Министерство образования и науки Российской Федерации
Государственное образовательное учреждение высшего профессионального образования
Магнитогорский государственный технический университет им. Г.И.Носова

И.И. Баранкова, Т.Н. Носова

Применение СКМ MathCAD в моделировании

Утверждено Редакционно-издательским советом университета в качестве учебного пособия

Магнитогорск
2010

УДК 519.6

Рецензенты:

Заведующий кафедрой ВТ и СУ АНО КЦПК «Персонал»
А.Н. Бурыкин

Заведующий кафедрой информационных технологий ОАНО ВПО института «Магнитогорская высшая школа бизнеса», кандидат технических наук
А.В. Леднов

Баранкова И.И., Носова Т.Н.
Применение СКМ MathCAD в моделировании: учеб. пособие. - Магнитогорск: Изд-во Магнитогорск. гос. техн. ун-та им.Г.И.Носова, 2010. - 99 с.
ISBN 978-5-9967-0161-2

В данном учебном пособии подробно рассматриваются вопросы использования пакета MathCad в инженерных и математических расчетах. Рассмотрены вопросы общего и прикладного использования пакета при решении задач матричной алгебры, задач высшей математики, оптимизации, физических задач и др. Особое внимание уделено использованию системы в математическом моделировании. Рассмотрены вопросы использования MathCad при описании статических и динамических математических моделей, построения моделей по экспериментальным данным, решения задач интерполяции, поиска аппроксимирующих функций.

Пособие рекомендовано для студентов дневной и заочной форм обучения специальностей 150108, 150106, 150105,190603.

УДК 519.6

ISBN 978-5-9967-0161-2

Введение

Широкую известность и заслуженную популярность еще в середине 80-х годов приобрели интегрированные системы для автоматизации математических расчетов класса MathCAD, разработанные фирмой MathSoft (США). Основным достоинством этих систем является возможность описания решения математических задач с помощью привычных математических формул и знаков. Такой же вид имеют и результаты вычислений.

С момента своего появления системы класса MathCAD имели удобный пользовательский интерфейс – совокупность средств общения с пользователем в виде масштабируемых и перемещаемых окон, клавиш и иных элементов. У этой системы есть и эффективные средства типовой научной графики, они просты в применении и интуитивно понятны.

MathCAD – математически ориентированная универсальная система. Помимо собственно вычислений она позволяет решать задачи, которые с трудом поддаются популярным текстовым редакторам или электронным таблицам. С ее помощью можно не только качественно подготовить тексты статей, книг, диссертаций, научных отчетов, дипломных и курсовых проектов, но и облегчить набор самых сложных математических формул, получить результаты в наглядном графическом представлении, в виде двумерных, пространственных или анимированных изображений.

Последние версии системы MathCAD дают новые средства для подготовки сложных документов. В них предусмотрено красочное выделение отдельных формул, многовариантный вызов одних документов из других, возможность закрытия "на замок" отдельных частей документов, гипертекстовые и гипермедиа-переходы и т.д. Это позволяет создавать превосходные обучающие программы и целые книги по любым курсам, базирующимся на математическом аппарате. Здесь же реализуется удобное и наглядное объектно-ориентированное программирование сложнейших задач, при котором программа составляется автоматически по заданию пользователя, а само задание формулируется на естественном математическом языке общения с системой.

Принятые условные обозначения

Обозначение	Расшифровка обозначения
	Обозначает любую информацию, на которую следует обратить внимание
	Ниже следует список вопросов для самоконтроля
	Рассматривается пример
	Ниже будут приведены контрольные задания
	Обозначение для контрольных заданий первого уровня сложности
	Обозначение для контрольных заданий второго (повышенного) уровня сложности

Раздел 1. ОСНОВЫ РАБОТЫ С СКМ MathCAD

1.1. Входной язык системы

Самым важнейшим свойством MathCAD является возможность описания математических алгоритмов в естественной математической форме с применением общепринятой математической символики. Такой подход значительно облегчает восприятие математической сущности решаемой задачи и избавляет пользователя от изучения некоторого промежуточного языка программирования.

Этот факт, однако, не означает, что в системе нет своего языка программирования. В действительности он есть, но это математически ориентированный язык программирования сверхвысокого уровня, используемый в основном как входной язык для диалога с системой.

По существу входной язык системы – промежуточное звено между скрытым от пользователя истинным языком программирования и текстом документа. По мере того, как пользователь средствами встроенных редакторов создает объекты (тексты, формулы, таблицы и графики), система сама составляет программу, которая хранится в ОЗУ до тех пор, пока не будет сохранена на диске в виде файла с расширением **.mcd.**

Работа с системой приобрела характер визуального объектно-ориентированного программирования. *При этом решающим является задание объектов, а программу система составляет са-*

ма. Более того, в MathCAD эффективно решена *проблема сквозной передачи данных* от одного объекта к другому (от выражения к другому выражению, от него к таблицам и графикам). Поэтому изменение в любой формуле или в задании исходных данных тут же ведет к пересчету задачи по всей цепи взаимодействия объектов.

