшиеся в последний раз перед данным расчетом. Необходимо проверить эти значения. Следует набрать имя параметра и нажать клавишу (=) (рисунок 3).

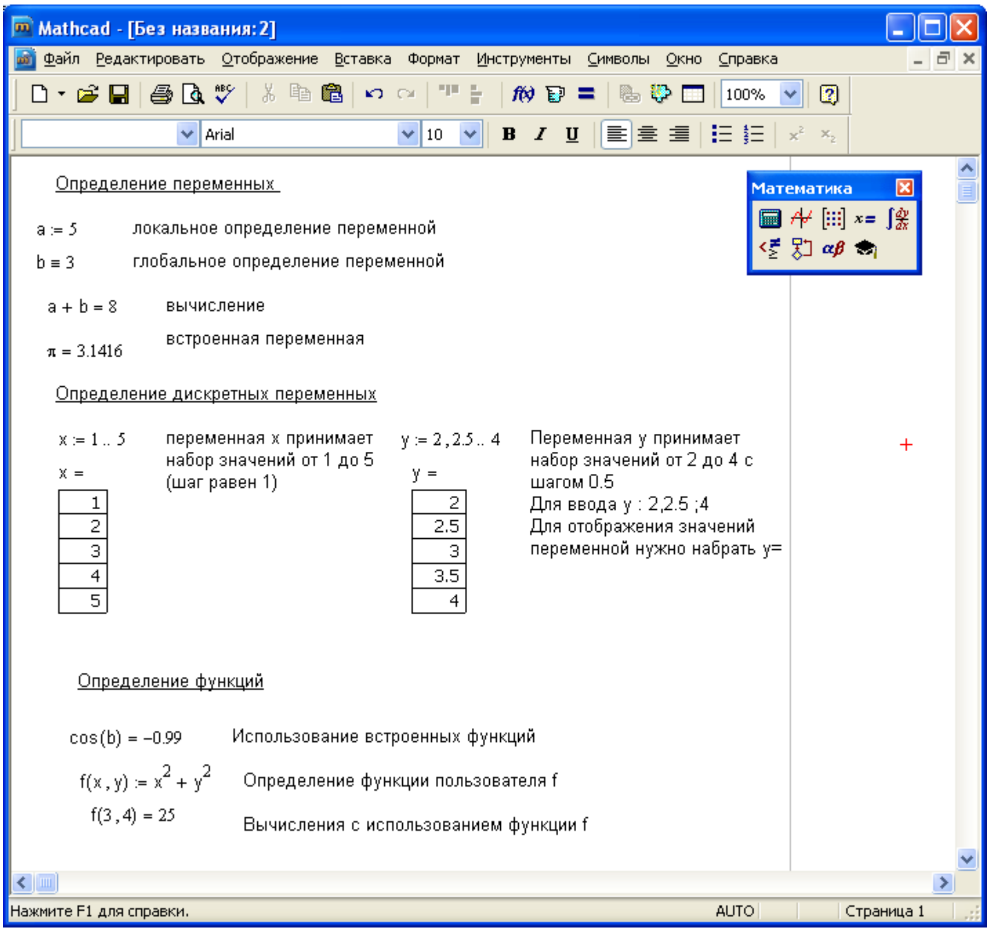


Рис.3. Определение переменных

1.1.9 Вычисления с комплексными числами

Mathcad работает не только с действительными, но и с комплексными числами. Комплексные числа представляются в форме: a + b i, где a и b – обычные числа с плавающей точкой; i – мнимая единица (это обозначение используется по умолчанию).

Чтобы записать комплексное число, например 4.68+5.19i, выполните следующие действия:

Введите действительную часть числа; нажмите [+]; введите мнимую часть числа: 5.19i. Обратите внимание, что перед “i” знак умножения не ставится, в противном случае Mathcad интерпретирует “i” как имя переменной. Вы также можете выбрать мнимую единицу из палитры “Calculator”.

1.1.10 Предопределенные переменные

Предопределенные (системные) переменные – особые переменные, которым изначально системой присвоены начальные значения, приведены в таблице 1.

Таблица 1

Переменная	Ввод	Назначение	Значение по умолчанию
π	Ctrl + Shift +p	Число π	3.14159
e	e	Основание натур. логарифма	2.718
∞	Ctrl + Shift + z	Системная бесконечность	10^{307}
i или j	1i или 1j	Мнимая единица	
%		Процент	0.01
TOL		Погрешность численных методов	0.001
ORIGIN		Нижняя граница индексации массивов	0

1.1.11 Операторы

Операторы- элементы языка, с помощью которых можно создавать математические выражения. К ним, например, относятся символы ариф-метических и логических операций, знаки вычисления сумм, произве-дений, производной и интеграла и т. д. (см табл. 2).

Операторы, обозначающие основные арифметические действия, вводятся с панели Calculator (Калькулятор, Арифметика).

Вычислительные операторы вставляются в документы при помощи панели инструментов Calculus. При нажатии любой из кнопок в документе появляется символ соответствующего математического действия, снабженный несколь-кими местозаполнителями. Количество и расположение местозапол-нителей определяется типом оператора и в точности соответствует их общепринятой математической записи.

Результатом действия логических, или булевых, операторов являются только числа 1 (если логическое выражение, записанное с их помощью, истинно) или 0 (если логическое выражение ложно).

Вычислительные операторы сгруппированы на панели Evaluation (Вычисления):

-Численный вывод (Evaluate Numerically) =

-Символьный (аналитический) вывод (Evaluate Symbolically)→

-Присваивание (Definition) :=

-Глобальное присваивание (Global Definition) ≡

Таблица 2

Оператор	**Клавиша**	**Назначение оператора**
X := Y	X : Y	Локальное присваивание X значения Y
X ≡Y	X ~ Y	Глобальное присваивание X значения Y
X =	X =	Вывод значения X
X + Y	X + Y	Сложение X с Y
X - Y	X - Y	Вычитание из X значения Y
X · Y	X * Y	Умножение X на Y
$\frac{X}{Y}$	X/Y	Деление X на Y
X ÷ Y	Ctrl + /	Линейное деление
a$\frac{b}{c}$	Ctrl + Shift + +	Дробь (смешанный номер)
z^k	Z^k	Возведение z в степень k
$\sqrt{z}$	z|	Вычисление квадратного корня из z
n!	n!	Вычисление факториала

Bn	B [n	Ввод нижнего индекса n
An,m	A [n , m	Ввод двойного нижнего индекса
$A^{<n>}$	Ctrl + 6 n	Ввод верхнего индекса (для векторов)

1.1.12 Организация символьных вычислений

MathCad выполняет многие математические преобразования и операции математического анализа. Для этих целей разработан специальный символьный редактор, который активизируется щелчком по пункту **Symbolics** (символьные вычисления) главного меню. Символьные операции выполняются только с выделенными фрагментами. Меню **Symbolics** содержит операции символьной математики. Приведём некоторые из них:

Expand (развернуть) – раскрыть скобки, перемножить и привести подобные;

Solve – найти значения выделенной переменной, при которых содержащее её выражение становится равным нулю;

Symplify (упростить) – выполнить арифметические операции, привести подобные, сократить дроби, использовать для упрощения основные тождества (формулы сокращённого умножения, тригонометрические тождества и т.п.);

Substitute – заменить выделенную переменную содержимым буфера обмена;

Series – разложить в ряд Тейлора в окрестности заданной точки (в позиции ввода шаблона надо набрать переменную и её значение);

Faktor (разложить на множители) – представить, если возможно, выражение в виде произведения простых сомножителей.

Следующие примеры выполните самостоятельно:

1. Вычислите: $\int x^3 dx$; $\lim_{x\to 0}\frac{\sin x}{x}$; $\frac{d^2}{dx^2}x^3$; $\sum_{n=0}^{\infty}\frac{x^{n-1}}{(2n)!}$.
2. Раскройте скобки: $(x^2-y^2)(x^3-y^3)$
3. Разложите на множители: (x^3-y^3)
4. Упростите выражение: $(\frac{3}{4x-2})(2-\frac{8x^2}{4x+2})$
5. Разложите выражения f(x)=$\cos(x)$ *и* f(x)=$\sin(\frac{x}{x+2})$ в ряд Тейлора в окрестности точки x=0.

1.1.13 Определение функций

Функция – выражение, согласно которому проводятся некоторые вычисления с его аргументами и определяется его числовое значение.

Функции в пакете MathCAD могут быть встроенные и определенные пользователем.

В MathCAD имеется множество встроенных функций. Для их ввода используется команда меню Вставка → Функция или кнопка на панели инструментов $f(x)$. В диалоговом окне нужно выбрать Категорию и соответствующую функцию.

Функция пользователя вначале должна быть определена, а затем к ней может быть произведено обращение.

Функция пользователя определяется следующим образом:

Имя_функции (Переменная1, Переменная2, ...) := Выражение

Задается имя функции, в скобках указывается список аргументов функции - это перечень используемых в выражении переменных, разделяемых запятыми. Затем записывается знак присваивания, справа от которого записывается выражение. Выражение - это любое арифметическое выражение, содер-жащее доступные системе операторы и функции с операндами и аргументами, указанными в списке аргументов.

Вычисление значения функции в одной точке осуществляется набором функции в общем математическом виде f(x):= <выражение> и оператором обращения к ней при конкретном значении аргумента. Например,
f(x):=x+3.5 f(5)=

В функциях многих переменных аргументы разделяются запятой. Например,

F(x, y):=<выражение> F(3.5, 2)=

1.1.14 Тригонометрические функции

1. Вычисление значений в радианах.

1.1Значение аргумента: $x = 3.34$

1.2 Соответствующие значения тригонометрических функций:

$\sin(3.34) = -0.197$	$\cos(3.34) = -0{,}98$
sec(3.34) =−1.02	csc(3.34) =−5.073
tan(3.34) = 0.201	cot(3.34) = 4.974

2. Вычисление значений в градусах.

2.1 Значение аргумента: x=. 300 x = x.deg $x = .300\ x = x.deg$

2.2 Соответствующие значения тригонометрических функций:

$\sin(x) = 0.866$	$\cos(x) = 0.5$
$sec(x) = 2$	$csc(x) = 1.155$
tan(x)=1.732	$cot(x) = 0.577$

1.1.15 Логарифмы

1. Вычисляет натуральные и по основанию 10 логарифмы от положи-

тельных вещественных чисел.

1.1 Ввод положительного числа: $x = 5.67$

1.2 Вычисление натурального логарифма: ln(x) = 1.735

1.3 Вычисление логарифма по основанию10: log(x) = 0.754

2. Логарифмы по произвольному основанию.

2.1 Ввод положительного числа: $x = 12.78$ x

2.2 Ввод основания: $b = 2$

2.3 Вычисление логарифма по данному основанию:

logb(b,x)=ln(x)/ln(b); logb(b, x) = 3.676

1.1.16 Редактирование формул

Замена символа:

Если необходимо исправить несколько символов:

1. Щѐлкнуть на имени или числе. Если необходимо, нажать клавишу [↓], чтобы превратить выделяющую рамку в маркер ввода. Для перемещения маркера использовать клавиши [←] и [→].

2. Напечатать буквы или цифры. Чтобы удалить символ слева от маркера, нажать [Backspace].

Другой способ:

1. Щѐлкнуть на имени или числе. Если необходимо, нажать клавишу [↑], чтобы превратить маркер ввода в выделяющую рамку и заключить в неѐ имя или число.

2.Напечатать буквы или цифры. Чтобы удалить символ слева от маркера, нажать [**Backspace]**.

3.Щѐлкнуть в поле ввода и напечатать новое имя или число.

Вставка оператора:

1. Щѐлкнуть на операнде, чтобы заключить его в выделяющую рамку. Если нужно увеличить рамку следует использовать клавишу [↑].

2. Набрать комбинацию клавиш (сумма, разница, умножить, разделить), задающую оператор.

3. Чтобы вставить оператор перед выделенным выражением нужно нажать клавишу **[Ins]** прежде, чем начинать печатать.

Замена оператора:

1. Выделить операнд перед/после заменяемого оператора.

2. Нажать [**Backspace]/[Del]**, чтобы удалить оператор.

3. Напечатать новый оператор.

Вставка и удаление скобок:

1. Заключить выражение в выделяющую рамку.

2. Нажать апостроф [**'**] чтобы вставить скобки.

3. Нажать [**Del**], чтобы удалить пару скобок.

Вставка и удаление знака:

1.Щѐлкнуть на выражении, чтобы заключить его в выделяющую рамку. Использовать [↑], если нужно увеличить выделяющую рамку.

2. Для вставки знака нажать [**Ins**].

3. Нажать знак минус.

4. Для удаления знака заключить выражение целиком, включая знак минус,

в выделяющую рамку. Далее нажать [**Backspace].**

Перемещение частей формулы:

1. Заключить выражение, которое следует переместить, в выделяющую рамку.

2. Выполнить команду Вырезать или Копировать из меню Правка, чтобы вырезать или скопировать выражение в буфер обмена.

3. Щѐлкнуть в свободном пространстве или в поле ввода, куда необходимо вставить выражение.

4. Выполнить команду Вставить из меню Правка, чтобы вставить выражение.

Применение функции к выражению:

1. Заключить выражение в выделяющую рамку.

2. Нажать апостроф [**'**].

3. Нажать [**Пробел**] чтобы заключить скобки в рамку.

4. Нажать [**Ins**].

5. Напечатать имя функции.

1.1.17 Форматирование чисел

В Mathcad на результат расчета повлиять нельзя, но можно изменить формат вывода чисел. Mathcad вычисляет все выражения с точностью 20 знаков, но выводит на экран не все значащие цифры.

Установив указатель мыши на нужном численном результате расчета, нужно сделать двойной щелчок левой кнопкой мыши. Откроется окно форматирования чисел ***Result Format*** *(Формат результата),* открытое на пункте **Number Format** *(Формат чисел)* (рис. 4). В этом окне можно выбрать следующие форматы:

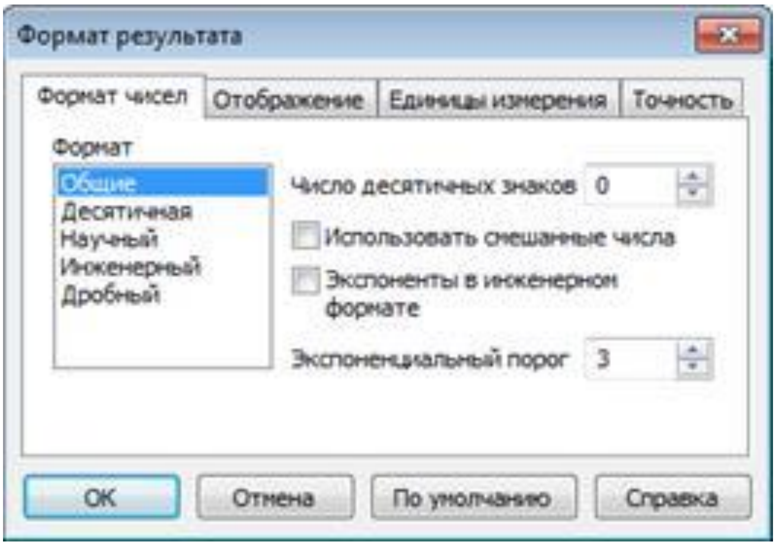


Рис. 4. Окно «Формат результата»

General *(Общий)* – принят по умолчанию. Числа отображаются с порядком. Количество знаков перед запятой определяется в пункте *Exponential threshold* (*Порог экспоненты).*

Scientific (*Научный*)- числа отображаются только с порядком $1.22 * 10^5$

Decimal (*Десятичный*) – десятичное представление чисел с плавающей запятой: 12,2564.

Engineering (*Инженерный*) – числа отображаются только с порядком, кратным 3: $1{,}22 \cdot 10^6$

Fraction **(***Дробь*) – в виде правильной или неправильной дроби: / 1 .

Вдробном формате можно выбрать уровень точности (*Level of accuracy*) исмешанные числа (*Use fixed number*).

Выбранные установки могут быть применены только к выделенному числу (если выбрать пункт *ОК*) или по умолчанию ко всем числам данного документа *(если выбрать пункт Set as default (Применить по умолчанию)).*

1.1.18 Редактирование текста

Создание текстовых областей:

1. Щѐлкнуть в месте создания области.

2. Выбрать команду *Регион текста* из меню *Вставка*. Появится текстовая рамка.

3. По мере ввода текста текстовая рамка будет увеличиваться. Она исчезнет, как только если щѐлкнуть вне текстовой области.

4. Чтобы покинуть текстовую область, необходимо щелкнуть вне еѐ . Не стоит нажимать **[Enter]**, это приведѐт к переходу на новую строку.

Замена символа:

-щѐлкнуть на имени или числе;

-для перемещения маркера необходимо использовать клавиши [←] и [→];

-напечатать буквы или цифры. Чтобы удалить символ слева от маркера, нужно нажать [**Backspace**].

Скопировать или вырезать текст:

1.Выделить область пунктирным выделяющим прямоугольником.

2.Выполнить команду *Копировать* из меню *Правка*, чтобы скопировать текст в *Буфер обмена* или выполнить команду *Вырезать*, чтобы вырезать.

Вставить текст:

1.Поместить курсор в то место, куда нужно вставить текст.

2.Выполнить команду *Вставить* из меню *Правка*.

Перемещение текста:

2.Выделить область пунктирным выделяющим прямоугольником.

3.Поместить указатель мыши внутрь прямоугольника.

4.Нажав и удерживая левую кнопку мыши, переместить мышь.

Прямоугольник переместится вслед за ней.

Изменение стиля текстовой области:

Чтобы настроить стиль текстовой области необходимо:

1. В главном меню выберите команду:

Style → Normal → Modify → Font

(Стиль → Обычный → Изменить → Шрифт).

2. В появившемся диалоговом окне выберите нужные шрифт (*Font*), форму шрифта (*Style Font*) и размер (*Size*).

1.2 Задания к лабораторной работе №1

Задание 1.2.1. Отвечать на контрольные вопросы

1. Как действует оператор присваивания?
2. Как выглядит оператор получения результата?
3. Как ввести греческие буквы?
4. Перечислить арифметические операторы.
5. Объяснить использование переменных.
6. Как записываются комплексные переменные.
7. Объяснить использование заглавных и строчных букв в именах переменных и функций?
8. Как записываются имена переменных с подстрочным обозначением

Задание 1.2. 2.

1. Рассчитать выражения в соответствии с вариантом, используя встроен-ные функции, вывести на экран вспомогательные слова. Ответ должен содержать m знаков после запятой; переменную x определить в соответствии с областью определения.

2. Получите таблицу значений функции на интервале [a, b] с шагом h.

Вариант 1

$$y = \frac{1+\sin^2\left(8+x^3\right)}{\sqrt[3]{8+x^3}}$$, ***m = 4, a = –5, b = 5, h = 1.***

Вариант 2

$$y = \frac{\sqrt[3]{x+16}}{\lg^2 x}$$, ***m = 3, a = 10, b = 14, h = 0.5.***

Вариант 3

$$y = \frac{1+\lg^2\frac{x}{10}}{1-e^{\frac{x}{2}}}$$, ***m = 2, a = 2, b = 8, h = 0.5.***

Вариант 4

$$y = \sqrt[4]{\left|x^2-2{,}5\right|}+\sqrt[3]{\lg x^2}$$, ***m = 4, a = 10, b = 15, h = 1.***

Вариант 5

$$y = \frac{2^x-3^x}{\lg\left|\frac{2}{3}\right|}\sqrt[3]{x}$$, ***m = 3, a = 3, b = 8, h = 1.***

Вариант 6

$$y = \frac{27+\sin^2 3x}{\arccos(2x)+e^{-x/2}}$$, ***m = 2, a = –3, b = 2, h = 1.***

Вариант 7

$$y = \frac{\lg\left(x^2-1\right)}{\log_5\left(4x^2-9\right)}$$, ***m = 4, a = 4, b = 11, h = 1.***

Вариант 8

$$y = \frac{\arccos\left(x^2-25\right)}{\arcsin\left(x^2-4\right)}$$, ***m = 3, a = –2, b = 3, h = 1.***

Вариант 9

$y = \arcsin\left(x^4\right) + \arccos\left(x^3\right)$, ***m = 2, a = –5, b = 5, h = 1.***

Вариант 10

$y = 5^{x^2-1} - \lg\left(x^2-1\right) + \sqrt[3]{x^2-1}$, ***m = 4, a = 5, b = 10, h = 1.***

Литература

1. Акимов, П.А. Информатика и прикладная математика: Учебное пособие / П.А. Акимов, А.М. Белостоцкий, Т.Б. Кайтуков и др. - М.: АСВ, 2016. - 588 с.

2. Квасов, Б.И. Прикладная математика в системе MATHCAD: Учебное пособие / Б.И. Квасов. - СПб.: Лань П, 2016. - 352 с.

3. Мышкис, А.Д. Прикладная математика для инженеров: Специальные курсы / А.Д. Мышкис. - М.: Физматлит, 2007. - 688 с.

4. Охорзин, В., А. Прикладная математика в системе MATHCAD: Учебное пособие / В. А. Охорзин. - СПб.: Лань, 2008. - 352 с.

5. Охорзин, В.А. Прикладная математика в системе MATHCAD / В.А. Охорзин. - СПб.: Лань, 2009. - 352 с.

6. Поршнев, С. В., Беленкова, И. В. Численные методы на базе Mathcad (+ CD) / С.В Поршнев, И.В. Беленкова. -С-Пб.: БХВ-Петербург, 2005. -456 с.

2. Лабораторная работа №2.
МАССИВЫ

2.1 Работа с массивами в среде MathCad

2.1.1 Создание массивов (матриц и векторов)

Математический программный пакет MathCad с большим успехом можно использовать для решения задач связанных с массивами. Ознакомимся с возможностями MathCad в создании и преобразовании массивов.

Пусть требуется создавать матрицу. Для этого поступают так:

1.Установить курсор в то место, где надо создать матрицу.

2.Щелкнуть на кнопке математической панели *Vector and Matrix Toolbar* (*Векторы и матрицы*), а в появившейся панели инструментов щелкнуть на кнопке *Matrix or Vector* (*Матрица или вектор*). Можно в главном меню

Mathcad выбрать команду *Insert → Matrix* (*Вставка → Матрица*) или нажать комбинацию клавиш [Ctrl]+[m].

3. В появившемся диалоговом окне вписать число строк (*Rows*) и число столбцов (*Columns*), а затем щелкнуть на кнопке *ОК*. На месте курсора появится шаблон матрицы.

4. В каждое место ввода вписать число, буквенную константу или функцию. Переход от одного места ввода к другому осуществляется клавишами со стрелками. Можно также щелкнуть мышью на нужном месте ввода, но это не так удобно, как перемещение с помощью клавиш.

2.1.2 Изменение размеров матрицы

Можно изменять размер матрицы, вставляя и удаляя строки и столбцы:

1. Щелкнуть мышью на крайнем элементе матрицы. Синий уголок курсора должен находиться справа от выделенного элемента.

2. Выбрать значок матрицы в математическом меню или нажать комбинацию клавиш [**Ctrl**]+[**m**]. В появившемся окне вписать число строк и столбцов, которые необходимо вставить или удалить.

3. Щелкнуть на кнопке *Insert* (*Вставить*) или *Delete* (*Удалить*), затем на *ОК*. Если необходимо вставить или удалить одну строку, то надо вписать количество строк – 1, количество столбцов – 0 и выбрать пункт *Insert* или - *Delete*.

Mathcad вставляет строки или столбцы вправо и вниз от выделенного элемента матрицы, удаляет строки и столбцы вправо и вниз, включая выделенный элемент.

Чтобы удалить всю матрицу, следует выделить ее черным цветом и нажать клавишу [**Delete**] (для безвозвратного удаления) или выбрать *Cut* на стандартной панели Mathcad (чтобы вырезать в буфер). Можно также выбрать команду *Cut* в контекстном меню, появляющемся при щелчке на выделенной матрице правой кнопкой мыши.

2.1.3 Нумерация элементов матрицы

Элементы матрицы определяются двумя нижними индексами, элементы вектора – одним.

Для ввода нижнего индекса можно щелкнуть в математической панели на кнопке *Xn Subscript* (Нижний индекс), но лучше использовать клавишу [**[**] (открывающая квадратная скобка), так как во время работы с матрицами вводить нижний индекс приходится очень часто.

Чтобы из матрицы выделить вектор (один из столбцов матрицы), используется верхний индекс. Для выполнения данной операции необходимо:

1.Ввести имя матрицы и выделить его синим уголком курсора.

2.В математическом меню щелкнуть на кнопке *Matrix and Vectors Toolbar* (*Панель векторов и матриц*) и щелкнуть на кнопке *M<> Matrix Column(Столбец матрицы).*

3. В появившемся месте ввода вписать номер столбца.

Пример:

$$\mathbf{M} := \begin{pmatrix} 2 & 3 & 1 \\ 4 & 2 & 0 \end{pmatrix} \quad \mathbf{V1} := \mathbf{M}^{\langle 0 \rangle} \quad \mathbf{V2} := \mathbf{M}^{\langle 2 \rangle} \quad \mathbf{V1} = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$$

2.1.4 Встроенная переменная ORIGIN

Начало нумерации элементов в векторах и матрицах определяется встроенной переменной ORIGIN. По умолчанию ORIGIN=0, то есть первый элемент вектора, первая строка и первый столбец матрицы имеют индекс нуль.

Чтобы изменить нумерацию индексов в первой строке документа, необходимо набрать (прописными буквами) ORIGIN:=l. Можно переопределить встроенную переменную ORIGIN другим способом. Для этого необходимо выбрать в главном меню команду *Tools → Worksheet Options-* (*Инструменты- → Параметры документа*), в открывшемся окне перейти на вкладку *Built-in Variables (Встроенные переменные) и в поле Array Origin (ORIGIN)* ввести индекс первого элемента массива. Переменной ORIGIN можно присваивать разные значения, в том числе отрицательные.

В Mathcad есть встроенные функции, в алгоритм которых по умолчанию включены с нулевым индексом, например, **mean** (вычисление среднего значе ния) или **fft** (разложение в ряд Фурье). Если **ORIGIN**:=l введено с клавиатуры, неизбежна ошибка в расчетах.

2.1.5 Определение параметров массивов

В Mathcad есть следующие встроенные функции для определения параметров матриц и векторов:

Функции, относящиеся к матрицам и векторам:

rows(M) – число строк в матрице или векторе;

max(M) и **min(M)** – максимальное и минимальное значения элементов в матрице или векторе;

Функции, относящиеся только к матрицам:

cols(M) – число столбцов в матрице;

tr(M) – сумма диагональных элементов квадратной матрицы, называемая следом матрицы.

Функции, относящиеся только к векторам:

last(M) – индекс последнего элемента в векторе;

length(M) – число элементов вектора; сумма элементов вектора вычисляется нажатием кнопки *ΣV* на панели *Matrix*.

Пример 1. Определение числа строк и столбцов матрицы:

$$M := \begin{pmatrix} 2 & 1 \\ 4 & 3 \\ 5 & 2 \end{pmatrix} \qquad n := rows(M) \qquad m := cols(M) \qquad n = 3 \qquad m = 2$$

Пример 2. Суммирование элементов матрицы:

$$A := \begin{pmatrix} 1 & -2 & 9 \\ 3 & 8 & -7 \\ 7 & 4 & 3 \end{pmatrix} \qquad i := 0..2 \qquad j := 0..2 \qquad \sum_j A_{i,j} = \begin{array}{|r|}\hline 8 \\ \hline 4 \\ \hline 14 \\ \hline\end{array} \qquad \sum_i A_{i,j} = \begin{array}{|r|}\hline 11 \\ \hline 10 \\ \hline 5 \\ \hline\end{array}$$

Очень часто при решении задач приходится иметь дело с элементами массива (матрицы или вектора), а именно с индексами элементов матрицы. MathCad позволяет разобраться с ними

Пример 3. Выделение элемента из матрицы:

$$ORIGIN := 1$$

$$A := \begin{pmatrix} 1 & -2 & 9 \\ 3 & 8 & -7 \\ 7 & 4 & 3 \end{pmatrix} \qquad A_{1,1} = 1 \quad A_{2,2} = 8$$

Пример 4. Определение минимального и максимального элемента векто-ра:

$$V := \begin{pmatrix} 4 \\ 3 \\ 1 \\ -5 \end{pmatrix} \qquad \begin{matrix} MV := \max(V) & MV = 4 \\ MV := \min(V) & MV = -5 \end{matrix}$$

Пример 5. Определение суммы элементов на диагонали квадратной матрицы:

$$M := \begin{pmatrix} 2 & 4 & -1 \\ 3 & -2 & 7 \\ 8 & -2 & -3 \end{pmatrix} \qquad tr(M) = -3$$

2.1.6 Образование новых матриц из уже существующих

Для образования новых матриц из уже существующих пользуются функциями:

а) Объединение матриц

Stack(A, B) - объединяет матрицы друг над другом.

Пример 6. Объединение матриц «друг над другом»:

$$M1 := (1 \ \ 2 \ \ 2)$$

$$M2 := (4 \ \ 5 \ \ 6) \qquad M := stack(M1, M2) \qquad M = \begin{pmatrix} 1 & 2 & 2 \\ 4 & 5 & 6 \end{pmatrix}$$

б) Сортировка векторов и матриц

В Mathcad имеется несколько встроенных функций для сортировки элементов массива в порядке возрастания или убывания:

sort(V) – сортировка элементов вектора в порядке возрастания;

Пример 7. Сортировка элементов вектора по возрастанию:

$$V := \begin{pmatrix} 5 \\ 2 \\ 4 \\ 1 \end{pmatrix} \qquad V := sort(V) \qquad V = \begin{pmatrix} 1 \\ 2 \\ 4 \\ 5 \end{pmatrix}$$

в) Транспонирование матрицы

Транспонированием называется операция, в результате которой столбцы исходной матрицы становятся строками, а строки – столбцами.

Для выполнения этой операций необходимо:

1.Набрать имя матрицы.

2.Щелкнуть на кнопке со значком матрицы на математической панели.

3.На панели *Matrix* (*Матрицы*) щелкнуть мышью на значке соответствующей операции, в данном случае MT.

Пример 8. Транспонирование вектора и матрицы:

$$\mathbf{V} := \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} \qquad \mathbf{VT} := \mathbf{V}^{\mathbf{T}} \qquad \mathbf{VT} = (2 \;\; 3 \;\; 1)$$

$$\mathbf{M} := \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 12 & 11 & 12 \end{pmatrix} \qquad \mathbf{M} := \mathbf{M}^{\mathbf{T}} \qquad \mathbf{M} = \begin{pmatrix} 1 & 5 & 9 \\ 2 & 6 & 12 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{pmatrix}$$

2.1.9 Вычисление определителя матрицы

Для нахождения определителя заданной матрицы на панели *Matrix* необходимо выбрать значок |x|, меющий тройное значение:

$$\mathbf{M} := \begin{pmatrix} 2 & 3 \\ 1 & 2 \end{pmatrix} \qquad \mathbf{NM} := |\mathbf{M}| \qquad \mathbf{NM} = 1$$

2.1.7 Нахождение матрицы, обратной заданной

Для нахождения матрицы, обратной заданной, необходимо выбрать значок X^{-1} на панели Matrix.

Произведение прямой матрицы на обратную, т.е. $X \cdot X^{-1}$ есть единичная матрица E. Иногда единичная матрица необходима для решения матричных уравнений. Для создания единичной матрицы в Mathcad есть встроенная функция **identity(n)**, где n – размер квадратной матрицы.

Пример 9. Нахождение обратной матрицы:

$$\mathbf{M} := \begin{pmatrix} 1 & 3 & 4 & 6 \\ 8 & 5 & 4 & 2 \\ 2 & 1 & 4 & 6 \\ 1 & 2 & 9 & 1 \end{pmatrix} \qquad \mathbf{M} := \mathbf{M}^{-1} \qquad \mathbf{M} = \begin{pmatrix} -0.223 & 0.109 & 0.192 & -0.035 \\ 0.389 & 0.055 & -0.404 & -0.017 \\ -0.068 & -0.021 & 0.053 & 0.127 \\ 0.055 & -0.032 & 0.134 & -0.07 \end{pmatrix}$$

2.1.8 Действия над массивами

а) Сложение и вычитание матриц

Складывать и вычитать можно только матрицы одинакового размера. Сложение и вычитание матриц есть операция нахождения матрицы А, все элементы которой равны попарной сумме всех соответствующих элементов матриц В и С, то есть каждый элемент матрицы равен *cij aij bij*

Пример 10. Сложение и вычитание матриц:

$$A := \begin{pmatrix} 2 & 8 & 5 \\ 4 & 3 & 8 \\ 7 & 1 & 6 \end{pmatrix} \quad B := \begin{pmatrix} 7 & 6 & 4 \\ 2 & 5 & 8 \\ 1 & 7 & 8 \end{pmatrix} \quad A + B = \begin{pmatrix} 9 & 14 & 9 \\ 6 & 8 & 16 \\ 8 & 8 & 14 \end{pmatrix} \quad A - B = \begin{pmatrix} -5 & 2 & 1 \\ 2 & -2 & 0 \\ 6 & -6 & -2 \end{pmatrix}$$

б) Умножение матриц

Для умножения_матриц используется традиционный знак умножения (клавиша со звездочкой). Знак умножения в Mathcad-документе по умолчанию обозначается точкой.

Пример 11. Умножение матриц:

$$A := \begin{pmatrix} 2 & 8 & 5 \\ 4 & 3 & 8 \\ 7 & 1 & 6 \end{pmatrix} \quad B := \begin{pmatrix} 7 & 6 & 4 \\ 2 & 5 & 8 \\ 1 & 7 & 8 \end{pmatrix} \quad A \cdot B = \begin{pmatrix} 35 & 87 & 112 \\ 42 & 95 & 104 \\ 57 & 89 & 84 \end{pmatrix}$$

Для этого необходимо щелкнуть правой кнопкой мыши на выражении, в котором есть знак умножения. В открывшемся контекстном меню найти пункт *View Multiplication as* (*Вид знака умножения*). Далее следует щелкнуть мышью на одном из предлагаемых семи пунктов – значок изменится.

в) Возведение матрицы в степень

Эта операция представляет собой простое перемножение матриц, в кото-ром первый и второй множители равны между собой.

Пример 12. Возведение матрицы в степень:

$$A := \begin{pmatrix} 1 & 2 & 1 & 3 \\ 3 & 2 & 1 & 2 \\ 4 & 1 & 3 & 2 \\ 1 & 3 & 4 & 3 \end{pmatrix} \quad A := A^2 \quad A = \begin{pmatrix} 14 & 16 & 18 & 18 \\ 15 & 17 & 16 & 21 \\ 21 & 19 & 22 & 26 \\ 29 & 21 & 28 & 26 \end{pmatrix}$$

г) Скалярное произведение векторов

Скалярное произведение векторов (vector inner product) определяется как скаляр, равный сумме попарных произведений соответствующих элементов. Векторы должны иметь одинаковую размерность, скалярное произведение имеет ту же размерность.

Примеры 13.

$$V1 := \begin{pmatrix} 2 \\ 3 \\ 5 \end{pmatrix} \qquad V2 := \begin{pmatrix} 3 \\ 2 \\ -1 \end{pmatrix} \qquad C := V1 \cdot V2 \qquad C = 7$$

$$V1 := \begin{pmatrix} 2 \\ 3 \\ 5 \end{pmatrix} \qquad V2 := \begin{pmatrix} 3 & 2 & 1 \end{pmatrix} \qquad M := V1 \cdot V2 \qquad M = \begin{pmatrix} 6 & 4 & 2 \\ 9 & 6 & 3 \\ 15 & 10 & 5 \end{pmatrix}$$

д) Векторное произведение двух векторов

Векторное произведение можно вычислить только для векторов с тремя элементами.

Пример 14. Векторное произведение векторов:

$$V1 := \begin{pmatrix} 2 \\ 3 \\ 5 \end{pmatrix} \qquad V2 := \begin{pmatrix} 4 \\ 5 \\ 5 \end{pmatrix} \qquad H := V1 \cdot V2 \qquad H = 48$$

2.1.9 Оператор векторизации

Векторная алгебра MathCAD включает несколько необычный оператор, который называется *оператором векторизации* (vectorize operator). Этот оператор предназначен, как правило, для работы с массивами. Он позволяет провести однотипную операцию над всеми элементами массива (т. е. матрицы или вектора), упрощая тем самым программирование циклов. Например, иногда требуется умножить каждый элемент одного вектора на соответствующий элемент другого вектора. Непосредственно такой операции в MathCAD нет, но ее легко осуществить с помощью векторизации . Для этого:

1. Введите векторное выражение.

2. Переместите курсор таким образом, чтобы линии ввода выделяли все выражение, которое требуется подвергнуть векторизации .

3. Введите оператор векторизации, нажав кнопку **Vectorize-** (Векторизация) на панели **Matrix** (Матрица) или сочетанием клавиш <Ctrl>+<->.

4. Введите <=>, чтобы получить результат.

Оператор векторизации можно использовать только с векторами и матрицами одинакового размера.

Пример 15. $v := \begin{pmatrix}1\\2\\3\end{pmatrix}$ $\quad v \cdot \begin{pmatrix}1\\2\\3\end{pmatrix} = 14 \quad \overrightarrow{\left[v \cdot \begin{pmatrix}1\\2\\3\end{pmatrix}\right]} = \begin{pmatrix}1\\4\\9\end{pmatrix}$.

Большинство неспецифических функций MathCAD не требуют векторизации для проведения одной и той же операции над всеми элементами вектора. Например, аргументом тригонометрических функций по определению является скаляр. Если попытаться вычислить синус векторной величины, MathCAD осуществит векторизацию по умолчанию, вычислив синус каждого элемента и выдав в качестве результата соответствующий вектор.

Пример 16. $\sin(v) = \begin{pmatrix}0.841\\0.909\\0.141\end{pmatrix} \qquad sin(\vec{v}) = \begin{pmatrix}0.841\\0.909\\0.141\end{pmatrix}$

2.1.10 Ранг матрицы

Рангом (rank) матрицы называют наибольшее натуральное число k, для которого существует не равный нулю определитель k-го порядка подматрицы, составленной из любого пересечения k столбцов и k строк матрицы.

Для вычисления ранга в MathCAD предназначена функция **rank**:

- rank (A) - ранг матрицы.

Пример17. Ранг матрицы:

$$\operatorname{rank}\begin{pmatrix}1 & 2\\3 & 6\end{pmatrix} = 1 \qquad \operatorname{rank}\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix} = 2$$

$$\operatorname{rank}\begin{pmatrix}1 & 2\\3 & 6\\1 & 2\end{pmatrix} = 1 \qquad \operatorname{rank}\begin{pmatrix}1 & 2\\3 & 4\\5 & 6\end{pmatrix} = 2$$

2.2 Задания к лабораторной работе №2

Задание 2.2.1 Отвечать на контрольные вопросы

1. Как создать матрицу, вектор - строку, вектор- столбец?
2. Какие операторы есть для работы с матрицами?
3. Перечислите команды панели инструментов Матрицы.
4. Как вставить матричные функции?
5. Как выполнять вычисления, если матрица задана в символьном виде?

Задание 2.2.2

1. Задать два вектора X и Y с пятью элементами двумя способами. Элементы первого вектора – произвольно заданные числа. Элементы 2 – го вектора вычисляются по формуле: $y := 3 \cdot i + 5$.

2.Выполнить следующие преобразования: элементы вектора X удвоить, элементы вектора Y увеличить на 1.

3. Получить вектор Z, равный сумме векторов X и Y. Найти скалярное произведение векторов X и Y.

4. Задать два вектора X и Y из трех элементов. Найти их скалярное и векторное произведения, используя соответствующие «кнопки» на панели инструментов «**Matrix**».

5. Задать квадратные матрицы А и В одинаковых размеров (не больше 5). Получить:

а) сумму матриц, разность матриц, произведение, определитель каждой из матриц;

б) найти обратные матрицы матрицам А и В. Произвести проверку того, что обратные матрицы вычислены верно.

в) найти скалярное произведение первой строчки матрицы А на ее последний столбик;

г) найти суммы элементов в каждом из столбиков матрицы А;

д) найти суммы элементов в каждом из строк матрицы А;

е) найти сумму всех элементов матрицы А.

Литература

1. Кирьянов Дмитрий Mathcad 15/Mathcad Prime 1.0; БХВ-Петербург - Москва, 2012.

2. Макаров Евгений Инженерные расчеты в Mathcad 15. Учебный курс; Питер - Москва, 2011. - 400 с.

3. Охорзин В. А. Компьютерное моделирование в системе Mathcad; Финансы и статистика - , 2006. - 144 с.

4. Макаров Е.Г. Mathcad: учебный курс (+). - СПб. Питер. 2009. - 384 с.

5. Ракитин В.И. Руководство по методам вычислений и приложения MathCad: учебное пособие. — М.: ФИЗМАТЛИТ, 2005. - 264 с.

6. Плис А.И. MathCad-2000. Математический практикум. / А.И. Плис, Н.А. Сливина. - М.: Финансы и статистика, 2002. - 654 с.

7. Кудрявцев Е.М. Справочник по MathCad 11. - М.: ДМК Пресс, 2005. - 180 с.

3. Лабораторная работа №3.
ПОСТРОЕНИЕ ГРАФИКОВ В MATHCAD

3.1 Возможности MathCad для построения графиков

3.1.1 Построение двумерных графиков

Одним из многих достоинств MathCAD является легкость построения графиков. В среде MathCAD можно построить два вида двумерных графиков: в декартовых и полярных координатах. Для построения двумерных графиков необходимо задать два множества: множество значений аргументов и соответствующее ему множество значений функции. Форма и способ задания множеств аргумента и функции индивидуальны и зависят от особенностей графика. Графический процессор может автоматически создавать эти массивы при быстром построении графиков, если в полях ввода аргумента и функции задавать функции.

Панель графиков вызывается нажатием кнопки с изображением графиков на математической панели (рисунок 1).

Рис. 1. Панель графиков

На панели графиков расположены девять кнопок с изображением различных типов графиков: **X-Y Plot** - графики в декартовых координатах, **Polar Plot** - графики в полярных координатах**, 3D Bar Chart** - столбиковые диаграммы, **Surface Plot** - трехмерный график, **Cunter Plot** - карта линий уровня (изолиний), **Vector Field Plot** - векторное поле, **3D Scatter Plot** - трехмерный точечный график. Сначала нас будет интересовать левая верхняя кнопка **X-У** графиков в декартовой системе координат.

Задача 1. Вычислить функцию $y(x) = 4x^2 + 5x + 8$ и получить решение в виде графика (рисунок 2).

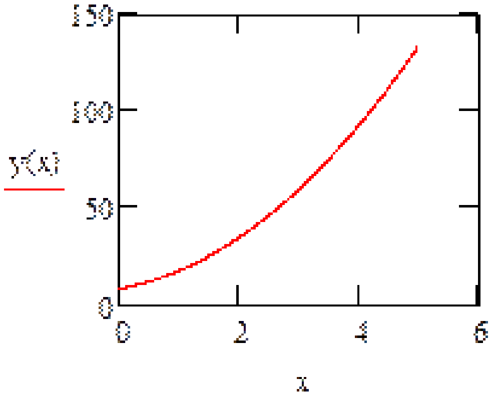


Рис.2. Построение графика

Для построение графика данной функции нужно:

1) набрать условие задачи;

2) провести ранжировку x, набрав пределы его изменения x, например $x := 0, 0.01 .. 5$. Чем меньше шаг изменения аргумента, тем более гладким получается график.

3) вызвав панель графиков, нажать на кнопку с изображением декартовых графиков. Появятся два вложенных друг в друга квадрата, внутри которых есть несколько точек.

4) Сначала нужно подвести курсор к средней точке оси абсцисс и набрать там аргумент x.

5) Затем следует подвести курсор к средней точке около оси y и набрать там наименование функции в виде y (x).

6) Щелкнем несколько раз мышью вне графика. На экране появится график параболы. Поместим курсор внутрь графика и щелкнем левой клавишей мыши. Появится окно, показанное на рисунке 3. Оно состоит из трех страниц. На рисунке 3 представлена первая страница.

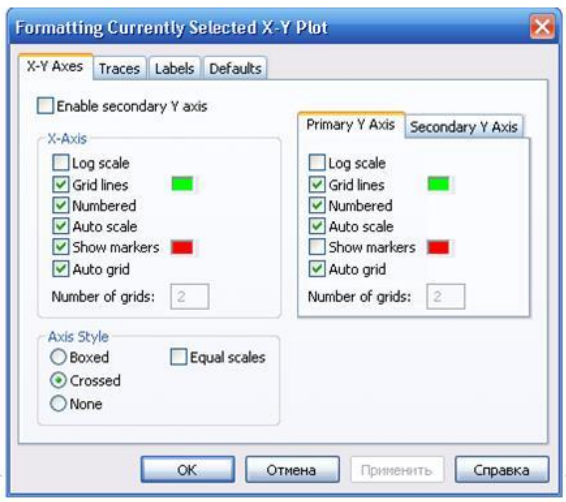


Рис.3. Первая страница окна формирования графиков

В левом нижнем углу страницы имеются точки **Boxed** (коробочка), **Crossed**(оси), **None**(нет). Нажав на первую точку, введем в график оси координат. На первой странице имеются строки **X-Axes** (**ось X**) и **Primary Y-Axes** (**первая ось У**), а под ними ряд надписей, левая часть которых относится к оси X, а правая – к оси У: **Log Scale** (логарифмическая шкала) вводит логарифмический масштаб для соответствующей оси; **Grid lines** (сетка) – ее нажатие вводит сетку на график; Numbered(оцифровка) – оцифровка сетки; **Auto scale** (автоматическая оцифровка); **Show markers** (показ маркеров); **Auto grid** (автоматическое разбиение сетки). Наличие надписей **Enable secondary Y-Axes** (возможность второй оси У) и **Secondary YAxes** (вторая ось У) дает возможность формировать графики различного масштаба для различных функций

На рисунке 4 представлена вторая страница того же окна.

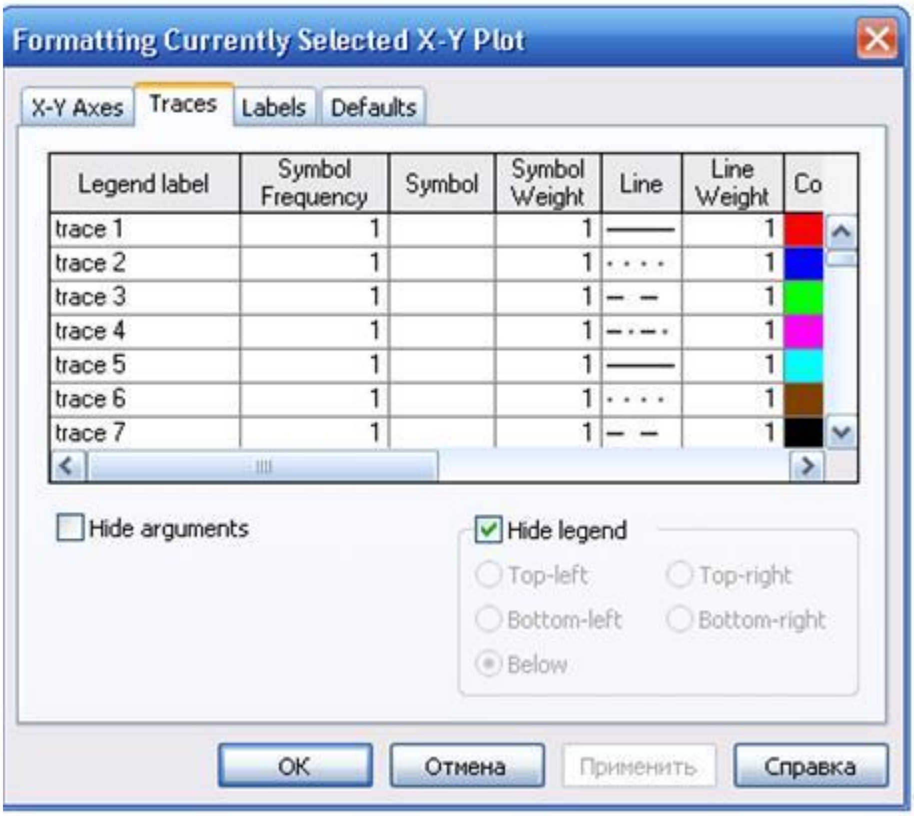


Рис.4. Вторая страница окна формирования графика

Из ее левого столбца (**trace** (след) 1, **trace** 2 и т.д.) следует, что на одном графике можно наносить до 16 различных функций.

Вводя соответствующие значения в остальные столбцы, можно изменять вид (сплошная линия, пунктир, точки), цвет, толщину и т.д. каждой функции. На третьей странице окна задается заголовок (**Title**), место его расположения **Above**(сверху), **Belove** (снизу), наименования осей (**Axis Labels**) .

Выбрав те или иные требования к графику, нажмем ОК и получим желаемый график (рисунок 5).

Задача 2. Изменить на построенных ранее графиках:

а) толщину линии;

б) заменить сплошную линию пунктиром;

в) изменить цвет графика на зеленый;

г) произвести нанесение осей координат, оцифровку осей;

д) расположить над графиком заголовок «ВЫЧИСЛЕНИЕ ФУНКЦИИ».

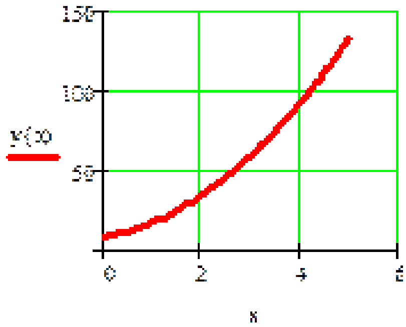


Рис.5 График функции с осями и сеткой.

На одном графике можно строить до 16 различных кривых. Построим на том же графике еще одну кривую (рисунок 6).

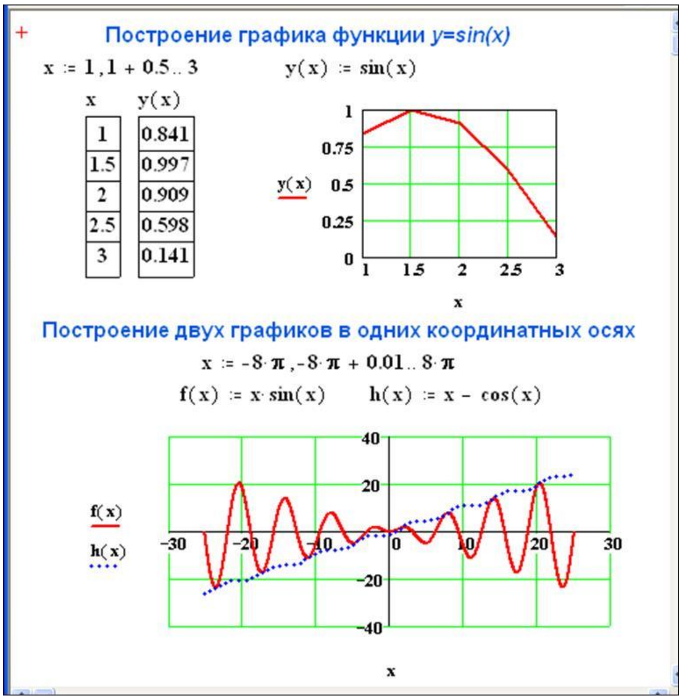


Рис. 6.

Для этого:

1) проведем ранжировку аргумента x;

2) наберем обе функции;

3) введем первую функцию, как было описано выше;

4) затем подведем курсор к записи на оси y и нажмем клавишу **Ctrl+«запятая»** клавиатуры. Под записью y(x) появится маркер, в который введем имя второй функции.

На рисунке 7 показан порядок построения графиков трех функций в одной системе координат

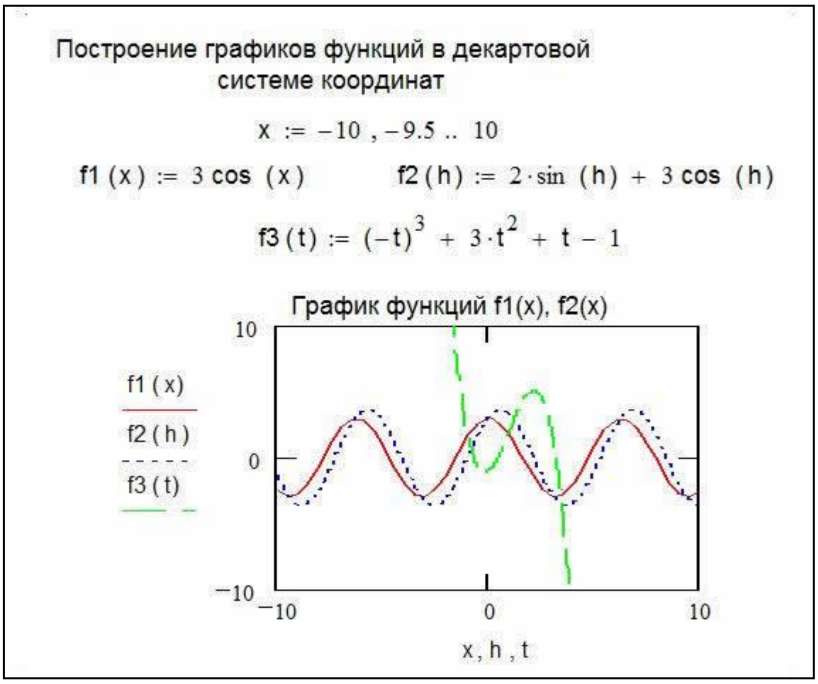


Рис. 7.

Задача 3. Изменить на построенных ранее графиках:

а) толщину линии

б) заменить сплошную линию пунктиром

в) изменить цвет графика на зеленый

г) произвести нанесение осей координат, оцифровку осей

д) расположить над графиком заголовок «ВЫЧИСЛЕНИЕ ФУНКЦИИ»

Задача 4. Построить самостоятельно график двух функций в одной системе координат

$y(x)=2\sin^2(x)$

$z(x)=5\cos^3(x)$

в пределах $0 \leq x \leq 20$

3.1.2 Графики в полярных координатах(Polar Plot)

Такие графики задаются нажатием кнопки с изображением графика в полярных координатах. Обозначение переменных не изменяется. На рисунке 8 приведено построение графика функции в полярных координатах.

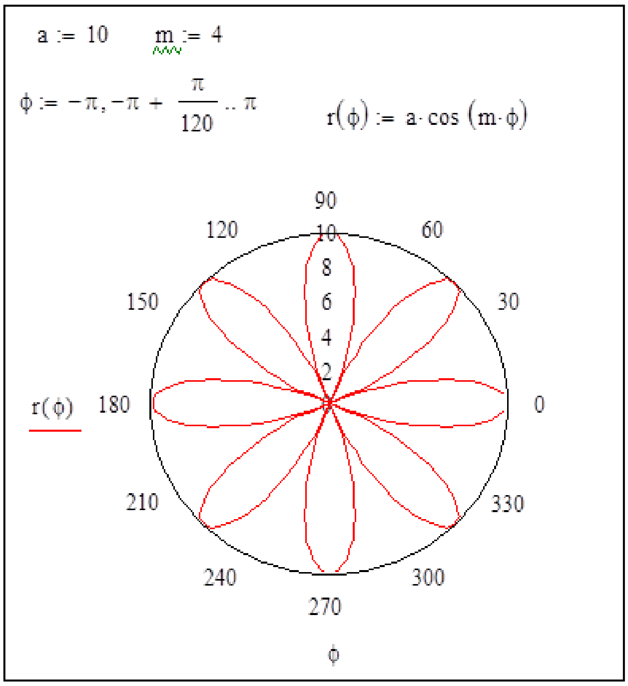


Рис.8 График функции в полярных координатах.

1.1.3 Трехмерные графики

В Маткад можно строить различные трехмерные графики: поверхности, уровней, столбиковые диаграммы и т. п. Чтобы создать трехмерный график, требуется нажать кнопку с изображением одного из типов трехмерных графиков на панели инструментов Graph (График). В результате появится пустая область графика с тремя осями (рисунок 9) и единственным полем ввода в нижнем левом углу. В этот поле ввода следует ввести данные, описывающие график.

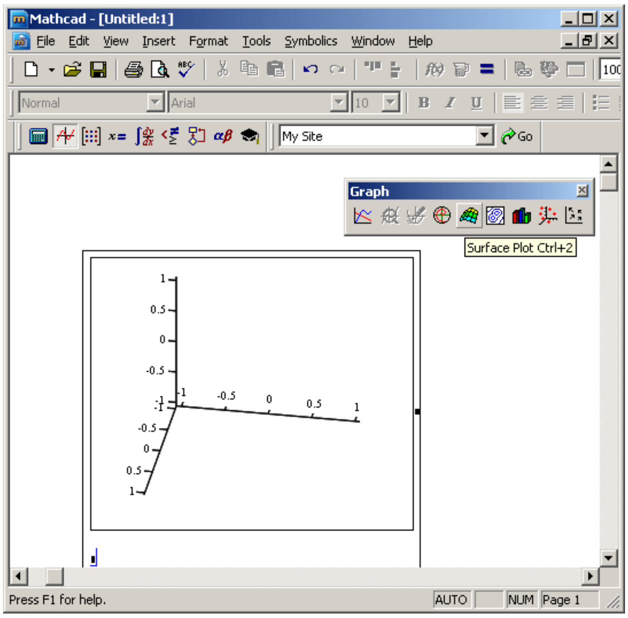


Рис.9 Построение трехмерных графиков

Для трехмерных графиков предусмотрено три основных формата задания входных данных – координат точек графика:

А. Функция пользователя двух переменных – применяется для быстрого построения графиков. Эта функция должна описывать значение координаты Z в зависимости от значений координат X и Y. Расчет массивов координат выполняется автоматически. Достоинством этого формата является простота, а недостатком – ограниченные возможности.

Б. Матричная переменная – определяет трехмерный график, у которого координата x принимается равной номеру столбца матрицы, координата y – номеру строки, а числовые значения элементов соответствуют координате z.

В. Три векторных или матричных переменных – каждая переменная соответствует своей координате. Наиболее часто применяется в точечных графиках. В других видах графиков этот формат поддерживается не всегда корректно. Номер элемента в векторах задает номер определяемой точки. Векторы должны записываются в поле ввода в круглых скобках через запятую. Если применить матричные переменные, можно строить более сложные трехмерные поверхности .

Создавать массивы входных данных можно любым известным способом.

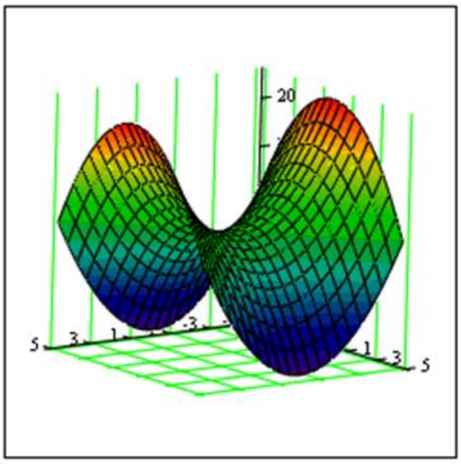
Рассмотрим технику построения трехмерных графиков различных типов с использованием различных способов подготовки данных.

График поверхности **Surface Plot** позволяет графически отобразить функцию двух переменных **Z(x,y)** в трехмерном пространстве. Для построения графика поверхности можно воспользоваться двумя способами:

1. Если вам надо только посмотреть общий вид поверхности, то MathCAD предоставляет возможность быстрого построения подобных графиков. Для этого достаточно определить функцию **f(x,y)** и выполнить команду **Insert -> Graph -> Surface Plot** или нажать соответствующую кнопку наборной панели **Graph** (сочетание клавиш **[Ctrl+7]**). В появившейся графической области под осями на месте шаблона для ввода надо указать имя (без аргументов) функции (в нашем случае буква g, см рис.10). MathCAD автоматически построит график поверхности. Независимые переменные **x** и **y** принимают значения из промежутка [-5,5]. При необходимости этот промежуток может быть уменьшен или увеличен. Для этого необходимо выделить график и воспользоваться командой **Format -> Graph -> 3D Plot** или щелкнуть ПРАВОЙ кнопкой мыши по выделенному графику и в контекстном меню выбрать команду **Format**. В появившемся окне **3-D Plot**

Format на вкладке **QuickPlot Data** можно установить другие параметры изменения независимых переменных **x** и **y**.

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



g

Рис.10

2. Для построения графика поверхности в определенной области изменения независимых переменных или с конкретным шагом их изменения необходимо сначала задать узловые точки $\mathbf{x_i}$ и $\mathbf{y_j}$, в которых будут определяться значения функции. После (а можно и до) этого надо определить функцию **f(x,y)**, график которой хотите построить. После этого необходимо сформировать матрицу значений функции в виде: $\mathbf{A_{i,j}=f(x_i,y_j)}$.

Теперь после выполнения команды **Insert -> Graph -> Surface Plot** в появившейся графической области достаточно ввести имя матрицы (без индексов). Если вы хотите, чтобы узловые точки были расположены через равные промежутки, воспользуйтесь формулами, изображенными на рис. 11.

$N := 15$

$i := 0..N \qquad j := 0..N$

$x_{min} := -1.5 \qquad x_{max} := 1.5$

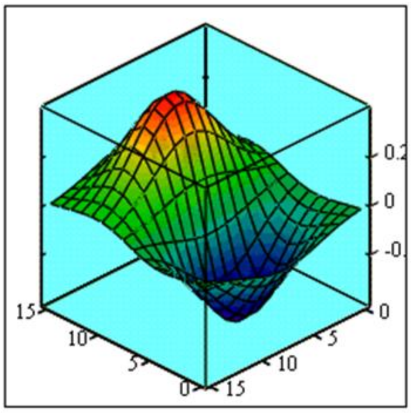
$y_{min} := -1.5 \qquad y_{max} := 1.5$

$x_i := x_{min} + \frac{i}{N}\cdot(x_{max} - x_{min})$

$y_j := y_{min} + \frac{j}{N}\cdot(y_{max} - y_{min})$

$f(x,y) := \sin(x)\cdot\exp(-x^2 - y^2)$

$A_{i,j} := f(x_i, y_j)$



A

Рис.11.

Для построения графика линий уровня данной функции необходимо поступать также как это было описано выше, только вместо команды (Поверхности) следует выбрать команду **Contour Plot** (Контурный) (рис.12,13).

Аналогично, при помощи команды **3D Bar Plot** (3D Диаграммы) можно построить трехмерный столбчатый график данной функции, при помощи команды **3D Scatter Plot** (3D Точечный) - трехмерный точечный график, а при помощи команды **3D Patch Plot** (3D Лоскутный) - трехмерный график поверхности в виде несвязанных квадратных площадок - плоскостей уровня для каждой точки данных, параллельных плоскости X-Y (рис.14,15).

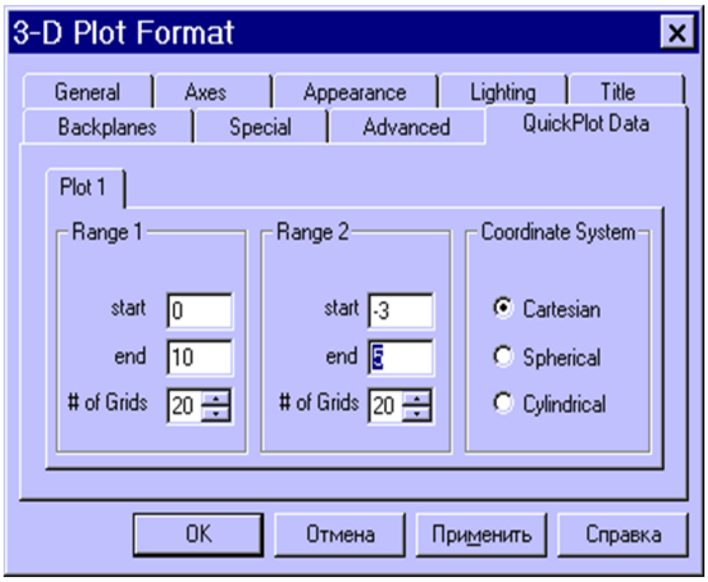


Рис.12.

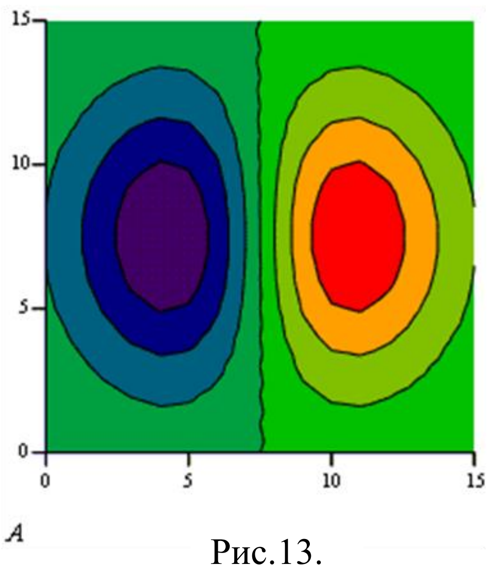


Рис.13.

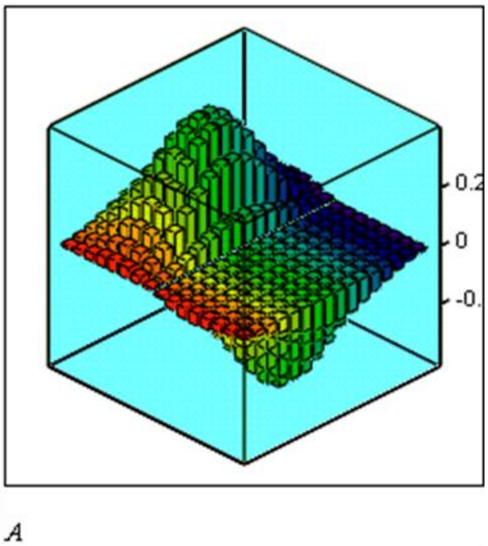


Рис.14.

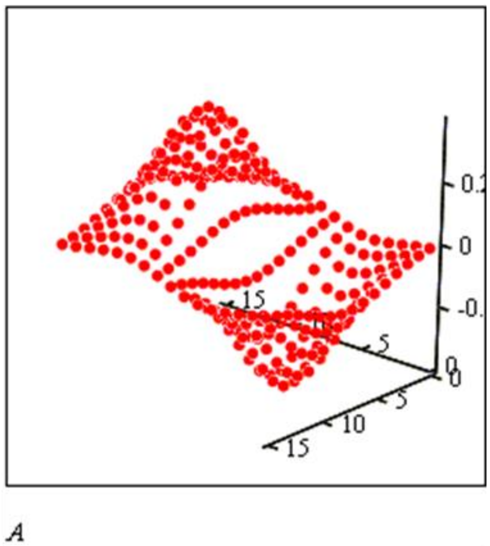


Рис.15.

Если вас не устраивает внешний вид созданного трехмерного графика, вы можете изменить его, выполнив команду **Format -> Graph -> 3D Plot** или выполнив двойной щелчок мышкой на графической области. В результате на экране появится диалоговое окно **3-D Plot Format** (рис.16), позволяющее изменять параметры отображения графика. Мы рассмотрим здесь основные опции. Разобраться во всех тонкостях управлением видом графика вы можете самостоятельно, построив график и поэкспериментировав, выбирая те или иные опции.

Диалоговое окно **3-D Plot Format** содержит несколько вкладок. Некоторые из них мы рассмотрим более подробно, а для других - опишем лишь их функциональное назначение.

На вкладке **General** (Общие свойства) вы можете:

- в области **View** задать направление взгляда наблюдателя на трехмерный график. Значение в поле **Rotation** определяет угол поворота вокруг оси Z в плоскости X-Y. Значение в поле **Tilt** задает угол наклона линии взгляда к плоскости X-Y. Поле **Zoom** позволяет увеличить (уменьшить) графическое изображение в число раз, равное цифре, указанной в поле.

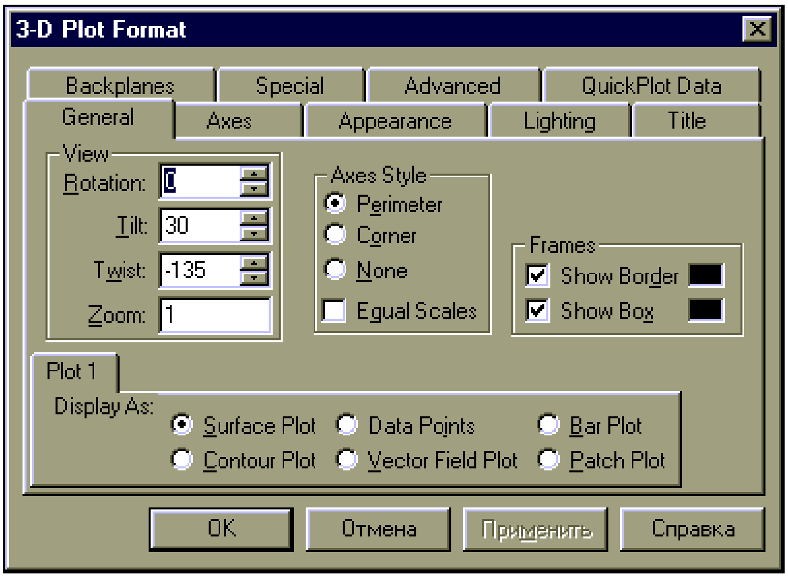


Рис.16.

- в области **Axes Style-** (Стиль оси) задать вид осей, выбрав селекторную кнопку **Perimetr** (Периметр) или **Corner-** (Угол). В первом случае оси всегда находятся на переднем плане. При выборе кнопки **Corner** точка пересечения осей *Ox* и *Oy* задается элементом $A_{0,0}$ матрицы *A*.
- в области **Frames** опция **Show box** (Каркас) предназначена для отображения вокруг графика куба с прозрачными гранями, а опция **Show**

border (Границы) позволяет заключить график в прямоугольную рамку.

•в области **Plot 1 (Plot 2...) Display as** (График/ несколько графиков Отобразить как) - имеются селекторные кнопки для представления графика в друих видах (контурный, точечный, векторное поле и др.)

Элементы вкладки **<u>Axes</u>** (Ось) позволяют изменять внешний вид осей координат (рисунок 17).

Посредством опций области **Grids** (Сетки) можно отобразить на графике линии, описываемые уравнениями- $x,y,z=$ *const*. Если установлены опции **Show Numbers-** (Нумерация), отображаются метки на осях и подписи к ним.

При этом рядом с осями *Ox* и *Oy* указываются не значения узловых точек x_i, y_j, а значения индексов *i* и *j*, в то время как ось *Oz* размечается в соответствии с промежутком, которому принадлежат элементы матрицы значений $A_{i,j}$.

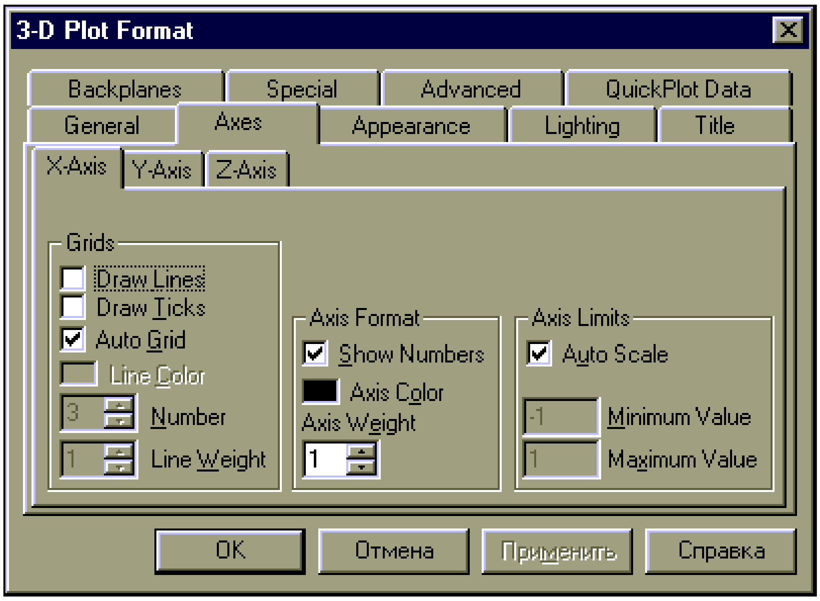


Рис.17

• Если установлена опция **Auto Grid** (Автосетка), программа самостоятельно задает расстояние между соседними отметками на осях. Вы можете сами указать число линий сетки, если отключите указанную опцию.

• Если установлена опция **Auto Scale** (Авошкала) , то MathCAD сам определяет границы построкния графика и масштабы по осям. Можно отключить данную опцию и для каждой оси самостоятельно задать пределы изменения переменных в полях **Minimum Value**(Минимум) и **Maximum Value** (Максимум).

Вкладка **<u>Appearance</u>** (Внешний вид) позволяет изменять для каждого графика вид и цвет заливки поверхности (область **Fill Options**); вид, цвет и толщину дополнительных линий на графике (область **Line Options**); наносить на график точки данных (опция **Draw Points** области **Point Options**), менять их вид, размер и цвет.

Вкладка **Lighting** (Освещение) при включении опции **Enable Lighting** (Наличие подсветки) позволяет выбрать цветовую схему для освещения, "установить" несколько источников света, выбрав для них цвет освещения и определив его направление.

Вкладка **Backplanes** (Задние плоскости) позволяет изменить внешний вид плоскостей, ограничивающих область построения: цвет, нанесение сетки, определение ее цвета и толщины, прорисовка границ плоскостей.

На вкладке **Special** (Специальная) можно изменять параметры построения, специфичные для различных типов графиков.

Вкладка **Advansed** позволяет установить параметры печати и изменить цветовую схему для окрашивания поверхности графика, а также указать *направление* смены окраски (вдоль оси *Ox*, *Oy* или *Oz*). Включение опции **Enable Fog** (Наличие Тумана) делает график нечетким, слегка размытым (полупрозрачным).

3.2 Примеры выполнения лабораторной работы

3.2.1 Построение графиков двух функций на одном поле

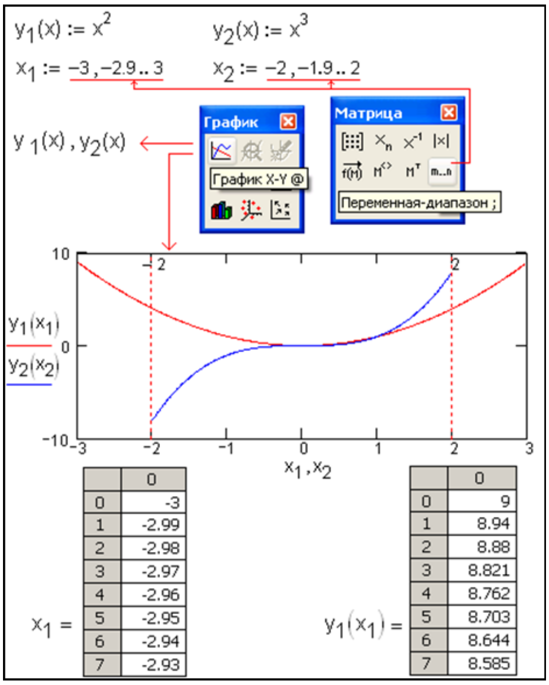


Рис.18.

3.2.2 Построение графиков функции и двух её производных на одном поле

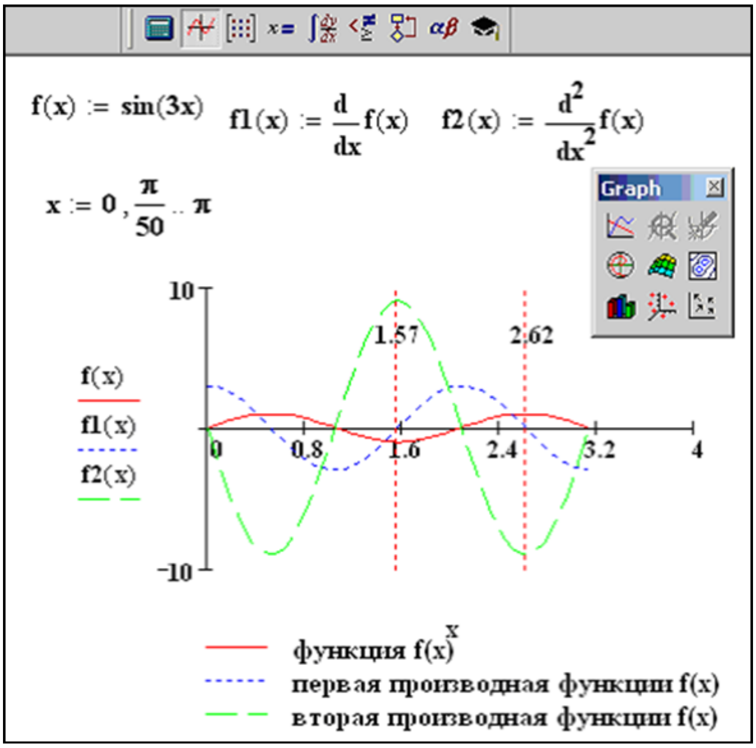


Рис.19.

3.2.3 Построение поверхностей

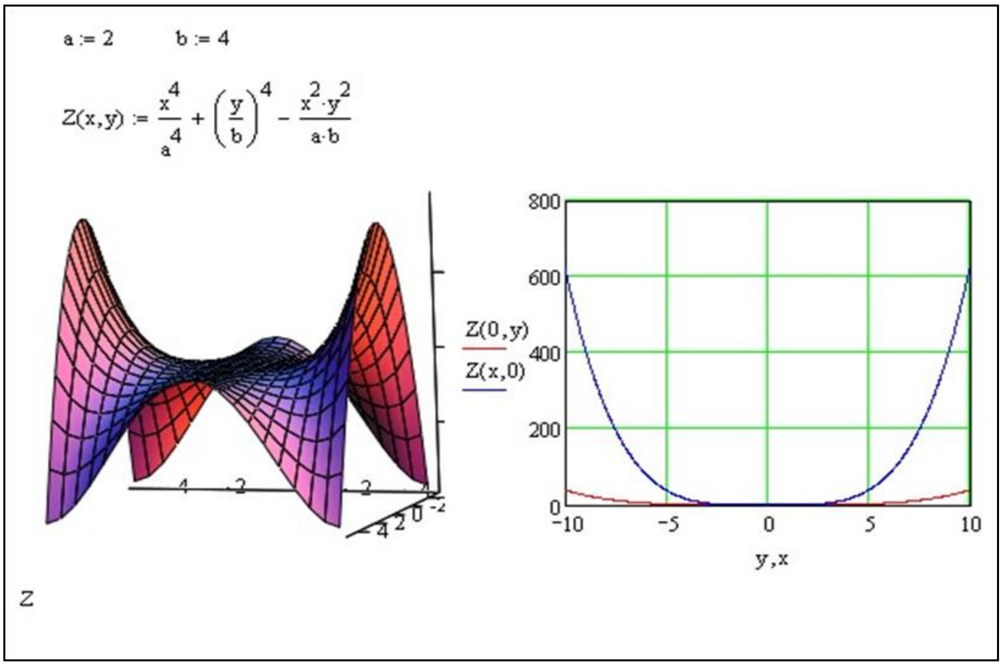


Рис.20

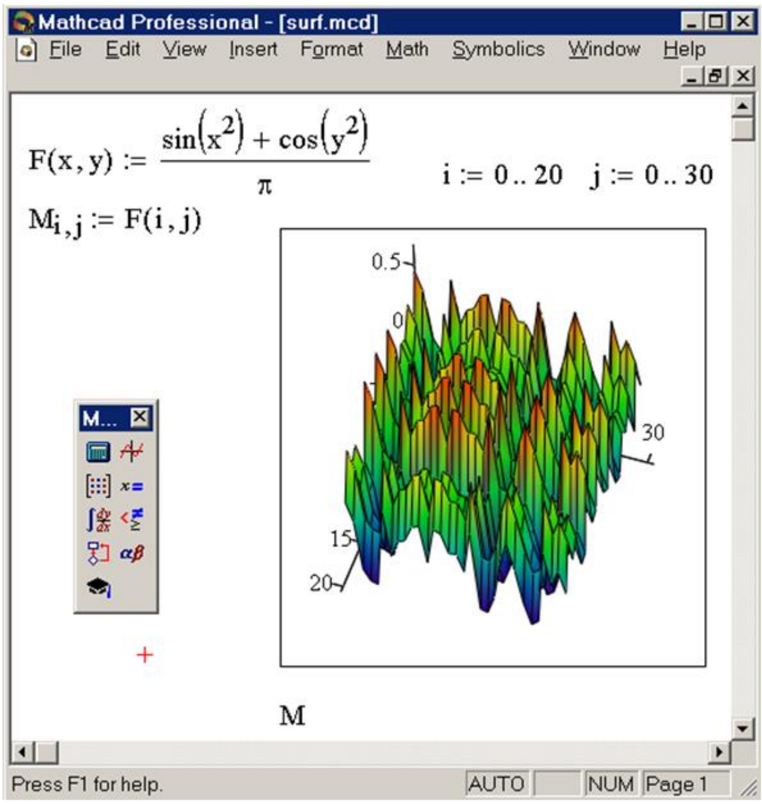


Рис.21.

3.2.4 Построение пересечения двух поверхностей

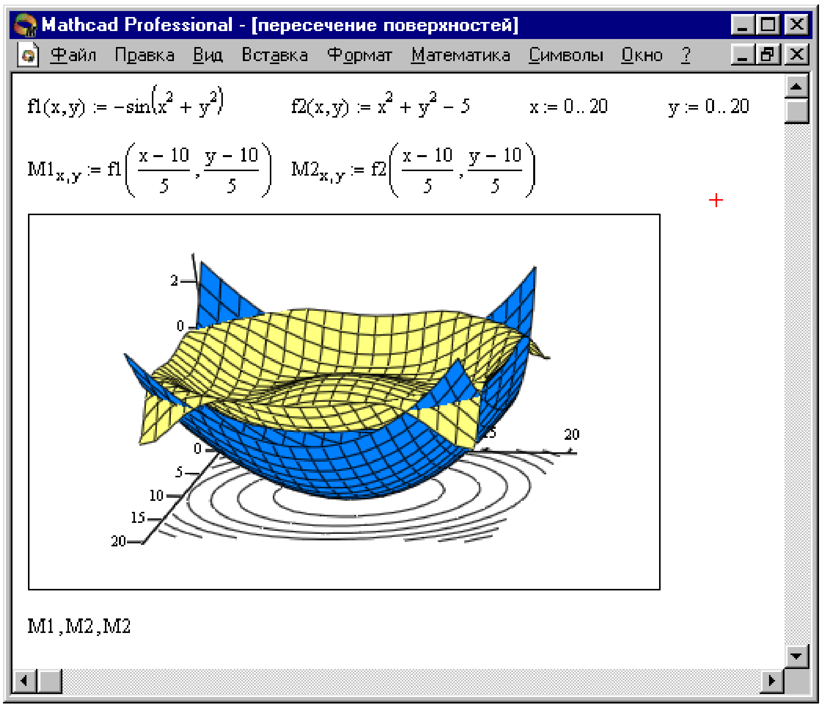


Рис.22.

3.3 Задания к лабораторной работе

Задание 3.3.1 Отвечать на контрольные вопросы

1. Как построить декартовый график?
2. Как построить несколько графиков в одной системе координат?
3. Как построить график в полярной системе координат?
4. Как построить график поверхности?
5. Что делать, если при быстром построении графиков они получаются хаотической формы и не соответствуют заданным формулам?
6. Почему при построении двумерных графиков применяют векторные переменные, а для трехмерных – матричную переменную?
7. Какие значения откладывает графический процессор по осям x и y, если при построении трехмерного графика применяется матричная переменная? Как отформатировать построенный график?
8. Для чего используются функции CreateMesh, CreateSpace?

Задание 3.3.2. В соответствии с номером своего варианта задания выбрать функцию одной переменной из таблицы 1. Произвести табулирование указанной функции f(x) на отрезке $x_0...x_n$ с указанным шагом dx. Используя результаты табулирования, построить двумерный график функции f(x).

Таблица 1

№ варианта	Функция	Интервал $x_0..x_n$	Шаг dx
1	$f(x) = \sin(x) + 2 \cdot \cos(x - 1)$	$0...2 \cdot \pi$	***0,1***
2	$f(x) = \ln(x) + \cos^2(x)$	$0.1...\pi$	***0,2***
3	$f(x) = x^{(1-2 \cdot x)} + e^{\sqrt{x}}$	***0…3***	***0,1***
4	$f(x) = 2 \cdot \cos(1 - x) + ctg(x^2)$	$-\frac{\pi}{4}...2 \cdot \pi$	***0,1***
5	$f(x) = 2 \cdot \ln(x) + \sqrt{\sin(x)}$	***1…10***	***0,3***
6	$f(x) = x^3 - 10x^2 + x$	***-10…10***	***1***
7	$f(x) = \frac{1 + \cos(x)}{2^x}$	***-10…10***	***0,1***
8	$f(x) = 2 \cdot \cos(1 - x) + tg(x^2)$	$-\frac{\pi}{4}...2 \cdot \pi$	***0,1***
9	$f(x) = \sin(x^2) + \cos(2 \cdot x - 1)$	$-2 \cdot \pi...3 \cdot \pi$	***0,1***
10	$f(x) = \frac{\sin(x^2)}{2^x} + \cos(2 \cdot x^2 - 1)$	$-2 \cdot \pi...3 \cdot \pi$	***0,1***

Задание 3.3.3. В соответствии с номером своего варианта задания выбрать из таблицы 2 функцию двух переменных. Построить трехмерный график.

Таблица 2

№ варианта	Функция	Интервал
1	***f(x,y)=sin(y)*cos(x)***	***0<x<=5*** ***1<y<=7***
2	***f(x,y)=*** $\boldsymbol{\sin^2(x+1)+y}$	$0\le x\le 5$ $1\le y\le 7$
3	***f(x,y)=*** $\boldsymbol{\ln(x+1)-y^2}$	$0\le x\le 4$ $1\le y\le 5$
4	***f(x,y)=sin(x)+cos(y)***	$1\le x\le 5$ $1\le y\le 7$
5	***f(x,y)=2sin(x)-cos(y)***	$0\le x\le 5$ $1\le y\le 4$
6	***f(x,y)=ln(x+1)*sin(y)***	$0\le x\le 5$ $1\le y\le 7$
7	***f(x,y)=cos(x)+2sin(y+1)***	$0\le x\le 5$ $1\le y\le 7$
8	***f(x,y)=*** $\boldsymbol{\cos^2(x)-\sin(y)}$	$0\le x\le 5$ $1\le y\le 7$
9	***f(x,y)=2sin(x)-cos(y)***	$0\le x\le 5$ $1\le y\le 7$
10	***f(x,y)=ln(x+2)sin(x)***	$0\le x\le 5$ $1\le y\le 7$

Литература

1. Макаров Е.Г. Mathcad: учебный курс (+). - СПб. Питер. 2009. - 384 с.

2. Дъяконов В.П. Энциклопедия MathCad 2000i и MathCad 11. - М.: СОЛОН-Пресс, 2004. - 832 с.

3. Дьяконов В. MathCaD 2001 : учеб. курс / В. Дьяконов. – СПб.: Питер, 2001. – С. 184 – 231. – ISBN 5-318-00367-2.

4. Ракитин В.И. Руководство по методам вычислений и приложения MathCad: учебное пособие. - М.: ФИЗМАТЛИТ, 2005. - 264 с.

5. Плис А.И. MathCad-2000. Математический практикум. / А.И. Плис, Н.А. Сливина. - М.: Финансы и статистика, 2002. - 654 с.

6. Кудрявцев Е.М. Справочник по MathCad . - М.: ДМК Пресс, 2005. -180 с.

4. Лабораторная работа № 4.
ПРОГРАММИРОВАНИЕ В СРЕДЕ MATHCAD

4.1. Основы теории

Выполняя предыдущие лабораторные работы мы заметили, что MathCAD есть система, ориентированная на пользователя – не программиста. Создатели MathCAD изначально поставили перед собой такую задачу, чтобы дать возможность всем профессионалам самостоятельно проводить нужные расчеты, не обращаясь за помощью к программистам.

4.1.1. Программирование в MathCAD.

В очень ранних версиях MathCAD встроенного языка программирования не было. В последних версиях MathCAD такой язык имеется. Для написания программ в MathCAD предусмотрена специальная панель инструментов, которая так и называется – **Programming** (рис.1).

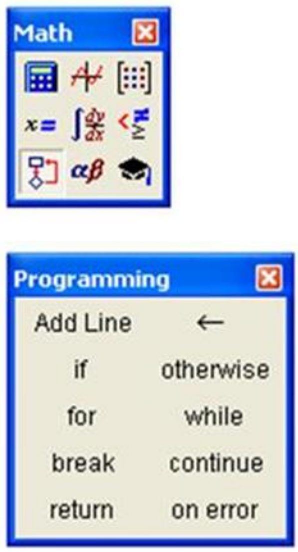


Рис.1

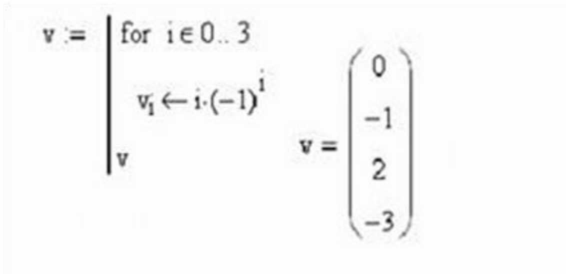
На этой панели совсем немного кнопок, каждая из них понадобится неоднократно. Первая из всех кнопок по порядку – ***Add Line***. С ее помощью добавляется новая строка в программный код и создается новая программа. Программный код отличается по своему внешнему виду от обычных математических операторов присутствием жирной вертикальной полосы слева от кода. Самый важный оператор при программировании это присваивание значения переменной $\boxed{\leftarrow}$.

Операторы программирования могут быть введены только с панели **Programming** (или с помощью горячих клавиш, которые приведены в тексте всплывающей подсказки), ***но никак не с клавиатуры***!

Создание программного блока (модуля) начинается с команды ***Add Line***. Нажатие этой клавиши приведет к тому, что в рабочей области документа появится вертикальная черта, а справа от нее – два пустые поля ввода:

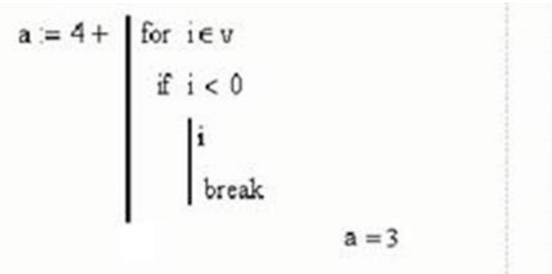
Вертикальная черта означает, что строки, находящиеся справа от нее, образуют линейную программную последовательность операций. В полях ввода можно ввести две первые строки программы. Если требуется большее число строк, то необходимо нажимать клавишу ***Add Line*** столько раз, сколько необходимо строк.