1.2. Основы пользовательского интерфейса

Под интерфейсом пользователя подразумевается совокупность средств графической оболочки MathCAD, обеспечивающих легкое управление системой как с клавиатуры, так и с помощью мыши. Под управлением понимается и просто набор необходимых символов, формул, текстовых комментариев и т.д., и возможность полной подготовки в среде MathCAD документов (Worksheets) и электронных книг с последующим их запуском в реальном времени.

Пользовательский интерфейс системы создан так, что пользователь, имеющий элементарные навыки работы с Windows-приложениями, может сразу начать работу с MathCAD.

Вверху окна на рис. 1 показаны несколько строк с типовыми элементами интерфейса. Верхняя строка – титульная. Она отображает название загруженного или создаваемого документа.

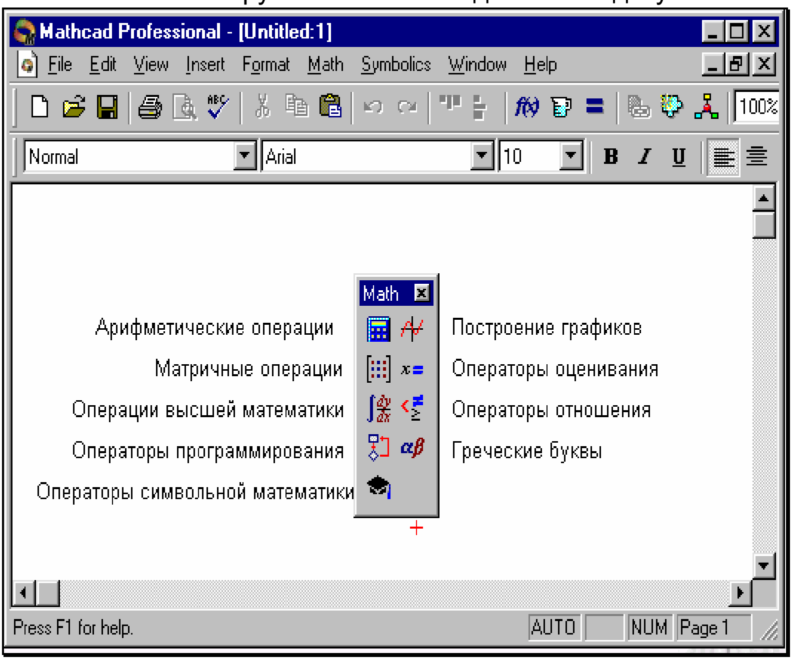


Рис. 1. Основные элементы интерфейса системы MathCAD

Вторая строка окна системы – главное меню. Далее следуют панели инструментов. Работа с документами MathCAD обычно не требует обязательного использования команд главного меню, так как основные из них дублируются кнопками на панелях инструментов, находящихся под строкой главного меню. Обычно при загрузке MathCAD имеются две панели: панель инструментов ***Standard***, содержащая несколько групп кнопок (пиктограмм), каждая из которых дублирует одну из важнейших операций главного меню, и ***Formating*** – панель форматирования.

Панель инструментов ***Formatting*** содержит типовые средства управления шрифтами: пиктограммы выбора типа символов, размеров шрифтов, начертания, выравнивания текста.

С перечнем основных пиктограмм и их назначением можно ознакомиться в приложении 1.

Панели инструментов можно выводить на экран или убирать с помощью соответствующих опций ***View*** (Вид)***/Toolbars*** (Панели инструментов) главного меню, или с помощью панели ***Math***.

Назначение кнопок панели ***Math*** показано на рис. 2. Нажатие каждой кнопки на этой панели инструментов выводит отдельную палитру инструментов специального назначения.

Рис. 2. Обзор основных панелей инструментов MathCAD

Вопросы для самоконтроля

1. Какое расширение имеют документы, созданные в системе MathCAD?
2. В системе MathCAD необходимо открыть ранее созданный документ. Какими способами это можно сделать?
3. Какая панель инструментов является основной, предназначенной для вызова в рабочую область документа всех остальных панелей?
4. Для построения графика в декартовой системе координат требуется панель инструментов “Graph”. Перечислите способы вызова этой панели.
5. С помощью какой панели инструментов можно изменить размер и начертание шрифта в документе MathCAD?

1.3. Основные приемы работы

В простейшем случае работа с системой MathCAD сводится к подготовке в документе задания на вычисление и заданию вида (формата) результатов.

!	*Каждый объект в документе оформляется в виде отдельного блока.*

Для подготовки блоков используются три встроенных редактора: текстовый, формульный и графический.

Для запуска каждого редактора достаточно установить курсор мыши в любом свободном месте окна редактирования и щелкнуть левой клавишей. Появляется визир в виде маленького красного крестика, указывающий левый верхний угол блока.

Текстовый редактор начинает работать при наборе текста, **формульный** – при наборе вычислительных блоков с использованием шаблонов заполнения и специальных математических символов; **графический редактор** вызывается, если дана команда на построение графика. Как только блок опознается, система автоматически запускает внутренние подпрограммы выполнения необходимых действий (вычисления по формуле, вывод таблицы значений вектора, построение графика и т.п.).

Порядок вычисления блоков демонстрируется на рис. 3.