Важно знать: программный блок является полноценным выражением MathCAD. Это выражение может быть использовано для присвоения значения векторной или скалярной переменной (рисунки 2 и 3) или для определения функции (рисунок 4), или даже в составе другого выражения, в том числе и другой программы.



Результат вычисления программного блока присваивается вектору

Рис.2



Результат вычисления программного блока присваивается скаляру. Причем программный блок входит в состав арифметического выражения

Рис.3

Внутри программы можно использовать глобальные переменные, но изменить их значения никак нельзя. Можно создать в программе другие переменные, доступ к которым может осуществляться только из самой программы. Эти переменные называются локальными переменными. Локальные переменные «не видны» извне. Локальная переменная создается с помощью знака локального присвоения ← с панели Programming. Для оператора локального присваивания, так же как и для операторов обычного := и глобального ≡ присваивания, можно изменить внешний вид так, чтобы он выглядел как обычный знак равенства. Для этого достаточно вызвать контекстное меню этого оператора и в нем выбрать команду **View Definition As/Equal.**

Важно знать: последняя строка любой программы не должна содержать никаких управляющих операторов. Эта строка задает значение, возвращаемое программой.

Последняя строка программы может содержать имя локальной переменной либо некоторое математическое выражение, куда входят как локальные, так и глобальные переменные, либо вектор или матрицу.

Для определения функции это выглядит следующим образом :

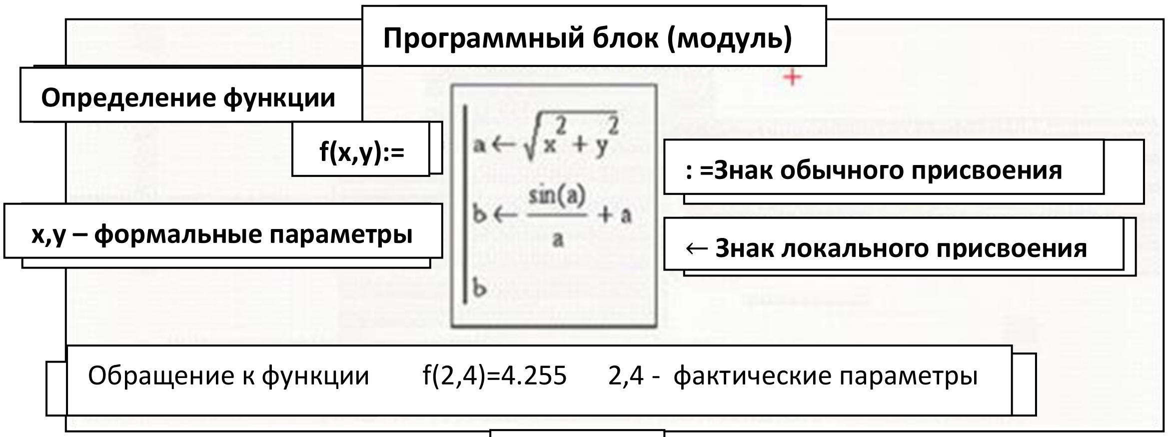


Рис.4

На следующем примере продемонстрируем элементы программирования в MathCAD.

Пример 1. Требуется подготовить описание функции $y(x, g)$ для вычис-ления ее значения при $x = 4.15$ и $g = 1.854$.

Решение. Прежде всего отметим, что при вычислении y можно обойтись без описания и использования пользовательской функции и тем более без программирования, как это показано на рисунке 5, a. Для нахождения значения функции с помощью программы значения x и g следует передать во внутрь программы. На рис. 5, b, c приведены варианты таких решений поставленной задачи. В данном примере введена вспомогательная локальная переменная t для вычисления значения функции. И эта переменная расположена в последней строчке программного блока.

a)	b)	c)
$x := 4.15$ $g := 1.854$ $R := \sin\left(\frac{x}{g}\right)$ $R = ▪$	$T := \left\vert \begin{array}{l} t1 \leftarrow \frac{x}{g} \\ \sin(t1) \end{array} \right.$	$D := \left\vert \begin{array}{l} t1 \leftarrow \frac{x}{g} \\ t2 \leftarrow \sin(t1) \\ t2 \end{array} \right.$

Рис.5

Подготовка описания функции $y(x, g)$ и ее выполнение представлено на рис. 6.

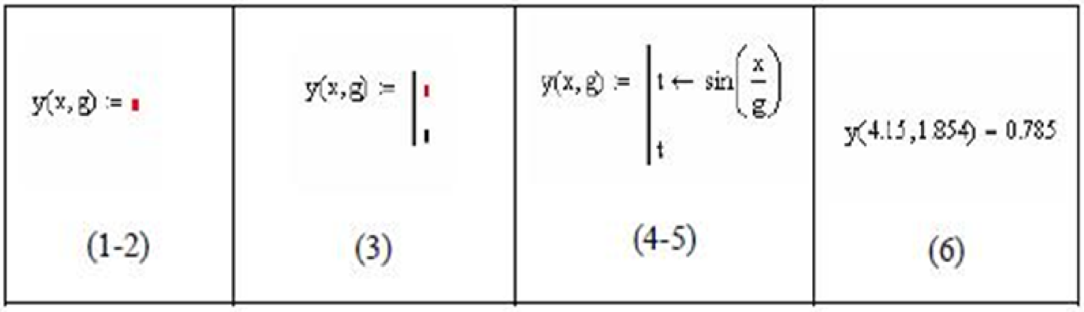


Рис.6

Вставить строку программного кода в уже созданную программу можно в любой момент с помощью кнопки ***Add Line***. Для этого следует поместить на нужное место внутри программы линии ввода синего цвета (рис. 7):

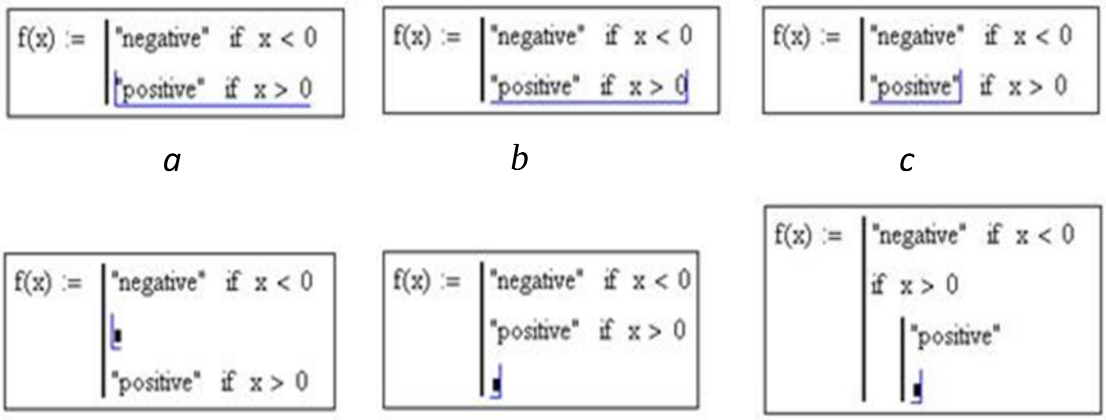


Рис.7

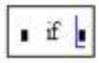
– если вертикальная линия ввода синего цвета находится в начале текущей строки, то нажатие кнопки ***Add Line*** приведет к появлению новой строки перед текущей строкой (рис.7а);

–если линия ввода находится в конце строки, то нажатие кнопки ***Add Line*** приведет к появлению новой строки после текущей (рис.7 b);

– если выделить только часть текущей строки, то это повлияет на положение новой строки в программе (рис.7с).

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

4.1.2. Проверка условий в программах

Программы в MathCAD могут быть не только линейными, но и разветвленными. Одним из вариантов ветвления в программах является проверка условия. Условия могут проверять значения как локальных, так и глобальных переменных, а также выражений, содержащих эти переменные. Для проверки условий в программах MathCAD служит оператор ***if***. При нажатии на этот оператор на экране появится . В ***поле ввода справа*** нужно ввести условие. Для ввода условий служит панель ***Boolean*** (Рис.8).

Панель Boolean (Булевы).

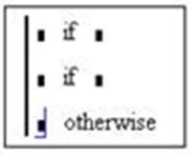
Рис.8

На этой панели есть кнопки, предназначенные для проверки условий = < > ≤ ≥ ≠ , а также кнопки, предназначенные для вставки логических операций ¬ ∧ ∨ ⊕ (логическое отрицание «не», логическое произведение «и», логическое сложение «или», операция «исключающее или»). Вторые представляют собой булевы операции и позволяют создавать сложные условия. В ***поле ввода слева*** нужно ввести строку программы, которая должна выполняться, если введенное ***условие истинно***.

Если для условия «истинно» необходимо выполнение нескольких строк программы, надо воспользоваться кнопкой ***Add Line.***

Если *невыполнение* условия должно привести к выполнению какого-либо иного программного кода, можно в строке, следующей за оператором ***if***, вставить оператор ***otherwise.***

В ***поле ввода слева*** от этого оператора необходимо ввести строку программы, которая будет выполняться только в том случае, если не выполнилось условие, заданное в операторе if.

Важно знать: если в программе введено подряд несколько строк с оператором ***if*** 

то выражение слева от оператора ***otherwise*** будет выполнено только в том случае, если не выполняются условия, заданные во всех операторах $\boldsymbol{if}$**.** Текст такой программы представлен на рисунке 9.

```
f(x) := | "negative"  if x < 0
        | if x > 0
        |   | "positiv"
        |   | "big positive"  if x > 1000
        | "zero"  otherwise

f(1) = "positiv"

f(10^5) = "big positive"
```

Рис.9.

Пример 2. Составить модуль программы для построения графика функции $y = x^2 - 1$. Для этого составляем программу:

```
x := -2,-1.8..2          F(x) := x^2 - 1

G(x) := | F(x)  if -2 ≤ x ≤ -1.3
        | (-F(x))  if -1.3 ≤ x ≤ 0
        | F(x)^2  if 0 ≤ x ≤ 1.4
        | x  otherwise
```

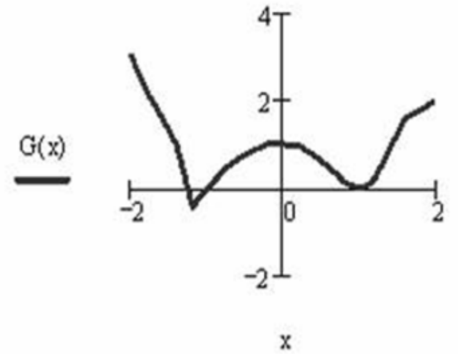


Рис.10

4.1.3. Создание циклов.

При создании нелинейных программ появляется необходимость создавать циклы. Они позволяют повторять несколько раз выполнение одного и того же программного блока. Для создания циклов в MathCAD предусмотрено два оператора $\boldsymbol{for}$ и $\boldsymbol{while}$.

Цикл for.

В таких циклах создается некоторая **переменная – счетчик**, значение которой изменяется после каждого выполнения тела цикла. Выход из цикла происходит при достижении этой переменной заданного значения. Этот оператор вводится с панели Programming клавишей $\boldsymbol{for}$

```
for ∎ ∈ ∎
  ∎
```

В *поле ввода после слова* ***for*** следует указать имя переменной – счетчика. Это может быть любое имя, которое не использовалось ранее в программе. Внутри цикла можно использовать эту переменную в любых выражениях, нельзя только присваивать ей никакого значения. *В поле ввода после знака* ∈ следует указать диапазон значений переменной-счетчика. Вводить диапазон в данном случае следует так же, как и *при создании ранжированной переменной*. Вместо диапазона в данном поле ввода можно *указать имя некоторого массива* (вектора или матрицы). В этом случае переменная-счетчик будет последовательно принимать значения всех элементов этого массива. Возможность перебора элементов массива не может быть реализована с помощью цикла ***while***, поэтому именно в таких случаях цикл ***for*** и является незаменимым. В поле ввода под словом ***for*** следует ввести тело цикла. (Если выполняемая программа зациклилась, то ее можно остановить, нажатием клавиши **Esc).**

Пример 3.

$$x := \left| \begin{array}{l} z \leftarrow 0 \\ \text{for } i \in 0..6 \\ \quad z \leftarrow z + 2\cdot i \end{array} \right.$$

$$x = 42$$

Цикл while позволяет программе работать до тех пор, пока выполняется определенное условие.

В *поле ввода справа* от слова ***while*** следует ввести условие. Это условие строится по тем же правилам, что и в операторе ***if.*** Оно будет проверяться после каждого выполнения тела цикла и в тот момент, когда условие перестанет выполняться, повторение тела цикла прекратится. В *поле ввода ниже* слова ***while*** следует ввести тело цикла (напомним, что для ввода нескольких строк в теле цикла надо воспользоваться кнопкой ***Add Line***). Продемонстрируем изложенное на следующих двух примерах.

Пример 4.

$$x := \left| \begin{array}{l} z \leftarrow 1 \\ \text{while } z < 10 \\ \quad z \leftarrow z + 1 \end{array} \right.$$

$$x = 10$$

Пример 5. Найти первый элемент вектора, превосходящий заданное значение. Составим программу:

Создание вектора

$m := 0..2500 \qquad v_m := 1 + \sin(m)$

$$t(v, thres) := \left| \begin{array}{l} j \leftarrow 0 \\ \text{while } v_j \le thres \\ \quad j \leftarrow j + 1 \\ j \end{array} \right.$$

инициализация счетчика

возврат значения

$t(v, 1.98) = 8$

Величина 1,98 впервые превосходится восьмым элементом вектора.

4.1.4. Использование операторов break и continue.

Иногда возникает необходимость повлиять на выполнение цикла некоторым образом, например, прервать его выполнение по какому-либо условию или выполнять некоторые итерации не так, как другие. Для этого и служат операторы ***break*** и ***continue***.

Оператор break, если он расположен внутри цикла, означает немедленное прекращение выполнения текущей итерации и выход из цикла. Если есть необходимость прекратить выполнение цикла по какому-либо условию, то следует использовать конструкцию следующего вида ***break if*** *(условие)*.

Например, в следующих двух примерах, как только значение переменной цикла i достигнет 2, цикл, благодаря оператору break в последней строке программного модуля, прерывается. Соответственно, значение переменной x в первом примере остается равным x=0+1+2=3, а во втором примере 6.

$$x := \left| \begin{array}{l} z \leftarrow 0 \\ \text{for } i \in 0..5 \\ \quad \left| \begin{array}{l} z \leftarrow z + i \\ (\text{break}) \ \text{ if } i = 2 \end{array} \right. \end{array} \right. \qquad x := \left| \begin{array}{l} z \leftarrow 1 \\ \text{while } z < 10 \\ \quad \left| \begin{array}{l} z \leftarrow z + 1 \\ (\text{break}) \ \text{ if } z > 5 \end{array} \right. \end{array} \right.$$

$x = 3 \qquad x = 6$

Оператор continue используется для того, чтобы немедленно перейти в начало цикла и начать следующую итерацию. Этот оператор также обычно используется в составе конструкции вида ***continue if*** *(условие)*. Оператор ***continue*** используется в случаях, когда необходимо чтобы некоторые вычисления производились для одних итераций и не производились для других.

Пример 6. Требуется заполнить элементы квадратной матрицы в шахматном порядке.

```
m(n) := | x ← 0
        | for i ∈ 0..n
        |   for j ∈ 0..n
        |     | continue  if  mod(i + j, 2) = 0
        |     | x ← x + 1
        |     | m[i,j] ← x
        | m
```

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

4.1.5. Оператор **return** *(возврат значения).*

Как мы уже указывали, результат выполнения программного модуля помещается, как правило, в последней его строке. Но можно прервать выполнение программы в любой ее точке (например, с помощью условного оператора) и выдать некоторое значение, применив оператор ***return.*** В этом случае при выполнении указанного условия значение, введенное в поле ввода после ***return***, возвращается в качестве результата, и никакой другой код больше не выполняется. Вставляется в программу оператор ***return*** с помощью одноименной кнопки панели Programn return ▪

***Пример* 7.**

```
f(x) := | z ← x^2
        | return "zero"  if  x = 0
        | return 1  if  x = 1
        | z
```

f(0) = "zero"

f(1) = 1

f(3) = 9

4.1.6. Обработка ошибок.

MathCAD предоставляет пользователю некоторый контроль над ошибками, которые могут возникнуть при вычислении выражений или при выполнении программ. Для этой цели служит оператор on error, который можно вставить с помощью кнопки панели Programming

В *поле ввода справа* следует ввести выражение или программу, которые необходимо вычислить (известно, что это выражение может содержать ошибку при определенных значениях входных параметров). В *поле ввода слева* следует ввести выражение, которое будет выполнено вместо правого выражения, если при выполнении последнего возникнет ошибка.

Например, рассмотрим случай, когда аргументу функции присвоено нулевое значение, а на ноль делить нельзя. В таких случаях за счет оператора ***on error*** сообщение не выводится, а функции в этой точке присваивается значение, указанное слева от оператора ***on error*** – значение машинной бесконечности.

Пример 8.

$$f(x) := \infty \;\; \text{on error} \;\; \left| \begin{array}{l} z \leftarrow 1 \\ \text{for } i \in 1..100 \\ \quad z \leftarrow z + \left(\frac{1}{x}\right)^{i} \\ z \end{array} \right. \qquad \begin{array}{l} f(5) = 1.25 \\ f(0) = 1 \times 10^{307} \end{array}$$

В поле ввода слева может быть введено текстовое выражение, сообщающее об ошибке

$$f(x) := \text{"can't divide by zero"} \;\; \text{on error} \;\; \frac{1}{x} \qquad \begin{array}{l} f(2) = 0.5 \\ f(0) = \text{"can't divide by zero"} \end{array}$$

Иногда бывает необходимо, чтобы при определенных условиях результатом выражения было сообщение об ошибке, хотя в действительности при этом не возникает ни одной стандартной ошибки MathCAD. Для таких случаев в MathCAD предусмотрена встроенная функция ***error.*** В качестве аргумента этой функции нужно в кавычках указать текст сообщения об ошибке, который должен быть выведен. Таким образом, если необходимо, чтобы программа возвращала ошибку при определенном условии, то следует использовать конструкцию вида: **error («текст ошибки») if (условие).**

Пример 9.

$$f(x) := \left| \begin{array}{l} \text{error("только положительные x")} \;\; \text{if} \;\; x \le 0 \\ \frac{1}{x} \;\; \text{otherwise} \end{array} \right. \qquad \begin{array}{l} f(2) = 0.5 \\ f(-2) = \blacksquare \end{array}$$

только положительные x

Как было отмечено выше, очень часто при составлении программ приходится передавать переменные во внутрь подпрограмм. Как это делается показано на рисунке 11.

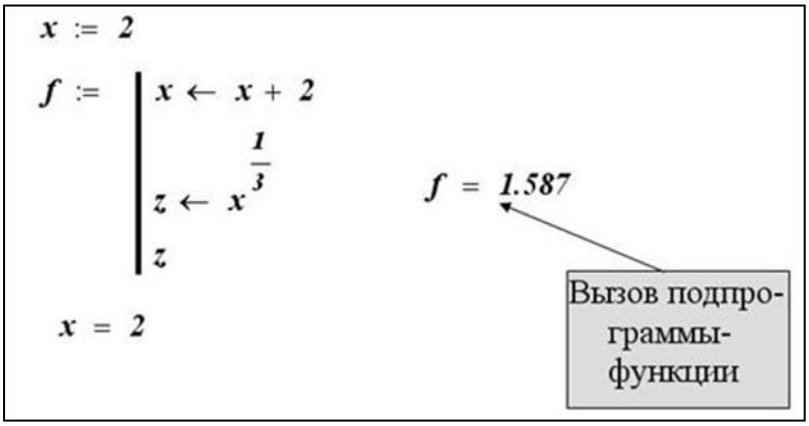
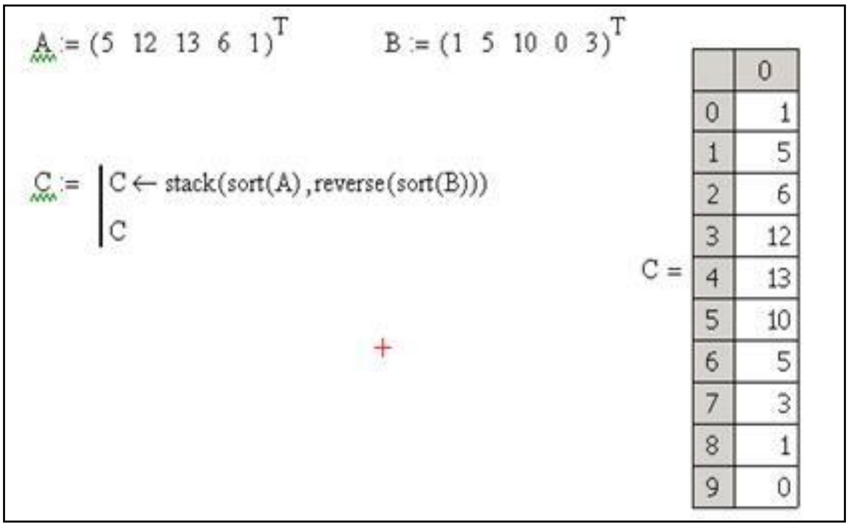


Рис.11

Важно знать: Для того, чтобы иметь возможность нормально вводить текст на русском языке в аргумент функции **error** (а также во все другие функции со строками), следует изменить шрифт, который используется во встроенном стиле **Constant**. Для того чтобы этот стиль правильно отображал русские буквы, установите курсор на любом числе или строковом выражении в формульном блоке. При этом в поле на панели инструментов **Formatting**, отображающем текущий стиль, должно быть написано – **Constant**. Теперь выберите из раскрывающегося списка шрифтов шрифт, поддерживающий кириллицу.

4.2. Примеры выполнения лабораторной работы.

Пример 1. Даны массивы A(5) и B(5). Получить массив C, в который записаны сначала элементы массива A в порядке возрастания, а затем элементы массива B порядке убывания. Для этого составляем программу:



Пример 2. По введенным значениям коэффициентов A, B, C определить корни квадратного уравнения $A \cdot cos^2 + B \cdot cosx + C = 0$.

$$f(a,b,c) := \left| \begin{array}{l} d \leftarrow \sqrt{b^2 - 4a \cdot c} \\ x1 \leftarrow \dfrac{-b+d}{2a} \\ x2 \leftarrow \dfrac{-b-d}{2a} \\ y1 \leftarrow acos(x1) \\ y2 \leftarrow acos(x2) \\ \begin{pmatrix} y1 \\ y2 \end{pmatrix} \end{array} \right.$$

$$f(2,10,5) = \begin{pmatrix} 2.169 \\ 3.142 - 2.17i \end{pmatrix}$$

$$a \cdot cos(x)^2 + b \cdot cos(x) + c \left| \begin{array}{l} solve, x \\ substitute, a = 2 \\ substitute, b = 10 \\ substitute, c = 5 \\ float, 3 \end{array} \right. \rightarrow \begin{pmatrix} 2.17 \\ 3.14 - 2.17 \cdot i \end{pmatrix}$$

Пример 3. Вычислить значение функции sin(x) с точностью e $=10^{-6}$.

Прежде чем составлять программу, данную функцию следует разложить в ряд Фурье и найти общий член последовательности:

$$sin(x)\ series, x = 0, 10 \rightarrow 1 \cdot x - \frac{1}{6} \cdot x^3 + \frac{1}{120} \cdot x^5 - \frac{1}{5040} \cdot x^7 + \frac{1}{362880} \cdot x^9$$

$$t(x,n) := (-1)^{n-1} \cdot \frac{x^{2n-1}}{(2n-1)!} \qquad e := 10^{-6}$$

$$fsin(x,e) := \left| \begin{array}{l} n \leftarrow 1 \\ s \leftarrow 0 \\ while\ \ |t(x,n)| \ge e \\ \quad \left| \begin{array}{l} s \leftarrow s + t(x,n) \\ n \leftarrow n + 1 \end{array} \right. \\ s \end{array} \right.$$

$$fsin(0.5, e) = 0.479426$$
$$sin(0.5) = 0.479426$$
$$fsin(1.1, e) = 0.891207$$
$$sin(1.1) = 0.891207$$

Пример 4. Составить программу решения уравнения $y = x^3 - 9$ методом половинного деления (дихотомии):

```
PolDel(f, a, b, eps) := | while |b − a| ≥ eps
                        |   | c ← (a + b)/2
                        |   | break  if f(c) = 0
                        |   | d ← f(a)·f(c)
                        |   | a ← c  if d > 0
                        |   | b ← c  if d < 0
                        | c ← | c  if f(c) = 0
                        |     | (a + b)/2  otherwise
                        | c

f(x) := x^3 − 9     a := 0     b := 9     eps := 0.0001     x := PolDel(f, a, b, eps)

      Корень уравнения:          Проверка:
        x = 2.0801               f(x) = 7.459× 10^−5
```

Пример 5. Построить график кусочно-непрерывной функции

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

Составление программы и построение графика функции приведено на рисунке 12.

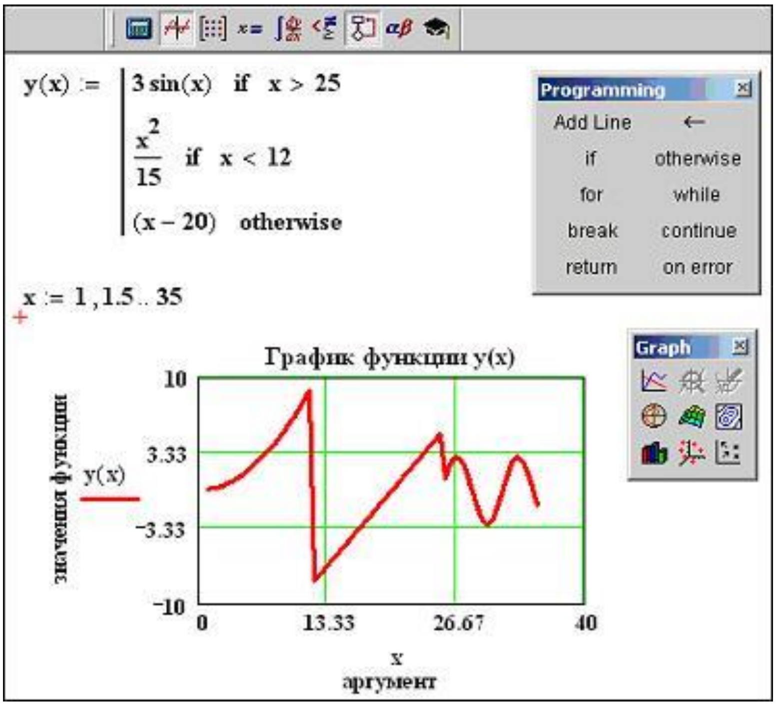


Рис.12

4.3 Задания к лабораторной работе

Задание 1. Отвечать на контрольные вопросы

1. Какая панель служит для вставки программного кода в документ MathCAD?

2. Можно ли операторы программирования набрать с клавиатуры?

3. С какой команды начинается создание программного блока? Как с ее помощью можно создавать разветвленный программный блок?

4. Что такое определение программного блока? Обращение к программному блоку?

5. Что такое глобальные и локальные переменные для программного блока? 6. Что может содержать последняя строка программного блока?

7. Как работает оператор if в программном блоке? Приведите пример.

8. Создание цикла с параметром в программном блоке. Приведите пример.

9. Создание цикла while в программном блоке. Приведите пример.

10. Для чего служат операторы break, continue в программном блоке? Приведите примеры.

11. Как работает оператор return в программном блоке? Приведите пример.

12. Как осуществляется обработка ошибок в программном блоке? Приведите пример.

Задание 2. Построить график кусочно-непрерывной функции

Таблица 1

№ варианта	**Функция**	**№ варианта**	**Функция**
1	$\begin{cases} x^2-1, & x\le -2 \\ \frac{1}{x+2}, & -2<x\le 3 \\ \sqrt{x}, & x>3 \end{cases}$	2	$\begin{cases} x^2-1, & x\le -1 \\ \frac{1}{x+1}, & -1<x\le 3 \\ \sin x, & x>3 \end{cases}$
3	$\begin{cases} x^2+1, & x\le 1 \\ \frac{1}{x-1}, & 1<x\le 5 \\ \sqrt{x-5}, & x>5 \end{cases}$	4	$\begin{cases} \sqrt{x^2-1}, & x\le -2 \\ \frac{1}{x+2}, & -2<x\le 3 \\ \sin x, & x>3 \end{cases}$
5	$\begin{cases} x^2-4, & x\le -2 \\ \frac{3}{x+2}, & -2<x\le 3 \\ \cos x, & x>3 \end{cases}$	6	$\begin{cases} (x-1)^2, & x\le -3 \\ \frac{-2}{x+3}, & -3<x\le 4 \\ \sqrt{x}, & x>4 \end{cases}$
7	$\begin{cases} \boldsymbol{2x+1}, \text{если } \boldsymbol{x} < 0 \\ \boldsymbol{-1,5x-1}, \text{если } \boldsymbol{0} \le \boldsymbol{x} < 2, \\ \boldsymbol{x-4}, \text{если } \boldsymbol{x} \ge \boldsymbol{2} \end{cases}$	8	$\begin{cases} \boldsymbol{x-0.5}, \text{если } \boldsymbol{x} < -2 \\ \boldsymbol{-2x-6.5}, \text{если } -\boldsymbol{2} \le \boldsymbol{x} \le \boldsymbol{-1}, \\ \boldsymbol{x-3.5}, \text{если } \boldsymbol{x} > -12 \end{cases}$

Литература

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.

5. Лабораторная работа № 5.

ПРИБЛИЖЕННЫЕ ЧИСЛА. АБСОЛЮТНАЯ И ОТНОСИТЕЛЬНАЯ ПОГРЕШНОСТИ

5.1 Основы теории погрешностей

5.1.1 Источники ошибок.

При решении практических задач мы всегда имеем дело с приближенными числами. Приближённое число отличается от точного на погрешность (ошибку), допущенную при решении конкретной задачи, и заменяет точное число в дальнейших расчетах.

Ошибки появляются :

1) из-за неточностей в исходных данных. Данная погрешность является неустранимой;

2) из-за использования итерационных методов, где используются большое число шагов вычислений, и каждый шаг вносит свою ошибку;

3) из-за погрешностей используемых методов.

Арифметические действия с приближёнными числами следует производить также приближённо, ограничиваясь той степенью точности, которая необходима для данной задачи.

5.1.2 Абсолютная погрешность

Обозначим буквами a_0, b_0, $c_0 \ldots$ и т.д. точные значения искомых величин, а приближённые значения этих величин обозначим через a_i, где i = 1,2,3 и т.д.

Абсолютная величина разности между точным и приближённым значением числа, т.е. $\Delta = |a - a_0|$, называется *истинной абсолютной погрешностью* этого числа [1- 3].

Например, точное значение числа $a_0 = 245{,}2$, а приближенное значение этого числа $a = 246$. Тогда абсолютная погрешность числа $|a - a_0| = |245{,}2 - 246| = 0{,}8$.

Число Δa называется *границей абсолютной погрешности.* Если задана Δa, то говорят, что число a есть приближённое значение числа a_0 с точностью до Δa, и пишут $a_0 = a \pm \Delta a$, или в виде двойного равенства: $a - \Delta a \le a_0 \le a + \Delta a$.

Например, если число $a_0 = 9{,}3 \pm 0{,}5$, то с помощью двойного неравенства его можно записать следующим образом: $9{,}3 - 0{,}5 \le a_0 \le 9{,}3 + 0{,}5$; $8{,}8 \le a_0 \le 9{,}8$.

5.1.3 Запись приближённых чисел. Значащие цифры

Приближенные числа могут содержать верные и сомнительные цифры.

Определение 1. Некоторая цифра приближённого числа считается *верной,* если его абсолютная погрешность Δa не превосходит единицы того разряда, в котором стоит эта цифра. В противном случае цифра считается *сомнительной* [1-4]. Очевидно, что если какая-либо цифра числа верна, то и все предшествующие ей цифры также являются верными.

Например, требуется найти верные и сомнительные цифры числа $a_0 = 945{,}673 \pm 0{,}03$.

Здесь $a = 945{,}673$, $\Delta a = 0{,}03$. Рассмотрим цифру 6. Она представляет собой цифру десятых долей, и, поэтому, единица разряда этой цифры равна 0,1. Очевидно, что $0{,}1 > 0{,}03$, то есть абсолютная погрешность числа а не превосходит единицы разряда, в котором стоит цифра 6. Следовательно, цифра 6 – верная. Очевидно, что все цифры стоящие перед цифрой 6, также являются верными.

Рассмотрим цифру 7- цифру сотых долей. Единица этого разряда = 0,01. Сравнивая эту единицу с погрешностью числа заметим, что $0{,}01 < 0{,}03$. Отсюда следует, что абсолютная погрешность числа больше единицы разряда, в котором стоит цифра 7. Следовательно, цифра 7 – сомнительная. Очевидно, что цифра 3 также является сомнительной.

В записи приближённых чисел принято соблюдать правила [1-5]:

1. Оставлять в записи приближённого числа только верные цифры.

2.Если приближенное число дано в виде десятичной дроби, где последние верные цифры нули, то их не следует отбрасывать.

3.Если приближенное число содержит в конце n сомнительных нулей, то их следует записывать как 10^n.

Например, требуется записать правильно следующие числа: а) $a = 0{,}075 \pm 0{,}000005$; б) $a = 746000000 \pm 5000$. Для первого числа погрешность не превосходит 0,00001. Следовательно, это число должно быть записано в виде $a = 0{,}07500$. Для второго числа первой верной цифрой является цифра десятков тысяч, поскольку погрешность числа не превосходит 10000. Значит, число должно быть записано в виде $a = 74600 \cdot 10^4$.

Приведем пример по III правилу. Требуется указать абсолютную погрешность приближённых чисел двух типов: а) $a = 2175000$; б) $a = 173 \cdot 10^4$. В первом из этих чисел указаны все нули разряда сотен, десятков, единиц. Все они верные цифры. Следовательно, абсолютная погрешность числа не превосходит единицы наименьшего разряда, в котором стоят верные цифры, т.е. $\Delta a = 1$; Во втором числе, согласно правилу III, на 10^4 заменены нули, не являющиеся верными цифрами. Следовательно, первой верной цифрой является цифра 3 в разряде десятков тысяч. Итак, Δa = 10000.

Определение 2. *Значащими* цифрами числа называют все его верные цифры, за исключением нулей, стоящих левее первой цифры, отличной от нуля [1,2]. Например, число 0,712 содержит три значащие цифры: 7, 1, 2; число 0,0016 – две значащие цифры: 1, 6; число 45,03 – четыре значащие цифры: 4, 5, 0, 3.

5.1.4 Округление приближённых чисел

При записи приближённых чисел требует их округление. Чтобы округлить число с точностью до указанного разряда, нужно цифры, стоящие правее указанного разряда, отбросить (в дробной части числа) или заменить нулями (в целой части числа).

При этом, если первая из отброшенных цифр:

1) меньше 5, то оставшиеся десятичные знаки сохраняются без изменения;

2) больше 5, то к последней оставшейся цифре прибавляется единица;

3) равна 5 и среди остальных отброшенных есть неравные нулю, то последняя оставшаяся цифра увеличивается на единицу;

4) равна 5 и все остальные отброшенные цифры равны нулю, то последняя оставшаяся цифра остается неизменной, если она четная, и увеличивается на единицу, если она нечетная (правило четной цифры).

В конце, в приложении 1 приведена блок-схема алгоритма округления чисел по заданной величине абсолютной погрешности.

5.1.5 Относительная погрешность

Определение 3. Относительной погрешностью приближённого числа **а**- называется отношение абсолютной погрешности этого числа к самому числу **а** [1-4], т.е. $\boldsymbol{\delta} = \frac{\Delta a}{a}$.

Так как абсолютная погрешность обычно неизвестна, то на практике используют понятие границы относительной погрешности числа.

Границей относительной погрешности $\boldsymbol{\delta_a}$ приближённого значения ***a*** называется отношение границы абсолютной погрешности $\boldsymbol{\Delta a}$ к модулю числа ***a***, т.е $\frac{\Delta a}{|a|}$. Отсюда следует, что чем меньше граница относительной погрешности, тем выше качество измерения.

Например, границу относительной погрешности числа a = 142,5 при Δa = 0,05 равна: $\delta_a = \frac{\Delta a}{a} \cdot 100\% = \frac{0{,}05}{142{,}5} \cdot 100\% = 0{,}03\%$.

Если требуется найти границу абсолютной погрешности числа $|a|$ = 1348 при δ_a= 0,04% , то следует пользоваться формулой $\Delta a = |a| \cdot \delta_a$.

Откуда Δa = 1348 · 0,0004 = 0,539 ≈ 0,5. Отсюда $\boldsymbol{a}$ = 1348 ± 0,5.

5.1.6 Правила подсчета цифр

На практике при выполнении действий с приближенными числами пользуются более простыми правилами, называемыми правилами подсчёта цифр.

1) При сложении и вычитании приближённых чисел в результате сохраняют столько десятичных знаков, сколько их в наименее точном числе.

2) Если в вычислениях точность задана заранее, то вычисления ведут с запасным знаком, который в результате округляют.

Например, при сложении приближённых чисел: 14,5 + 113,76 +12,783 +11,2161, округляем все числа по наименее точному числу (14,5), оставляя запасной знак, и производим сложение: 14,5 + 113,76 + 12,78 + 11,22 = 152,26. Запасной знак округляем и получаем ответ: 152, 3.

3) При умножении и делении приближённых чисел в результате сохраняют столько значащих цифр, сколько их в числе с меньшим количеством значащих цифр.

Например, требуется найти произведение двух приближённых чисел: 0,3862 · 0,85.

Для этого округляем первое число, оставляя один запасной знак, так как второе число содержит две значащие цифры. Таким образом, 0,3862 · 0,85 = 0,386 · 0,85 = 0,3281 ≈ 0,33.

4) При возведении в степень в результате сохраняют столько значащих цифр, сколько их в основании степени. Например, требуется вычислить $3{,}27^3$.$3{,}27^3$ = 34,965 ≈ 35,0. В результате оставлены три значащих цифры, так как столько значащих цифр содержит основание степени.

5) При извлечении корня сохраняют столько значащих цифр, сколько их в подкоренном выражении. Например, требуется вычислить $x = \sqrt(3{,}27\ )/3{,}284$.

Извлекая квадратный корень из числа 3,27, получим $\sqrt{3{,}27}$ = 1,81. Здесь оставлены три значащих цифры, так как столько значащих цифр содержит подкоренное выражение. Округляем остальные числа до трёх значащих цифр. В результате находим $x \approx 0{,}246$.

6) При выполнении промежуточных действий оставляют на один знак больше, чем требуют правила, а в результате запасной знак округляют.

5.1.7 Погрешность произведения

Относительная погрешность произведения нескольких приближенных чисел, отличных от нуля, не превышает суммы относительных погрешностей этих чисел [1,2].

Обозначим через $u = a_1 a_2 \cdots a_n$ произведение приближенных чисел, а через $U = A_1 \cdot A_2 \cdots A_n$, произведение соответствующих точных чисел , где $a_i > 0,\ A_i > 0, i = \overline{1,n}$.

Можно показать [1,6] , что:

$$\frac{\Delta u}{u} = \frac{\Delta a_1}{a_1} + \frac{\Delta a_2}{a_2} + \ldots + \frac{\Delta a_n}{a_n}, \text{ а } \left|\frac{\Delta u}{u}\right| \le \left|\frac{\Delta a_1}{a_1}\right| + \left|\frac{\Delta a_2}{a_2}\right| + \ldots + \left|\frac{\Delta a_n}{a_n}\right|.$$

Обозначая $\left|\frac{\Delta u}{u}\right|$ через δ, имеем: $\delta \le \delta_1 + \delta_2 + \ldots + \delta_n$, что требовалось доказать.

Это правило остается верным, если сомножители $a_i,\ i = \overline{1,n}$ имеют различные знаки.

Предельная относительная погрешность произведения равна сумме предельных относительных погрешностей сомножителей:

$\delta_u = \delta_{a_1} + \delta_{a_2} + \ldots + \delta_{a_n}$.

При умножении приближенного числа a на точный множитель $k \ne 0$ предельная относительная погрешность не изменяется ($\delta_u = \delta_a$), а предельная абсолютная погрешность увеличивается в модуль k раз $\Delta_u = |k|\Delta_a$.

5.1.8 Погрешность частного

Относительная погрешность частного не превышает суммы относительных погрешностей делимого и делителя [1, 3].

Пусть $u = \frac{x}{y}$, где x, y – приближенные числа. Тогда $\ln u = \ln x - \ln y$.

Отсюда

$$\frac{\Delta u}{u} = \frac{\Delta x}{x} - \frac{\Delta y}{y} \text{ или } \left|\frac{\Delta u}{u}\right| \le \left|\frac{\Delta x}{x}\right| + \left|\frac{\Delta y}{y}\right|.$$

Следовательно, $\delta \le \delta_1 + \delta_2$.

Предельная относительная погрешность частного равна сумме предельных относительных погрешностей делимого и делителя $\delta_u = \delta_x + \delta_y$.

5.1.9 Погрешность степени

Предельная относительная погрешность k-й степени числа x в k раз больше относительной погрешности самого этого числа [1,7].

Пусть $u = x^k$, где k – натуральное число. Тогда, $\dfrac{\Delta u}{u} = k\dfrac{\Delta x}{x}$. Следовательно, $\left|\dfrac{\Delta u}{u}\right| = k\left|\dfrac{\Delta x}{x}\right|$, или $\delta_u = k\delta_x$.

5.1.10 Погрешность корня

Предельная относительная погрешность корня k-й степени числа x в k раз меньше предельной относительной погрешности самого числа x [1,6].

Пусть $u = \sqrt[k]{x}$, где k – натуральное число. Тогда $\dfrac{\Delta u}{u} = \dfrac{1}{k}\dfrac{\Delta x}{x}$.

Следовательно, $\left|\dfrac{\Delta u}{u}\right| = \dfrac{1}{k}\left|\dfrac{\Delta x}{x}\right|$, или $\delta_u = \dfrac{1}{k}\delta_x$.

5.1.11 Погрешность функции

1. Пусть на отрезке $[a,b]$ дана непрерывная **функция одной переменной** $y = f(x)$ и пусть эта функция на этом же отрезке $[a,b]$ имеет непрерывную первую производную. Тогда для любых двух точек x, $x+\Delta x$ этого отрезка имеем:

$$f(x+\Delta x) - f(x) = \Delta f = f'(\xi)(x+\Delta x - x) = f'(\xi)\Delta x,$$

где $x < \xi < x + \Delta x$, $\Delta x > 0$. Отсюда следует, что

$$|\Delta f| = |f'(\xi)\Delta x| = |f'(\xi)| \cdot |\Delta x|$$

т.е. **абсолютная погрешность функции одной переменной пропорцио-нальна абсолютной погрешности аргумента с коэффициентом пропорцио-нальности** $|f'(\xi)|$.

Например, если дана функция $y = \cos x$, и известно, что $\Delta x > 0$, то абсолютная погрешность $\Delta y = |-\sin x|\,\Delta x = |\sin x|\Delta x$.

2. Относительная погрешность функции одной переменной $\delta = \dfrac{\Delta y}{|y|}$. Но из математического анализа следует, что $|dy| \approx \Delta y = |y'|\Delta x = |f'(x)|\Delta x$. Отсюда следует, что:

$$\delta = \frac{|f'(x)|\,\Delta x}{|f(x)|} = \left|(\ln f(x))'\right|\Delta x.$$

Таким образом, **относительная погрешность функции одной переменной пропорциональна абсолютной погрешности аргумента с коэффициентом пропорциональности** $(\ln f(x))'$.

3. Пусть дана функция многих переменных $U = f(x_1, x_2, ..., x_n)$. Обозначим через

$|\Delta x_i|$ – абсолютные погрешности аргументов x_i, где $i = \overline{1,n}$. Тогда :

$|\Delta U| = |f(x_1 + \Delta x_1, x_2 + \Delta x_2, ..., x_n + \Delta x_n) - f(x_1, x_2, ..., x_n)|$.

Можно показать, что в этом случае **абсолютная и относительная погрешности** функции определяются по формулам:

$$|\Delta U| \le \sum_{i=1}^{n} \left|\frac{\partial f}{\partial x_i}\right| \cdot |\Delta x_i|, \quad \delta \le \sum_{i=1}^{n} \left|\frac{\partial \ln U}{\partial x_i}\right| \cdot |\Delta x_i|$$

Предельная абсолютная и относительная погрешности функции определяются по формулам

$$\Delta_U = \sum_{i=1}^{n} \left|\frac{\partial f}{\partial x_i}\right| \cdot \Delta_{x_i} \quad \delta_U = \sum_{i=1}^{n} \left|\frac{\partial \ln U}{\partial x_i}\right| \cdot \Delta_{x_i}$$

В последних формулах Δ_{x_i} есть предельные абсолютные погрешности аргументов.