!	*Просмотр блоков и вычисления производятся сверху вниз и слева направо.*

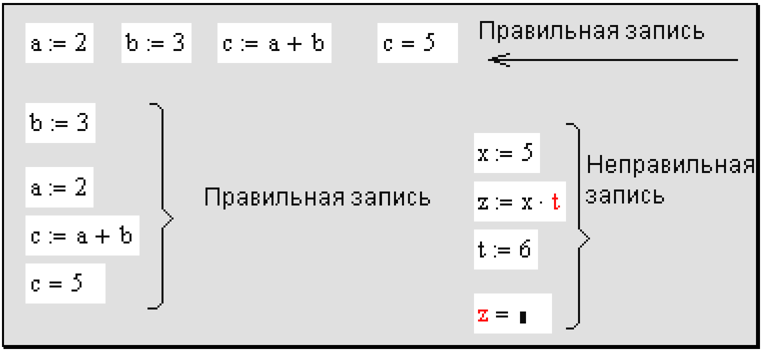


Рис. 3. Порядок вычисления блоков в системе MathCAD

ВЫДЕЛЕНИЕ И РЕДАКТИРОВАНИЕ ОБЪЕКТОВ

Важной при редактировании математических выражений является возможность выделения их целиком или в виде отдельных объектов. По существу это означает замену одномерного маркера в виде синей вертикальной черты на маркер двумерный в виде выделяющего часть выражения прямоугольника (выделяющая рамка). Для увеличения размера выделяющей рамки пользуются клавишей **«Пробел»**. Для изменения ориентации выделяющей рамки используется клавиша **“Insert”.**

При создании формул пользуются следующим основным правилом:

⚠	*То, что заключено в выделяющую рамку, становится операндом для следующего вводимого знака операции.*

На рис. 4 показано как положение выделяющего маркера влияет на применение операции деления при составлении математического выражения.

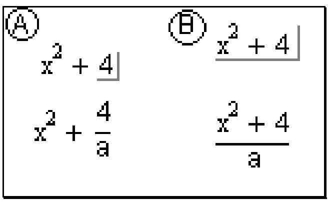


Рис. 4. Примеры ввода математических выражений

Вопросы для самоконтроля

1. Какие типы редакторов встроены в систему MathCAD?
2. В каком порядке происходит вычисление блоков в математическом выражении?
3. Что такое выделяющая рамка? Как ею пользоваться?
4. Когда вызывается графический редактор MathCAD?

Контрольные задания

Контрольные задания I уровня сложности

1. Запишите в документе MathCAD два математических выражения $e^x \cdot \sin(x^2)$ и $e^{x \cdot \sin(x^2)}$.Какие отличия наблюдаются при создании этих выражений?
2. При вычислении значения переменной t получена ошибка (рис. 5). Как ее исправить?

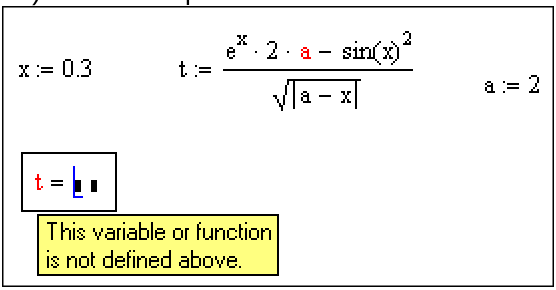


Рис. 5. Иллюстрация к контрольному заданию

Контрольные задания II уровня сложности

1. Вычислите значение выражения

$$\frac{\sqrt{\sqrt[5]{x^4}+\sqrt[5]{\left|x^{4-x}\right|}}+\ln\left|\sin\frac{\pi x}{x+2/3}\right|}{e^{|2\cdot x|}+tg\left(\left|\frac{x}{25}\right|\right)}.$$

1.4. Алфавит, константы и переменные системы

Алфавит входного языка системы определяет совокупность специальных знаков и слов, которые используются при задании команд. Алфавит системы MathCAD содержит:

- малые и большие латинские буквы;
- малые и большие греческие буквы;
- малые и большие буквы кириллицы;
- арабские цифры от 0 до 9;
- системные переменные;
- имена встроенных функций;
- операторы;
- спецзнаки.

К укрупненным элементам системы относятся типы данных, операторы, функции и управляющие структуры. К типам данных относятся числовые константы, обычные и системные переменные, массивы (векторы и матрицы) и данные файлового типа.

1.4.1. Операторы

Операторы – элементы языка, с помощью которых можно создавать математические выражения. К ним относятся, например, символы математических операций (+ - / * ^), знаки сравнений $(\leq \ \geq \ = \ \neq \ > \ <)$, знаки вычисления сумм, произведений, производной, интегралов и т.д.

Скалярные операторы предназначены для выполнения арифметических действий над скалярными величинами (табл.1).

Таблица 1

Скалярные и логические операторы

Оператор	Клавиши	Назначение оператора
X+Y	X+Y	Суммирование X с Y
X-Y	X-Y	Вычитание из X значения Y
$X \cdot Y$	X*Y	Умножение X на Y
$\frac{X}{Y}$	X/Y	Деление X на Y
X^{Y}	X^Y	Возведение X в степень Y
$\sqrt{X}$	\X	Вычисление квадратного корня из X
X!	X!	Вычисление факториала

Окончание табл. 1

Оператор	Клавиши	Назначение оператора
$\lvert X \rvert$	|X	Вычисление модуля числа X
X_n	X[n	Ввод нижнего индекса n
X**:=**Y	X:Y	Локальное присваивание X значения Y
$X \equiv Y$	X~Y	Глобальное присваивание X значения Y
X**=**	X=	Вывод значения переменной
X **=** Y (жирное равно)	X Ctrl =Y	Оператор сравнения X с Y или оператор приближенного равенства (тождества)
$X \rightarrow$		Вывод значения X в символьном виде

1.4.2. Константы

Константами *называются объекты, хранящие некоторые значения, которые не могут быть изменены в ходе вычислений.*

MathCAD работает с двумя видами констант: числовыми и строковыми. Числовые константы задаются своими числовыми значениями (к примеру, значения констант 0 и 1 есть соответственно ноль и единица). В системе MathCAD используются числовые константы, значения которых представляют собой числа определенного типа – десятичные, восьмеричные или шестнадцатеричные.

Числовые десятичные константы задаются с помощью арабских цифр, **десятичной точки** (а не запятой) и знака минус (-). Например:

123 –целочисленная десятичная константа;

12.6 –десятичная константа с дробной частью;

$1.25 \cdot 10^{-5}$ – десятичная константа с мантиссой (1.25) и порядком –5.

***Комплексные* числа**

Большинство вычислений система выполняет как с действительными, так и с комплексными числами, которые обычно представляются в алгебраическом виде

Z = ReZ+ i* ImZ или **Z** = ***ReZ***+ *j* ImZ.*

Здесь *ReZ* – действительная часть комплексного числа Z, *ImZ* – его мнимая часть, а символы *i* или *j* обозначают мнимую единицу, т.е. корень квадратный из –1. Такое представление характерно и для системы MathCAD. Итак, если *ReZ* = 2, а *ImZ* = 3, то комплексная числовая константа в системе MathCAD должна быть задана в виде 2+ i* 3 или 2+ *j* 3.*

Однако система не всегда знает, какой символ применить для обозначения мнимой единицы. Поэтому перед использованием любых операций с комплексными числами полезно вначале определить *i* или *j* как мнимую единицу (т.е. присвоить им значение квадратного корня из –1).

Строковые константы

При работе со строковыми константами следует заключить их в двойные английские кавычки, например:

"My_name" или "My first example". В строковую константу могут входить один или несколько символов либо слов.

1.4.3. Переменные

Переменные *являются поименованными объектами, имеющими некоторое значение, которое может изменяться по ходу выполнения вычислений.*

Имена констант, переменных и иных объектов называют идентификаторами. Тип переменной определяется ее значением; переменные могут быть числовыми, строковыми, символьными и т.д.

1.4.4. Правила именования объектов

В именах переменных, констант и функций допускается использование прописных и строчных латинских, прописных и строчных греческих букв, цифр от 0 до 9, специальных символов (% _ `).

На имена объектов накладываются следующие ограничения:

- имя объекта не должно начинаться с цифры или спецзнака;
- имя объекта не должно содержать пробелов;
- все символы в имени должны быть напечатаны шрифтом одной гарнитуры, размера и начертания;
- MathCAD не делает различий между именами переменных и именами функций. Поэтому, если определить в начале *f(x)*, а затем переменную *f*, окажется невозможным использовать *f(x)* где-либо ниже определения *f*;

- некоторые имена используются MathCAD для встроенных констант, единиц измерения и функций. Хотя эти имена можно переопределить, нужно иметь в виду, что в этом случае будут утеряны их встроенные значения. Например, если определить переменную с именем sin, то встроенная функция MathCAD sin(x) уже не сможет использоваться;
- MathCAD различает в именах символы верхнего и нижнего регистров, поэтому, например, dd и DD – это разные переменные. MathCAD различает также начертание шрифтов, поэтому DD и ***DD*** – также разные переменные.

Идентификаторы в системе MathCAD могут иметь практически любую длину, в них могут входить любые латинские и греческие буквы, а также цифры. Однако начинаться идентификатор может только с буквы, например: x, xl, alfa, X_coordinate. Кроме того, идентификатор не должен содержать пробелов. Некоторые спецсимволы (например, знак подчеркивания _) могут входить в состав идентификаторов, другие (например, знаки операторов арифметических действий) – недопустимы. Строчные и прописные буквы в идентификаторах различаются. Идентификаторы должны быть уникальными, т.е. они не могут совпадать с именами встроенных или определенных пользователем функций.

Для присвоения значения переменной используется знак присваивания **:=** (вводится знаком **:**).

Например:

a:=2 *z:=–8.97*

Использование неинициализированной переменной (т.е. не получившей первоначального значения) является ошибкой.

Переменная, проинициализированная с использованием глобального оператора присваивания ≡, может быть объявлена в любом месте листа, в том числе и после своего использования. Например:

$$c := 2 \cdot a$$

…………….

$$a \equiv 5$$

Виды переменных MathCAD

В MathCAD имеются переменные следующих видов: скалярные, ранжированные, индексированные и системные.

Скалярные – переменные, способные воспринять одно значение. Например : *b:= 5* *x:=cos(3.5)* *z:=2*x.*

***Системные переменные** – небольшая группа особых объектов, имеющих предопределенные системой начальные значения.*

В табл.2 указаны эти объекты и приведены их начальные значения (в скобках).