5.1.12 Обратная задача теории погрешностей

На практике встречаются задачи, когда нужно определить абсолютные погрешности аргумента функции Δ_{x_i}, чтобы абсолютная погрешность самой функции $U = f(x_1, x_2, ..., x_n)$ не превышала заданной величины. При решении этих задач пользуются *принципом равных влияний* [1,6-8]. Это означает, что частные дифференциалы $\frac{\partial f}{\partial x_i}\Delta_{x_i}$ в приведенных выше формулах одинаково влияют на образование предельной абсолютной погрешности

$$\Delta_U = \sum_{i=1}^{n} \left|\frac{\partial f}{\partial x_i}\right| \cdot \Delta x_i, \text{ т.е.:}$$

$$\left|\frac{\partial U}{\partial x_1}\right| \Delta_{x_1} = \left|\frac{\partial U}{\partial x_2}\right| \Delta_{x_2} = \cdots = \left|\frac{\partial U}{\partial x_n}\right| \Delta_{x_n} = \frac{\Delta_U}{n}. \text{ Отсюда:}$$

$$\Delta_{x_i} = \frac{\Delta_U}{n\left|\frac{\partial U}{\partial x_i}\right|}.$$

5.2 Примеры выполнения лабораторной работы

Пример 1. Вычислить абсолютную и относительную погрешности числа x*=2.5732, заданного всеми своими верными цифрами в строгом (узком) смысле. Абсолютная погрешность Δ(x)=0,00005. Вычисления выполнить в среде MathCad.

Решение примера в MathCad:

$x := 2.5732$

$\Delta x := 0.00005$

$\delta(x) := \Delta (x)/X$

$\delta(x) = 1.943 \cdot 10^{-5}$

Пример 2. Округлить число x*=2.5732, заданного всеми своими верными значащими цифрами в широком смысле, до 3 значащих цифр. Округлённое значение x1=2.57 и погрешность округления Δокр =0.0032. Выполнить округление числа x*=2.5732 в MathCad.

Решение примера в MathCad:

$x := 2.5732$

Округленное число x1=2.57

$x1 := 2.57$

По определению верной цифры числа в широком смысле абсолютная погрешность $\Delta(x^*) = 0.0001$.

$\Delta x := 0.0001$

Погрешность округления определяется по формуле

$\Delta окр = |x - x1|$

$\Delta окр := x - x1$

$\Delta окр \; 3.2 \cdot 10^{-3}$

Абсолютная погрешность округленного числа определяется по формуле

$\Delta x1 = \Delta x + \Delta окр$

$\Delta x1 := \Delta x + \Delta окр \quad \Delta x = 3.3 \cdot 10^{-3}$

$\Delta x1 = 3.3 \cdot 10^{-3} < 1 \cdot 10^{-2}$

Это означает, что в числе x три цифры (2,5,7) верны в строгом смысле по абсолютной погрешности. Относительная погрешность округленного числа определяется по формуле $\delta x1 = \Delta x1/x1$

$\delta x1 := \Delta x1/x1 \qquad \delta x1 = 1.284 \; 10^{-3}$

$\Delta x1 = 3.3 \cdot 10^{-3} < 1 \cdot 10^{-2}$

Это означает, что в числе x три цифры (2,5,7) верны в строгом смысле.

Пример 3. Вычислить абсолютные и относительные погрешности арифметических операций (сложения, умножения, деления, вычитания, возведения в степень и извлечения корня) с числами **a=0.7219** и **b=135.347**, заданными всеми своими значащими цифрами в широком смысле в среде Mathcad. Формулы для вычисления абсолютной и относительной погрешностей арифметических операций приведены в таблице 1.

Таблица 1

Арифметические действия	**Абсолютная погрешность**	**Относительная погрешность**
x + y	Δx + Δy	$\frac{\|x\|}{\|x+y\|}\delta x + \frac{\|y\|}{\|x+y\|}\delta y$
x - y	Δx + Δy	$\frac{\|x\|}{\|x-y\|}\delta x + \frac{\|y\|}{\|x-y\|}\delta y$
x*y	$\|x\|\Delta y + \|y\|\Delta x$	$\delta x + \delta y$
x/y	$\frac{\|x\|\Delta y + \|y\|\Delta x}{y^2}$	$\delta x + \delta y$
x^n	$nx^{n-1}\Delta x$	$n\,\delta x$
$\sqrt[n]{x}$	$\frac{1}{nx^{\frac{n-1}{n}}}$	$\frac{\delta x}{n}$

Решение примера в MathCad:

a := 0.7219 b := 135.347

Δa := 0.0001 Δb := 0.001

δa :=Δa/a δa = 0.000139

δb := Δb /b δb = 0.000007

3.1 Сложение

c := a + b   c = 136.0689

Δc := Δa + Δb   Δc = 0.0011

δc:= Δc/ c δc = 0.00000808

3.2 Вычитание

c1 := a – b   c1 = -134.6251

Δc1 := Δa + Δb   Δc1 = 0.0011

$\delta c1$:= Δc1 /$|c1|$ $\delta c1$ = 0.00000817

3.3 Умножение

c2 := a×b   c2 = 97.7070

Δc2 := $|a|$ ×Δb + $|b|$ ×Δa   Δc2 = 0.0143

δc2:= Δc2/ c2 δc2= 0.000146

3.4 Деление

c3:=a/b   c3 = 0.0053

Δc3 :=$\frac{|a| \times \Delta b + |b| \times \Delta a}{b^2}$ Δc3=0.00000078

δc3:= Δc3/ c3 δc3= = 0.00014591

3.5 Возведение в степень

c4:= a ^{3} c4 = 0.376

Δc4:= 3a^2· Δa Δc4 = 0.000156

δc4:= Δc4/ c4 δc4 = 0.000416

3.6 Извлечение корня

c5 := $\sqrt{a}$ c5 = 0.8496

Δc5 := $\frac{1}{2\sqrt{a}}\Delta$a Δc5 = 0.000059

δc5:= Δc5/ c5 δc5 = 0.000069

Пример 4. Оценить абсолютную и относительную погрешность вычисления функции y= $\sqrt{x-1}$·cos(x) при x =2.5732, заданного всеми своими верными значащими цифрами в строгом смысле, в системе Mathcad. Формулы расчета абсолютной и относительной погрешностей основных элементарных функций приведены в таблице 2.

Таблица 2

f (x)	**Δf (x)**	**δf (x)**
$\sqrt{x}$	$\frac{\Delta x}{2\sqrt{x}}$	$\frac{\delta x}{2}$
$\frac{1}{x}$	$\frac{\Delta x}{x^2}$	δx
sin x	$\lvert cos\ x\rvert \cdot \Delta$x	$\lvert xctgx\rvert\delta x$ ×
cos x	$\lvert \sin x\rvert \cdot \Delta$x	$\lvert xtgx\rvert\delta x$
tg x	$\frac{\Delta x}{\cos^2 x}$	$\frac{2\lvert x\rvert}{\lvert\sin 2x\rvert}\delta x$
ctg x	$\frac{\Delta x}{\lvert\sin^2 x\rvert}$	$\frac{2\lvert x\rvert}{\lvert\sin 2x\rvert}\delta x$
ln x	$\frac{\Delta x}{x}$	$\frac{\delta x}{\lvert\ln x\rvert}$
lg x	$\frac{\Delta x}{x\ln 10}$	$\frac{\delta x}{\lvert\ln x\rvert\ln 10}$
e^x	$e^x\Delta$x	$\lvert x\rvert\delta x$
10^x	10^x ln 10Δx	ln 10$\lvert x\rvert\delta x$
a^x	a^x lna Δx	ln a $\lvert x\ \rvert\delta x$
arcsin x	$\frac{\Delta x}{\sqrt{1-x^2}}$	$\frac{\lvert x\rvert}{\lvert\arcsin x\rvert\sqrt{1-x^2}}\ \delta x$

arccos x	$\frac{\Delta x}{\sqrt{1-x^2}}$	$\frac{\lvert x\rvert}{\lvert \arccos x\rvert\sqrt{1-x^2}}\ \delta x$
arctg x	$\frac{\Delta x}{1+x^2}$	$\frac{\lvert x\rvert}{\lvert \mathrm{arctg}\, x\rvert(1+x^2)}\ \delta x$
arcctg x	$\left\lvert\frac{1}{1+x^2}\right\rvert \Delta x$	$\frac{\lvert x\rvert}{\lvert \mathrm{arcctg}\, x\rvert(1+x^2)}\ \delta x$
x^n	$\lvert nx^{n-1}\rvert \Delta x$	n δx

Решение примера в MathCad

Исходная функция: y= $\sqrt{x-1}\cdot\cos(x)$

x:=2.5732

Округлим число до 4 значащих цифр: x1:=2.573

Найдём абсолютную погрешность:

Δ x :=$\frac{0.0001}{2}$ Δ x = 0.00005

Найдём погрешность округления:

xokr:=|x-x1| xokr=0.0002

Погрешность округления числа равна сумме погрешности исходного числа и погрешности округления:

Δ x 1:= xokr + Δ x Δ x 1 = 0.00025

Найдём относительную погрешность x:

$\delta x := \frac{\Delta\, x1}{x1}$ δx = 0.00009716

Найдём абсолютную погрешность функции:

$\Delta y(x) := \left\lvert\frac{\Delta}{\Delta x} y(x)\right\rvert \Delta x1$ $\Delta y(x)$ = 0.000253

Найдём относительную погрешность функции:

$\delta y(x) := \frac{\Delta y(x)}{\lvert y(x)\rvert}$ $\delta y(x)$ = 0.000239

$y(x) = -1.05706$ $y(x1) = -1.0569$

5.3. Задания к лабораторной работе

Задание 5.3.1 Отвечать на контрольные вопросы

1. Как классифицируются виды ошибок?
2. Дайте определения и приведите примеры устранимой и неустранимой погрешностей.
3. Что такое погрешность округления? Какова ее связь с разрядностью ЭВМ?

4. Что такое абсолютная и относительная погрешности?

5. Как вычислить относительную погрешность, зная абсолютную?

6. Как по абсолютной погрешности вычислить относительную погрешность?

7. Что значит цифра, верная в строгом, широком смыслах?

8. Как определить количество верных цифр по относительной погрешностиприближенного числа?

9. Как осуществить оценку погрешности значений элементарных функций одной переменной и многих переменных?

10. Как формулируется обратная задача теории погрешности?

Задание 5.3.2.

1. Вычислить абсолютную и относительную погрешности чисел, заданных всеми своими верными цифрами в строгом (узком) смысле. Испытуемые числа приведены в таблице 1.

2. Округлить числа заданные всеми своими верными значащими цифрами в широком смысле, до 3 значащих цифр. Испытуемые числа приведены в таблице 3

Таблица 3

Вариант	**Исследуемое число**	**Вариант**	**Исследуемое число**
1	23.3748, δ =0.27%.	11	0.66385, δ =0 .34%.
2	13.5726 ± 0.0072	12	4.884 45 ± 0.000 52
3	0.088748, δ = 0.56%.	13	2.8867, δ = 0.43%.
4	4, 57633 ± 0, 00042	14	5.6483± 0.0017
5	46.7843, δ = 0.32%.	15	3.7542, δ = 0.32%.
6	0.38725 ± 0.00112	16	0.8647 ± 0.0013
7	45.7832, δ = 0.18%	17	0.3944, δ = 0.15%.
8	0.752 44 ± 0.00013	18	3.6878 ± 0.0013
9	46.453, δ = 0.15%	19	0.85638, δ = 0.22%.
10	0.66385 ± 0.00042	20	13.6853± 0.0023

Задание 5.3.3.

Вычислить абсолютные и относительные погрешности арифметических операций (сложения, умножения, деления, вычитания, возведения в степень и извлечения корня) с числами **a** и **b**, заданными всеми своими значащими цифрами в широком смысле в среде Mathcad. Значения исследуемых чисел **a** и **b** приведены в таблице 4.

Таблица 4

Вариант	**Исследуемые числа**		**Вариант**	**Исследуемые числа**	
	a	**b**		**a**	**b**
1	23.3748	0,351	11	0.66385	23.653
2	13.5726	0,675	12	4.884 45	0.6754
3	0.088748	12.456	13	2.8867	985.234
4	4, 57633	567.34	14	5.6483	87.237
5	46.7843	85.345	15	3.7542	0.0456
6	0.38725	456.012	16	0.8647	345,564
7	45.7832	0.0123	17	0.3944	78.123
8	0.752 44	67.345	18	3.6878	65.056
9	46.453	45,037	19	0.85638	0.6754
10	0.66385	8.765	20	13.6853	879.23

Задание 5.3.4.

Оценить абсолютную и относительную погрешность вычисления функции $f(x)$ при значении x, заданном в таблице 5 всеми своими верными значащими цифрами в строгом смысле, в системе Mathcad.

Таблица 5

Вариант	$\boldsymbol{f(x)}$	$\boldsymbol{x}$	**Вариант**	$\boldsymbol{f(x)}$	$\boldsymbol{x}$
1	$lnx + \cos 2x$	3,758	11	$\dfrac{2+\sqrt{3x}}{x \cdot \cos x}$	2,0764
2	$\dfrac{\sqrt{3x}}{x^3} + tgx$	1,82	12	$\sqrt[3]{\sin 3x}$	8,6386
3	$\dfrac{\sqrt{3x}}{x^2} + 2sinx$	2,056	13	$\sqrt[3]{\sin x + 2x}$	0,958
4	$2^{\sqrt{\tan x}}$	1,78	14	$ln(x + \sqrt{1 + x^2}$	2,635
5	$\ln(sin3x)$	3,456	15	$\sqrt{cos5x + x^5}$	2,001

6	$\sqrt{1+\sin x}$	0,02	16	$\sqrt{2x+\cos x}$	3,996
7	$\sqrt{4x+ln3}$	1,97	17	$\sin 2x+\cos 3x$	8,506
8	$\frac{x}{x+1}-sin4x$	4,6354	18	5^{2x+4}	1,56
9	$(x^2+1)lnx$	2,808	19	$x^2\sin(3x+1)$	1,607
10	$\sin 2x\cdot lnx$	3,809	20	$(x^3+2)3^{3x}$	0,87

Литература

1. Демидович Б.П., Марон И.А. Основы вычислительной математики: Учеб. пособие. −М.: Наука, 2007. −672с.

2. Демидович Б.П. Марон И.А., Шувалов Э.З. Численные методы анализа. – М.: Наука, 1967. −368с.

3.Рыжиков Ю.И. Вычислительные методы: − СПб.: Изд-во БХВ−Петербург, 2007. −400с.

4. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы: Учеб. пособие. −М.: Наука, 2001. −632с.

5. Вержбицкий В.М. Численные методы (линейная алгебра и нелинейные уравнения). −М.: Высшая школа, 2000. −370с.

6.Вержбицкий В.М. Численные методы (математический анализ и обыкновенные дифференциальные уравнения). −М.: ОНИКС 212 век, 2005. −400с.

7. Лапчик М.П., Рагулина М.И., Хеннер Е.К. Численные методы. −М.: Academia, 2004. −384с.

8. Самарский А.А., Гулин А.В. Численные методы. −М.: Наука, 1989. −432с.

6. Лабораторная работа № 6.

ЧИСЛЕННОЕ РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ В ПАКЕТЕ MATHCAD

6.1. Теоретические основы

Нелинейным уравнением называется уравнение содержащее нелинейную функцию. Такими являются: нелинейная алгебраическая функция, трансцендентная функция или комбинирование этих функций.

Решением нелинейного уравнения называется такое значение $\boldsymbol{x}$, которое при подстановке в уравнение обращает его в тождество.

На практике не всегда удается найти точное решение такого уравнения. В этом случае решение нелинейного уравнения находят с применением приближенных (численных) методов.

Решение нелинейного уравнения состоит из двух этапов: на первом этапе производит отделение корней, а на втором этапе – их уточнение.

6.1.1 Методы отделения корней уравнения

Пусть дано нелинейное уравнение

$$f\ (x)\ =\ 0 \tag{1}$$

Корни этого уравнения представляют собой абсциссы точек пересечения, касания или других «особых» точек графика функции и оси ОХ (рис. 1).

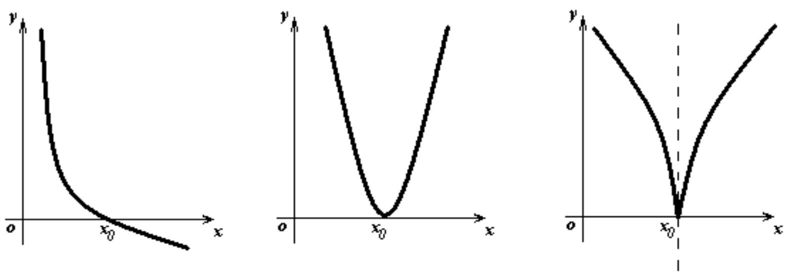


Рис.1

Для определения корней уравнения (1), прежде всего необходимо отделить их друг от друга. Это значит найти такие конечные промежутки на оси ОХ, внутри которых находится один и только один корень данного уравнения. Существуют графический и аналитический методы отделения корней.

а) Графический метод отделения корней

Отделение корней уравнения (1) можно выполнить графически, построив график функции y = f(x), по которому можно судить о том, в каких промежутках находится точка пересечения его с осью OX.

В случаях, когда функция имеет сложный вид, целесообразно представить уравнение f(x)=0 в виде

$$f_1(x) = f_2(x) \qquad (2)$$

с таким расчетом, чтобы графики функций $y = f_1(x)$ и $y = f_2(x)$ строились по возможности проще. В этом случае корень уравнения (2) представляет собой абсциссу точки пересечения графиков $y = f_1(x)$ и $y = f_2(x)$.

б) Метод исследования отрезков

Если на отрезке $[a\,;\,b]$ функция *y = f(x)* непрерывна, $f'(x)$ сохраняет свой знак (является монотонной), а значения *f(x)* на концах этого отрезка имеют разные знаки, то на этом отрезке имеется один и только один корень уравнения.

Оценку погрешности приближенного корня производится согласно следующей **теореме.** Пусть ζ — точный, а $\overline{x}$ — приближенный корни уравнения $f(x) = 0$, находящиеся на одном и том же отрезке $[a, b]$, причем $|f'(x)| \geq m_1 > 0$ при $a \leq x \leq b$. Тогда справедлива оценка:

$$|\overline{x} - \zeta| \leq \frac{|f(\overline{x})|}{m_1}, \qquad (3)$$

где $m_1 -$ наименьшее значение $f'(x)$ *при* $a \leq x \leq b$.

6.1.2 Методы уточнения корней

Существуют несколько способов уточнения корней. К ним относятся: метод половинного деления, метод простых итераций, метод пропорциональных частей (метод хорд), метод касательных Ньютона, комбинированный метод (хорд и касательных) и др. Рассмотрим их в краткой форме.

а) Метод простых итераций

Если каким нибудь способом получено приближенное значение x_0 корня уравнения, то уточнение приближения можно осуществить методом итераций (методом последовательных приближений).

Пусть задано уравнение *f(x) = 0,* представим его в виде $x = \varphi(x)$, где $|\varphi'(x)| \leq r < 1$ всюду на отрезке [a;b], содержащем единственный корень ξ. Исходя из некоторого начального значения $x_0 \in [a; b]$ можно построить последовательность:

$$x_1 = \varphi(x_0)\,,\quad x_2 = \varphi(x_1)\,,\quad x_3 = \varphi(x_2)\ \ldots\ldots\ x_n = \varphi(x_{n-1})\ \ldots$$

Можно показать, что пределом этой последовательности является единственный корень уравнения (1) на отрезке [a ; b], т.е.:

$$\lim_{n\to\infty} x_n = \varphi(\lim_{n\to\infty} x_{n-1}), \quad \text{или } \boldsymbol{\zeta = \varphi(\zeta)}.$$

Для практического применения метода итерации нужно выяснить выполнение достаточного условия сходимости итерационного процесса:

$$|\varphi'(x)| \le r < \mathbf{1} \tag{4}$$

При выполнении условия (4) процесс итерации $x_n = \varphi(x_{n-1})$, где n=1,2,… сходится независимо от выбора начального значения $a \le x_0 \le b$.

В этом случае оценка точности приближения производится по формуле:

$$|\zeta - x_n| \le \frac{\gamma^n}{1-\gamma} |x_1 - x_0| \tag{5}$$

Из этой формулы можно заметить, что итерационный процесс будет тем быстрее, чем меньше число γ. В некоторых случаях для оценки приближения применяется неравенство вида

$$|\zeta - x_n| < \varepsilon, \tag{6}$$

где ε — заданная предельная абсолютная погрешность корня. Процесс итерации следует продолжать до тех пор, пока для двух последовательных приближений x_{n-1} и x_n не будет обеспечено выполнение неравенства

$$x_n - x_{n-1} \le \frac{1-q}{q}\,\varepsilon. \tag{7}$$

б) Метод пропорциональных частей (метод хорд)

Метод хорд приближенного решения уравнения (1) имеет следующую геометрическую иллюстрацию: вместо точки пересечения оси OX и графика функции y = f(x), входящей в это уравнение, рассматривается точка пересечения данной оси и отрезка прямой, соединяющей концы дуги графика (рис.2).

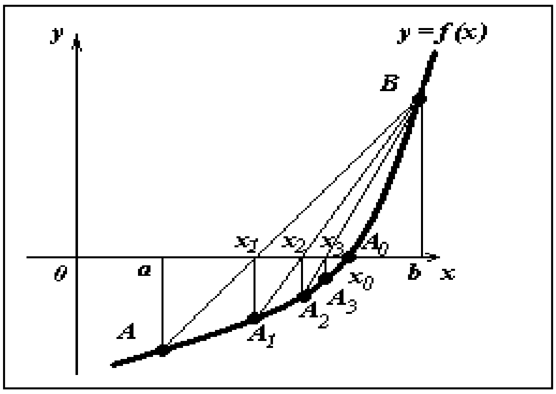


Рис.2

Пусть требуется вычислить действительный корень уравнения *f(x) = 0*, изолированный на отрезке [a; b].

Рассмотрим график функции *y = f(x).* Пусть *f (a) < 0,* а *f(b) > 0.* Точки А *(a; f(a))* и *B (b; f(b))*соединим хордой. Найдем точку x_1:

$$x_1 = a - \frac{(b-a)f(a)}{f(b)-f(a)} \tag{8}$$

Если $f(x_1) < 0$, то за новый, более узкий, интервал изоляции можно взять отрезок [x_1 ; b]. Соединив точки $A_1(x_1; f(x_1))$ и *B (b; f(b))*, получим в точке пересечения хорды с осью второе приближение x_2, которое вычислим по формуле:

$$x_2 = x_1 - \frac{(b-x_1)f(x_1)}{f(b)-f(x_1)} \tag{9}$$

и т. д. Последовательность чисел a, x_1, x_2 ....стремится к искомому корню.

В рассматриваемом случае неподвижен конец b, а последовательные приближения:

$$x_0 = a;$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f(b)-f(x_n)}(b - x_n) \quad (n=0,1,2,\ldots) \tag{10}$$

образуют ограниченную монотонно возрастающую последовательность

$$x_0 < x_1 < x_2 < \cdots < x_n < x_{n+1.} < \zeta < b.$$

(Если f (a) > 0, а f(b) < 0, то получим монотонно убывающую последова тельность).

Обобщая результаты, заключаем:

1) неподвижен тот конец, для которого знак функции f(x) совпадает со знаком ее второй производной $f''(x)$;

2) последовательные приближения x_n лежат по ту сторону корня ζ, где функция $f(x)$ имеет знак, противоположный знаку ее второй производной $f''(x)$. В обоих случаях каждое следующее приближение x_{n+1} ближе к корню ζ чем предшествующее x_n. Переходя к пределу в равенстве (10), будем иметь:

$$\bar{\zeta} = \bar{\zeta} - \frac{f(\bar{\zeta})}{f(b)-f(\bar{\zeta})}(b - \bar{\zeta}). \tag{11}$$

Отсюда $f(\bar{\zeta}) = 0$. Так как по предположению уравнение f(x) =0 имеет единственный корень ζ на интервале (a, b), то, следовательно, $\bar{\zeta} = \zeta$, что требовалось доказать.

Оценку точности приближения производим по формуле (3)

$$|x_n - \zeta| \le \frac{f'(x_n)}{m_1}, \text{ где } |f'(x_n)| \ge m_1 \text{ при } a \le x \le b.$$

Приведем еще одну формулу, позволяющую оценивать абсолютную погрешность приближенного значения x_n , если известны два последовательных приближения x_{n-1} и x_n :

$$|\zeta - x_n| \le \frac{M_1 - m_1}{m_1} |x_n - x_{n-1}|. \quad (12)$$

В этой формуле m_1, M_1 наименьшее и наибольшее значения модуля производной $f'(x)$ на отрезке (a,b).

В методе простых итераций вычисления следует вести до тех пор, пока не перестанут изменяться те десятичные знаки, которые мы хотим сохранить в ответе (т.е. пока не будет достигнута заданная степень точности).

в) Метод Ньютона (касательных)

Метод касательных отличается от метода хорд тем, что здесь рассматривается не секущая, соединяющая концы дуги графика, а касательная к графику (рис. 3). Точка пересечения касательной с осью OX дает приближенное значение корня.

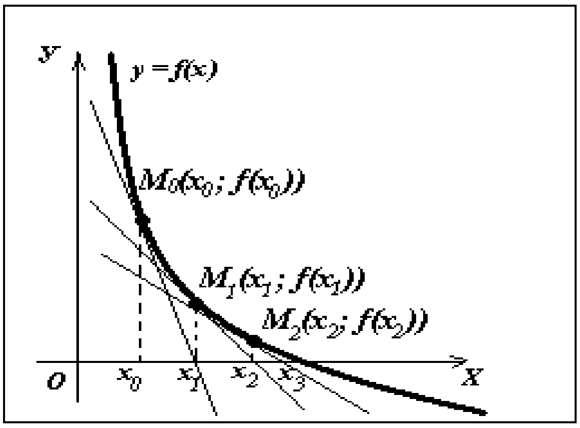


Рис.3

Пусть действительный корень уравнения f(x) = 0 на отрезке [a;b].

Выберем на этом отрезке такое число x_0, при котором $f(x_0)$ имеет тот же знак что и $f''(x_0)$, т.е. выполняется условие

$$f(x_0)f''(x_0) > 0 \quad (13)$$

Проведем в точке M_0 (x_0; $f(x_0)$) касательную к кривой $y = f(x)$. За приближенное значение корня примем абсциссу точки пересечения этой касательной с осью OX. Это приближенное значение корня найдется по формуле: $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$. Точно также находим: $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$.

Применив этот метод многократно получим:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (14)$$

Это есть итерационная формула метода Ньютона.

Применяя метод Ньютона, следует руководствоваться следующим правилом: **в качестве исходной точки x_0 выбирается тот конец интервала (a, b), которому отвечает ордината того же знака, что и знак $f''(x)$.**

Для оценки погрешности n-го приближения x_n можно воспользоваться формулой (7).

6.2. Средства MathCad для решения нелинейных уравнений

6.2.1. Численное и символьное решение квадратных уравнений

Пусть нелинейное уравнение *f(x)=0* есть квадратное уравнение. Для решения такого уравнения в MathCad необходимо описать функцию в виде **f(x):=<выражение>** и пользоваться следующими командами:

Тогда команда ***solve*** Возвращает аналитическое и (или) численное значение корней уравнения *f(x)=0.*

Решение уравнений в символьном виде позволяет найти точные или приближенные корни уравнения.

Если решаемое уравнение имеет параметр, то решение в символьном виде может выразить искомый корень непосредственно через параметр. Поэтому вместо того, чтобы решать уравнение для каждого нового значения параметра, можно просто заменять его значение в найденном символьном решении.

Если нужно найти все комплексные корни полинома со степенью меньше или равной 4, символьное решение даст их точные значения в одном векторе или в аналитическом или цифровом виде.

Команда **Символы / Переменные / Вычислить** позволяет решить уравне-ние относительно некоторой переменной и выразить его корни через остальные параметры уравнения. Как было отмечено выше, при численном и символьном решении нелинейных уравнений пользуются командой ***solve*** **.**

Пример 1. Получение численного решения для квадратного уравнения

$$5x^2 + 3x - 2 = 0:$$

Вариант I:

$$5\cdot x^2 + 3\cdot x - 2 \text{ solve}, x \rightarrow \begin{pmatrix} -1 \\ \frac{2}{5} \end{pmatrix}$$

Вариант II:

$$f(x) := 5\cdot x^2 + 3\cdot x - 2 \qquad f(x) \text{ solve}, x \rightarrow \begin{pmatrix} -1 \\ \frac{2}{5} \end{pmatrix}$$

Вариант III:

$$a := 5 \qquad b := 3 \qquad c := -2$$

$$f(x) := a\cdot x^2 + b\cdot x + c$$

$$f(x) \text{ solve}, x \rightarrow \begin{pmatrix} -1 \\ \frac{2}{5} \end{pmatrix}$$

Чтобы решить уравнение символьно необходимо:

– напечатать выражение (для ввода знака равенства используйте комбинацию клавиш **[Ctrl]=**).

– выделить переменную, относительно которой нужно решить уравнение, щелкнув на ней мышью.

– выбрать пункт меню **Символы / Переменные / Вычислить**.

Нет необходимости приравнивать выражение нулю. Если Mathcad не находит знака равенства, он предполагает, что требуется приравнять выражение нулю. На следующих двух примерах покажем порядок применения этой команды.

Пример 2. Символьное решение задачи нахождения точки пересечения двух прямых:

Given

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

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

$$\text{Find}(x, y) \rightarrow \begin{bmatrix} \dfrac{-(a - 2\cdot b\cdot\pi)}{(-1 + 8\cdot\pi)} \\ \dfrac{(-b + 4\cdot a)}{(-1 + 8\cdot\pi)} \end{bmatrix}$$

Пример 3. Символьное решение квадратного уравнения:

Вариант I:

$$a \cdot x^2 + b \cdot x + c \text{ solve}, x \rightarrow \begin{bmatrix} \frac{1}{2 \cdot a} \cdot \left[-b + \left(b^2 - 4 \cdot a \cdot c\right)^{\frac{1}{2}} \right] \\ \frac{1}{2 \cdot a} \cdot \left[-b - \left(b^2 - 4 \cdot a \cdot c\right)^{\frac{1}{2}} \right] \end{bmatrix}$$

Вариант II:

$$f(x) := a \cdot x^2 + b \cdot x + c \qquad f(x) \text{ solve}, x \rightarrow \begin{bmatrix} \frac{1}{2 \cdot a} \cdot \left[-b + \left(b^2 - 4 \cdot a \cdot c\right)^{\frac{1}{2}} \right] \\ \frac{1}{2 \cdot a} \cdot \left[-b - \left(b^2 - 4 \cdot a \cdot c\right)^{\frac{1}{2}} \right] \end{bmatrix}$$

6.2.2. Вычислительный блок Given

Блок предназначен для приближённого нахождения действительных решений уравнений и систем и имеет структуру:

Начальные значения:

для уравнения - значение переменной;

для систем - вектор значений переменных.

Начальные значения задаются в виде: **<переменная> := <значение>.**

Given

Уравнение (я). Знак равенства набирается жирным знаком (=) из палитры знаков соотношений: **Ctrl(=).**

Ограничительные условия на неизвестные (могут отсутствовать).

Выражения с функциями ***Find*** или ***Minerr.***

Между функциями ***Find*** и ***Minerr*** существует принципиальные различия. Первая функция используется, когда решение реально существует. Вторая функция пытается найти максимальное приближение даже к несуществующему решению путём минимизации среднеквадратической погрешности решения. При использовании этих функций рекомендуется дополнить блок проверкой решения.

Для выбора алгоритма уточнения корня необходимо щелкнуть правой кнопкой мыши на имени функции ***Find(x)*** и в появившемся контекстном меню (см. рисунок 4) выбрать подходящий алгоритм.

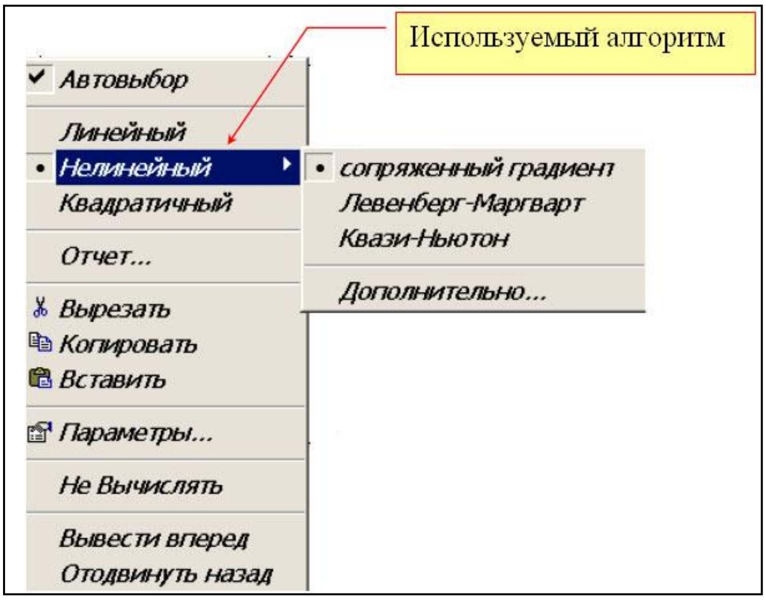


Рис.4.

Аналогично можно задать алгоритм решения и для функции ***Minerr(x).*** Использование численных методов в функциях *Find(x), Minerr(x)* требует перед блоком *Given* задать начальные значения переменным, по которым осуществляется поиск корней уравнения.

6.2.3. Численное решение нелинейных уравнений в пакете MathCad

Численное решение нелинейного уравнения осуществляется в два этапа. На первом этапе производится ***отделение корней***, т.е. установление интервалов, в каждом из которых находится только один корень. На втором этапе осуществляется ***уточнение отделенных корней***, т.е. доведение их значений до заданной точности.

Пример 4. Отделение корней уравнения на основе построения графика:

Пусть уравнение имеет вид: $2x^3 - 10\sin(x) + 3 = 0$

На рисунке 5 приведены: график функции, интервалы отделенных корней и рекомендации по заданию начальных приближений к корням.

$x := -2, 1.5 .. 2$

$y(x) := 2 \cdot x^3 - 10 \cdot \sin(x) + 3$

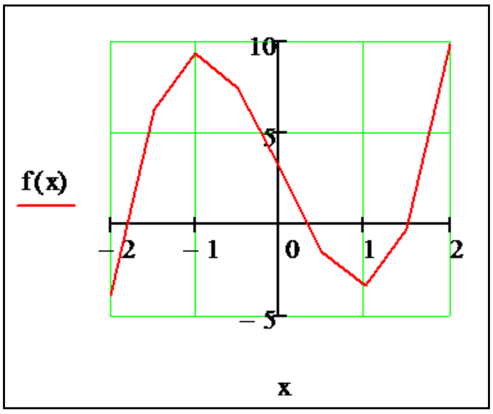


Рис.5

На интервале [-2,2] уравнение имеет три действительных корня, принадлежащие подинтервалам: [-2,-1.5],[0,0.5],[1.2,2]. За начальное приближение к корню на каждом подинтервале можно взять любое значение x, наиболее близкое к истинному значению корня, принадлежащему этому подинтервалу.

Пример 5. Уточнение отделённых корней для уравнения

$$2 \cdot x^3 - 10 \cdot \sin(x) + 3 = 0$$

Уточнение корня на интервале [-2, -1.5] при начальном значении равном -2 и ограничении x≤0

$F(x) := 2 \cdot x^3 - 10 \sin(x) + 3$

$x := -2$ Given

$F(x) = 0 \quad x \le 0$

$x1 := \mathrm{Find}(x)$

$x1 = -1.84783 \quad x1 = -1.848$ (с точностью 0.00

Проверка: $F(x1) = 1.16 \times 10^{-6}$

Точно таким же образом уточним корень на интервале [1.2; 2].

Уточнение корня на интервале [1.2, 2] при начальном значении равном 1.5 и ограничении 1.2≤x≤2

$x := 1.5$ Given

$F(x) = 0 \quad 1.2 \le x \le 2$

$x2 := \mathrm{Find}(x)$

$x2 = 1.51726 \quad x2 = 1.517$ (с точностью 0.00

Проверка: $F(x2) = -2.71 \times 10^{-8}$

6.2.4. Функция root .Функция реализует процесс уточнения корня любого нелинейного уравнения (не обязательно только алгебраического) итерационным методом. Перед её применением надо задать начальное приближение к корню, которое может быть определено по результатам табулирования функции или на основе построения графика. Функцию ***root*** можно применять в виде *x1:=* ***root****(F(x),x)* или ***root****(F(x), x,a,б)* = , где *F(x)* – имя функции или арифметическое выражение, соответствующее решаемому нелинейному уравнению, x – скалярная переменная, относительно которой решается уравнение, a и b – границы интервала локализации корня.

Если функцию root записать в виде **root (F(x), x, a, b)** , где **a, b** – пределы интервала изоляции корня, то нет необходимости задавать начальное значение **x**, так как оно определено в интервале [**a,b**] .

Демонстрация вариантов применения функции ***root*:**

Пример 6. Решить уравнение $\left(\frac{x}{2}\right)^3 + sin^2(2x) = 3ln|x|$.

Решение данного уравнения будем проводить в два этапа: отделение корней уравнения графически, уточнение корней уравнения.

Определим функцию *f*(*x*), равную левой части данного уравнения, когда правая равна нулю:

$$\boxed{f(x) := \left(\frac{x}{2}\right)^3 + \sin(2x)^2 - 3 \cdot \ln(|x|)}$$

Зададим ранжированную переменную *x* на некотором диапазоне с мелким шагом, например: x:=-5,-4.9..5 и построим график.

График функции *f(x)* будет выглядеть следующим образом (рисунке 6):

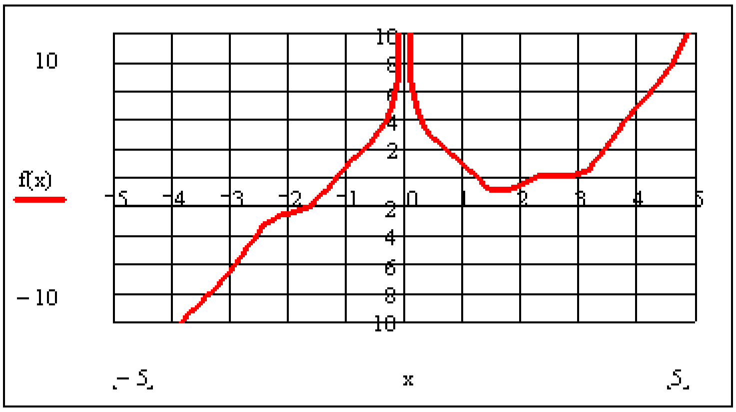


Рис. 6

Из графика функции *f(x)* видно, что уравнение $\left(\frac{x}{2}\right)^3 + sin^2(2x) = 3ln|x|$ – имеет три корня, которые приблизительно равны: $x_1 \approx$ -1; $x_2 \approx$ 1; $x_3 \approx$ 2,5.

Этап отделения корней завершен.

Уточним теперь корни уравнения с помощью функции ***root.***

Присвоим начальное приближение переменной *x* и укажем точность поиска корня: x:= -1 TOL:= 0.0001

Уточним заданное приближение к значению корня с помощью функции root: x_1:= root(f(x),x) x_1= -1.1395

Выполним проверку, подтверждающую, что первый корень найден с заявленной точностью:

$$\boxed{f(x1) = -2.8866 \times 10^{-14}}$$

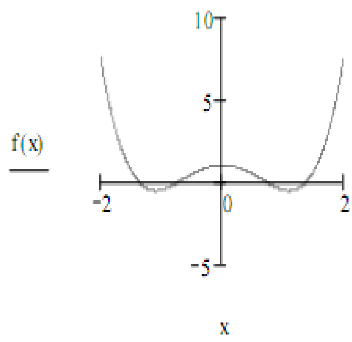
Начальное приближение можно не задавать при использовании в качестве аргументов *root* границ отрезка нахождения корня, например, второй корень можно уточнить:

$\boxed{\mathrm{root}(f(x), x, 1, 2) = 1.232}$

Пример 7. На рисунке 7 приведена схема уточнения корней нелинейного уравнения

Пусть уравнение имеет вид $x^4 - 2\cdot x^2 + \cos(x) = 0$

График функции $f(x) := x^4 - 2\cdot x^2 + \cos(x)$



Решение:

1) Начальное приближение: $x := -1$

$\mathrm{koren1} := \mathrm{root}\left(x^4 - 2\cdot x^2 + \cos(x), x\right)$

$\mathrm{koren1} = -1.378$

2) Начальное приближение: $x := -2$

$\mathrm{koren2} := \mathrm{root}(f(x), x)$

$\mathrm{koren2} = -1.378$

3) Начальное приближение: $x := 2$

$\mathrm{root}(f(x), x) = 1.378$

4) Начальное приближение: $x := 1.5$

$\mathrm{koren4} := \mathrm{root}(f(x), x)$

$\mathrm{koren4} = 1.378$

Остальные корни уточняются аналогично.

Рис. 7

Пример 8. Используя функцию **root(F(x),x),** найти все три корня уравнения $x^3 - 6\cdot x^2 + 21\cdot x + 52 = 0$, включая и два комплексных.

Замечание: *для вычисления всех трех корней использовался прием понижения порядка алгебраического уравнения. На рисунке 8 показан пример решения уравнения, имеющего комплексные корни.*

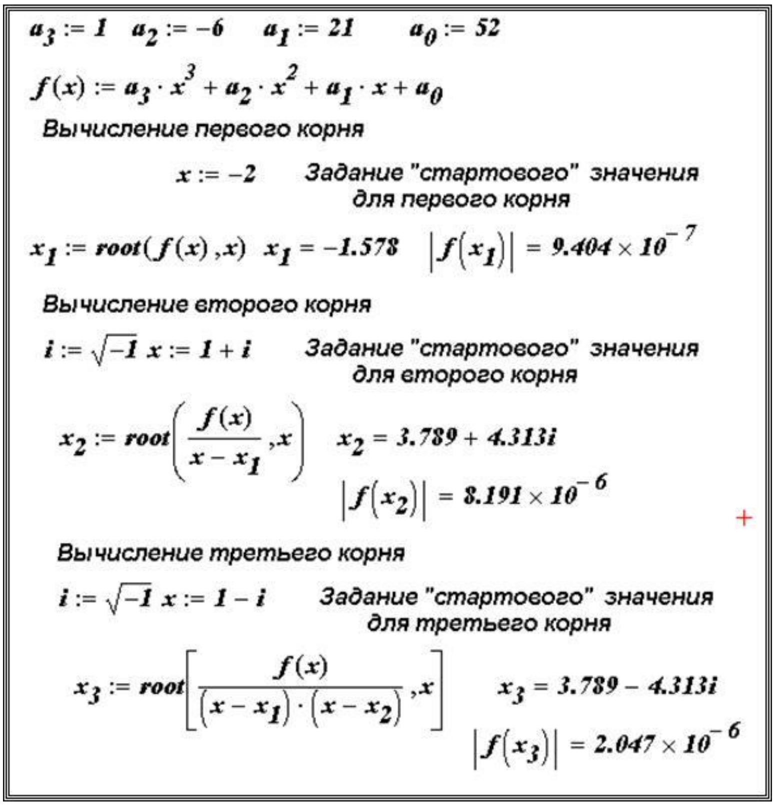


Рис. 8

Пример 9. Используя функцию ***root****(F(x),x)*, решить трансцендентное уравнение $\cos x = x + 0.2$. Схема решения этого примера приведена на рисунке 9.

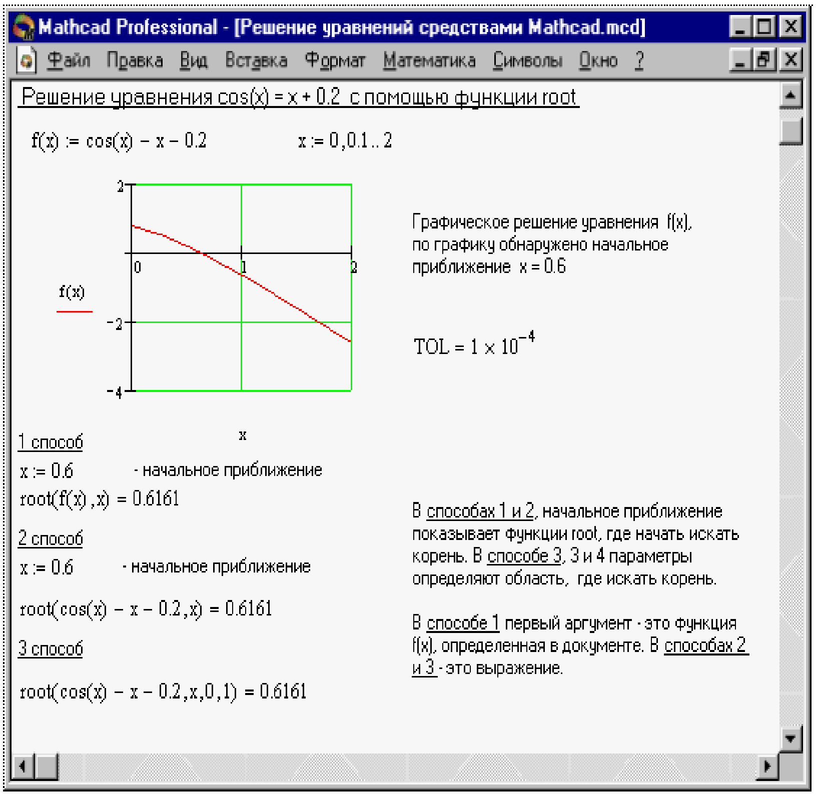


Рис. 9

Замечание!

Отсутствие сходимости функции root

Если после многих итераций Mathcad не находит подходящего приближения, то появится сообщение Can't converge to a solution. *(отсутствует сходимость). Эта ошибка может быть вызвана следующими причинами:*

Уравнение не имеет корней.

Корни уравнения расположены далеко от начального приближения.

Выражение имеет локальные max и min между начальным приближением и корнями.

Выражение имеет разрывы между начальными приближениями и корнями.

Выражение имеет комплексный корень, но начальное приближение было вещественным.