Необходимо отметить, что значения системных переменных, как и обычных, могут быть в дальнейшем изменены путем присвоения им новых значений. Например, значение переменной *е* можно изменить так, что эта переменная будет означать заряд электрона, а не основание натурального логарифма. Однако рекомендуется этого не делать во избежание двойного истолкования таких переменных.

Таблица 2

Системные переменные

Объект	Ввод	Назначение
π	Ctrl+ P	Число "пи" (3.14159..)
e	e	Основание натурального логарифма (2. 71..)
%	%	Процент (0. 01)
TOL		Погрешность численных методов (по умолчанию 0.001)
ORIGIN		Нижняя граница индексации массивов (по умолчанию 0)
PRNCOLWIDTH		Число десятичных знаков, используемых для записи численных данных в операторе **WRITEPRN** (8)
PRNPRECISSION		Число столбцов оператора **WRITEPRN (**4**)**
FRAME		Счетчик кадров при работе с анимационными рисунками (0)

Пять последних переменных вводятся набором их имен.

***Размерные переменные** – это переменные, которые помимо своего числового значения характеризуются еще и указанием физической величины.*

Для задания значения размерных переменных используются единицы измерения, выбранные из одной из категорий физических величин: Time (время), Length (длина), Mass (масса), Charge (заряд), Force(сила), Illuminosity (освещенность) и др.

Для создания размерной переменной необходимо указать ее имя, знак присваивания, задать ее числовую величину, затем указать знак *(умножить), в меню ***Insert/Unit*** выбрать категорию физической величины (***Demension***) и единицу измерения (***Unit***). Например:

$$dl := 14 \cdot km \qquad\qquad netto := 200 \cdot kg$$

Индексированные переменные *– это элементы векторов и матриц.*

Перед использованием таких переменных необходимо сначала создать массив значений с использованием панели инструментов ***Matrix***. Обращение к элементу вектора производится с использованием одного индекса; к элементу матрицы – с использованием двух индексов – номера строки и номера столбца (рис. 6).

$$\mathrm{ORIGIN} := 1$$

$$\mathrm{Vect} := \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix} \qquad \mathrm{Vect}_1 = 2$$

$$\mathrm{M} := \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \qquad \mathrm{M}_{1,2} = 2$$

Рис. 6. Способы обращения к элементам векторов и матриц

Более подробно работа с элементами массивов рассматривается в разд. 7 «Массивы, векторы и матрицы».

Глобальные переменные – это переменные, значения которым присваиваются в документе в первую очередь. Эти переменные видны и могут использоваться в любом месте документа MathCAD, вне зависимости от места их инициализации (инициализация – присвоение первоначального значения).

Присвоение значения глобальным переменным производится с использованием знака глобального присваивания ($\equiv$) в любом месте документа, необязательно перед первым использованием. Например,

$$\mathrm{x} := 0.3$$
$$\mathrm{K} := \mathrm{a} \cdot \mathrm{x}$$
$$\ldots\ldots$$
$$\mathrm{a} \equiv 5$$

Дискретные (ранжированные) переменные – особый класс переменных, который в системе MathCAD зачастую заменяет управляющие структуры, называемые циклами (однако полноценной такая замена не является).

> ***Дискретные (ранжированные) переменные*** – *переменные, имеющие ряд фиксированных значений (либо целочисленных, либо в виде чисел, изменяющихся от начального до конечного значения с определенным шагом).*

Дискретные переменные позволяют создавать упорядоченные ряды числовых данных, используются для табулирования функций (вычисления значений функции при различных значениях аргумента) и др.

Для создания дискретной переменной используется следующее выражение.

Name:=Nbegin,(Nbegin+Step)..Nend

Здесь ***Name*** – имя ранжированной переменной;

Nbegin,Nend – начальное и конечное значения переменной;

Step – шаг изменения переменной. Он может быть целым и дробным, положительным и отрицательным. Если шаг не указан, то он берется по умолчанию равным 1, если *Nbegin<Nend,* и –1, если *Nbegin>Nend;*

.. – символ, указывающий на изменение переменной в заданных пределах (вводится знаком “точки с запятой” (;)или пиктограммой  на панели инструментов **Matrix**).

Например

$$i := 0..5 \quad x := 0,0.5..10 \quad t := -10,-9.5..10$$

Дискретные переменные широко применяются для представления числовых значений функций в виде таблиц, а также для построения их графиков. Использование знака = после имени ранжированной переменой инициирует вывод таблицы значений этой переменной. Например,

$i =$

0
1
2
3
4
5

Полезно учитывать некоторые свойства таблиц вывода:

- если таблица вывода содержит много строк, то выводятся первые 16 строк и таблица вывода содержит полосу прокрутки;
- числа в таблицах можно задавать в требуемом формате с помощью операций задания формата чисел;
- при использовании в таблице единиц размерности все данные таблицы будут содержать единицы размерности (поделите результат с размерными переменными на размерность для указания ее только в заголовке таблицы вывода).