Чтобы установить причину ошибки, исследуйте график f(x).

Рекомендации по использованию функции root

Для изменения точности, с которой функция root ищет корень, нужно изменить значение системной переменной TOL. Если значение TOL увеличивается, функция root будет сходиться быстрее, но ответ будет менее точен. Если значение TOL уменьшается, то функция root будет сходиться медленнее, но ответ будет более точен. Чтобы изменить значение TOL в определенной точке рабочего документа, используйте определение вида $TOL := 0.01$. *Если два корня расположены близко друг от друга, следует уменьшить TOL, чтобы различить их.*

6.2.5. Функция polyroots. Для вычисления всех корней алгебраического уравнения порядка **n** (**n** $\le$ **5**) рекомендуется использовать функцию ***polyroots.***

Обращение к этой функции имеет вид ***polyroots(v)***, где ***v*** – вектор, состоящий из $n+1$ проекций, равных коэффициентам алгебраического уравнения, т.е.

$$v_0 = a_0,\ v_1 = a_1, \dots, v_n = a_n.$$

Эта функция не требует проведения процедуры локализации корней.

Пример 10. Решить уравнение $x^4 - 2x^2 - 7x + 9 = 0$.

Напечатаем левую часть уравнения, не приравнивая выражение к 0, и выделим синим курсором переменную *x*:

$$\boxed{x^4 - 2\cdot x^2 - 7\cdot x + 9}$$

Выберем из главного меню **Символы / Полиномиальные коэффициенты**. Появившийся вектор коэффициентов полинома выделим целиком синим курсором и вырежем в буфер обмена, используя кнопку **Вырезать** на панели инструментов (***Форматирование***) или комбинацию клавиш **Ctrl + X**.

Напечатаем v := и вставим вектор из буфера обмена, используя кнопку-**Вставить** на панели инструментов или комбинацию клавиш **Ctrl + V**.

Для получения результата напечатаем ***polyroots(v)* =:**

$$v := \begin{pmatrix} 9 \\ -7 \\ -2 \\ 0 \\ 1 \end{pmatrix} \qquad X := \text{polyroots}(v) \qquad X = \begin{pmatrix} -1.455 - 1.523i \\ -1.455 + 1.523i \\ 1.16 \\ 1.749 \end{pmatrix}$$

Пример 11. На рисунке 10. показано как команда ***polyroots(v)*** реализуется при решении уравнения $0{,}75 \cdot x^3 - 8 \cdot x + 5 = 0$.

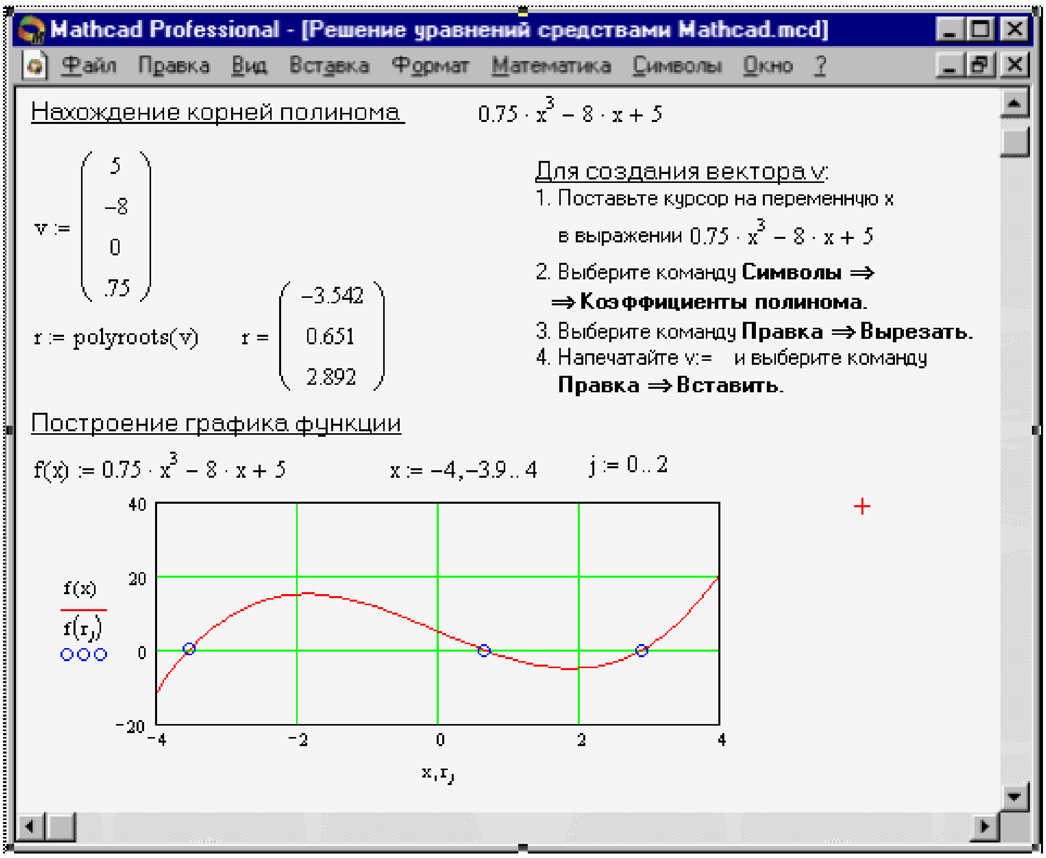


Рис.10

6.2.6. *Численное решение нелинейного уравнения итерационными методами в пакете MathCad*

Задача нахождения корня уравнения $f(x) = 0$ итерационными методами состоит в следующем:

1) отделение корней - отыскание приближенного значения корня (например, графическим методом);

2) уточнение корней - доведение их значений до заданной степени точности ε.

Кратко напомним, что при использовании *метода Ньютон* необходимо задаться начальным приближением x_0, расположенным достаточно близко к точному значению корня. Итерационный процесс в этом случае строится по формуле:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)},\ f'(x_i) \neq 0,\ i = 0,1,\ldots \tag{15}$$

Метод *простых итераций* решения уравнения $f(x) = 0$ состоит в замене исходного уравнения эквивалентным ему уравнением $x = \varphi(x)$ и построении итерационной последовательности по формуле:

$$x_{i+1} = \varphi(x_i), \quad i = 0,1,\ldots \tag{16}$$

Достаточным условием сходимости рассмотренных итерационных процессов является выполнение неравенства

$$|x_i - x_{i-1}| \leq \varepsilon \tag{17}$$

Это условие следует проверять на каждом шаге итерации.

Для решения нелинейных уравнений в MathCad имеется функция **until(*a*,*z*)**. Она возвращает ***z***, пока выражение ***a*** не становится отрицательным; ***a*** должно содержать дискретный аргумент.

Рисунок 11 иллюстрирует использование функции ***until*** для реализации *метода Ньютона*.

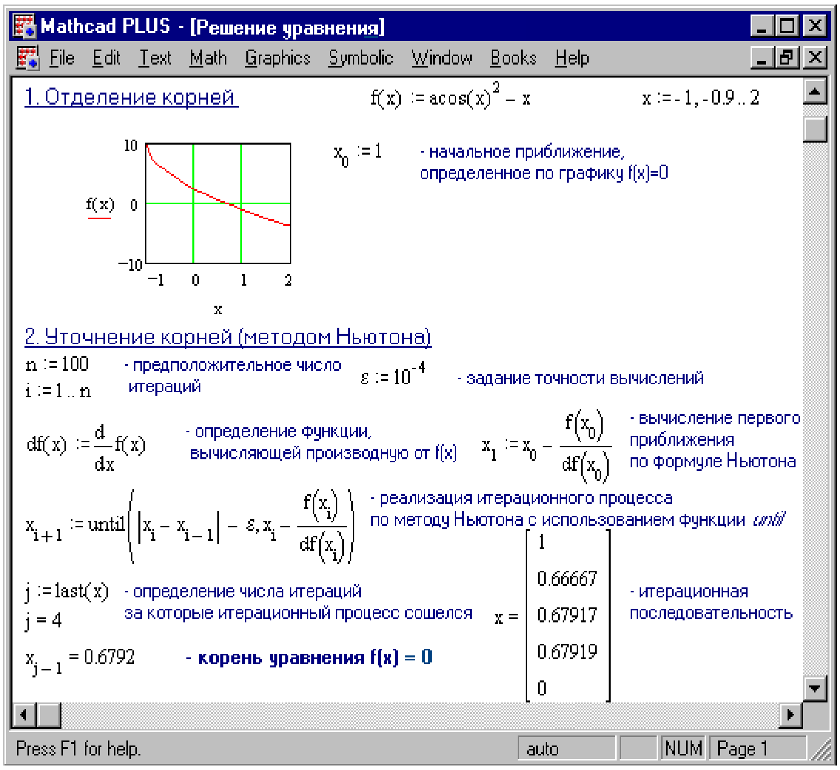


Рис.11

6.3. Задания к лабораторной работе

Задание 6.3.1. Отвечать на контрольные вопросы

1. В чем отличие алгебраического уравнения от трансцендентного?
2. Сущность и физический смысл процедуры отделения корней.
3. Обладает ли метод дихотомии гарантированной сходимостью?
4. Как выбирается начальное приближение в методе Ньютона?
5. Для каких функций не рекомендуется применять метод Ньютона?
6. Может ли в методе хорд интервал находиться с одной стороны корня?
7. Назовите условие выбора интервала в методе секущих?
8. К какому виду нужно преобразовать уравнение для метода итераций?
9. Можно ли воспользоваться методом итераций при невыполнения условий сходимости?
10. Какие функции могут быть использованы для решения нелинейных уравнений в пакете MathCAD ?
11. Какие функции для решения одного уравнения в MathCAD вы знаете? В чем их отличие?
12. Какая системная переменная отвечает за точность вычислений?
13. Как системная переменная TOL влияет на решение уравнения с помощью функции root?

Задание 6.3.2.

1) Отделите любой корень каждого уравнения Вашего варианта (Табл.1).

2) Найдите решение с точностью 0,0001 всеми приведенными методами.

3) Проведите проверку полученных решений.

4) Сравните методы решения и сделайте вывод о том, какими методами можно решать уравнения, имеющие действительные и/или комплексные корни.

Таблица 1

№ Варианта	Уравнение 1	Уравнение 2
1	$x - \sin x = 0{,}25$	$x^3 - 3x^2 + 9x - 8 = 0$
2	$tg(0{,}58x + 0{,}1) = x^2$	$x^3 - 6x - 8 = 0$
3	$x - \cos(0{,}387x) = 0$	$x^3 - 3x^2 + 6x + 3 = 0$
4	$tg(0{,}4x + 0{,}4) = x^2$	$x^3 - 0{,}1x^2 + 0{,}4x - 1{,}5 = 0$
5	$3x - \cos x - 1 = 0$	$x^3 + 0{,}2x^2 + 0{,}5x - 1{,}2 = 0$
6	$x + \lg x = 0{,}5$	$x^3 + 3x + 1 = 0$
7	$tg(0{,}5x + 0{,}1) = x^2$	$x^3 + 0{,}2x^2 + 0{,}5x - 1{,}2 = 0$

8	$x^2 + 4\sin x = 0$	$x^3 - 3x^2 + 12x - 9 = 0$
9	$ctg1{,}05x - x^2 = 0$	$x^3 - 0{,}2x^2 + 0{,}3x - 1{,}2 = 0$
10	$tg(0{,}4x + 0{,}3) = x^2$	$x^3 - 3x^2 + 6x - 2 = 0$
11	$x \lg x - 1{,}2 = 0$	$x^3 - 0{,}1x^2 + 0{,}4x - 1{,}5 = 0$
12	$1{,}8x^2 - \sin 10x = 0$	$x^3 + 3x^2 + 6x - 1 = 0$
13	$tg(0{,}3x + 0{,}4) = x^2$	$x^3 + 4x - 6 = 0$
14	$x^2 - 20\sin x = 0$	$x^3 + 0{,}2x^2 + 0{,}5x + 0{,}8 = 0$
15	$tg(0{,}47x + 0{,}2) = x^2$	$x^3 - 0{,}2x^2 + 0{,}3x + 1{,}2 = 0$
16	$2x - \lg x - 7 = 0$	$x^3 - 3x^2 + 6x - 5 = 0$
17	$tg(0{,}3x + 0{,}4) = x^2$	$x^3 - 0{,}2x^2 + 0{,}3x + 1{,}2 = 0$
18	$3x - \cos x - 1 = 0$	$x^3 - 3x^2 + 6x - 5 = 0$
19	$x^2 + 4\sin x = 0$	$x^3 - 0{,}1x^2 + 04x + 2 = 0$
20	$tg(0{,}36x + 0{,}4) = x^2$	$x^3 - 0{,}2x^2 + 0{,}4x - 1{,}4 = 0$
21	$x + \lg x = 0{,}5$	$x^3 + 0{,}4x^2 + 0{,}6x - 1{,}6 = 0$
22	$2^x + 5x - 3 = 0$	$2x^3 - 9x^2 - 60x + 1 = 0$

Задание 6.3.3. Построить график функции $f(x)$ и приблизительно определить один из корней уравнения вашего варианта (табл. 2).

Решить уравнение $f(x) = 0$ с точностью $\varepsilon = 10^{-4}$:

1) с помощью встроенной функции Mathcad ***root***;

2) методом Ньютона (касательных), используя функцию ***until***;

3) методом итераций, используя функцию ***until***.

Определить число итераций в каждом методе, с помощью функции ***last***.

Таблица 2

№ варианта	$f(x)$	№ варианта	$f(x)$
1	$3\sin(\sqrt{x}) + 0.35x - 3.8$ $x \in [2, 3]$	8	$3x - 4\ln x - 5$ $x \in [2, 4]$
2	$x - \dfrac{1}{3 + \sin(3.6x)}$ $x \in [0, 1]$	9	$e^x - e^{-x} - 2$ $x \in [0, 1]$
3	$\arccos x - \sqrt{1 - 0.3x^3}$ $x \in [0, 1]$	10	$\sqrt{1 - x} - \operatorname{tg} x$ $x \in [0, 1]$

4	$\sqrt{1-0.4x^2} - \arcsin x$ $x \in [0, 1]$	11	$\cos\left(\frac{2}{x}\right) - 2\sin\left(\frac{1}{x}\right) + \frac{1}{x}$ $x \in [1, 2]$
5	$3x - 14 + e^x - e^{-x}$ $x \in [1, 3]$	12	$0.1x^2 - x\ln x$ $x \in [1, 2]$
6	$0.25x^3 + x - 2$ $x \in [0, 2]$	13	$1 - x + \sin x - \ln(1 + x)$ $x \in [0, 2]$
7	$\arccos \frac{1-x^2}{1+x^2}$ $-x$ $x \in [2, 3]$	14	$e^{x-1} - x^3 - x$ $x \in [0, 1]$

Литература

1. Амосов, А. А.. Вычислительные методы : / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова. — Москва: Лань", 2014. — 672 с.

2. Мастяева И.Н., Семенихина О.Н. Численные методы: Учебное пособие. –М.; МЭСИ, 2011.

3. Мастяева И.Н., Семенихина О.Н. Численные методы: Практикум. -М; МЭСИ, 2011.

4. Демидович, Б.П.. Численные методы анализа. Приближение функций, дифференциальные и интегральные уравнения : учеб. пособие / Б.П. Демидович, И.А. Марон, Э.З. Шувалова; под ред. Б.П. Демидовича. — Москва: Лань, 2010. — 400 с.

5.Бахвалов Н.С. Численные методы в задачах и упражнениях: Учебное пособие/ Н.С. Бахвалов, Е.В. Чижонков, А.В. Лапин. – М.: БИНОМ. ЛЗ,2013-240с.

6.Бахвалов,Н.С.Численные методы. Решения задач и упражнения: Учебное пособие/ Н.С.Бахвалов, А.А.Корнев, Е.В. Чижонков.-М.:Бином,2016.-352с.

7. Лабораторная работа № 7.

РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ И НЕЛИНЕЙНЫХ УРАВНЕНИЙ В ПАКЕТЕ MATHCAD

7.1 Теоретические основы

7.1.1. Решение СЛАУ методом Крамера

Пусть система линейных уравнений содержит столько уравнений, каково количество независимых переменных, т.е. имеет вид

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{1n}x_n = b_1, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{n1}x_1 + a_{n2}x_1 + \cdots + a_{nn}x_n = b_n. \end{cases} \quad (1)$$

Такие системы линейных уравнений называются квадратными. Определитель, составленный из коэффициентов при независимых переменных системы (1), называется главным определителем системы. Мы будем обозначать его греческой буквой Δ. Таким образом,

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}. \quad (2)$$

Метод Крамера предназначен для решения систем линейных алгебраических уравнений (СЛАУ), у которых определитель матрицы системы отличен от нуля. Естественно, при этом подразумевается, что матрица системы квадратная. Решение системы уравнений методом Крамера проходит за три шага простого алгоритма:

1. Составить определитель матрицы системы и убедиться, что он не равен нулю, т.е. $\Delta \neq 0$.

2.Для каждой переменной $x_i (i = \overrightarrow{1,n})$ необходимо составить определитель Δ_{x_i} полученный из определителя Δ заменой i-го столбца столбцом свободных членов заданной СЛАУ.

3. Найти значения неизвестных по формуле $x_i = \frac{\Delta xi}{\Delta}\ (i = \overrightarrow{1,n})$:

$$x_1 = \frac{\Delta x_1}{\Delta},\ \ x_2 = \frac{\Delta x_2}{\Delta}, \dots,\ x_n = \frac{\Delta x_n}{\Delta}. \quad (3)$$

7.1.2. Решение СЛАУ методом обратной матрицы

Иногда этот метод называется матричным методом решения СЛАУ. Предположим, что требуется решить СЛАУ с *n* неизвестными:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{1n}x_n = b_1, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{n1}x_1 + a_{n2}x_1 + \cdots + a_{nn}x_n = b_n. \end{cases}$$

Как следует из теории, её легко перевести в матричную форму:

$$\boldsymbol{A} \cdot \boldsymbol{X} = \boldsymbol{B} \qquad (4)$$

где $\boldsymbol{A}$ — основная матрица системы, $\boldsymbol{B}$ и $\boldsymbol{X}$ — столбцы свободных членов и решений системы соответственно, т.е:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}, B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}, X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Умножим это матричное уравнение слева на A^{-1} — обратную матрицу к матрице A:

$$\boldsymbol{A^{-1}} \, \boldsymbol{(A \cdot X) = A^{-1} \cdot B}.$$

Так как $\boldsymbol{A^{-1}A = E}$, то $\boldsymbol{X = A^{-1}B}$. Правая часть уравнения дает столбец решений начальной системы. Условием применимости матричного метода есть невырожденность матрицы A (неравенство нулю определителя матрицы A).

7.1.3. Решение СЛАУ методом **Гаусса**

Одним из универсальных и эффективных методов решения линейных алгебраических систем является метод Гаусса, состоящий в последовательном исключении неизвестных.

Напомним, две системы уравнений называются эквивалентными (равносильными), если множества их решений совпадают. Другими словами, системы уравнений эквивалентны, если каждое решение одной из них является решением другой и наоборот. Эквивалентные системы уравнений получаются при элементарных преобразованиях их матриц.

К элементарным преобразованиям, сохраняющим равносильность СЛАУ, относятся следующие преобразования:

- смена мест уравнений СЛАУ;
- отбрасывание одного из двух одинаковых уравнений СЛАУ;
- умножение обеих частей какого-либо уравнения на число, отличное от нуля;
- замена одного из уравнений СЛАУ уравнением, полученным его почленным сложением с другим уравнением СЛАУ.

Пусть дана система уравнений

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_{1,} \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_{2,} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots. \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_{m.} \end{cases}$$

Процесс решения этой системы по методу Гаусса состоит из двух этапов. На первом этапе (прямой ход) система с помощью элементарных преобразований приводится к **ступенчатому,** или **треугольному** виду, а на втором этапе (обратный ход) идет последовательное, начиная с последнего по номеру переменного, определение неизвестных из полученной ступенчатой системы.

Предположим, что коэффициент данной системы $a_{11} \neq 0$, в противном случае в системе первую строку можно поменять местами с любой другой строкой так, чтобы коэффициент при x_1 был отличен от нуля.

Преобразуем систему, исключив неизвестное x_1 во всех уравнениях, кроме первого. Для этого умножим обе части первого уравнения на $-\frac{a_{21}}{a_{11}}$ и сложим почленно со вторым уравнением системы. Затем умножим обе части первого уравнения на $-\frac{a_{31}}{a_{11}}$ и сложим с третьим уравнением системы. Продолжая этот процесс, получим эквивалентную систему

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_{1,} \\ a_{22}{}^{(1)}x_2 + \cdots + a_{2n}{}^{(1)}x_n = b_2{}^{(1)}, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots. \\ a_{m2}{}^{(1)}x_2 + \cdots + a_{mn}{}^{(1)}x_n = b_1{}^{(1)}. \end{cases}$$

Здесь $a_{ij}^{(1)}, b_i^{(1)}$ $(\, i,j = \overline{2,m})$ – новые значения коэффициентов и свободных членов, которые получаются после первого шага.

Аналогичным образом, считая главным элементом $a_{22}^{(1)} \neq 0$, исключим неизвестное x_2 из всех уравнений системы, кроме первого и второго. Продолжим этот процесс, пока это возможно, в результате получим ступенчатую систему

$$\begin{cases} \tilde{a}_{11}x_1 + \tilde{a}_{12}x_2 + \cdots + \tilde{a}_{1n}x_n = b_1, \\ \qquad\qquad \tilde{a}_{22}x_2 + \cdots + \tilde{a}_{2n}x_n = b_2\,, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots. \\ \qquad\qquad \tilde{a}_{kk}x_k + \cdots + \tilde{a}_{kn}x_n = b_{k.} \end{cases}$$

где $\tilde{a}_{11}, \tilde{a}_{22}, \dots, \tilde{a}_{kk}$ – главные элементы системы (k$\leq n$).

Если в процессе приведения системы к ступенчатому виду появятся уравнения $0 \cdot x_1 + 0 \cdot x_2 + \cdots + 0 \cdot x_n = 0$, т. е. равенства вида $0 = 0$, то их отбрасывают, так как им удовлетворяют любые наборы чисел $(x_1, x_2, \dots, x_m)$.

Если же при $b_i \neq 0$ появится уравнение вида $0 \cdot x_1 + 0 \cdot x_2 + \cdots + 0 \cdot x_n = b_i$, которое не имеет решений, то это свидетельствует о несовместности системы.

При обратном ходе из последнего уравнения преобразованной ступенчатой системы выражается первое неизвестное x_k через все остальные неизвестные $x_{k+1}, \dots, x_n$, которые называют ***свободными.*** Затем выражение переменной x_k из последнего уравнения системы подставляется в предпоследнее уравнение и из него выражается переменная x_{k-1}. Аналогичным образом последовательно определяются переменные $x_{k-1}, \dots, x_1$.

Переменные $x_1, x_2, \dots, x_k$, выраженные через свободные переменные, называются ***базисными*** (зависимыми). В результате получается общее решение системы линейных уравнений.

7.1.4. Общее и базисное решения СЛАУ

Теорема Кронекера-Капелли. Система линейных уравнений совместна (имеет решение) тогда и только тогда, когда ранг матрицы равен рангу расширенной матрицы этой системы.

Вопрос о количестве этих решений разрешим с помощью следствия из теоремы Кронекера. Согласно ему, если ранг матрицы равен рангу расширенной матрицы ($rangA = rang\tilde{A} = n$, где n – количество неизвестных), то СЛАУ имеет единственное решение. Если же $rangA = rang\tilde{A} < n$, то количество решений заданной СЛАУ бесконечно.

Особый интерес представляет именно случай $rangA = rang\tilde{A} < n$, которым и займёмся в этой теме. Так как $rangA = rang\tilde{A}$, то обозначим эти ранги просто буквой r, т.е. $rangA = rang\tilde{A} = r$. Итак, $r < n$ и система неопределена, т.е. имеет бесконечное количество решений.

Что означает фраза «ранг матрицы равен r»? Она означает, что в определителе матрицы системы есть хотя бы один минор -го порядка, который не равен нулю. Напомню, что такой минор называется базисным. Базисных миноров может быть несколько. При этом все миноры, порядок которых выше r, равны нулю или не существуют.

Если коэффициенты при r переменных совместной СЛАУ образуют базисный минор матрицы системы А, то эти r переменных называют *базисными* или основными. Остальные $n - r$ переменных именуют *свободными* или неосновными.

Выбрать r базисных переменных в общем случае можно различными способами.

Решение СЛАУ, в котором все свободные переменные равны нулю, называется *базисным*.

Так как каждому разбиению переменных соответствует одно решение, а число способов разбиения не превосходит число сочетаний $C_n^r = \frac{n(n-1)\dots(n-r+1)}{1*2*3*\dots r}$, то и базисных решений имеется не более C_n^r.

Таким образом, совместная система m уравнений с n переменными (m<n) имеет бесчисленное множество решений, среди которых базисных решений конечное число, не превосходящее $C_n^r,\ где\ r \le m$.

7.1.5. Методы решения систем нелинейных уравнений

Системами нелинейных уравнений (СНУ) называются системы вида:

$$\begin{cases} f_1(x_1, x_2 \dots x_n) = 0, \\ f_2(x_1, x_2 \dots x_n) = 0, \\ \dots \\ f_n(x_1, x_2 \dots x_n) = 0, \end{cases} \tag{5}$$

где хотя бы одна из функций $f_i(x_1,...,x_n)$, $i = \overline{1,n}$ нелинейна. Решение СНУ является задачей более сложной, чем решение одного уравнения.

Наиболее распространенными методами уточнения корней для СНУ являются метод простых итераций и метод Ньютона. При этом полагают, что корни уравнения отделены.

а) Метод простых итераций для решения СНУ

Особенностью решения СНУ является переход от исходной системы (5) путем элементарных преобразований к эквивалентной системе вида:

$$\begin{cases} x_1 = \Phi_1(x_1, \dots, x_n) \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ x_n = \Phi_n(x_1, \dots, x_n) \end{cases} \tag{6}$$

Итерационный процесс начинается с помощью формул:

$$\begin{cases} x_1^{(k+1)} = \ \Phi_1\left(x_1^{(k)}, \dots, x_n^{(k)}\right) \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ x_n^{(k+1)} = \ \Phi_n\left(x_1^{(k)}, \dots, x_n^{(k)}\right) \end{cases}, \quad k = 0,1,2,\dots \tag{7}$$

При этом необходимо задавать начальное приближение:

$$x^{(0)} = \left(x_1^{(0)}, \dots, x_n^{(0)}\right).$$

Достаточным условием сходимости итерационного процесса является выполнение одного из двух условий :

$$\sum_{j=1}^{n}\left|\frac{\partial\Phi_i}{\partial x_j}\right| < 1, \forall i = \overline{1,n}, \quad \text{или} \quad \sum_{i=1}^{n}\left|\frac{\partial\Phi_i}{\partial x_j}\right| < 1, \forall j = \overline{1,n} \qquad (8)$$

Рассмотрим на примере, как система (1) приводится к виду (2), допускающему сходящийся итерационный процесс.

Пусть задана СНУ второго порядка (9):

$$\begin{cases} f_1(x,y)=0, \\ f_2(x,y)=0. \end{cases} \qquad (9)$$

Умножим первое уравнение системы (9) на неизвестную постоянную α, второе - на β, затем сложим их и прибавим к обеим частям уравнения x. Получим первое уравнение системы :

$$x+\alpha f_1(x,y)+\beta f_2(x,y)=x, \quad \text{где}$$

$$\Phi_1(x,y)=x+\alpha f_1(x,y)+\beta f_2(x,y). \qquad (10)$$

Далее, умножим первое уравнение системы (9) на неизвестную постоянную γ, второе - на δ, затем сложим их и прибавим к обеим частям уравнения y. Тогда второе уравнение системы будет иметь вид:

$$y+\gamma f_1(x,y)+\delta f_2(x,y)=y, \quad \text{где}$$

$$\Phi_2(x,y)=y+\gamma f_1(x,y)+\delta f_2(x,y). \qquad (11)$$

Уравнения (10) и (11) являются уравнениями системы (6).

Таким образом мы показали, что систему (5) можно свести к системе (6).

Неизвестные $\alpha,\beta,\gamma,\delta$ в этих двух уравнениях определим из достаточных условий сходимости (8):

$$\left|\frac{\partial\Phi_1}{\partial x}\right|+\left|\frac{\partial\Phi_2}{\partial x}\right|<1 \text{ и } \left|\frac{\partial\Phi_1}{\partial y}\right|+\left|\frac{\partial\Phi_2}{\partial y}\right|<1.$$

Отсюда:

$$\left|1+\alpha\frac{\partial f_1}{\partial x}+\beta\frac{\partial f_2}{\partial x}\right|+\left|\gamma\frac{\partial f_1}{\partial x}+\delta\frac{\partial f_2}{\partial x}\right|<1,$$

$$\left|\alpha\frac{\partial f_1}{\partial y}+\beta\frac{\partial f_2}{\partial y}\right|+\left|1+\gamma\frac{\partial f_1}{\partial y}+\delta\frac{\partial f_2}{\partial y}\right|<1.$$

Полагая равными нулю выражения под знаком модуля, получим систему, состоящую из четырех линейных алгебраических уравнений с четырьмя неизвестными $\alpha,\beta,\gamma,\delta$:

$$\begin{cases} 1+\alpha\dfrac{\partial f_1}{\partial x}+\beta\dfrac{\partial f_2}{\partial x}=0, \\ \gamma\dfrac{\partial f_1}{\partial x}+\delta\dfrac{\partial f_2}{\partial x}=0, \\ \alpha\dfrac{\partial f_1}{\partial y}+\beta\dfrac{\partial f_2}{\partial y}=0, \\ 1+\gamma\dfrac{\partial f_1}{\partial y}+\delta\dfrac{\partial f_2}{\partial y}=0. \end{cases} \qquad (12)$$

Решая эту систему определяем $\alpha,\beta,\gamma,\delta$, подставляя их в (10) и (11), находим рабочие итерационные формулы для метода простых итераций.

б) Метод Ньютона для решения СНУ

Пусть дана система нелинейных уравнений (5) и требуется решить эту систему приближенно, т.е. найти корни этой системы с заранее заданной точностью.

Пусть известно приближение $x_i^{(k)}$ решения этой системы x_i^*. Обозначим через Δx_i разницу между решением и его приближением:

$\Delta x_i = x_i^* - x_i^{(k)}$. Отсюда $x_i^* = x_i^{(k)} + \Delta x_i$, $i=\overline{1,n}$

Подставим полученное выражение для x_i^* в исходную систему.

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

Решая эту систему получим приближенные значения поправок Δx_i для данного приближения, т.е. $\Delta x_i^{(k)}$. Эти поправки не позволяют сразу получить точное решение x_i^*, но дают возможность приблизиться к решению, т.е. получить новое приближение решения $x_i^{(k+1)} = x_i^{(k)} + \Delta x_i^{(k)}$, $i=\overline{1,n}$.

Для этого разложим функции f_i в ряды Тейлора в окрестности $x_i^{(k)}$, и ограничимся первыми членами разложения. В результате получим систему вида:

$$\sum_{i=1}^{n}\frac{\partial f_i(x_1^{(k)},x_2^{(k)},...,x_n^{(k)})}{\partial x_i}\Delta x_i^{(k)} = -f_j(x_1^{(k)},x_2^{(k)},...,x_n^{(k)}),\quad j=\overline{1,n}$$

Коэффициенты этого уравнения можно вычислить, используя последнее приближение решения $x_i^{(k)}$.

Для решения системы линейных уравнений порядка n=2, 3 можно использовать формулы Крамера, а при большей размерности системы - метод последовательного исключения Гаусса.

Значения поправок используются для оценки достигнутой точности решения. Если максимальная по абсолютной величине поправка меньше заданной точности **ε**, расчет завершается. Таким образом, условие окончания расчета задается неравенством:

$$\delta = \max_{i=1,n} \left|\Delta x_i^{(k)}\right| \le \varepsilon \tag{13}$$

В матричной форме систему нелинейных уравнений можно записать так:

$$W\left(X^{(k)}\right) \cdot \Delta X^{(k)} = -F\left(X^{(k)}\right). \tag{14}$$

В выражении (14):

$$W(X) = \left(\frac{\partial f_j}{\partial x_i}\right)_{n,n} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix},$$ - матрица Якоби;

$$\Delta X^{(k)} = \begin{bmatrix} \Delta x_1^{(k)} \\ \Delta x_2^{(k)} \\ \dots \\ \Delta x_n^{(k)} \end{bmatrix}$$ - вектор поправок;

F(X) - вектор-функция;

$W(X^{(k)})$ – матрица Якоби, вычисленная для очередного приближения;

$F(X^{(k)})$ – вектор-функция, вычисленная для очередного приближения.

Выразим вектор поправок $\Delta X^{(k)}$ из выражения (14):

$$\Delta X^{(k)} = -\,W^{-1}\left(X^{(k)}\right) \cdot F\left(X^{(k)}\right),$$

где W^{-1} – матрица, обратная матрице Якоби.

Отсюда, с учетом $x_i^{(k+1)} = x_i^{(k)} + \Delta x_i^{(k)}$, окончательно получим формулу последовательных приближений метода Ньютона решения СНУ в матричной форме:

$$X^{(k+1)} = X^{(k)} - W^{-1}\left(X^{(k)}\right) \cdot F\left(X^{(k)}\right). \tag{15}$$

Отметим, что метод Ньютона сходится очень быстро, если $det|W| \neq 0$ и начальное приближение $X^{(0)}$ выбрано близким к решению.

Таким образом, алгоритм решения СНУ методом Ньютона состоит в следующем:

1) Задается размерность системы **n**, требуемая точность **ε**, начальное приближенное решение $\overline{x}^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}\right)$;

2) Вычисляются элементы матрицы Якоби $W = \left(\dfrac{\partial f_i}{\partial x_i}\right)_{n,n}$;

3) Вычисляется обратная матрица W^{-1};

4) Вычисляется вектор функция $F = (f_i)_n$, $f_i = f_i(x_1, x_2, \ldots, x_n)$, $i = \overline{1, n}$;

5) Вычисляются вектор поправок $\Delta X = W^{-1} \cdot F$.

6) Уточняется решение $X_{n+1} = X_n + \Delta X$.

7) Оценивается достигнутая точность $\delta = \max\limits_{i=\overline{1,n}} \left|\Delta x_i^{(k)}\right|$;

8) Проверяется условие завершения итерационного процесса **δ≤ε.**

Если оно не соблюдается, алгоритм выполняется снова с пункта 2.

7.2. Средства Mathcad для решения систем линейных и нелинейных алгебраических уравнений

7.2.1. Точные методы решений

К точным методам решения СЛАУ относятся метод Крамера, метод Гаусса и метод обратной матрицы. В MathCad решение СЛАУ задается функцией $lsolve\ (A, B)$. При выполнении этой функции возвращает вектор решения, такой, что $\boldsymbol{A \cdot X = B}$.

7.2.2. Итерационные методы.

Итерационные методы позволяют получить решение системы с заданной точностью путем сходящихся итерационных процессов (метод итерации, метод Зейделя и др.). Вследствие неизбежных округлений результаты даже точных методов являются приближенными. При использовании итерационных методов, сверх того, добавляется погрешность метода.

Эффективное применение итерационных методов существенно зависит от удачного выбора начального приближения и быстроты сходимости процесса.

MathCAD дает возможность решать СЛАУ и неравенств. Максимальное число уравнений и переменных в системе равно 50. Для решения таких систем уравнений необходимо выполнить следующее:

•Задать начальное приближение для всех неизвестных, входящих в систему уравнений. Mathcad решает систему с помощью итерационных методов.

•Напечатать ключевое слово ***Given.*** Оно указывает Mathcad, что далее следует система уравнений.

•Введите уравнения и неравенства в любом порядке. Используйте- **[Ctrl]=** для печати символа **=.** Между левыми и правыми частями неравенств может стоять любой из **символов < , >, ≤ , ≥** .

•Введите любое выражение, которое включает функцию ***Find,***
например: ***a*:= *Find*(*x, y*).**

***Find*(*z*1, *z*2, . . .)** возвращает точное решение системы уравнений (число аргументов должно быть равно числу неизвестных).

Ключевое слово ***Given***, уравнения и неравенства, которые следуют за ним, и какое-либо выражение, содержащее функцию *Find*, называют ***блоком решения уравнений.***

Внутри блока решения недопустимы следующие выражения:

•Ограничения со знаком **≠.**

•Дискретный аргумент или выражения, содержащие дискретный аргумент в любой форме.

•Неравенства вида $a < b < c$.

Блоки решения уравнений не могут быть вложены друг в друга, каждый блок может иметь только одно ключевое слово *Given* и имя функции *Find*.

Функция, которая завершает блок решения уравнений, может быть использована аналогично любой другой функции. Можно произвести с ней следующие три действия:

• вывести найденное решение, напечатав выражение вида: ***Find*(*var*1, *var*2,…) =.**

• определить переменную с помощью функции *Find*: ***a* := *Find*(*x*)** – скаляр, ***var* := *Find*(*var*1, *var*2,…)** - вектор.

•Определить функцию с помощью ***Findf*(*a, b, c, …*) := *Find*(*x, y, z, …*).**

Сообщение об ошибке No solution was found. Try changing the guess value or the value of TOL or CTOL. (Решение не найдено) при решении уравнений появляется, когда:

•Поставленная задача может не иметь решения.

•Для уравнения, которое не имеет вещественных решений, в качестве начального приближения взято вещественное число и наоборот.

•В процессе поиска решения последовательность приближений попала в точку локального минимума невязки. Для поиска искомого решения нужно задать различные начальные приближения.

•Возможно, поставленная задача не может быть решена с заданной точностью. В таких случаях попробуйте увеличить значение **TOL**.

Имеются некоторые задачи, для которых возможности Mathcad позволяют находить решения линейных систем уравнений **в символьном** (аналитическом) виде. Решение таких систем в символьном виде позволяет найти точные или приближенные решения. Чтобы решить систему уравнений в символьном виде, необходимо выполнить следующее:

•Напечатать ключевое слово ***Given.***

•Напечатать уравнения в любом порядке ниже слова *Given*. Удостоверьтесь, что для ввода знака = используется **[Ctrl]=**.

•Напечатать функцию ***Find***, соответствующую системе уравнений.

•Нажать **[Ctrl].** (клавиша CTRL, сопровождаемая точкой). Mathcad отобразит символьный знак равенства ® .

•Щелкнуть мышью на функции *Find*.

•На рисунке 1 иллюстрируется решение СЛАУ в Mathcad в численном и символьном виде, используя блоки ***Given … Find.***

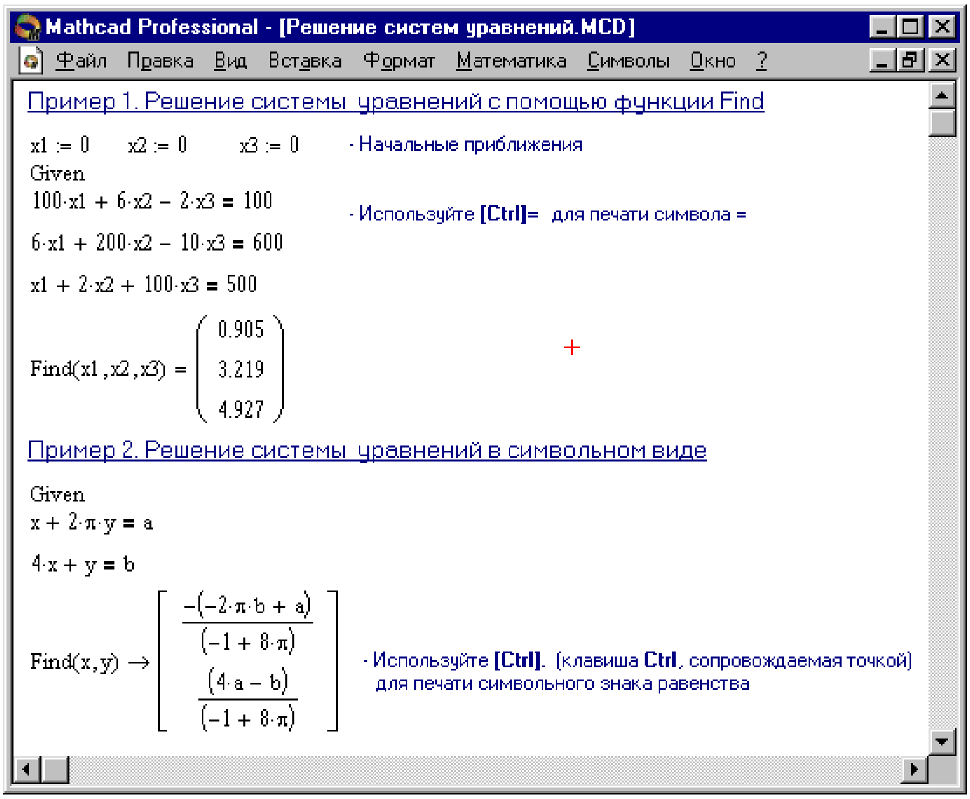


Рисунок 1

В MathCad решение систем уравнений реализуется с помощью функций ***Find*** *и* ***Minner*.** Правила использования функции *Minner* такие же, как и функции *Find*.

Minerr(*z*1, *z*2, . . .) возвращает приближенное решение системы уравнений. Число аргументов должно быть равно числу неизвестных.

Если ***Minner*** используется в блоке решения уравнений, необходимо всегда включать дополнительную проверку достоверности результатов.

Пример 1. Решить систему линейных уравнений используя блоки ***Given…Minerr***.

$$\begin{cases} 3x + 4y - 5z = 1; \\ 7x - 4y + z = -2; \\ 2y - 9z = 5. \end{cases}$$

Сделать проверку.

Решение:

Порядок решения системы этим способом аналогичен порядку использования блока ***Given … Find*** и представлен ниже вместе с проверкой:

$x := 0 \quad y := 0 \quad z := 0$

Given

$3x + 4y - 5z = 1$

$7x - 4y + z = -2$

$2y - 9z = 5$

$X := \mathrm{Minerr}(x, y, z) \quad X = \begin{pmatrix} -0.345 \\ -0.257 \\ -0.613 \end{pmatrix}$

$3X_1 + 4X_2 - 5X_3 = 1$

$7X_1 - 4X_2 + X_3 = -2$

$2X_2 - 9X_3 = 5$

7.2.3. Решение систем линейных уравнений методом обратной матрицы

Рассмотрим систему n линейных алгебраических уравнений относительно n неизвестных $x_1, x_2, \ldots, x_n$:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \\ a_{n1}x_1 + a_{n2}x_1 + \cdots + a_{nn}x_n = b_n \end{cases}$$

Системы линейных уравнений удобно решать с помощью функции функции ***lsolve*.**

В соответствии с правилом умножения матриц рассмотренная система линейных уравнений может быть записана в матричном виде: A·X=B, где

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix}, \; b = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{bmatrix}$$

Матрица A, столбцами которой являются коэффициенты при соответствующих неизвестных, а строками - коэффициенты при неизвестных в соответствующем уравнении, называется матрицей системы; матрица-столбец b, элементами которой являются правые части уравнений системы, называется матрицей правой части или просто правой частью системы. Матрица-столбец x, элементы которой - искомые неизвестные, называется решением системы.

Если матрица A - неособенная, то есть det A не равен 0 то система, или эквивалентное ей матричное уравнение, имеет единственное решение.

В самом деле, при условии det A не равно 0 существует обратная матрица A^{-1}. Умножая обе части уравнения на матрицу A^{-1} получим:

$$\boldsymbol{A^{-1}Ax = A^{-1}b, \quad x = A^{-1}b.}$$

Это есть решение системы уравнений и оно единственное.

Как было указано выше, системы линейных уравнений удобно решать с помощью функции ***lsolve***.

lsolve(*A, b*) возвращает вектор решения *x* такой, что *Ax* = *b*.

Аргументы:

A - квадратная, не сингулярная матрица.

b - вектор, имеющий столько же рядов, сколько рядов в матрице ***A***.

На рисунке 2 показано решение системы трех линейных уравнений относительно трех неизвестных.

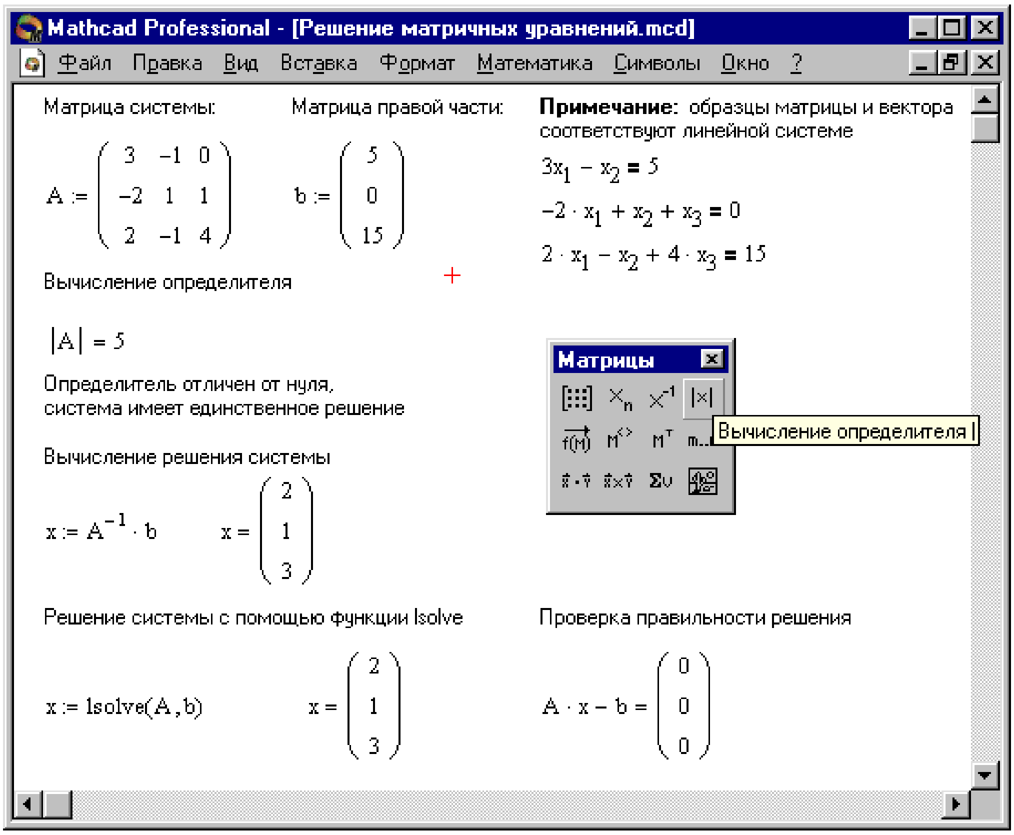


Рис. 2

7.2.4. Решение системы линейных уравнений матричным способом

Создадим матрицу *A*, состоящую из коэффициентов при неизвестных системы. Для этого напечатаем A := , вызовем окно создания массивов (**Ctrl + M**). Число строк (**Rows**) и столбцов (**Columns**) матрицы данной системы равно 3. Заполним пустые места шаблона матрицы коэффициентами при неизвестных системы, как показано ниже:

$$A := \begin{pmatrix} 3 & 4 & -5 \\ 7 & -4 & 1 \\ 0 & 2 & -9 \end{pmatrix}$$

Зададим вектор *b* свободных членов системы. Сначала напечатаем *b* :=, затем вставим шаблон матрицы(**Ctrl + M**), где количество строк (**Rows**) равно 3, а количество столбцов (**Columns**) равно 1. Заполним его:

$$b := \begin{pmatrix} 1 \\ -2 \\ 5 \end{pmatrix}$$

Решим систему матричным способом по формуле

$$\boxed{X := A^{-1} b \qquad X = \begin{pmatrix} -0.345 \\ -0.257 \\ -0.613 \end{pmatrix}}$$

Решим систему с помощью функции **lsolve**:

$$X := \text{lsolve}(A, b) \quad X = \begin{pmatrix} -0.345 \\ -0.257 \\ -0.613 \end{pmatrix}$$

Для проверки правильности решения системы, полученного матричным способом, достаточно вычислить произведение A · X, которое должно совпасть с вектором-столбцом свободных членов *b*:

$$A \cdot X = \begin{pmatrix} 1 \\ -2 \\ 5 \end{pmatrix}$$

7.2.5. Решение систем линейных уравнений методом Гаусса

Метод Гаусса состоит в том, что систему линейных уравнений приводят последовательным исключением неизвестных к эквивалентной системе с треугольной матрицей (к ступенчатой матрице):

$$\begin{cases} x_1 + \alpha_{12}x_2 + \ldots + \alpha_{1n}x_n = \beta_1, \\ \qquad x_2 + \ldots + \alpha_{2n}x_n = \beta_2, \\ \qquad\qquad \ldots\ldots\ldots\ldots\ldots \\ \qquad\qquad\qquad x_n = \beta_n, \end{cases}$$

решение которой находят по рекуррентным формулам:

$$x_n = \beta_n, \quad x_i = \beta_i - \sum_{j=i+1}^{n} \alpha_{ij}x_j, \quad (i = n-1, n-2, \ldots, 1)$$

В матричной записи это означает, что сначала (прямой ход метода Гаусса) элементарными операциями над строками приводят расширенную матрицу системы к ступенчатому виду:

$$A_p = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & \alpha_{12} & \ldots & \alpha_{1n} & \beta_1 \\ 0 & 1 & \ldots & \alpha_{2n} & \beta_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & \beta_n \end{bmatrix},$$

а затем (обратный ход метода Гаусса) эту ступенчатую матрицу преобразуют так, чтобы в первых *n* столбцах получилась единичная матрица:

$$\begin{bmatrix} 1 & 0 & \ldots & 0 & x_1 \\ 0 & 1 & \ldots & 0 & x_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & x_n \end{bmatrix}.$$

Последний, (*n* + 1) столбец этой матрицы содержит решение системы.

В Mathcad прямой и обратный ходы метода Гаусса выполняет функция ***rref*(*A*).**

На рисунке 3 показано решение системы линейных уравнений методом Гаусса, в котором используются следующие функции: **rref(*A*)** - возвращает ступенчатую форму матрицы *A*;

augment(*A, B*) - возвращает массив, сформированный расположением- ***A*** и ***B*** бок о бок. Массивы ***A*** и ***B*** должны иметь одинаковое число строк;

submatrix(*A, ir, jr, ic, jc*) - возвращает субматрицу, состоящую из всех элементов с ***ir*** по ***jr*** и столбцах с ***ic*** по ***jc***.

Удостоверьтесь, что ***ir*** ≤ ***jr*** и ***ic*** ≤ ***jc***, иначе порядок строк и (или) столбцов будет обращен.

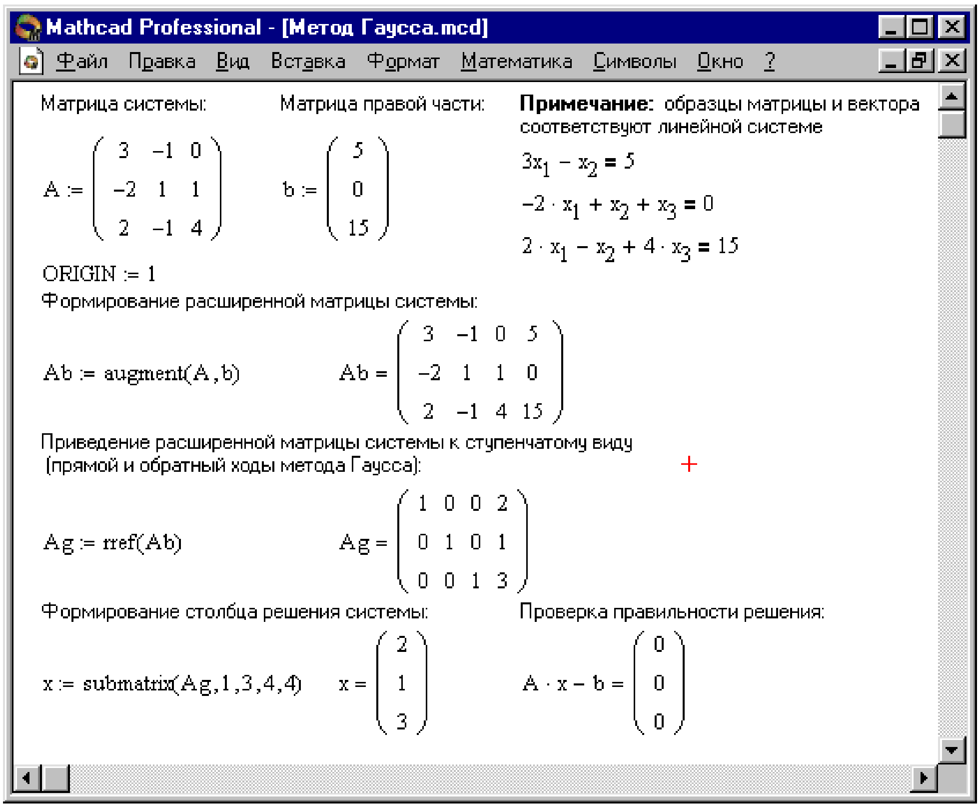


Рис. 3

7.2.6. Решение систем нелинейных уравнений

MathCAD дает возможность находить решение системы нелинейных уравнений (СНУ) численными методами.

Система нелинейных уравнений имеет вид:

$$\begin{cases} F1(x_1, x_2, \dots, x_n) = 0, \\ F2(x_1, x_2, \dots, x_n) = 0, \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ F\,n(x_1, x_2, \dots, x_n) = 0. \end{cases}$$

Для решения системы таких уравнений необходимо выполнить следующие этапы.

Задание начального приближения для всех неизвестных, входящих в систему уравнений. При небольшом числе неизвестных этот этап можно выполнить графически, как показано в следующем примере.

Пример 3. Дана система уравнений:

$$\begin{cases} y = x^2, \\ y = 8 + 3x. \end{cases}$$

Определим начальные приближения для решений этой системы. Для этого построим графики функций (Рис.4).

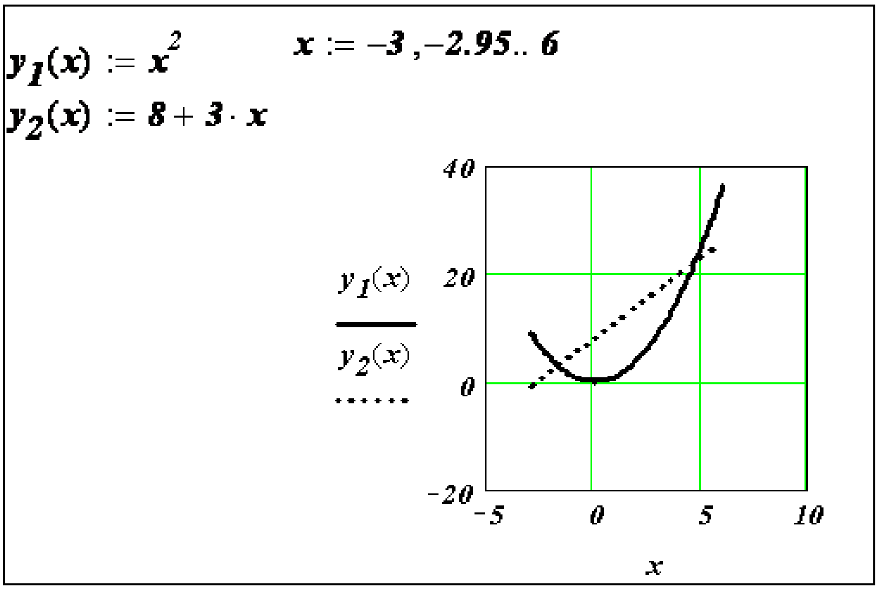


Рис.4

Видно, что система имеет два решения: для первого решения в качестве начального приближения может быть принята точка (-2, 2), т. е при x=2, y=2, а для второго решения – точка (5, 20).

Вычисление решения системы уравнений с заданной точностью.

Для этого используется уже известный вычислительный блок ***Given.***

Функция ***Find*** вычисляет решение системы уравнений с заданной точностью, и вызов этой функции имеет вид ***Find(x),*** где *x* – список переменных, по которым ищется решение. Начальные значения этим переменным задаются в блоке < Начальные условия >. Число аргументов функции должно быть равно числу неизвестных.

Пример 4. Используя блок *Given,* вычислить все решения системы предыдущего примера. Выполнить проверку найденных решений .

Решение (приведено на рисунке 5).

x:=-2 y:=2 ***Начальное приближение для первого решения***
Given
y=8+3*x
y=x²
s_A:=Find(x,y)
s_A= $\begin{pmatrix} -1.702 \\ 2.895 \end{pmatrix}$ ***Проекции первого решения***
x:=5 y:=20 ***Начальное приближение для второго решения***
Given
y=8+3*x
y=x²
x>0 ***Ограничительность на положительность проекции x для второго решения***
s_B:=Find(x,y)
s_B= $\begin{pmatrix} 4.702 \\ 22.105 \end{pmatrix}$ ***Проекции второго решения***

Рис.5

Пример 5. Используя функцию ***Minerr***, вычислите решение системы урав-нений $\begin{cases} x + y = 0{,}95, \\ (x^2+1)^2 + (y^2+1)^2 = 5.5. \end{cases}$

Решение:

x:=0 y:=0
Given
(x²+1)²+(y²+1)²=5.5 ***Знак равенства жирный***
x+y=0.95 ***Знак равенства жирный***
z:=Minerr(x,y)
z=(-0.106; 1.056) Найденное решение

Рис.6

Пример 6. Решить СНУ уравнений с помощью функции ***find.***

$$\begin{cases} \boldsymbol{\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1} - \text{Эллипс}, \\ y - x^2 = -1 - \textit{Парабола}. \end{cases}$$

Решение. Решение этой системы приведено на рисунках 7a и 7b.

Первое решение найдено приняв ограничение x<0 (рис. 7 *a*), второе - приняв ограничение x>0 (рис. 7 b). Следует отметить, что для решения системы уравнений указали начальные значения *x* и *y* (т.е. *x*:= -1 *y*: =1).

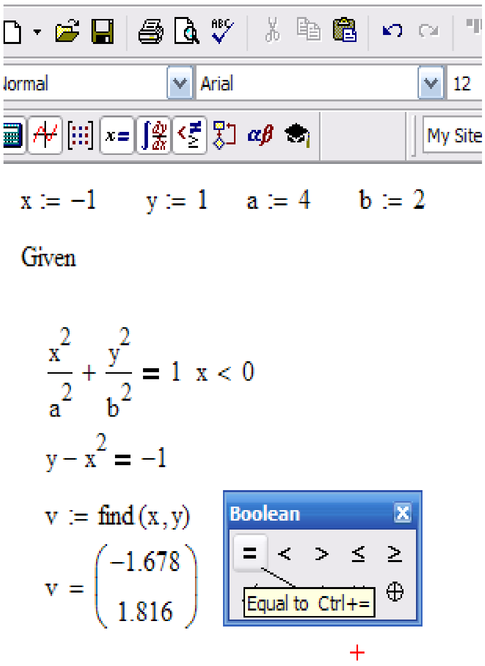


Рис. 7*a*

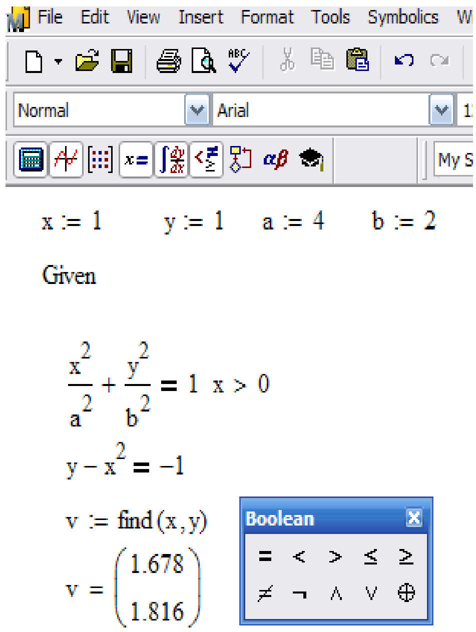


Рис. 7 *b*

Пример 7. Найти решение системы уравнений $\begin{cases} 4x + 5\cos(x-y) = 0 \\ 3y - 2\sin(x+y) = 0 \end{cases}$

Решение: Зададим функции *f(x,y)* и *g(x,y)*, соответствующие первому и второму уравнениям системы соответственно:

$$\begin{cases} f(x,y) := 4x + 5\cos(x-y) \\ g(x,y) := 3y - 2\sin(x+y) \end{cases}$$

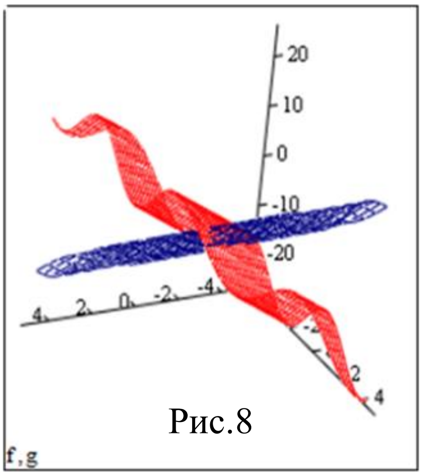


Рис.8

x0:=0   y0:=0

Given

f(x0,y0)=0   g(x0,y0)=0

z:=Find(x0,y0)

z=(-1.118; -0.653)

x0:=0   y0:=0

Рис.9

Построим графики поверхностей, описываемых этими уравнениями. На графике (рисунок 8) видно, что в качестве начального приближения можно выбрать, например, точку (0,0). Далее составляем вычислительный блок (рисунок 9) и решим систему:

7.3. Задания к лабораторной работе

Задание 7.3.1. Отвечать на контрольные вопросы

1. Что такое матрицы и векторы? Объяснить смысл основных операций над матрицами.

2. Показать, как выполняются операции над матрицами и векторами в программе MATHCAD.

3. Что такое система линейных алгебраических уравнений (СЛАУ) и какие основные группы методов их решения существуют?

4. Кратко охарактеризовать прямые и итерационные методы решения СЛАУ.

5. На каком принципе основаны итерационные методы решения систем уравнений?

6. Показать, как решается СЛАУ в MATHCAD прямыми и итерационными методами.

7. Что такое «начальное приближение» решения СЛАУ и как его выбирать? Сколько решений СЛАУ может существовать?

8. Что такое система нелинейных уравнений и какие этапы решения ее можно выделить?

9. Какова цель этапа локализации (отделения) различных решений системы нелинейных уравнений и зачем этот этап необходимо выполнять? Сколько решений систем нелинейных уравнений может существовать?

10. Объяснить действия, выполняемые программой MATHCAD.

11. В каких случаях MathCAD не может найти решение системы уравнений?

12.Дайте сравнительную характеристику функциям Find и Minerr.

Задание 7.3.2

1. Решите всеми указанными способами системы линейных уравнений Вашего варианта (таблица 1), и проведите проверку полученных решений.

2. Сравните способы решений и сделайте вывод о том, какими методами можно решать системы линейных уравнений.

Таблица 1

№	Система уравнений	№	Система уравнений
1	$\begin{cases} 3{,}14x_1 - 2{,}12x_2 + 1{,}17x_3 = 1{,}27 \\ 2{,}12x_1 + 1{,}32x_2 - 2{,}45x_3 = 2{,}13 \\ 1{,}17x_1 - 2{,}45x_2 + 1{,}18x_3 = 3{,}14 \end{cases}$	2	$\begin{cases} 1{,}17x_1 - 0{,}65x_2 + 1{,}54x_3 = -1{,}43 \\ -0{,}65x_1 + 1{,}16x_2 - 1{,}73x_3 = 0{,}68 \\ 1{,}54x_1 - 1{,}73x_2 + 2{,}15x_3 = 1{,}87 \end{cases}$
2	$\begin{cases} 1{,}65x_1 - 2{,}27x_2 + 0{,}18x_3 = 2{,}25 \\ -2{,}27x_1 + 1{,}73x_2 - 0{,}46x_3 = 0{,}93 \\ 0{,}18x_1 - 0{,}46x_2 + 2{,}16x_3 = 1{,}33 \end{cases}$	4	$\begin{cases} 1{,}17x_1 + 2{,}23x_2 - 0{,}77x_3 = 1{,}11 \\ 2{,}23x_1 - 0{,}81x_2 + 1{,}72x_3 = 1{,}88 \\ -0{,}77x_1 + 1{,}72x_2 - 0{,}65x_3 = 0{,}57 \end{cases}$
3	$\begin{cases} 0{,}93x_1 + 1{,}42x_2 - 2{,}55x_3 = 2{,}48 \\ 1{,}42x_1 - 2{,}87x_2 + 2{,}36x_3 = -0{,}75 \\ -2{,}55x_1 + 2{,}36x_2 - 1{,}44x_3 = 1{,}83 \end{cases}$	6	$\begin{cases} 1{,}42x_1 - 2{,}15x_2 + 1{,}07x_3 = 2{,}48 \\ -2{,}15x_1 + 0{,}76x_2 - 2{,}18x_3 = 1{,}15 \\ 1{,}07x_1 - 2{,}18x_2 + 1{,}23x_3 = 0{,}88 \end{cases}$
4	$\begin{cases} 2{,}23x_1 - 0{,}71x_2 + 0{,}63x_3 = 1{,}28 \\ -0{,}71x_1 + 1{,}45x_2 - 1{,}34x_3 = 0{,}64 \\ 0{,}63x_1 - 1{,}34x_2 + 0{,}77x_3 = -0{,}87 \end{cases}$	8	$\begin{cases} 2{,}16x_1 + 1{,}45x_2 - 0{,}89x_3 = 0{,}61 \\ 1{,}45x_1 - 2{,}44x_2 + 1{,}18x_3 = 1{,}05 \\ -0{,}89x_1 + 1{,}18x_2 - 2{,}07x_3 = -0{,}83 \end{cases}$
5	$\begin{cases} 0{,}78x_1 + 1{,}08x_2 - 1{,}35x_3 = 0{,}57 \\ 1{,}08x_1 - 1{,}28x_2 + 0{,}37x_3 = 1{,}27 \\ -1{,}35x_1 + 0{,}37x_2 + 2{,}86x_3 = 0{,}47 \end{cases}$	10	$\begin{cases} 0{,}83x_1 + 2{,}18x_2 - 1{,}73x_3 = 0{,}28 \\ 2{,}18x_1 - 1{,}41x_2 + 1{,}03x_3 = -1{,}18 \\ -1{,}73x_1 + 1{,}03x_2 + 2{,}27x_3 = 0{,}72 \end{cases}$
6	$\begin{cases} 2{,}74x_1 - 1{,}18x_2 + 1{,}23x_3 = 0{,}16 \\ -1{,}18x_1 + 1{,}71x_2 - 0{,}52x_3 = 1{,}81 \\ 1{,}23x_1 - 0{,}52x_2 + 0{,}62x_3 = -1{,}25 \end{cases}$	12	$\begin{cases} 1{,}35x_1 - 0{,}72x_2 + 1{,}38x_3 = 0{,}88 \\ -0{,}72x_1 + 1{,}45x_2 - 2{,}18x_3 = 1{,}72 \\ 1{,}38x_1 - 2{,}18x_2 + 0{,}93x_3 = -0{,}72 \end{cases}$
7	$\begin{cases} 1{,}48x_1 + 0{,}75x_2 - 1{,}23x_3 = 0{,}83 \\ 0{,}75x_1 - 0{,}96x_2 + 1{,}64x_3 = -1{,}12 \\ -1{,}23x_1 + 1{,}64x_2 - 0{,}55x_3 = 0{,}47 \end{cases}$	14	$\begin{cases} 2{,}16x_1 - 3{,}18x_2 + 1{,}26x_3 = 1{,}83 \\ -3{,}18x_1 + 0{,}63x_2 - 2{,}73x_3 = 0{,}54 \\ 1{,}26x_1 - 2{,}73x_2 + 3{,}15x_3 = 1{,}72 \end{cases}$
8	$\begin{cases} 0{,}63x_1 - 1{,}72x_2 + 3{,}37x_3 = -0{,}75 \\ -1{,}72x_1 - 2{,}27x_2 + 1{,}62x_3 = 1{,}27 \\ 3{,}27x_1 + 1{,}62x_2 - 0{,}43x_3 = 2{,}74 \end{cases}$	16	$\begin{cases} 0{,}64x_1 + 1{,}05x_2 - 2{,}93x_3 = 1{,}18 \\ 1{,}05x_1 - 1{,}41x_2 + 0{,}16x_3 = -0{,}27 \\ -2{,}93x_1 + 0{,}16x_2 - 1{,}51x_3 = 0{,}72 \end{cases}$
9	$\begin{cases} 2{,}32x_1 + 1{,}17x_2 - 0{,}28x_3 = 1{,}43 \\ 1{,}17x_1 - 1{,}43x_2 + 0{,}88x_3 = -0{,}47 \\ -0{,}28x_1 + 0{,}88x_2 - 1{,}45x_3 = 1{,}09 \end{cases}$	18	$\begin{cases} 0{,}75x_1 - 1{,}24x_2 + 1{,}56x_3 = 0{,}49 \\ -1{,}24x_1 + 0{,}18x_2 - 1{,}72x_3 = -0{,}57 \\ 1{,}56x_1 - 1{,}72x_2 + 0{,}79x_3 = 1{,}03 \end{cases}$

Задание 7.3.3.

1. Решите всеми указанными способами системы нелинейных уравнений Вашего варианта (таблица 2), и проведите проверку полученных решений.

2. Сравните способы решений и сделайте вывод о том, какими методами можно решать системы нелинейных уравнений.

Таблица 2

№	Система уравнений	№	Система уравнений
1	$\begin{cases}\sin(x+1)-y=1{,}2\\ 2x+\cos y=2\end{cases}$	2	$\begin{cases}\cos(y+0{,}5)+x=0{,}8\\ \sin x-2y=1{,}6\end{cases}$
3	$\begin{cases}\cos(x-1)+y=0{,}5\\ x-\cos y=3\end{cases}$	4	$\begin{cases}\sin(y-1)+x=1{,}3\\ y-\sin(x+1)=0{,}8\end{cases}$
5	$\begin{cases}\sin x+2y=2\\ \cos(y-1)+x=0{,}7\end{cases}$	6	$\begin{cases}2x-\cos(y+1)=0\\ y+\sin x=-0{,}4\end{cases}$
7	$\begin{cases}\cos x+y=1{,}5\\ 2x-\sin(y-0{,}5)=1\end{cases}$	8	$\begin{cases}\cos(y+0{,}5)-x=2\\ \sin x-2y=1\end{cases}$
9	$\begin{cases}\sin(x+0{,}5)-y=1\\ \cos(y-2)+x=0\end{cases}$	10	$\begin{cases}\sin(y+2)-x=1{,}5\\ y+\cos(x-2)=0{,}5\end{cases}$
11	$\begin{cases}\cos(x+0{,}5)+y=0{,}8\\ \sin y-2x=1{,}6\end{cases}$	12	$\begin{cases}\sin(x+1)-y=1\\ 2x+\cos y=2\end{cases}$
13	$\begin{cases}\sin(x-1)=1{,}3-y\\ x-\sin(y+1)=0{,}8\end{cases}$	14	$\begin{cases}\cos(x-1)+y=0{,}8\\ x-\cos y=2\end{cases}$
15	$\begin{cases}\sin y+2x=2\\ \cos(x-1)+y=0{,}7\end{cases}$	16	$\begin{cases}\sin(y+1)-x=1\\ 2y+\cos x=2\end{cases}$
17	$\begin{cases}\cos(x+0{,}5)-y=2\\ \sin y-2x=1\end{cases}$	18	$\begin{cases}\cos x+y=1{,}2\\ 2x-\sin(y-0{,}5)=2\end{cases}$

Литература

1. Трубников, С.В. Вычислительная математика: учеб. пособие / С.В.-Трубников, Б.В. Порошин. – Брянск: БГТУ, 2005. – 396 с.
2. Охорзин, В.А. Прикладная математика в системе MATHCAD: учебное пособие. – СПб.: Лань, 2009. – 352 с.
3. Кирьянов, Д.В. MATHCAD 14/ Д.В.Кирьянов. СПб.: ИРМ, 2007. – 704 с.
4. Шушкевич, Г.Ч. Введение в MATHCAD: учебное пособие/ Г.Ч. Шушкевич, С.В.Шушкевич. М., 2012. – 138 с.
5. Воскобойников, Ю.Е.Основы вычислений и программирования в пакете MATHCAD/ Ю.Е.Воскобойников, А.Ф.Задорожный, Л.А.Литвинов, Ю.Г.Черный. – Новосибирск, Новосибирский архитектурно-строительный университет, 2012 – 212 с.

6. Программирование в среде MathCAD: учеб.-метод. Пособие для бакалавров инженерных и физических специальностей / сост. В. К. Толстых. – Донецк: ДонНУ, 2010. – 128 с.

7. Ханова А.А., Макарова И.Г. Лабораторный практикум по математическому моделированию и методам в расчетах на ЭВМ. - Астрахань: Изд-во АГТУ, 1998. - 93 с.

8. Ханова А.А. Численное решение уравнений и систем. - Астрахань: Изд-во АГТУ, 2001. - 44 с.

8. Лабораторная работа №8.
ПРИБЛИЖЕНИЕ ФУНКЦИЙ

8.1 Основы теории и ее реализация в среде MathCad

8.1.1 Задача интерполяции

Основная задача *интерполяции* — нахождение значения таблично заданной функции в тех точках внутри данного интервала, где она не задана. Исходные табличные данные могут быть получены как экспериментально, так и расчетным путем по сложным зависимостям.

Решение задач интерполяции и экстраполяции обеспечивается построением интерполяционной функции $L(x)$, приближенно заменяющей исходную $f(x)$, заданную таблично, и проходящей через все заданные точки — *узлы интерполяции* (рис.1). С помощью этой функции можно рассчитать искомое значение исходной функции в любой точке.

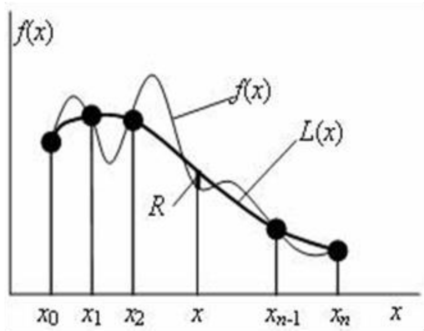


Рис. 1. Иллюстрация

В связи с интерполяцией рассматриваются три основные проблемы.

1) выбор интерполяционной функции $L(x)$;

2) оценка погрешности интерполяции $R(x)$;

3) размещение узлов интерполяции для обеспечения наивысшей возмож-ной точности восстановления функции (x_1, x_2,…,x_n).

Специальные методы интерполяции позволяют определить искомое значение функции без непосредственного прямого построения интерполяционной функции. В принципе все интерполяционные методы, базирующиеся на использовании в качестве интерполяционной функции полиномов, дают одни и те же результаты, но с разными затратами. Это объясняется тем, что полином n-й степени, содержащий $n+1$ параметр и проходящий через все заданные $n+1$ точки, — единственный. Кроме того, полином можно представить как усеченный ряд Тейлора, в который разложили исходную дифференцируемую функцию. Это, пожалуй, одно из главных достоинств полинома как интерполяционной функции. Поэтому чаще первая проблема интерполяции решается выбором в качестве интерполяционной функции именно полинома, хотя могут применяться и другие функции.

Выбор вида интерполяционной функции является в общем случае важной задачей, особенно если помнить, что через заданные точки можно провести любое количество функций. Следует отметить, что существует очевидный способ построения интерполяционной функции: из условия прохождения функции через все точки составляется система уравнений, из решения которой и находятся ее параметры. Однако этот путь далеко не самый эффективный, особенно при большом числе точек.

Принято различать локальную и глобальную интерполяцию. В том случае, когда полином един для всей области интерполяции, говорят, что интерполяция ***глобальная.*** В тех случаях, когда между различными узлами полиномы различны, говорят о ***кусочной*** или ***локальной интерполяции.***

8.1.2 Линейная интерполяция

Простейшим и часто используемым видом локальной интерполяции является ***линейная интерполяция.*** Она состоит в том, что заданные точки $M(x_i, y_i)$ $(i = 0, 1, \ldots, n)$ соединяются прямолинейными отрезками, и функция $f(x)$ приближается к ломаной с вершинами в данных точках (рис. 2).

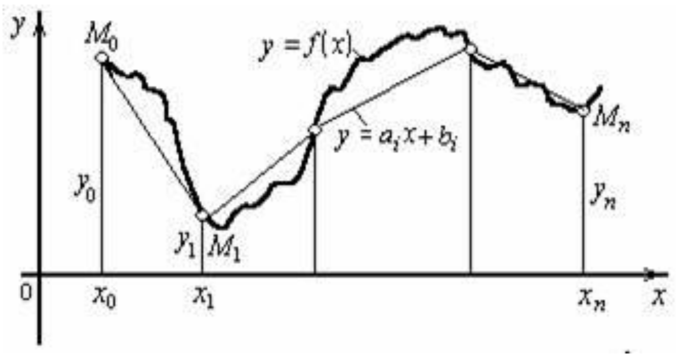


Рис. 2 Линейная интерполяция

Уравнения каждого отрезка ломаной линии в общем случае разные. Поскольку имеется n интервалов (x_i, x_{i+1}), то для каждого из них в качестве уравнения интерполяционного полинома используется уравнение прямой, проходящей через две точки. В частности, для i - го интервала можно написать уравнение прямой, проходящей через точки (x_i, y_i) и (x_{i+1}, y_{i+1}), в виде:

$$\frac{y - y_i}{y_{i+1} - y_i} = \frac{x - x_i}{x_{i+1} - x_i}.$$

Отсюда $y = a_i x + b_i, \;\; x_i \leq x \leq x_{i+1}$

$$a_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, \; b_i = \; y_i - a_i x_i \qquad (1)$$

Следовательно, при использовании линейной интерполяции сначала нужно определить интервал, в который попадает значение аргумента *x*, а затем подставить его в формулу (1) и найти приближенное значение функций в этой точке.

Mathcad предоставляет набор средств для линейной, сплайновой интерполяции и интерполяции по Лагранжу. При дальнейшем изложении использованы обозначения:

vx, vy – векторы – столбцы переменных ***x*** и ***y***;

vs – вектор – столбец вторых производных в используемых узлах.

Функции:

linterp* (*vx, vy, x*)** – возвращает значение функции по значению величины ***x****** при её линейной интерполяции.

interp (***vs, vx, vy, x****) – возвращает значение функции по значению x^*при её сплайн-интерполяции, где вектор ***vs*** вычисляется по функции:

cspline (vx, vy) при приближении к кубическому сплайну;

pspline (vx, vy) при параболической (квадратичному сплайну);

lspline (vx, vy) при приближении к прямой.

При линейной интерполяции Mathcad соединяет существующие точки данных прямыми линиями. Это выполняется функцией ***linterp***.

linterp(vx,vy,x) использует векторы данных ***vx*** *и* ***vy,*** чтобы возвратить линейно интерполируемое значение ***y***, соответствующее третьему аргументу ***x***. Аргументы ***vx*** *и* ***vy*** должны быть векторами одинаковой длины. Вектор ***vx*** должен содержать вещественные значения, расположенные в порядке возрастания.

Эта функция соединяет точки данных отрезками прямых, создавая таким образом ломаную. Интерполируемое значение для конкретного *x* есть ордината ***y*** соответствующей точки ломаной.

Для значений ***x***, расположенных перед первой точкой в **vx**, Mathcad продолжает ломаную прямой линией, проходящей через первые две точки данных. Для значений *x*, расположенных за последней точкой в **vx**, Mathcad продолжает ломаную прямой линией, проходящей через последние две точки данных.

Пример 1. Решим задачу использования линейной интерполяции в программе MathCAD (Рис.3). Для линейной интерполяции используется функция ***linterp***(*x*,*y*,*z*). Здесь *x*, *y* – исходные данные, *z* – точка, в которой находится значение функции.

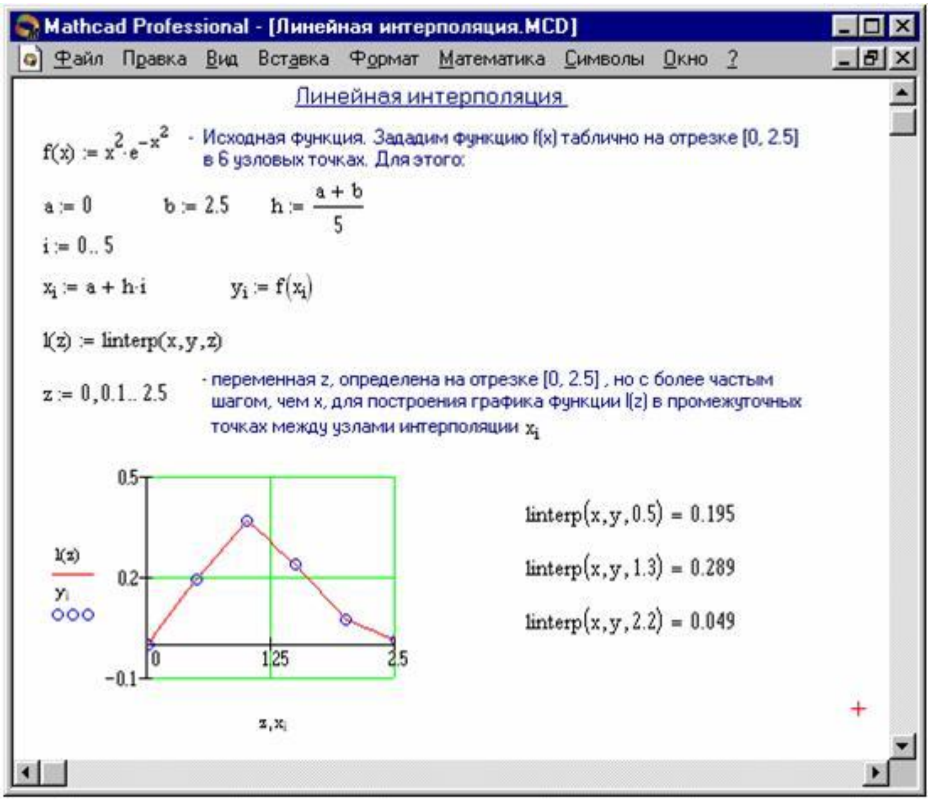


Рис. 3. Линейная интерполяция

Пример 2. Решим задачу линейного интерполирования, когда функциональная зависимость дана в виде таблицы (рисунок 4).

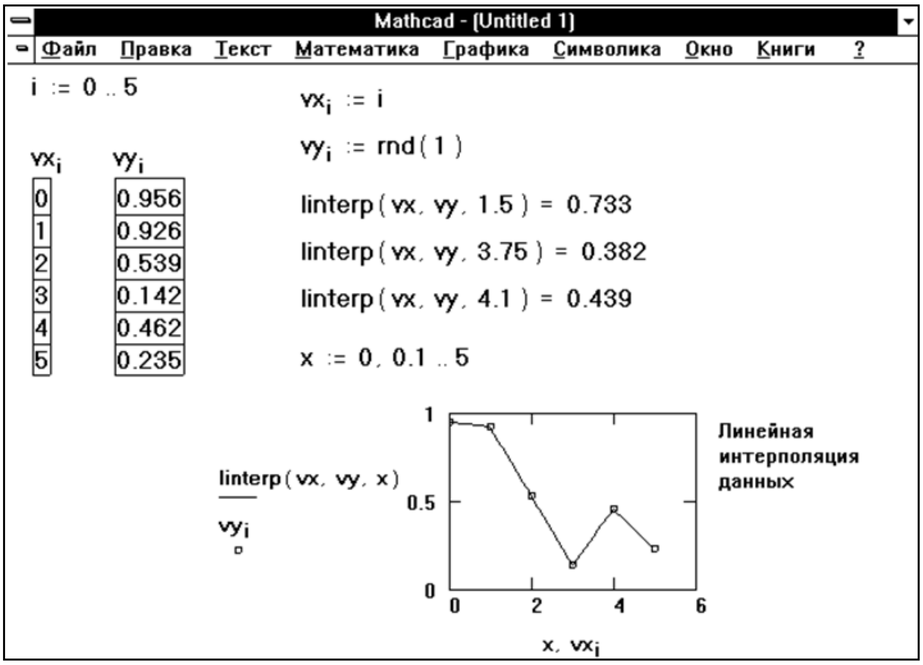


Рис.4. Пример линейной интерполяции

8.1.3 Квадратичная интерполяция

В случае *квадратичной интерполяции* в качестве интерполяционной функции на отрезке (x_{i-1}, x_{i+1}) принимается квадратный трехчлен. Уравнения квадратного трехчлена имеет вид

$$y = a_i x^2 + b_i x + c_i,\ x_{i-1} \le x \le x_{i+1}, \qquad (2)$$

и содержат три неизвестных коэффициента a_i, b_i, c_i, для определения которых необходимы три уравнения. Ими служат условия прохождения параболы (2) через три точки. Эти условия можно записать в виде:

(x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}).

$$a_i x_{i-1}^2 + b_i x_{i-1} + c_i = y_{i-1}$$

$$a_i x_i^2 + b_i x_i + c_i = y_i$$

$$a_i x_{i+1}^2 + b_i x_{i+1} + c_i = y_{i+1}$$

Интерполяция для любой точки $x \subset [x_0,\ x_n]$ проводится по трем ближайшим точкам.

8.1.4 Кубическая сплайн-интерполяция

В последние годы интенсивно развивается новый раздел современной вычислительной математики – теория ***сплайнов***. Сплайны позволяют эффективно решать задачи обработки экспериментальных зависимостей между параметрами, имеющих достаточно сложную структуру.

Рассмотренные выше методы локальной интерполяции, по существу, является простейшим сплайном первой степени (для линейной интерполяции) и второй степени (для квадратичной интерполяции).

Наиболее широкое практическое применение, в силу их простоты, нашли кубические сплайны. Основные идеи теории кубических сплайнов сформировались в результате попыток математически описать гибкие рейки из упругого материала. Известно, что рейка из упругого материала, закрепленная в некоторых точках и находящаяся в положении равновесия, принимает форму, при которой ее энергия является минимальной. Это фундаментальное свойство позволяет эффективно использовать сплайны при решении практических задач обработки экспериментальной информации.

В общем случае для функции $y = f(x)$ требуется найти приближение $y = j(x)$ таким образом, чтобы $f(x_i) = j(x_i)$ в точках $x = x_i$, а в остальных точках отрезка $[a, b]$ значения

функций $f(x)$ и $j(x)$ были близкими между собой. При малом числе экспериментальных точек (например, 6-8) для решения задачи интерполяции можно использовать один из методов построения интерполяционных полиномов. Однако при большом числе узлов интерполяционные полиномы становятся практически непригодными. Это связано с тем, что степень интерполяционного полинома лишь на единицу меньше числа экспериментальных значений функций. Можно, конечно, отрезок, на котором определена функция, разбить на участки, содержащие малое число экспериментальных точек, и для каждого из них построить интерполяционные полиномы. Однако в этом случае аппроксимирующая функция будет иметь точки, где производная не является непрерывной, т. е. график функции будет содержать точки “излома”.

Кубические сплайны лишены этого недостатка. Исследования теории балок показали, что гибкая тонкая балка между двумя узлами достаточно хорошо описывается кубическим полиномом, и поскольку она не разрушается, то аппроксимирующая функция должна быть, по меньшей мере, непрерывно дифференцируемой. Это означает, что функции $f(x)$, $f'(x)$, $f''(x)$ должны быть непрерывными на отрезке $[a, b]$.

Кубическим интерполяционным сплайном, соответствующим данной функции $f(x)$ и данным узлам x_i, называется функция $y(x)$, удовлетворяющая следующим условиям:

1. на каждом сегменте $[x_{i-1}, x_i]$, $i = 1, 2, ..., n$ функция $y(x)$ является полиномом третьей степени;

*2.*функция $y(x)$, а также ее первая и вторая производные непрерывны на отрезке $[a,b]$.

Кубический сплайн склеивается из полиномов третьей степени, которые для i-го участка записываются так:

$$y = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$$

Для всего интервала будет соответственно n кубических полиномов, отличающихся коэффициентами a_i, b_i, c_i, d_i. Чаще всего узлы при сплайновой интерполяции располагают равномерно, т.е. $x_{i+1} - x_i = const = h$ (хотя это и необязательно).

Необходимо найти четыре коэффициента при условии прохождения каждого полинома через две точки $(x_i,\ y_i)$ и $(x_{i+1},\ y_{i+1})$, следствием чего являются следующие очевидные уравнения:

$$y_i = d_i, \qquad i = 0, 1, 2, ..., n-1$$

$$y_{i+1} = a_i h^3 + b_i h^2 + c_i h + d_i, \qquad i = 0, 1, 2, ..., n-1$$

Первое условие соответствует прохождению полинома через начальную точку, второе — через конечную точку. Найти все коэффициенты из этих уравнений нельзя, так как условий меньше, чем искомых параметров. Поэтому указанные условия дополняют условиями гладкости функции (т.е. непрерывности первой производной) и гладкости первой производной (т.е. непрерывности второй производной) в узлах интерполяции. Математически эти условия записываются как равенства соответственно первой и второй производных в конце i-го и в начале $(i+1)$-го участков. Так как

$y' = 3a_i\left(x - x_i\right)^2 + 2b_i\left(x - x_i\right) + c_i$ и $y'' = 6a_i\left(x - x_i\right) + 2b_i$, то

$$3a_ih^2 + 2b_ih + c_i = c_{i+1}, \quad i = 0,1,2,\ldots,n-2$$

(***y'***(x_{i+1}) в конце i-го участка равна ***y'*** (x_{i+1}) в начале **(*i*+*1*)-го**),

$$6a_ih + 2b_i = 2b_{i+1}$$

(***y''***(x_{i+1}) в конце i-го участка равна ***y''*** (x_{i+1}) в начале (***i***+**1**)-го).

Получилась система линейных уравнений (для всех участков), содержащая $4n$ уравнения с $4n$ неизвестными (неизвестные $a1, a2, \ldots, an, b1, \ldots, dn$ — коэффициенты сплайнов):

$$1)\ \ y''(x_0) = y''(x_n) = 0, \qquad 3)\ \ y'(x_0) = y'(x_n) = 0,$$

$$2)\ \ y'(x_0) = g_0,\ y'(x_n) = g_n, \qquad 4)\ \ y'(x_0) = y'(x_n),\ y''(x_0) = y''(x_n)$$

Совместное решение $4n$ уравнений позволяет найти все $4n$ коэффициента.

Для восстановления производных можно продифференцировать на каждом участке соответствующий кубический полином. В случае необходимости определения производных в узлах существуют специальные приемы, сводящие определение производных к решению более простой системы уравнений относительно искомых производных второго или первого порядка. К важным

достоинствам интерполяции кубическими сплайнами относится получение функции, имеющей минимальную возможную кривизну. К недостаткам сплайновой интерполяции относится необходимость получения сравнительно большого числа параметров.