Применение дискретных переменных

Важно отметить что, в сущности, задание дискретных переменных эквивалентно заданию конечных циклов. Сами дискретные переменные являются векторами, что видно из выдачи их значений (столбец со всеми значениями переменных). Это означает, что объем памяти, занимаемый такими переменными, больше занимаемого обычными переменными.

Индексированные переменные могут применяться в последующих формульных блоках. Однако в этих блоках необходимо соблюдать соответствие результатов (конечных и промежуточных) векторному типу этих переменных.

Привыкшие к обычному программированию пользователи часто забывают, что дискретная переменная – вектор. Поэтому они пытаются выполнять с такими операциями действия, корректные лишь для обычных (скалярных) переменных. Например, задают выражение вроде f: =i* 2, используя обычную переменную f, что приведет к явной ошибке – система укажет (красным цветом), что f не соответствует векторному типу. Однако если использовать выражение, например, вида $f_i = i^2$, то будет получен новый вектор с именем f, элементы которого в нашем случае являются квадратами значений элементов вектора i. Более подробно особенности задания и применения векторов рассматриваются далее.

!	*Дискретные переменные широко применяются при построении графиков.*

Вопросы для самоконтроля

1. Что такое константа MathCAD? Приведите примеры.
2. Что такое переменная MathCAD? Дайте определение.

3. Какие типы переменных существуют в MathCAD?
4. Укажите правила использования дискретных переменных.
5. Укажите правила использования индексированных переменных.
6. Чем скалярная переменная отличается от размерной? Как при задании значения переменной добавить единицы измерения?
7. Укажите назначение глобальных переменных.
8. Что такое таблица вывода? Какими свойствами она обладает?
9. Как задать нумерацию элементов вектора, начиная с единицы?

Контрольные задания

1. Вычислите значение выражения $\cos(\pi\, x)$. Как задать значение числа π ?
2. Дан вектор $V := \begin{pmatrix} 2 \\ 3 \\ -6 \end{pmatrix}$. Как обратиться ко второй компоненте вектора?
3. Дан вектор $M := (-5 \quad 6 \quad 0)$. Как обратиться ко второй компоненте вектора?
4. Дана матрица $A := \begin{pmatrix} 1 & 2 \\ 0 & 5 \end{pmatrix}$. Присвойте переменной t значение элемента матрицы A, равного 0.
5. Определите в документе MathCAD переменную $y \in [-5;10] \quad \Delta x = 0.4$.
6. Определите в документе MathCAD переменные t со значением 10 миллисекунд и k со значением 500 килокалорий.

1.5. Работа с функциями

1.5.1. Встроенные функции

Система MathCAD содержит расширенный набор встроенных элементарных функций.

Общий синтаксис функций MathCAD:

Имя_функции (аргумент1, аргумент2, …)

Функции задаются своим именем и значением аргумента в круглых скобках. В ответ на обращения к ним функции возвращают вычисленные значения. Аргумент и значение функции могут быть действительными или комплексными числами.

Все функции MathCAD собраны в пункте меню ***Insert/Function*** и разделены на несколько категорий. Наиболее часто используемыми категориями являются:

- **All** – все функции;
- **Hyperbolic** – гиперболические функции;
- **Log and Exponential** – логарифмические и экспоненциальные функции;
- **Solving** – функции решения уравнений и систем уравнений;
- **Trigonometric** – тригонометрические функции;
- **Vector and Matrix** – функции работы с векторами и матрицами.

Перечень математических и тригонометрических функций см. в приложении 2.

Существует также ряд встроенных векторных и матричных функций. Они облегчают решение задач линейной алгебры и других сфер приложения векторов и матриц. Перечень функций, обрабатывающих векторы и матрицы, см. в приложении 3.

1.5.2. Задание функций пользователя

Несмотря на довольно широкий набор встроенных функций, часто возникает необходимость расширить систему новыми функциями, необходимыми для решения конкретной предметной задачи. Функции пользователя вводятся с применением следующих синтаксических правил.

Имя_функции (Список_параметров): = Выражение

Имя функции задается как любой идентификатор с соблюдением правил именования объектов (см. разд. 5.4). В скобках указывается список параметров функции. Это перечень используемых в выражении переменных, разделяемых запятыми.

Выражение – любое выражение, содержащее доступные системе операторы и функции.

Следует отметить особый статус переменных, указанных в списке аргументов функции пользователя. Эти переменные являются ***локальными***, поэтому они могут не определяться до задания тела функции. Фактически их указание в списке аргументов функции и является их объявлением.

Естественно, что локальные переменные могут использоваться только в выражении, описывающем функцию. Их имена могут совпадать с именами глобальных переменных, введенных ранее (рис. 7). Но при выходе из блока задания функции работа будет продолжаться с глобальными переменными.

$$x := 15$$

$$f(x) := 5 \cdot \sin\left(x^2\right)$$

$f(0.3) = 0.449$ ← Вычисление функции при x= 0.3

$$x = 15$$

Рис. 7. Примеры задания функций пользователя

1.5.3. Анализ условий. Функция if

Функция if полезна для управления ветвлениями и создания более сложных вычислительных алгоритмов.

Функция if используется для выбора одного из двух значений, определяемого условием.

Синтаксис:

$$if\,(< условие >, < true\_значение >, < false\_значение >)$$

Функция if возвращает $< true\_значение >$, если $< условие >$ отлично от нуля, и возвращает $< false\_значение >$, если условие равно нулю.