Пример 3. Решим задачу кубической сплайн-интерполяции с помощью программы MathCAD.

Для этого воспользуемся встроенной функцией ***interp(VS,x,y,z)***.

Переменные ***x*** и ***y*** задают координаты узловых точек, ***z*** является аргумен-том функции, ***VS*** определяет тип граничных условий на концах интервала.

Определим интерполяционные функции для трех типов кубического сплайна:

$$\mathrm{VSl} := \mathrm{lspline}(x, y), \qquad \mathrm{gl}(z) := \mathrm{interp}(\mathrm{VSl}, x, y, z)$$

$$\mathrm{VSp} := \mathrm{pspline}(x, y), \qquad \mathrm{gp}(z) := \mathrm{interp}(\mathrm{VSp}, x, y, z)$$

$$\mathrm{VSc} := \mathrm{cspline}(x, y), \qquad \mathrm{gc}(z) := \mathrm{interp}(\mathrm{VSc}, x, y, z)$$

Здесь: **cspline(VX,VY)** возвращает вектор VS вторых производных в опорных точках к кубическому полиному;

pspline(VX, VY) возвращает вектор **VS** вторых производных при приближении к опорным точкам к параболической кривой;

lspline(VX, VY) возвращает вектор **VS** вторых производных при приближении к опорным точкам прямой;

interp(VS, VX, VY, *x*) возвращает значение *y(x)* для заданных векторов **VS, VX, VY** и заданного значения *x*.

Вычисляем значения интерполяционных функций в заданных точках и сравниваем результаты с точными значениями:

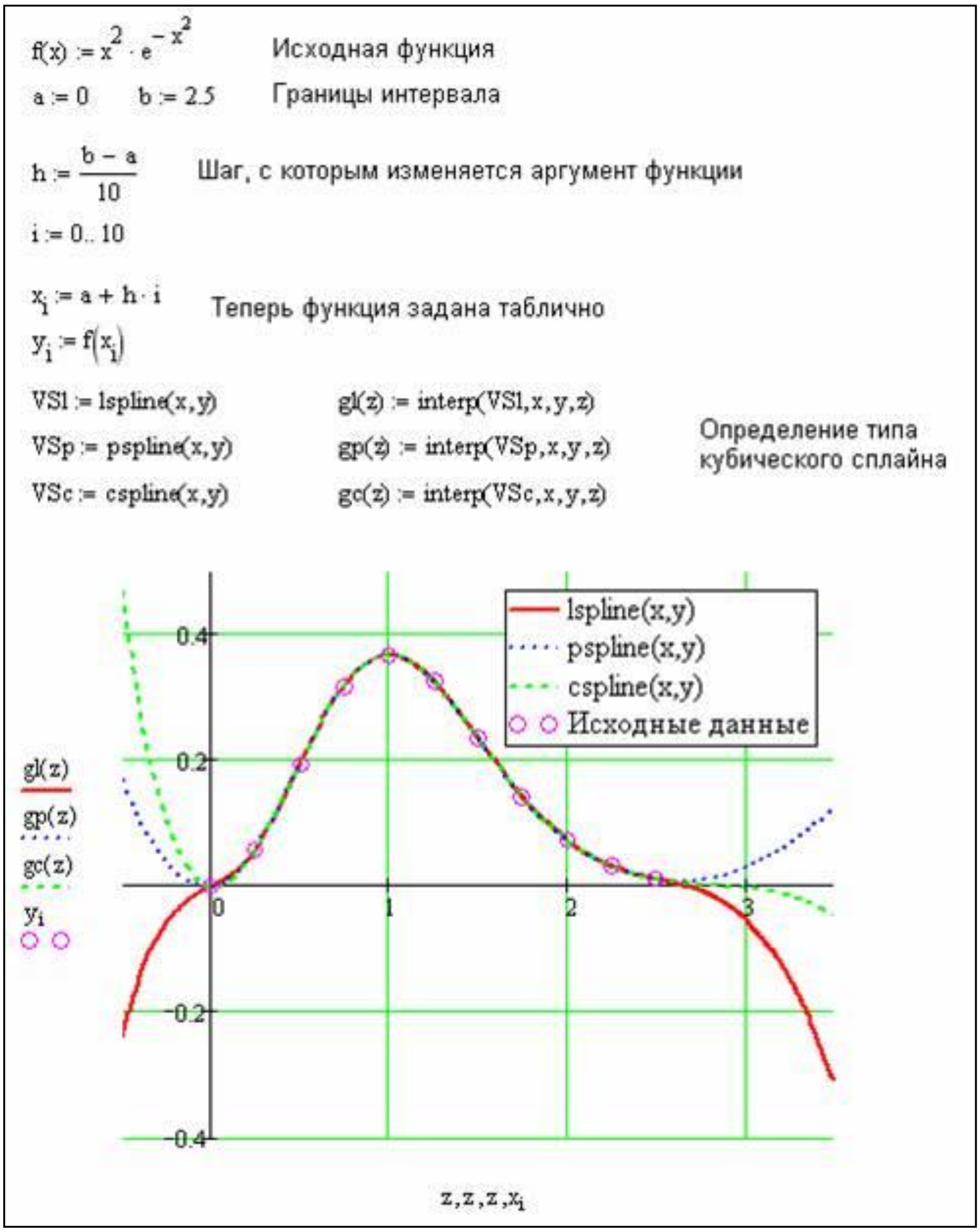


Рис. 5

Обратите внимание, что результаты интерполяции различными типами кубических сплайнов практически не отличаются во внутренних точках интервала и совпадают с точными значениями функции. Вблизи краев интервала отличие становится более заметным, а при экстраполяции за пределы заданного интервала различные типы сплайнов дают существенно разные результаты.

Если функция задана дискретно, то для интерполяции задаются матрицы данных. При глобальной интерполяции наиболее часто используется интерполяция полиномом *n*-ой степени или интерполяция Лагранжа.

8.1.5 Интерполяция по Лагранжу

Суть интерполяции по Лагранжу состоит в следующем.

Пусть имеется таблица данных (Таблица 1) , в которой x – независимая пере-менная, y – зависимая. В дальнейшем точку с координатами (x_i, y_i), принад-лежащую таблице, будем называть i-ым узлом.

Таблица1

x	x_0	x_1	x_2	…	x_i	x_{i+1}	…	x_n
y	y_0	y_1	y_2	…	y_i	y_{i+1}	…	y_n

В данном случае зависимость между x и y точечно. Требуется ***уплотнить таблицу, т.е.*** дополнить её новыми значениями $y_к = f(x_к{*})$ где $x_0 \le x_к{*} \le x_n$., т.е. в точках между узлами. Решить эту задачу можно на основе ***интерполяции***.

Установим зависимость $y(x)$ одного ряда чисел таблицы 1 от другого и построим новую функцию, которая с определенной степенью точности будет приближена к заданной.

Построим многочлен $P_n(x)$ таким образом, чтобы его значения совпали со значениями функции, заданными в таблице, для тех же аргументов, то есть

$$P_n(x_i) = y_i, i = \overline{0,n}. \tag{3}$$

Лагранж предложил строить многочлен n -й степени в виде:

$$\begin{aligned} P_n(x) = a_0(x - x_1)\ldots(x - x_n) + a_1(x - x_0)(x - x_2)\ldots(x - x_n) + \ldots + \\ + a_n(x - x_0)\ldots(x - x_{n-1}). \end{aligned} \tag{4}$$

Здесь в каждом слагаемом отсутствует скобка $(x - x_i)$, которой соответствует коэффициент a_i.

Найдем неизвестные коэффициенты $a_i, i = \overline{0,n}$, называемые коэффициентами Лагранжа, используя условие (1).

При $x = x_0$: $P_n(x_0) = y_0$.

$P_n(x_0) = a_0(x_0 - x_1)(x_0 - x_2)\ldots(x_0 - x_n) = y_0$.

Следовательно, коэффициент a_0 вычисляется по следующей формуле:

$$a_0 = \frac{y_0}{(x_0 - x_1)(x_0 - x_2)...(x_0 - x_n)}.$$

При $x = x_1$: $P_n(x_1) = y_1$.

$P_n(x_1) = a_1(x_1 - x_0)(x_1 - x_2)\ldots(x_1 - x_n) = y_1$.

Следовательно, коэффициент a_1 вычисляется по следующей формуле:

$$a_1 = \frac{y_1}{(x_1 - x_0)(x_1 - x_2)\ldots(x_1 - x_n)}.$$

Таким образом, коэффициенты $a_i, i = \overline{0,n}$ вычисляются по формулам:

$$a_i = \frac{y_i}{(x_i - x_0)...(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)}.$$

С учетом найденных коэффициентов интерполяционный полином Лагранжа запишется в виде

$$\boxed{\begin{aligned} &P_n(x) = y_0 \frac{(x - x_1)(x - x_2)\ldots(x - x_n)}{(x_0 - x_1)\ldots(x_0 - x_n)} + \ldots + \\ &+ y_n \frac{(x - x_0)(x - x_1)\ldots(x - x_{n-1})}{(x_n - x_0)\ldots(x_n - x_{n-1})} = \\ &= \sum_{i=0}^{n} y_i \frac{(x - x_0)...(x - x_{i-1})(x - x_{i+1})...(x - x_n)}{(x_i - x_0)...(x_i - x_{i-1})(x_i - x_{i+1})...(x_i - x_n)}. \end{aligned}} \tag{5}$$

Для интерполяционной формулы Лагранжа справедлива оценка погрешности:

$$|R_n(x)| = |f(x) - P_n(x)| \le \max_{a \le x \le b} \left| f^{(n+1)}(x) \right| \cdot \frac{|\Pi_{n+1}(x)|}{(n+1)!}, \tag{6}$$

где $\Pi_{n+1}(x)=(x-x_0)...(x-x_i)...(x-x_n)$. Если таблица 1, для которой построена формула Лагранжа, задана для равноотстоящих узлов $x_{i+1}=x_i+h, i=\overline{0,n-1}$, то формула Лагранжа упрощается. Действительно:

Обозначим через $q=\dfrac{x-x_0}{h}$. Тогда

$$\frac{x-x_1}{h}=\frac{x-(x_0+h)}{h}=\frac{x-x_0}{h}-1=q-1,$$

$$\frac{x-x_2}{h}=q-2,\ldots,$$

$$\frac{x-x_i}{h}=q-i, i=\overline{0,n}.$$

С учетом введенных обозначений формула Лагранжа запишется так:

$$P_n(q)=y_0\frac{(q-1)(q-2)\ldots(q-n)}{(-1)(-2)\ldots(-n)}+y_1\frac{q(q-2)\ldots(q-n)}{(-1)(-2)\ldots(-(n-1))}+$$

$$+\ldots+y_n\frac{q(q-1)\ldots(q-(n-1))}{n(n-1)\ldots1}.$$

Запишем формулу Лагранжа в случае, если $n=1$:

$$P_1(q)=y_0\frac{x-x_1}{x_0-x_1}+y_1\frac{x-x_0}{x_1-x_0}=-y_0(q-1)+y_1q=$$

$$=y_0-y_0q+y_1q=y_0+q(y_1-y_0)=y_0+q\Delta y_0.$$

Получили формулу линейной интерполяции:

$$\boxed{P_1(x)=y_0+q\Delta y_0} \tag{7}$$

При $n=2$ получаем формулу квадратичной интерполяции:

$$\boxed{P_2(x)=y_0+q\Delta y_0+\frac{q(q-1)}{2}\Delta^2 y_0} \tag{8}$$

Здесь $\Delta^2 y_i=\Delta y_{i+1}-\Delta y_i, i=1,2,...$ - табличные разности второго порядка, и так далее. Продолжая этот процесс, окончательно получим:

$$P_n(q)=y_0+q\Delta y_0+\frac{q(q-1)}{2!}\Delta^2 y_0+\ldots+\frac{q(q-1)\ldots(q-(n-1))}{n!}\Delta^n y_0.$$

Эта формула называется **первой интерполяционной формулой Ньютона**.

Если обозначить через $t=\dfrac{x-x_n}{h}$, то с учетом введенного обозначения, получим: $\dfrac{x-x_{n-1}}{h}=\dfrac{x-(x_n-h)}{h}=\dfrac{x-x_n}{h}+1=t+1,\ \ldots,\dfrac{x-x_{n-i}}{h}=t+i,\, i=\overline{0,n}$.
Тогда формула (3) примет вид:

$$P_n(t)=y_n+t\Delta y_{n-1}+\frac{t(t+1)}{2!}\Delta^2 y_{n-2}+\ldots+\frac{t(t+1)\ldots(t+n-1)}{n!}\Delta^n y_0. \qquad (9)$$

Эта формула называется **второй интерполяционной формулой Ньютона**.

Пример 4. С помощью калькулятора по заданной в таблице 2 системе точек построить интерполяционный многочлен Лагранжа второго порядка вида:

Таблица 2

x_i	$\frac{\pi}{6}=0.524$	$\frac{\pi}{4}=0.785$	$\frac{\pi}{2}=1.571$
y_i	0.5	0.707	1.0

$$P_2(x)=a_0(x-x_1)(x-x_2)+a_1(x-x_0)(x-x_2)+a_2(x-x_0)(x-x_1).$$

Коэффициенты этого многочлена будут вычислены по следующим формулам:

$$a_0=\frac{y_0}{(x_0-x_1)(x_0-x_2)}=\frac{0.5}{(0.524-0.785)(0.524-1.571)}=1.824,$$

$$a_1=\frac{y_1}{(x_1-x_0)(x_1-x_2)}=\frac{0.707}{(0.785-0.524)(0.785-1.571)}=-3.439,$$

$$a_2=\frac{y_2}{(x_2-x_0)(x_2-x_1)}=\frac{1.0}{(1.571-0.524)(1.571-0.785)}=1.216.$$

Тогда многочлен Лагранжа второго порядка будет иметь вид:

$$P_2(x)=1.824(x-x_1)(x-x_2)-3.439(x-x_0)(x-x_2)+$$
$$+1.216(x-x_0)(x-x_1)=-0.4x^2+1.32x-0.08.$$

Учитывая, что таблица приведена для функции $y=\sin(x)$, вычисленной в узловых точках $\frac{\pi}{6},\frac{\pi}{4},\frac{\pi}{2}$, сравним погрешность вычислений данной функции и построенного многочлена в контрольной точке $\frac{\pi}{3}$: $y\left(\frac{\pi}{3}\right)=\sin\left(\frac{\pi}{3}\right)\cong 0.866$. $P_2\left(\frac{\pi}{3}\right)=0.859$. Погрешность составляет $\left|\sin\left(\frac{\pi}{3}\right)-P_2\left(\frac{\pi}{3}\right)\right|=0.007$.

Ниже приведены графики функции $y = \sin(x)$ и построенного полинома Лагранжа на заданном интервале. Из рисунка 6 видно, что многочлен второго порядка обеспечивает достаточно высокую точность построения синусоиды на заданном отрезке.

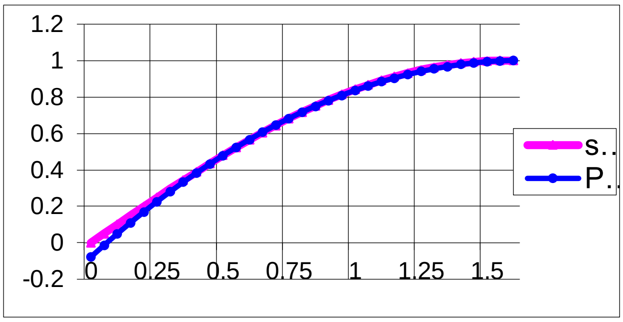


Рис.6.

Пример 5. Приведем схему интерполяции по Лагранжу в случае задания функции в виде таблицы. В этом случае табличные данные можно представить в виде двух векторов (или массивов). Процедуры применения возможностей MathCad приведены на рисунке 7.

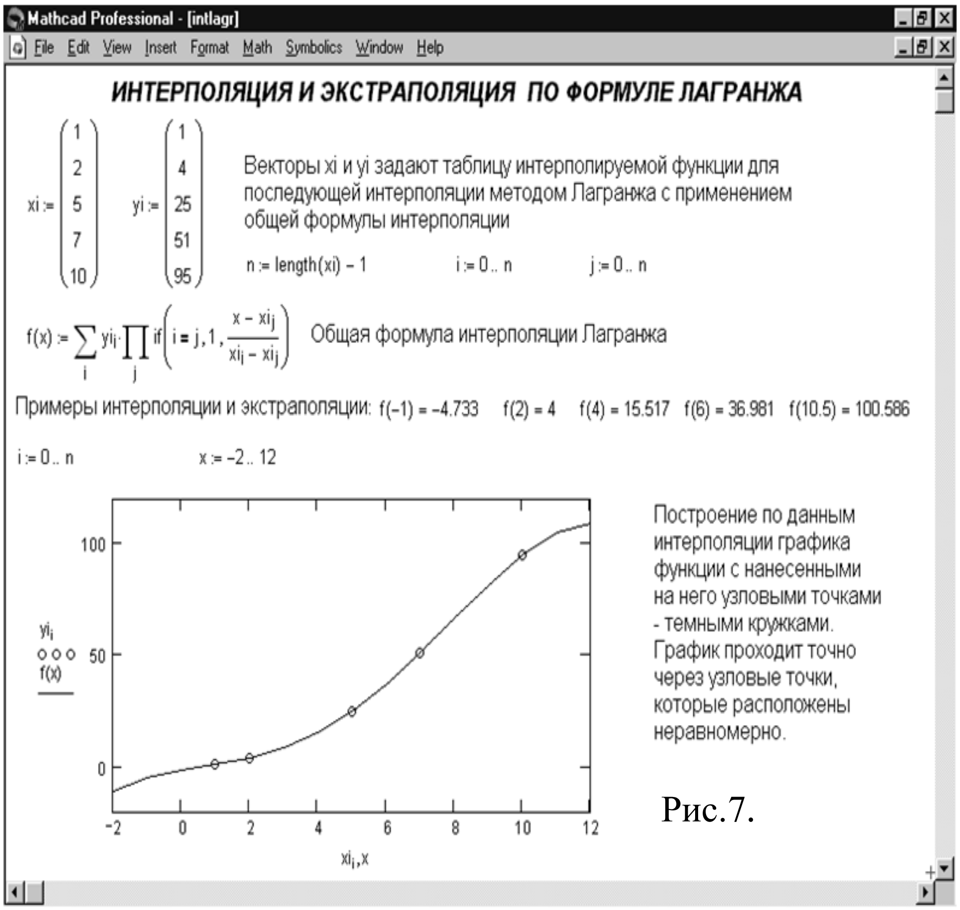


Рис. 7.

На рисунке 8 приведено сравнение методов аппроксимации. Внимательно проанализируйте этот рисунок.

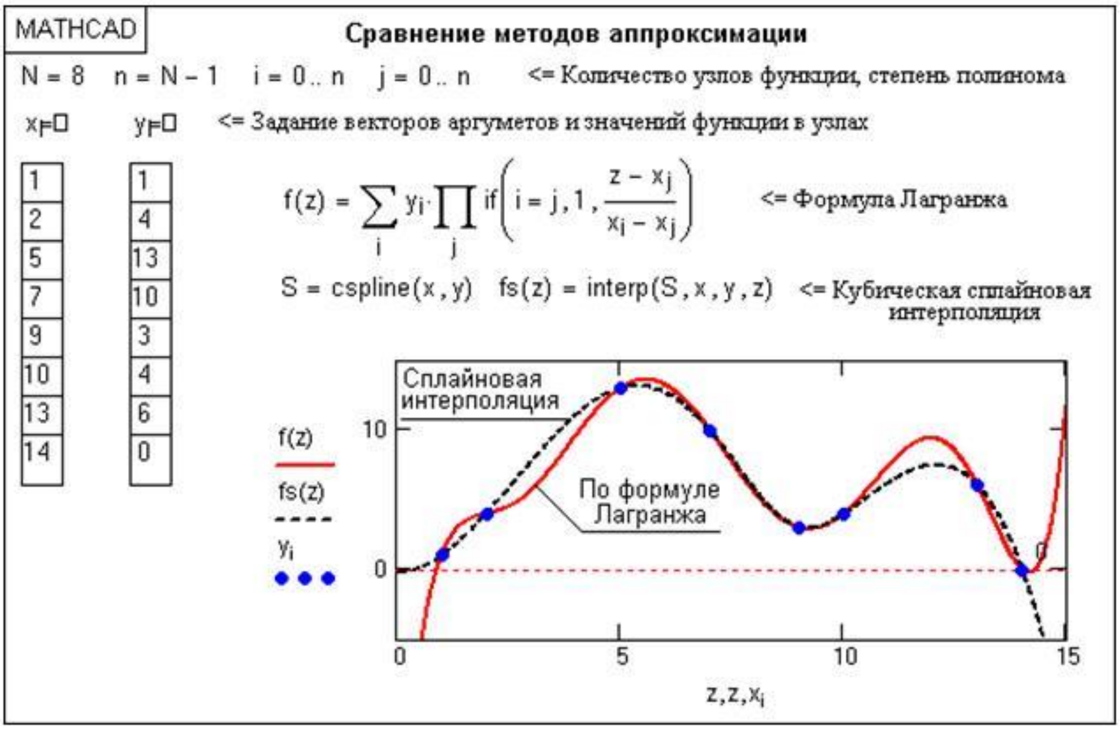


Рис. 8.

8.2. Задания к лабораторной работе

Задание 8.2.1. Отвечать на контрольные вопросы

1. Что такое интерполяция таблично (точечно) заданной функции?

2. Как соответствуют степень интерполирующего полинома (многочлена) и количество точек исходных данных?

3. Что такое аппроксимация таблично (точечно) заданной функции?

4. Что является количественной характеристикой качества аппроксимирующей функции?

5. Какие этапы можно выделить при построении качественной аппроксимирующей функции?

6. Чем различаются графики интерполирующей и аппроксимирующей функции для одних и тех же исходных данных?

7. Объяснить процедуру получения интерполирующих зависимостей с применением программы MATHCAD.

Задание 8.2.2:

1. Уплотнить линейной, сплайновой и Лагранжевой интерполяцией данные Вашего варианта (Таблица 3), вычислив дополнительные значения в

каждом промежутке при среднем и произвольном значениях *x* (выбирают сами).

2. Построить демонстрационные графики.

Таблица 3

№	i	0	1	2	3	4	5	6	7	8	9
1	x	-1,00	-0,80	-0,60	-0,40	-0,20	0,00	0,20	0,40	0,60	0,80
	y	0,81	0,97	1,17	1,40	1,67	2,00	2,39	2,87	3,43	4,11
2	x	-2,00	-1,00	0,00	1,00	2,00	3,00	4,00	5,00	6,00	7,00
	y	12,50	10,00	8,00	6,40	5,12	4,10	3,28	2,62	2,10	1,68
3	x	0,10	0,20	0,30	0,40	0,50	0,60	0,70	0,80	0,90	1,00
	y	-4,91	-2,83	-1,61	-0,75	-0,08	0,47	0,93	1,33	1,68	2,00
4	x	0,10	0,20	0,30	0,40	0,50	0,60	0,70	0,80	0,90	1,00
	y	0,00	0,01	0,04	0,10	0,19	0,32	0,51	0,77	1,09	1,50
5	x	-3,00	-2,70	-2,40	-2,10	-1,80	-1,50	-1,20	-0,90	-0,60	-0,30
	y	47,00	38,45	30,80	24,05	18,20	13,25	9,20	6,05	3,80	2,45
6	x	-3,00	-2,50	-2,00	-1,50	-1,00	-0,50	0,00	0,50	1,00	1,50
	y	143,00	86,13	48,00	24,88	13,00	8,63	8,00	7,38	3,00	-8,88
7	x	0,30	0,55	0,80	1,05	1,30	1,55	1,80	2,05	2,30	2,55
	y	6,00	4,18	3,50	3,14	2,92	2,77	2,67	2,59	2,52	2,47
8	x	-0,50	0,00	0,50	1,00	1,50	2,00	2,50	3,00	3,50	4,00
	y	2,24	1,50	1,01	0,67	0,45	0,30	0,20	0,14	0,09	0,06
9	x	-1,00	-0,50	0,00	0,50	1,00	1,50	2,00	2,50	3,00	3,50
	y	0,72	1,29	2,30	4,11	7,36	13,17	23,55	42,13	75,37	134,82
10	x	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50	0,55	0,60
	y	-3,69	-2,83	-2,16	-1,61	-1,15	-0,75	-0,40	-0,08	0,21	0,47
11	x	0,08	0,20	0,32	0,44	0,56	0,68	0,80	0,92	1,04	1,16
	y	0,00	0,01	0,05	0,13	0,26	0,47	0,77	1,17	1,69	2,34

12	x	-1,00	-0,35	0,30	0,95	1,60	2,25	2,90	3,55	4,20	4,85
	y	-36,29	-34,10	-30,64	-25,91	-19,91	-12,65	-4,11	5,68	16,75	29,08
13	x	-1,00	-0,80	-0,60	-0,40	-0,20	0,00	0,20	0,40	0,60	0,80
	y	13,00	10,56	9,08	8,32	8,04	8,00	7,96	7,68	6,92	5,44
14	x	0,38	0,58	0,78	0,98	1,18	1,38	1,58	1,78	1,98	2,18
	y	-1,16	-0,07	0,46	0,78	0,98	1,13	1,24	1,33	1,39	1,45
15	x	2,00	2,60	3,20	3,80	4,40	5,00	5,60	6,20	6,80	7,40
	y	1,11	1,41	1,80	2,29	2,91	3,69	4,70	5,97	7,59	9,65
16	x	-1,00	-0,50	0,00	0,50	1,00	1,50	2,00	2,50	3,00	3,50
	y	-0,85	-1,40	-2,30	-3,78	-6,21	-10,20	-16,77	-27,55	-45,27	-74,39

Литература

1. Трубников, С.В. Вычислительная математика: учеб. пособие / С.В.Трубников, Б.В. Порошин. – Брянск: БГТУ, 2005. – 396 с.
2. Воскобойников, Ю.Е.Основы вычислений и программирования в пакете MATHCAD/ Ю.Е.Воскобойников, А.Ф.Задорожный, Л.А.Литвинов, Ю.Г.Черный. – Новосибирск, Новосибирский архитектурно-строительный университет, 2012 – 212 с.
3. Программирование в среде MathCAD: учеб.-метод. пособие для бакалавров инженерных и физических специальностей / сост. В. К. Толстых. – Донецк: ДонНУ, 2010. – 128 с.

9. Лабораторная работа № 9. ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

9.1 Основы теории и ее реализация в MathCad

Большое число задач математики, техники, экономики связаны с вычислением определенного интеграла $\int_a^b f(x)dx$. Если для подынтегральной функции *f(x)* найдена первообразная *F(x)*, то интеграл вычисляется по известной формуле Ньютона-Лейбница:

$$\int_a^b f(x)dx = F(b) - F(a). \tag{1}$$

Однако на практике встречаются случаи, когда:

− первообразная функция *F(x)* не выражается через элементарные функции;

−аналитическое выражение подынтегральной функции *f(x)* неизвестно, т.е. оно задано таблицей или графиком.

В таких случаях для вычисления определенного интеграла применяют формулы приближенного интегрирования (формулы прямоугольников, трапеций, парабол и др).

Пусть требуется вычислить определенный интеграл

$$I = \int_a^b f(x)dx, \tag{2}$$

при условии, что a и b конечны, и $f(x)$ является непрерывной функцией во всем интервале $a \le x \le b$.

Определенный интеграл *I* представляет собой площадь, ограниченную кривой $f(x)$, осью X и прямыми $x = a$ и $x = b$. Будем вычислять интеграл (2), разбивая интервал от a до b на множество меньших интервалов, находя приблизительно площадь каждой полоски, получающейся при таком разбиении, и суммируя площади этих полосок.

9.1.1 Метод прямоугольников

Пусть требуется вычислить значение интеграла (2) на отрезке $[a, b]$. Для этого этот отрезок делится точками $x_0, x_1, \ldots, x_{n-1}, x_n$ на n равных отрезков длиной

$$\Delta x = \frac{b-a}{n}.$$

Обозначим через $y_0, y_1, \ldots, y_{n-1}, y_n$ значение функции $f(x)$ в точках

$$x_0, x_1, \ldots, x_{n-1}, x_n.$$

Далее составляем сумму $y_0 \Delta x + y_1 \Delta x + \ldots + y_{n-1} \Delta x + y_n \Delta x.$

Эта сумма приближенно выражает интеграл (2), т.е.:

$$\int_a^b f(x)dx \approx \frac{b-a}{n}(y_0 + y_1 + \cdots + y_{n-1}). \tag{3}$$

Если подынтегральная функция $f(x)$ положительная и возрастающая, то формула (3) выражает площадь ступенчатой фигуры, составленной из «входящих» прямоугольников. Чем меньше длина отрезков, на которые делится отрезок $[a, b]$, тем точнее значение, вычисляемое по (3).

Пример 1. Методами левых, правых, средних прямоугольников вычислить интеграл $\int_0^1 14^x dx.$ Аналитическое вычисление этого интеграла $\int_0^1 14^x dx$=4.926.

а) Метод левых прямоугольников

В этом случае, отрезок инт Рис.1 [a, b] разбивается на n элементарных отрезков равной длин... n . Тогда

$$\int_a^b f(x)dx = h(f(x_0) + f(x_1) + \cdots + f(x_{n-1})) \tag{4}$$

Этот метод в MathCad реализуется следующим образом (рисунок 1):

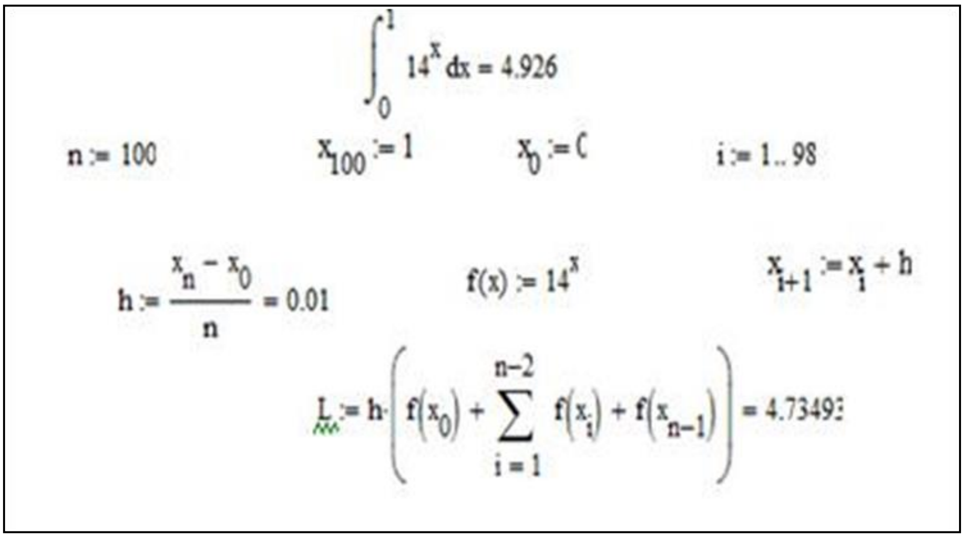


Рис. 1

б) Метод правых прямоугольников

В этом случае отрезок $[a, b]$ разбивается на n элементарных отрезков равной длины $h = \frac{(b-a)}{n}$. Тогда

$$\int_a^b f(x)dx = h(f(x_1) + f(x_2) + \cdots + f(x_n)) \tag{5}$$

Этот метод в MathCad реализуется следующим образом (см. рисунок 2):

$$\int_0^1 14^x\,dx = 4.926$$

$$z_{i+1} := z_i + d \qquad t := 1000 \qquad z_0 := 0 \qquad z_t := 1$$

$$f(z) := 14^z \qquad i := 1..998 \qquad d := \frac{z_t - z_0}{t} = 1 \times 10^{-3}$$

$$P := d\cdot\left(\sum_{i=1}^{t-1} f(z_i) + f(z_t)\right) = 4.91954$$

Рис. 2.

в) Метод средних прямоугольников

В этом случае, также как и в первых двух случаях, отрезок $[a,b]$ - разбивается на n элементарных отрезков равной длины $h = \frac{(b-a)}{n}$.

$$\int_a^b f(x)dx = h(f(x_0 + \tfrac{h}{2}) + f\left(x_1 + \tfrac{h}{2}\right) + \cdots + f(x_{n-1} + \tfrac{h}{2})) \qquad (6)$$

Реализация этого метода в MathCad показана на рис. 3.

Разница между указанными выше методами в том, в какой точке каждого отрезка на интервале интегрирования –левой, правой или в середине – берется значение функции $f(x)$. Сравните их и сделайте выводы.

$$\int_0^1 14^x\,dx = 4.926$$

$$n := 10000 \qquad x_0 := 0 \qquad x_n := 1 \qquad i := 0..9998$$

$$h := \frac{x_n - x_0}{n} = 1 \times 10^{-4}$$

$$x_{i+1} := x_i + h \qquad f(x) := 14^x$$

$$S := h\cdot\left(f\left(x_0 + \frac{h}{2}\right) + \sum_{i=0}^{n-2} f\left(x_i + \frac{h}{2}\right) + f\left(x_{n-1} + \frac{h}{2}\right)\right) = 4.9261$$

Рис. 3.

9.1.2 Правило трапеций

Разобьем интервал интегрирования на n равных частей, каждая длиной

$$h = (b - a) / n.$$

Рассмотрим теперь один из этих интервалов. Площадь, лежащая под кри-вой $y = f(x)$, между xi и xi+1 равна

$$I_i = \int_{x_i}^{x_{i+1}} f(x)dx.$$

Если h достаточно мало, то эту площадь без большой ошибки можно приравнять к площади трапеции

$$I_i \approx \frac{1}{2}h(y_i + y_{i+1}). \qquad (7)$$

Просуммировав площади по всем n интервалам, получим

$$I = \sum_{i=0}^{n-1} I_i, \qquad (8)$$

где x0=a, xn=b. Теперь, подставляя (7) в (8), получаем

$$I \approx \frac{h}{2}(y_0 + 2y_1 + ... + 2y_{n-1} + y_n). \qquad (9)$$

Формула (9) описывает хорошо известное правило трапеций для числен-ного интегрирования.

Пример 2. Правилом трапеций вычислить интеграл $\int_0^1 14^x dx$.

$$\int_a^b f(x)dx = h(\frac{f(x_0) + f(x_n)}{2} + f(x_1) + \cdots + f(x_{n-1}))$$

Для этого набираем:

$$\int_0^1 14^x\,dx = 4.926$$

$k := 100000 \qquad x_0 := 0 \qquad x_k := 1 \qquad j := 1..99999$

$$h := \frac{x_k - x_0}{k} = 1 \times 10^{-5} \qquad x_{j+1} := x_j + h$$

$$f(x) := 14^x$$

$$T := h \cdot \left(\frac{f(x_0) + f(x_n)}{2} + \sum_{j=1}^{k-1} f(x_j) \right) = 4.92711$$

Рис.4.

9.1.3 Метод Симпсона

Более точную формулу приближенного вычисления интеграла $\int_a^b f(x)dx$ можно получить, если заменить график функции на каждом отрезке $[x_{i-1}; x_i]$ не отрезками прямых, как в методах прямоугольников и трапеций, а дугами парабол (рисунок 5).

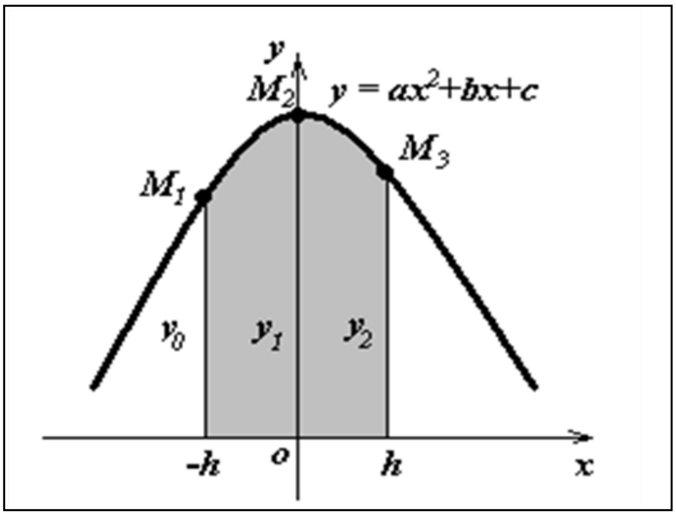


Рис.5.

Найдем вначале площадь S криволинейной трапеции, ограниченной сверху параболой $y = ax^2 + bx + c$, прямыми $x = -h$, $x = h$ и отрезком $[-h; h]$. Парабола проходит через точки $M_1(-h; y_0)$, $M_2(0; y_1)$ и $M_3(h; y_2)$. В этих точках

$$\begin{cases} y_0 = a(-h)^2 + b(-h) + c = ah^2 - bh + c \\ y_1 = a \cdot 0^2 + b \cdot 0 + c = c \\ y_2 = ah^2 + bh + c \end{cases} \qquad (9)$$

Тогда полученная площадь:

$$S = \int_{-h}^{h} (ax^2 + bx + c)dx = a\frac{x^3}{3} + b\frac{x^2}{2} + cx\Big|_{-h}^{h}$$

$$= a\frac{h^3}{3} + b\frac{h^2}{2} + ch - \left(a\frac{(-h)^3}{3} + b\frac{(-h)^2}{2} + c(-h)\right) = \frac{2}{3}ah^3 + 2ch \qquad (11)$$

Выразим полученное значение через y_0, y_1 и y_2. Используя формулы (10) получим $c = y_1$, $a = \frac{1}{2h^2}(y_0 - 2y_1 + y_2)$.

Отсюда $S = \frac{h}{3}(y_0 + 4y_1 + y_2)$ (12)

Если график функции $y = f(x)$ имеет более сложный вид (рис 6), то поступают следующим образом:

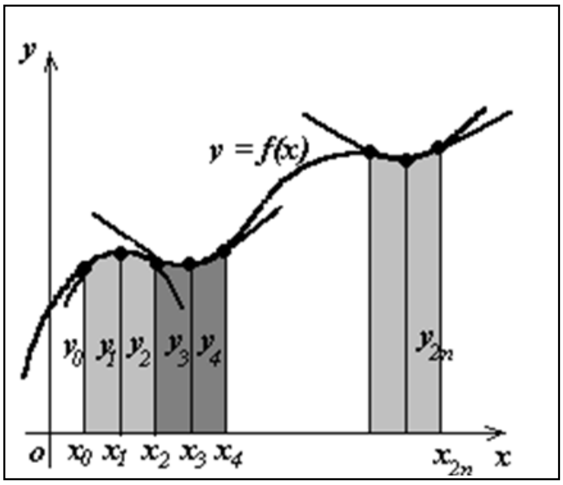


Рис.6.

1. Разбивают отрезок $[a, b]$ на $2n$ равных частей, длиной $h = \frac{b-a}{2n}$.

2. В точках деления вычисляют значения $y_0,\ y_1,\ y_2,\ \ldots\ldots,\ y_{2n-2},\ y_{2n-1},\ y_{2n}$..

3. Каждую пару соседних криволинейных трапеций заменяют параболическими трапециями с основаниями, равными $2h$. На отрезке $[x_0;\ x_2]$ парабола проходит через точки $(x_0;\ y_0)$, $(x_1;\ y_1)$, $(x_2;\ y_2)$. Тагда используя формулу (12) получим :

$$S_1 = \int_{x_0}^{x_2} f(x)dx = \frac{h}{3}\left(y_0 + 4y_1 + y_2\right)$$

Аналогично на отрезке $[x_2;x_4]$:

$$S_2 = \int_{x_2}^{x_4} f(x)dx = \frac{h}{3}\left(y_2 + 4y_3 + y_4\right)$$

$$S_n = \int_{x_{2n-2}}^{x_{2n}} f(x)dx = \frac{h}{3}\left(y_{2n-2} + 4y_{2n-1} + y_{2n}\right)$$

Следовательно:

$$\int_a^b f(x)dx = s_1 + s_2 + \cdots + s_n = \frac{h}{3}(y_0 + 4y_1 + y_2) +$$

$$\frac{h}{3}(y_2 + 4y_3 + y_4) + \cdots + \frac{h}{3}(y_{2n-2} + 4y_{2n-1} + y_{2n}) =$$

$$= \frac{h}{3}\left[(y_0 + y_{2n}) + 4(y_1 + y_3 + \cdots + y_{2n-1}) + 2(y_2 + y_4 + \cdots + y_{2n})\right]$$

Учитывая погрешность вычислений $|R_n|$и $h = \frac{b-a}{2n}$, получим формулу

Симпсона $\int_a^b f(x)dx = \frac{b-a}{6n}\left[(y_0 + y_{2n}) + 4(y_1 + y_3 + \ldots. + y_{2n-1}) + 2(y_2 + y_4 + \ldots + y_{2n})\right] \pm |R_n|$

Абсолютная погрешность метода оценивается соотношением

$$|R_n(f)| \le \frac{(b-a)^5 \cdot M_4}{180 \cdot (2n)^4} \qquad \text{где } M_2 = \max_{[a;b]} \left| f^{IV}(x) \right|$$

Пример 3. Методом Симпсона вычислить интеграл $\int_0^1 \mathbf{14}^{x} \boldsymbol{dx}$.

В данном методе отрезок интеграла $[\,a, b\,]$ разбивается не на n элементарных отрезков, а на n пар элементарных отрезков:

$h = (b-a)/2n$

$\int_a^b f(x)dx = \frac{h}{3}[(f(x_0)+ f(x_{2n}))+ 4(f(x_1)+ f(x_3)+\ldots+ f(x_{2n-1}))+2(f(x_2)+ f(x_4)+ \ldots+f(x_{2n-2}))]$.

Пример вычисления интеграла методом Симпсона на рисунке 7.

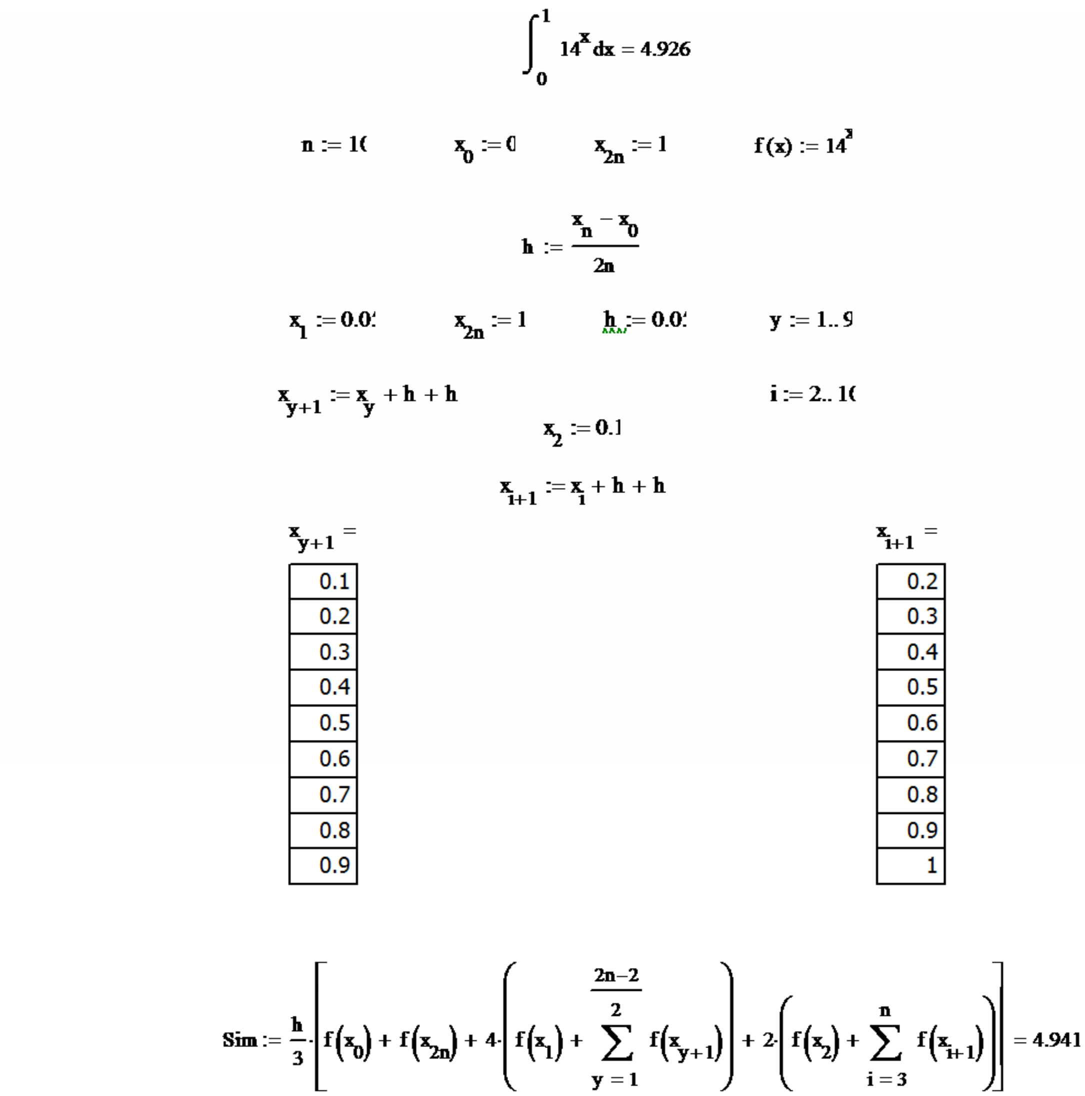


Рис.7.

9.1.4 Программирование вычисления определенного интеграла

Для этого вначале берем тестовую функцию, пределы интегрирования $[a, b]$ и число интервалов $\boldsymbol{n}$, на которое разбивается отрезок $[\boldsymbol{a}, \boldsymbol{b}]$**.** Величину шага $\boldsymbol{h}$ вычислим как $(\boldsymbol{b}-\boldsymbol{a})/\boldsymbol{n}$. Вычислим интеграл. Примерно такое же значение мы должны получать и при численном интегрировании.

Пример 4. Численным методом вычислить $\int_0^{3.2}\sqrt{x^4-x^3+8}$. Решение этого интеграла приведено на рисунке 8.

Вначале вычислим этот интеграл аналитическим методом:

$f(x) := \sqrt{x^4 - x^3 + 8}$ $\qquad a := 0$ $\qquad b := 3.2$ $\qquad n := 10$

Определяем величину шага h. Для контроля выводим "точное" решение (найденное встроенным в MAthCAD численным методом)

$h := \dfrac{b-a}{n}$ $\qquad h = 0.32$ $\qquad I := \displaystyle\int_a^b f(x)\,dx$ $\qquad I = 13.43187999195271$

Рис.8

Вычислим этот интеграл приближенными методами используя программирование в Mathcad (рисунки 9-11).

Метод левых прямоугольников

```
pr_l(a,b,n,h,f) := | I ← 0
                   | for i ∈ 0..n − 1
                   |   I ← I + f(a + i·h)
                   | I·h
```

$I1 := pr\_l(a,b,n,h,f)$ $\qquad I1 = 12.500377$

Метод правых прямоугольников

```
pr_p(a,b,n,h,f) := | I ← 0
                   | for i ∈ 1..n
                   |   I ← I + f(a + i·h)
                   | I·h
```

$I2 := pr\_p(a,b,n,h,f)$ $\qquad I2 = 14.45905$

Метод средних прямоугольников

```
pr_s(a,b,n,h,f) := | I ← 0
                   | for i ∈ 0..n − 1
                   |   I ← I + f(a + i·h + h/2)
                   | I·h
```

$I3 := pr\_s(a,b,n,h,f)$ $\qquad I3 = 13.407967$

Рис.9.

Метод трапеций

```
trap(a,b,n,h,f) := | I ← 0
                   | for i ∈ 1..n − 1
                   |   I ← I + f(a + i·h)
                   | (I + (f(a) + f(b))/2)·h
```

$I4 := trap(a,b,n,h,f) \qquad I4 = 13.479713$

Рис. 10.

Метод Симпсона (парабол)

```
simp(a,b,n,h,f) := | s1 ← 0
                   | s2 ← 0
                   | for i ∈ 1..n
                   |   | s1 ← s1 + f(a + i·h)  if mod(i,2) ≠ 0
                   |   | s2 ← s2 + f(a + i·h)  otherwise
                   | (((f(a) − f(b))/2 + 2·s1 + s2)/3)·2·h
```

$I5 := simp(a,b,n,h,f) \qquad I5 = 13.431921$

Метод Симпсона другим способом, без условного оператора

```
simp2(a,b,n,h,f) := | s ← 0
                    | for i ∈ 0..n − 1
                    |   | xi1 ← a + i·h
                    |   | xi ← xi1 + h
                    |   | s ← s + f(xi1) + 4·f((xi1 + xi)/2) + f(xi)
                    | s·h/6
```

$I6 := simp2(a,b,n,h,f) \qquad I6 = 13.431883$

Рис.11.

Ниже показаны оценки погрешностей для всех методов

$|I1 - I| = 0.931503093534339$

$|I2 - I| = 1.027169602736242$

$|I3 - I| = 0.0239126296301$

$|I4 - I| = o.\ 047833254600951$

$|I5 - I| = 0.0000407875508503$

$|I6 - I| = 0.000002665113579$

9.2. Задания к лабораторной работе

Задание 9.2.1. Отвечать на контрольные вопросы

1. Что такое шаг интегрирования?

2. По какой формуле вычисляется шаг равномерной сетки изменения x на отрезке $[a, b]$?

3. Какое влияние оказывает уменьшение числа разбиений на отрезке $[a; b]$ на погрешность интегрирования?

4. Каким образом вычисляется определенный интеграл в случае, если подынтегральная функция задана таблицей с переменным шагом?

5. Какой из изученных вами методов численного интегрирования обладает высшей степенью точности?

6. Зависит ли точность численного интегрирования от величины шага интегрирования?

7. Какие методы относятся к методам численного интегрирования?

8. Какой параметр должен быть известен, чтобы определить число разбиений отрезка $[a; b]$ при решении задачи численного интегрирования?

9. Что представляет собой формула для вычисления элементарного интеграла по формуле трапеций?

10.Что представляет собой формула для вычисления элементарного интеграла по формуле Симпсона?

11. Интерполяционным многочленом, какой степени заменяется подынтегральная функция в методе прямоугольников?

12. Как называется метод численного интегрирования, в котором подынтегральная функция заменяется полиномом нулевой степени?

13. В каком методе для вычисления интеграла необходимо выбирать количество интервалов разбиения кратное двум?

14. В каком методе при вычислении интеграла с заданной точностью потребуется меньшее количество интервалов разбиения?

15. Какой метод позволяет обеспечить вычисление интеграла с заданной точностью?

16. Какой метод численного интегрирования даст наиболее точный результат, если подынтегральная функция имеет вид $y = 5x^3$?

17. В каком методе численного интегрирования подынтегральная функция заменяется квадратичным полиномом?

18. Какой метод численного интегрирования даст точный результат, если подынтегральная функция имеет вид $f(x) = x^2$?

19. Какой метод интегрирования наилучшим образом подходит для вычисления интеграла линейной функции?

20. Обеспечивают ли методы трапеций и метод средних прямоугольников точность одного порядка?

21. Какой из известных вам методов интегрирования обладает наименьшей точностью?

22. Сколько шагов интегрирования содержит элементарный отрезок интегрирования в методе Симпсона?

23. Какому числу кратно количество интервалов разбиения в методе Симпсона?

Задание 9.2.2.

Вычислить определенный интеграл всеми вышеперечисленными методами. Индивидуальное задание выбрать из таблицы 1:

f(x) – подынтегральная функция;

a, b – пределы интегрирования;

h0 – шаг интегрирования.

Таблица 1

№	**f(x)**	**a**	**b**	h_0
1	$8\,e^{-x}\,Sin(-2x)$	2	3	0.25
2	$e^{-x}\,Sin(2x)$	0	2	0.5
3	$x^{3/2} – 2\,x\,Sin(x)$	3	4	0.25
4	$e^{-x}\,Cos(-2x)$	2	4	0.5
5	$Cos(2x) + 2\,Sin(x)$	1	3	0.5
6	$8\,Sin(2x) – x$	0.2	1.2	0.25
7	$5\,Cos(-2x)\,e^{-x}$	-0.5	0.5	0.25
8	$x\,Sin(x + 1) – Cos(x – 5)$	1	2	0.25
9	$8\,(x – 1)\;e^{-x^2/2}$	1.2	3.2	0.5
10	$Sin(2x) – 2\,Sin(x)$	3	5	0.5
11	$Sin(e^{x}) – e^{-x} +1$	0	1	0.25
12	$5\,x\,Sin(x + 1) + 2\,Cos(x)$	1	2	0.25
13	$5\,e^{-x} + 4\,x + x^3/3$	-1	1	0.5
14	$Sin(x – 1) – x\,Cos(x + 3)$	-4	-2	0.5

15	$5\ Sin^3(x) + Cos^3(x)$	1	2	0.25
16	$Sin(x^2) + 1 / (2 – x)$	-1.5	0.5	0.5
17	$X\ Sin(x) + Cos(x) + 5$	0	2	0.5
18	$(1 + x^2)^{1/2} + e^{-x}$	-1	2	0.75
19	$Sin(x + 1)\ e^{2\,/\,x}$	1	2	0.25
20	$2\ (1 + x)\ e^{-x} – 2\ Cos(x)$	1	4	0.75

Литература

1. Демидович Б.П., Марон И.А. Основы вычислительной математики. – М.: Наука, 1970. – 664 с.

2. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. – М.: Наука, 1987. – 432 с.

3. Самарский А.А., Гулин А.В. Численные методы. – М.: Наука, 1989. – 432 с.

4. Алексеев Е.Р., Чеснокова О.В., Рудченко Е.А. Scilab: Решение инженерных и математических задач. − М.: ALT Linux: БИНОМ. Лаборатория знаний, 2008. − 260 с.

5. Крылов, В. И. Приближенное вычисление интегралов / В. И. Крылов. – М. : ГИФМЛ, 1959. – 327 с.

6. Крылов, В. И. Вычислительные методы / В. И. Крылов, В. В. Бобков, Т. И. Монастырский. – Т. 1. – М. : Наука, 1976. – 303 с.

7. Никольский, С. М. Квадратурные формулы / С. М. Никольский. – М. : Наука, 1988. – 255 с.

10. Лабораторная работа №10.
ЧИСЛЕННОЕ РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ И ИХ СИСТЕМ В MATHCAD

10.1 Основы теории

Дифференциальные уравнения играют важнейшую роль в моделировании и анализе разнообразных явлений и процессов в науке и технике. В последние годы особую роль они начали играть при решении экономических задач. Основными методами решения дифференциальных уравнений являются аналитические, в результате которого получаются решения в виде конкретных аналитических функций. К сожалению, с помощью аналитических методов можно решать ограниченный круг задач. Для решения задач более сложного характера разработаны численные (приближенные) методы решения дифференциальных уравнений, с помощью которых искомые интегральные кривые получаются в виде таблиц их числовых значений. Такие методы позволяют получить приближенное решение дифференциального уравнения практически любой задачи с заранее заданной точностью.

Решить дифференциальное уравнение

$$y' = f(x, y) \qquad (1)$$

численным методом означает, что для заданной последовательности аргументов $x_0, x_1, x_2, \ldots, x_n$ и числа $y_0 = y(x_0)$ (задача Коши) не определяя аналитического вида функции $y = F(x)$, найти значения $y_1, y_2, \ldots, y_n$, удовлетворяющие условиям:

$$F(x_0) = y_0, \; y_k = F(x_k), \; k = \overline{1, n}.$$

Рассмотрим два, наиболее распространенных при решении практических задач, численных метода решения дифференциальных уравнений: **метод Эйлера и метод Рунге-Кутта.**

10.1.1 Метод Эйлера

Пусть дано дифференциальное уравнение с начальным условием (задача Коши)

$$y' = f(x, y), \; y_0 = y(x_0) \qquad (2)$$

и выполняются условия существования и единственности решения. Требуется найти решение $y(x)$ уравнения (2). Для этого:

1. Отрезок $[a, b]$ разделяют на n равных частей. Выбирают шаг h - достаточно малый, равный $h=(b-a)/n$, и строят систему равноотстоящих точек: x_0 , x_1 , . . . , x_n , $x_i = x_0 + ih,\ i=\overline{0,n}$.

2. Искомую интегральную кривую $y = y(x)$, проходящую через точку $M_0(x_0, y_0)$, приближенно заменяют ломаной Эйлера с вершинами $M_i(x_i y_i), i = 0,1,2....$

Звено ломаной M_iM_{i+1}, заключенное между x_i и x_{i+1}, наклонено к оси OX под углом α. Тангенс этого угла вычисляется по формуле

$$tg\alpha_i = \frac{y_{i+1} - y_i}{h} = y'(x_i) = f(x_i, y_i).$$

Отсюда, сделав преобразование, получим **формулу Эйлера:**

$$y_{i+1} = y_i + h\,f(x_i, y_i),\ i=\overline{0,n} \qquad (3)$$

Вычисление значений $y_1, y_2,\ . . . ,\ y_n$ осуществляется с использованием формулы (3) по следующему алгоритму:

1. По заданным начальным условиям $x_0 = a$ и y_0 полагая $i = 0$ в выражении (3) вычисляют значение

$$y_1 = y_0 + h\,f(x_0, y_0). \qquad (4)$$

2. Определяя значение аргумента x по формуле $x_1 = x_0 + h$, используя найденное значение y_1 и полагая в формуле (3) $i = 1$ вычисляют следующее приближенное значение интегральной кривой $y = F(x)$:

$$y_2 = y_1 + h\,f(x_1, y_1). \qquad (5)$$

3.Поступая аналогичным образом при $i = \overline{2, n-1}$ определяют все остальные значения y_i , в том числе и последнее значение $y_n = y_{n-1} + h\,f(x_{n-1}, y_{n-1})$, которое соответствует значению аргумента $x_n = b$.

Таким образом, соединяя на координатной плоскости точки $(x_0, y_0),\ (x_1, y_1),\ . . . ,(x_n,y_n)$ отрезками прямых, получаем ломаную линию с вершинами в точках $M_0(x_0, y_0), M_1(x_1, y_1),\ . . . ,M_n(x_n, y_n)$. Эту ломаную приближенно принимают за искомую интегральную кривую уравнения (1). Точность решения методом Эйлера оценивается по формуле

$$|y_{i+1} - \tilde{y}_{i+1}| \le \max_{x_i} \frac{h^2}{2!} y''(x_i, y_i) = \max_{x_i} \frac{h^2}{2!} f'(x_i, y_i). \qquad (6)$$

Достоинством метода Эйлера является его простота и высокая скорость поиска решения. Недостатками метода Эйлера являются малая точность и систематическое накопление ошибок.

Пример 1. Решим задачу Коши для дифференциального уравнения первого порядка $y' = f(x, y)$ методом Эйлера.

Пусть правая часть уравнения равна $f(x, y) = x * y$. Легко показать, что аналитическое решением этого уравнения имеет вид: $y(x) = \exp\left(\frac{x^2}{2}\right)$.

Решим это уравнение средствами MathCad: $f(x, y) := x * y$

Представим результат графически и сравним его с аналитическим решением. На рисунке 1 аналитическое решение показано в виде пунктирной линии.

$$f(x,y) := x \cdot y$$

$$n := 1($$

$$h := 0.1$$

$$i := 0..n$$

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} := \begin{pmatrix} x_i + h \\ y_i + h \cdot f(x_i, y_i) \end{pmatrix}$$

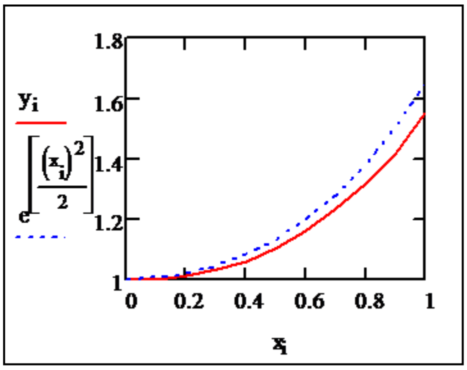


Рис. 1. Решение дифуравнения 1-го порядка методом Эйлера

Точное аналитическое решение и решение, полученное численно, отличаются в точке $x = 1$ на $y1(1) - y_n$= 0.102, т.е. относительная ошибка составляет

10.1.2 Метод Рунге – Кутта

Данный метод имеет более высокую точность (но невысокую скорость) поиска решения дифференциального уравнения.

Для поиска решения уравнения (2) этим методом, также как и в методе Эйлера, выбирают шаг h, $x_i = x_0 + ih$, $y_i = y(x_i)$, где $i = 0,1,\dots$. , и рассматривают числа:

$$k_{1i} = hf(x_i, y_i,), \qquad k_{2i} = hf(x_i + \frac{h}{2}, y_i + \frac{k_{1i}}{2}),$$

$$k_{3i} = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_{2i}}{2}\right), \quad k_{4i} = hf(x_i + h,\ y_i + k_{3i}). \tag{7}$$

Последовательные значения y_i искомой функции y определяют по итерационной формуле:

$$y_{i+1} = y_i + \frac{1}{6}(k_{1i} + k_{2i} + k_{3i} + k_{4i}). \tag{8}$$

Формулу (8) называют формулой Рунге-Кутта четвертого порядка точности. Помимо формулы (8) существуют и другие формулы типа Рунге-Кутта с иными порядками точности.

Пример 2. Решим еще раз задачу Коши для дифференциального уравнения первого порядка $\boldsymbol{f(x, y) = x * y}$ методом Рунге–Кутты используя средства MathCad.

Зададим границы изменения *x:* $\boldsymbol{x_{min} = 0}$ $\boldsymbol{x_{max} = 1}$.

Зададим число точек внутри интервала: $\boldsymbol{n = 10}$.

Зададим начальные условия $\boldsymbol{y_0 := 1}$

Обратите внимание на обозначения! Поскольку мы решаем только одно дифференциальное уравнение первого порядка, а не систему дифференциальных уравнений, матрица *y* содержит только один элемент, однако запись *y=1* была бы неправильной. Необходимо явно указывать на то, что величина *y* есть матрица, то есть писать индекс.

Функция ***rkfixed*** позволяет находить решение для дифференциального уравнения *n*-го порядка, а также систем дифференциальных уравнений 1-го порядка. Она возвращает матрицу, имеющую *n+1* столбец: первый столбец содержит точки, в которых ищется решение дифференциального уравнения, второй столбец содержит значения найденного решения в соответствующих точках, с третьего по *n+1*-ый – значения производных y', y'', ..., $y^{(n-1)}$.

Вызов функции имеет вид: ***rkfixed(y, x1, x2, k, D),*** *y* – вектор начальных условий размерности *n*, где *n* – порядок дифференциального уравнения или число уравнений в системе (если решается система уравнений). Для дифференциального уравнения 1-го порядка вектор начальных условий вырождается в одну точку $y_0=y(x1)$;

x1, x2 – граничные точки интервала, на котором ищется решение;

k – число точек (не считая начальной точки), в которой ищется приближенное решение. При помощи этого аргумента определяется число строк ($1+k$) в матрице, возвращаемой этой функцией;

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

Определим теперь матрицу производных. Эта матрица тоже состоит только из одного элемента. Этот элемент с точностью до обозначений совпадает с правой частью исходного дифференциального уравнения: $D(x,y) := y_o * x$

Решаем дифференциальное уравнение:

Z := rkfixed(y, $\boldsymbol{x_{min}, x_{max}, n, D}$**)**

k:=0..n $\boldsymbol{y1(x) = \exp\left(\frac{x^2}{2}\right)}$ **z:=0,0.1..1**

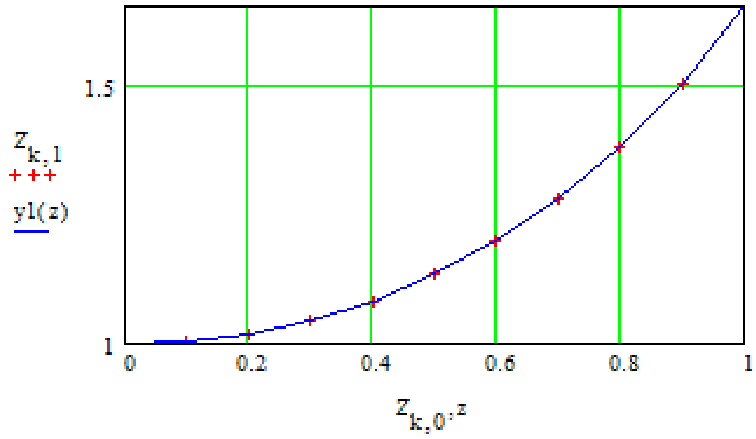


Рис. 2. Решение дифуравнения 1-го порядка методом Рунге–Кутты

Точное аналитическое решение и решение, полученное численно отличаются в точке $x = 1$ на $y1(1) - (Z^{<1>})_n = 2.636*10^{-7}$.

Относительная ошибка составляет $\frac{\left[y1(1) - \left(Z^{<1>}\right)_n\right]}{y1(1)} = 1.599 \cdot 10^{-5} \cdot \%$

Примечание: ***Mathcad*** *имеет еще две функции для решения задачи Коши. Это функции* ***Rkadapt*** *и* ***Bulstoer****. Эти функции имеют те же самые аргументы и возвращают решения в такой же форме, что и функция* ***rkfixed****. Первая из этих функций использует метод Рунге–Кутты с* *переменным шагом, что позволяет повысить точность вычислений и сократить их объем, если искомое решение имеет области, где ее значения меняются быстро, и области плавного изменения. Функция* ***Rkadapt*** *будет варьировать величину шага в зависимости от скорости изменения решения. Функцию* **Bulstoer(y,x1,x2,n,F)** *следует применять, если известно, что решение является гладкой функцией. Она дает более точное решение. Аргументы вышеуказанных функций таковы:*

y *- вектор начальных условий;*

x1,x2 *- границы интервала для поиска решения;*

n *- количество точек на интервале*

F(x,y) *- вектор-функция первых производных*

На рис.3. приведены решения полученные применением функций ***rkfixed и Rkadapt***

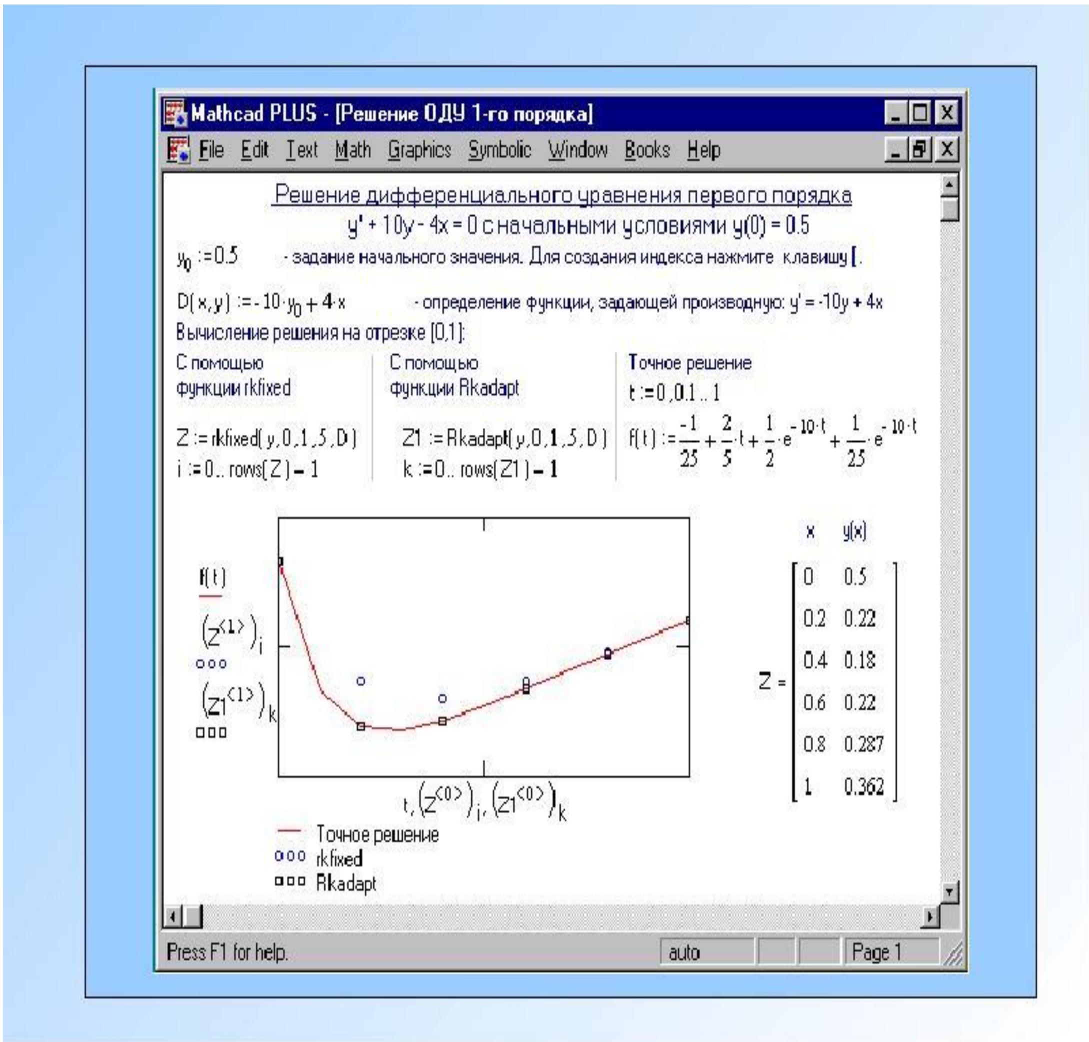


Рис.3. Сравнение решений полученных при использовании функций ***rkfixed и Rkadapt***

Пример 3. Найти решение уравнения $y' = -y^2 + x$, с начальным условием y(0)=1 точках на отрезке [0,100]. Решение приведено на рисунке 4.

ORIGIN:= 1

$Y_1 := 1$

$D(x,y) := -y_1 + 2\cdot x$

z := rkfixed(Y,0,2,100,D)

x y(x)

z =

	1	2
1	0	1
2	0.02	0.981
3	0.04	0.962
4	0.06	0.945
5	0.08	0.929
6	0.1	0.915
7	0.12	0.901
8	0.14	0.888
9	0.16	0.876
10	0.18	0.866
11	0.2	0.856
12	0.22	0.848
13	0.24	0.84
14	0.26	0.833
15	0.28	0.827
16	0.3	...

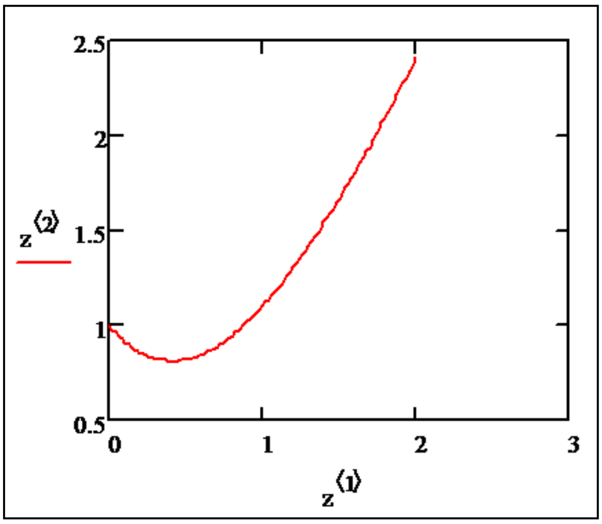


Рис.4. Решение дифференциального уравнения 1-го порядка

Пример 4. Найти решение дифуравнения второго порядка

$$y'' + \sin(x)\, y' - 3.5y = 1$$

с начальными условиями $y(1) = 0, y'(1) = 0.5$ в 8 точках отрезка [1,2].

Решение.

Приведем уравнение к системе уравнений 1-го порядка. Пусть $y' = t$.

Тогда $t' + \sin(x)\, t - 3.5y = 1$ с начальными условиями y(1)=0, t(1)=0.5

Начальные условия задаем по вектору $\boldsymbol{y}$, поэтому $\boldsymbol{y_0} = \mathrm{y}(1)$, $\boldsymbol{y_1} = \mathrm{t}(1)$.

Отсюда имеем (см. рисунок 5):

$$y := \begin{pmatrix} 0 \\ 0.5 \end{pmatrix}$$

$$D(x,y) := \begin{pmatrix} y_1 \\ 1 - \sin(x)\cdot y_1 - 3.5\cdot y_0 \end{pmatrix}$$

$$Z := \mathrm{rkfixed}(y,1,2,8,D)$$

$$Z = \begin{pmatrix} 1 & 0 & 0.5 \\ 1.125 & 0.066 & 0.553 \\ 1.25 & 0.137 & 0.568 \\ 1.375 & 0.207 & 0.55 \\ 1.5 & 0.273 & 0.505 \\ 1.625 & 0.332 & 0.438 \\ 1.75 & 0.382 & 0.357 \\ 1.875 & 0.421 & 0.268 \\ 2 & 0.449 & 0.176 \end{pmatrix}$$

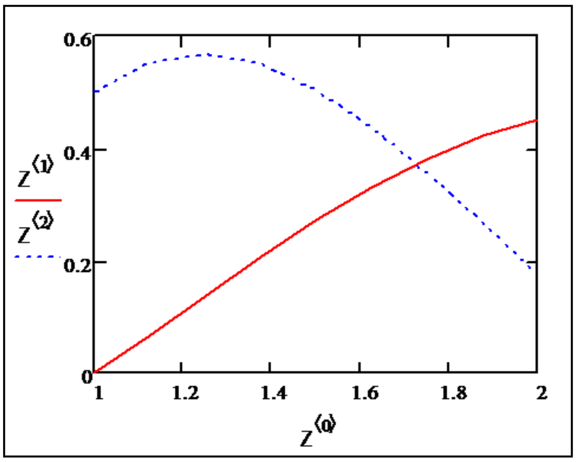


Рис.5. Решение дифференциального уравнения
2-го порядка

10.1.3 Численное решение систем дифференциальных уравнений

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

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

2) в документе MathCad сформировать вектор начальных условий, количество элементов которого равно количеству уравнений системы, присвоив его некоторой переменной например, $v := \begin{pmatrix} 0.1 \\ 1 \end{pmatrix}$;

3) 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;

– набрать в качестве элементов вектора правые части системы уравнений, в которых искомые функции представлены соответствующими элементами вектора-параметра, например,

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

4) 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.).

5) вывести матрицу, содержащую решение системы ДУ с помощь оператора «=», например: *Z* = ;

6) построить графики найденных функций, указав в качестве аргумента по оси абсцисс первый столбец матрицы решений, например, $Z^{\langle 1 \rangle}$, а в качестве значений функций по оси ординат – остальные столбцы матрицы через запятую, например, $Z^{\langle 2 \rangle}$, $Z^{\langle 3 \rangle}$ и т. д.

Пример 5. 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.

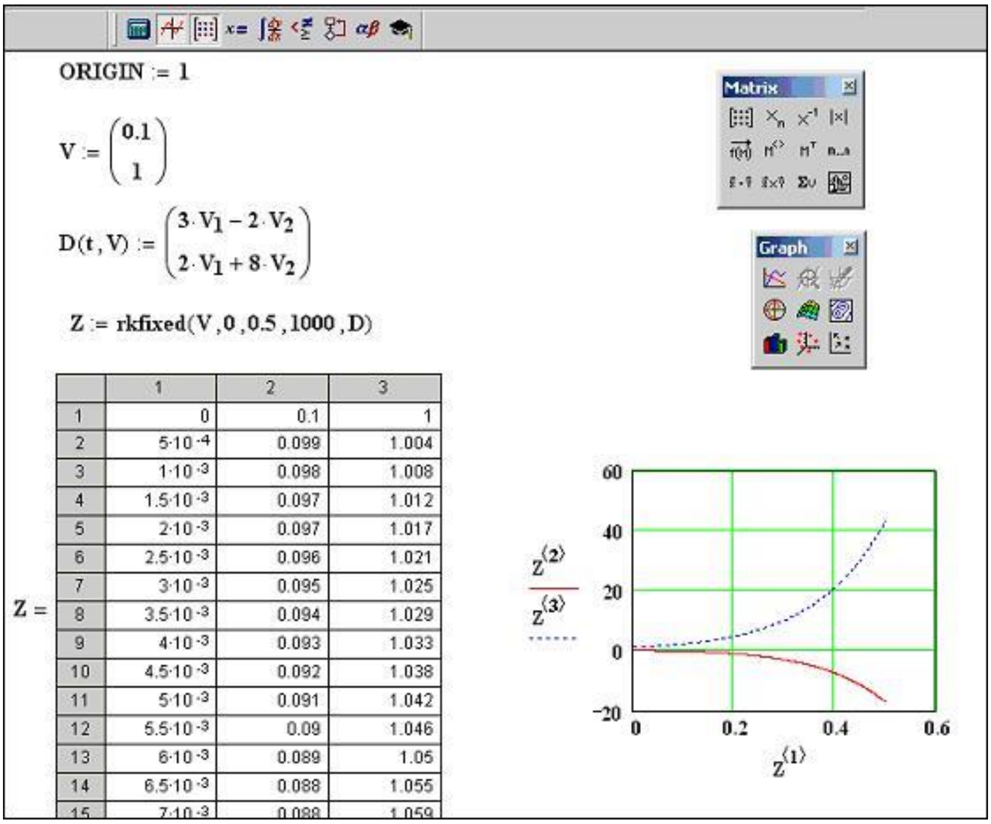


Рис.6. Решение системы дифференциальных уравнений 1-го порядка

На рисунке 7 показано численное решение систем дифференциальных уравнений 1-го порядка методом Эйлера.

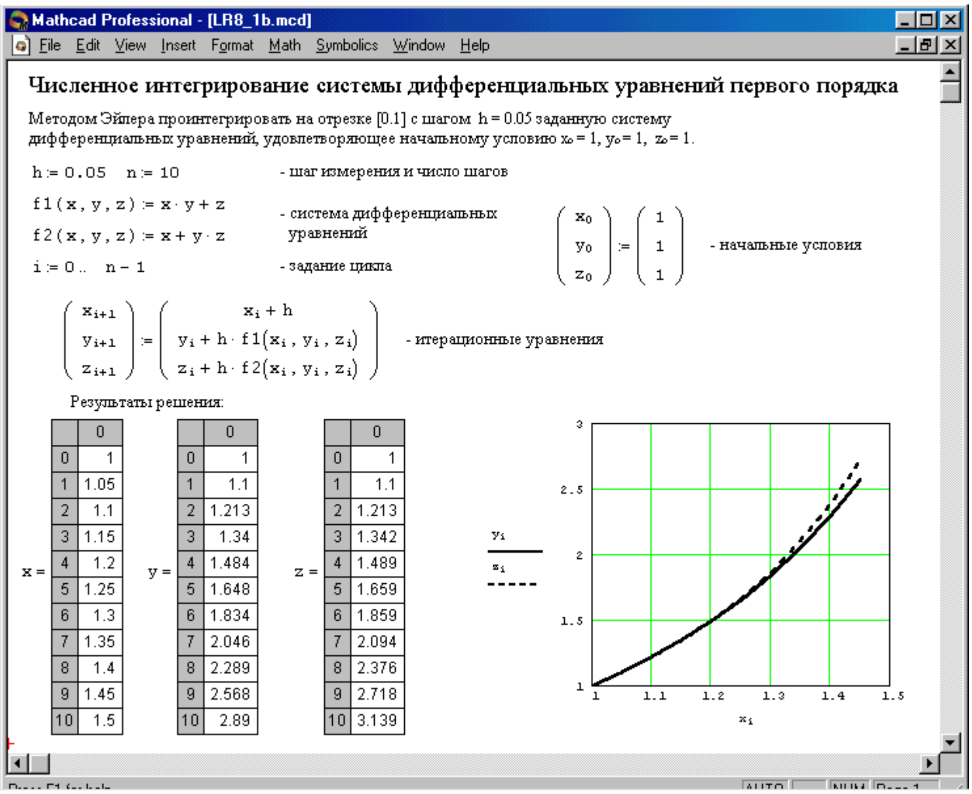


Рис.7.

10.1.4 Использование вычислительного блока Given/Odesolve

Ниже приведены примеры для решения дифуравнений первого и второго порядков с использованием вычислительного блока решения Given/Odesolve.

а) решение дифуравнения первого порядка в MathCad:

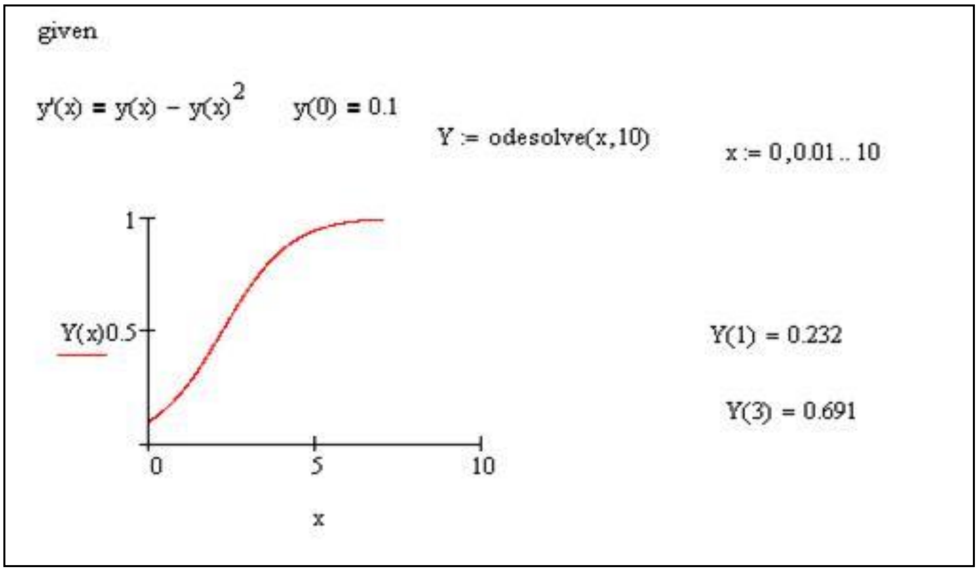


Рис.8. Решение дифференциального уравнения 1-го порядка

б) решение дифуравнения второго порядка в MathCad:

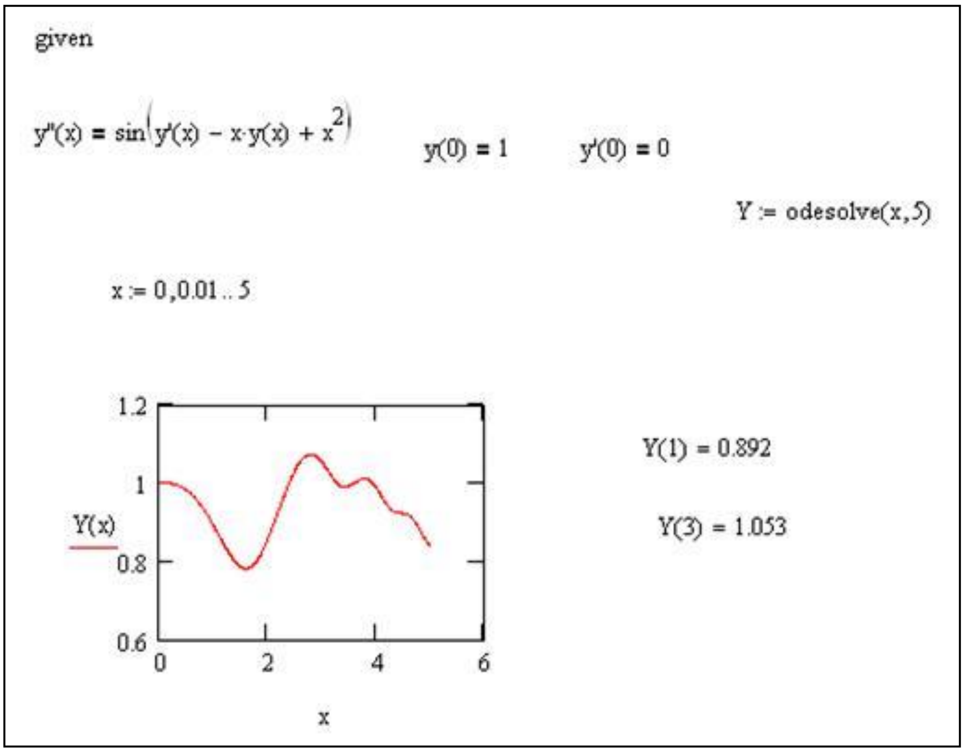


Рис.9. Решение дифференциального уравнения 2-го порядка

10.2. Задания к лабораторной работе

Задание 10.2.1. Отвечать на контрольные вопросы

1.Дайте определение дифференциального уравнения.

2. Сформулируйте постановку задачи Коши.

3. В чем разница между аналитическим и численным решением дифференциального уравнения?

4. В чем состоит основная идея методов Рунге- Кутта?

5. В чем достоинства и недостатки метода Рунге-Кута?

6. Приведите формулы методов Рунге- Кутта 1, 2 и 4-го порядка.

7. Приведите классификацию дифференциальных уравнений.

8. Определите локальную и глобальную погрешность задачи Коши.

9. В чем состоит идея решения дифференциального уравнения высокого порядка.

10. Как можно использовать методы Рунге- Кутта для решения системы дифференциальных уравнений?

11. Что такое функция rkfixed и как она работает?

Задание 10.2.2.

Согласно Вашего варианта (Таблица 1) найти решение дифференциального уравнения на интервале [0, 1] с шагом 0,1. Для чего:

– разработать блок-схемы и программы методов Эйлера и Рунге-Кутта;

– найти решение по разработанным программам и с использованием функции *rkfixed и других*. В результатах оставлять четыре цифры после запятой;

– построить графики. По результатам сделать вывод о полученных расхождениях.

Задание 10.2.3.

Согласно Вашего варианта (Таблица 2) найти решение дифференциального уравнения второго порядка на интервале [1, 2] с шагом 0,1. Для чего:

– привести уравнение к системе уравнений первого порядка;

– найти решение с использованием функции *rkfixed и других*;

– построить графики.

Таблица 1

№	Уравнение	Начальное условие	№	Уравнение	Начальное условие
1	$y' = x + y^2$	$y(0) = 0{,}5$	9	$y' = x^2 + 0{,}2xy$	$y(0) = 0{,}6$
2	$y' = 2x + 0{,}1y^2$	$y(0) = 0{,}2$	10	$y' = 3x^2 + 0{,}1xy$	$y(0) = 0{,}2$

3	$y' = 2x + y^2$	$y(0) = 0,3$	**11**	$y' = x^2 + 3xy$	$y(0) = 0,3$
4	$y' = x^2 + xy$	$y(0) = 0,2$	**12**	$y' = x^2 + 0,1y^2$	$y(0) = 0,7$
5	$y' = 0,2x + y^2$	$y(0) = 0,1$	**13**	$y' = 2x^2 + 3y^2$	$y(0) = 0,2$
6	$y' = x^2 + y$	$y(0) = 0,4$	**14**	$y' = 0,2x^2 + y^2$	$y(0) = 0,8$
7	$y' = x^2 + 2y$	$y(0) = 0,1$	**15**	$y' = 0,3x^2 + y^2$	$y(0) = 0,3$
8	$y' = xy + y^2$	$y(0) = 0,6$	**16**	$y' = xy + 0,1y^2$	$y(0) = 0,5$

Таблица 2

№	**Уравнение**	**Начальные условия**	**№**	**Уравнение**	**Начальные условия**
1	$y'' + xy' + y = x + 1$	$y(1) = 1$, $y'(1) = 1,2$	9	$y'' - 0,5xy' + y = 2$	$y(1) = 1,2$ $y'(1) = 1,4$
2	$y'' + 2y' - xy = x^2$	$y(1) = 0,7$ $y'(1) = 1$	10	$y'' + 2x^2y' + y = x$	$y(1) = 1$, $y'(1) = 3$
3	$y'' + 2xy' - y = 0,4$	$y(1) = 1$, $y'(1) = 2$	11	$y'' + 2xy' - 2y = 0,6$	$y(1) = 1$, $y'(1) = 1$
4	$y'' - 3xy' + 2y = 1,5$	$y(1) = 1,3$ $y'(1) = 2$	12	$y'' + 0,8y' - xy = 1,4$	$y(1) = 0,5$ $y'(1) = 1,7$
5	$y'' + 2xy' - 1,5 = x$	$y(1) = 2$, $y'(1) = 2,5$	13	$y'' + 2y' - 1,5xy = 2/x$	$y(1) = 1$, $y'(1) = 1$
6	$y'' + 0,6y' - 2y = 1$	$y(1) = 0,6$ $y'(1) = 3$	14	$y'' + 2xy' - 1,5 = x$	$y(1) = 2$, $y'(1) = 2,5$
7	$y'' - xy' + 2y = x + 1$	$y(1) = 2$, $y'(1) = 1$	15	$y'' - y' + 2y/x = x + 0,4$	$y(1) = 2$, $y'(1) = 4$
8	$y'' + 2y' - y/x = 3$	$y(1) = 2$, $y'(1) = 1$	16	$y'' + 1,5y' - xy = 0,5$	$y(1) = 1$, $y'(1) = 3$

Литература

1.Кирьянов, Д. В. MathCAD15/MathCAD Prime 1.0 / Д. В. Кирьянов. СПб. : БХВ-Петербург, 2015. – 432 с.

2. Назаров, Д. М. MathCAD 14: Основные сервисы и технологии / Д. М. Назаров, Г. И. Пожарская. М. : Национальный открытый университет «ИНТУИТ», 2016. – 139 с.

3. Агафонов, Е. Д. Прикладное программирование : учеб. пособие / Е. Д. Агафонов, Г. В. Ващенко. – Новосибирск: Сибирский федеральный университет, 2015. – 112 с.

4. Самоучитель по MathCAD 15. – URL: https://www.rk5.msk.ru/ Knig/Matycad11/pdf.5. Поршнев С.В., Беленкова *И.В.* Численные методы на базе MATHCAD. – СПб.: БХВ-Петербург, 2005. – 464 с.: ил.

5. Плисс А. И., Сливина Н. А. MathCad 2000. Математический практикум для экономистов и инженеров: Учеб. Пособие. – М.:Финансы и статистика, 2002.

6. 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.

7. 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.

8. 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.

9. 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.

10. Денисов-Винский, Н. Д. Mathcad при решении задач по курсу Математика. II курс / Н. Д. Денисов-Винский. – Москва : МИЭЭ, 2009. – 132 с.

11. Эдвардс, Ч. Г. Дифференциальные уравнения и краевые задачи: моделирование и вычисление с помощью Mathematica, Maple и MATLAB / Ч. Г. Эдвардс, Д. Э. Пенни. – Москва : Вильямс, 2008. – 1104 с.

12. Вельмисов, П. А. Дифференциальные уравнения : учеб. пособие / Вельмисов П. А., Покладова Ю. В., Распутько Т. Б.; Ульян. гос. техн. ун-т. – Ульяновск : УлГТУ, 2013. – 90 с.