Хотя $< условие >$ может быть любым выражением, удобнее использовать какое-либо логическое выражение, т.е. выражение, содержащее знаки сравнения.

Чтобы сохранить время, MathCAD вычисляет только те выражения, которые необходимы. Например, если $< условие >$ есть ложь, нет нужды вычислять $<true\_значение>$, так как его значение не будет возвращено. На рис. 8 график построен с использованием функции if.

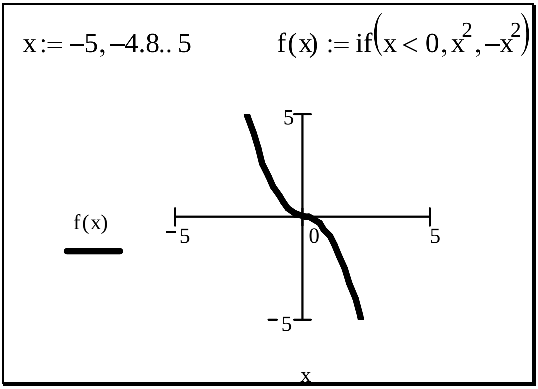


Рис. 8. Пример использования функции if

Для создания нелинейных алгоритмов можно также воспользоваться панелью инструментов «Программирование» (рис. 9).

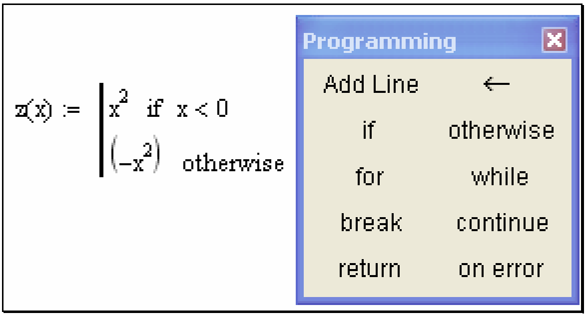


Рис. 9. Пример задания ветвящегося алгоритма с использованием панели инструментов «Программирование»

⚠	*Для анализа более сложных условий, логические выражения могут быть объединены с помощью знаков “**+**” и “*****”.*

Знак “*” обозначает операцию конъюнкции (логическое И). К аналогичному результату приводит использование пиктограммы ∧ на панели инструментов ***Boolean*** (рис. 10).

Знак “+” обозначает операцию дизъюнкции (логическое ИЛИ).

К аналогичному результату приводит использование пиктограммы $\boxed{\vee}$ на панели инструментов ***Boolean*** .

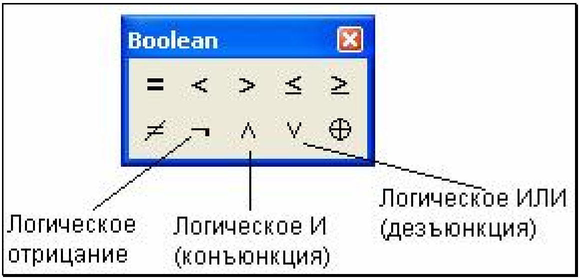


Рис. 10. Панель операций отношения

Например, составное логическое выражение $(x > 0) \cdot (y > 0)$ на рис. 11 возвращает не ноль (истину), только если истинны оба условия. Аналогично выражение $(x > 0) + (y > 0)$ действует подобно логическому “или”, возвращающему “истину”, если или $(x > 0)$, или $(y > 0)$.

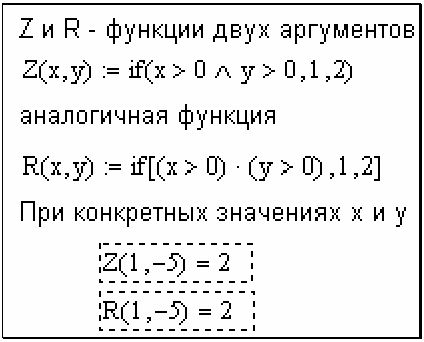


Рис. 11. Пример объединения логических выражений

Импульсные функции

Функцию if удобно применять для построения ступенчатых функций (рис.12).

Объявим функцию, способную возвращать одно из двух значений 0 или 1.

$F(x) := if(x < 0, 0, 1)$

$F(x) := if(x < 0, 0, 1)$

$pulse(x, w) := F(x) - F(x - w)$

Определим x на участке (-20;20)

$x := -20, -19.9 .. 20$

Ширина импульса $w := 2$

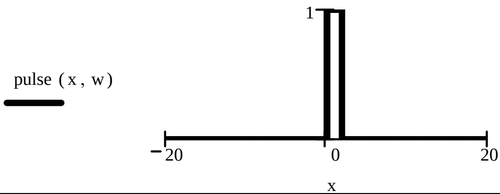


Рис. 12. Пример построения импульсной функции

Эта функция может быть использована для создания импульса шириной w.

Вопросы для самоконтроля

1. Что такое функция MathCAD? Как осуществляется вызов встроенной функции?
2. Какая команда главного меню MathCAD позволяет увидеть полный перечень встроенных функций?
3. С какими категориями функций можно работать в документе MathCAD?
4. Каким символом отделяются друг от друга аргументы функций MathCAD?
5. Как задаются пользовательские функции в документе MathCAD?
6. Каким образом в документе MathCAD можно реализовать нелинейный (ветвящийся) алгоритм вычислений?
7. Каким образом в документе MathCAD можно задать логические операции И, ИЛИ?

Контрольные задания

Задания I уровня сложности

1. Вычислите значение выражения при x=0.3; 0.5; 0.7.

$$k = \frac{\mathrm{arctg}(x^2) + \sinh(x)}{\mathrm{ctg}(x) + 2}.$$

2. Вычислите в документе MathCAD значение выражения

$$t = \sqrt[3]{\left|\frac{\ln(x^2+1)+e^{\sin(x)}}{4+\left|tg(x/25)\right|}\right|}.$$

3. Задайте исходные данные для построения изображения поверхности $Z = \sin(x^2 + y^2)^{1.1}$ на участках $x \in [0; 20]$ и $y \in [0; 20]$.

Задания II уровня сложности

4. Значение переменной k зависит от значения переменной d. Присвойте k значение d^2, если d – отрицательное число, и $d-5$, если неотрицательное.
5. Задайте исходные данные для построения графика кусочной функции $f(x) = \begin{cases} \cos^2 x, & \text{если } x \in [-5; 2]; \\ -x^2, \text{иначе.} \end{cases}$

1.6. Построение графиков

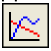
Для создания графиков в системе MathCAD имеется программный графический редактор. Он начинает свою работу при вставке на рабочий лист одного из шаблонов графиков: графика в декартовой системе координат, в полярной системе координат, графика трехмерной поверхности, графиков уровней и т.д.

1.6.1. Построение двумерных графиков

Для построения точки в двумерном пространстве необходимо знать две координаты: координату *x* – абсциссу и координату *y* – ординату. Точки соединяются друг с другом разнообразными линиями (сплошной, пунктирной и т.д.), могут быть показаны узловые точки графиков в виде маркеров (точек, квадратиков, кружков и т.д.). Возможно построение на одном графике нескольких кривых.

Порядок построения двумерного графика

1. Задать изменение аргумента функции на заданном интервале в виде дискретной переменной.

2. Определить функцию, график которой необходимо построить.
3. Установить курсор правее или ниже объявлений функций и дать команду ***Insert/Graph/X-Y Plot*** или нажать пиктограмму  на панели инструментов “Graph”.
4. Заполнить шаблон графика, указывая по оси ОХ название аргумента; по оси ОУ – полное имя функции.

Чтобы в одном шаблоне разместить насколько графиков, надо, набрав на оси ординат имя первой функции, нажать клавишу запятая (уголок курсора при этом обязательно должен находиться в конце имени функции) и ввести имя второй функции (рис. 13).

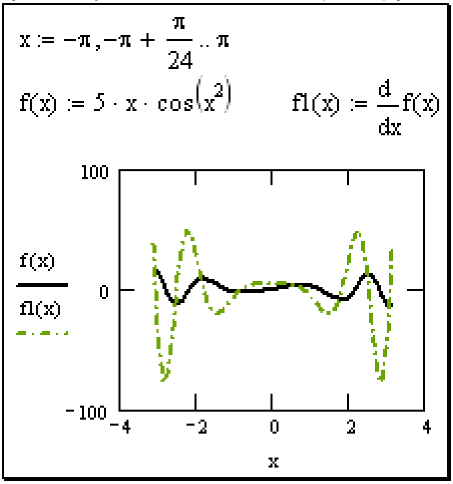


Рис. 13. Пример построения графиков функции и её первой производной на одном изображении в декартовой системе координат

Простые функции, если они в дальнейшем не используются, можно задать математическим выражением непосредственно в шаблоне графика (рис.14).

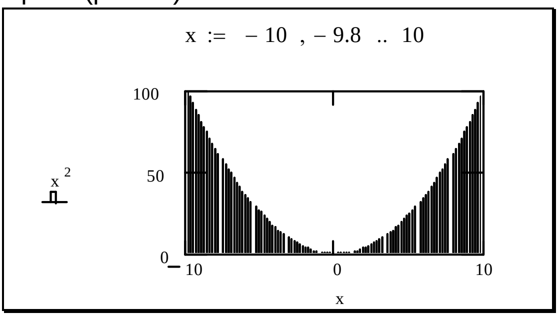


Рис. 14. Определение вида зависимости непосредственно в шаблоне графика

Построение графиков параметрически заданных функций

Отличие функций, заданных в каноническом виде, от функций, заданных в параметрическом виде, заключается в том, что при канонической записи значения ординаты имеют прямую функциональную зависимость от значений абсциссы, например, $y(x) = x^2$.

Для параметрически заданных функций обе координаты и абсцисса, и ордината выражаются через некоторый параметр. Например, $\begin{cases} x(t) = 1 - t^2; \\ y(t) = 1 - t^3. \end{cases}$

Этот параметр изменяется в границах некоторого диапазона с определенным шагом. При каждом значении параметра вычисляются значения координат X и Y.

Чаще всего параметрическая запись функции используется для описания важнейших алгебраических кривых, таких, как эллипсов, циклоид, спиралей и т.п. (рис. 15).

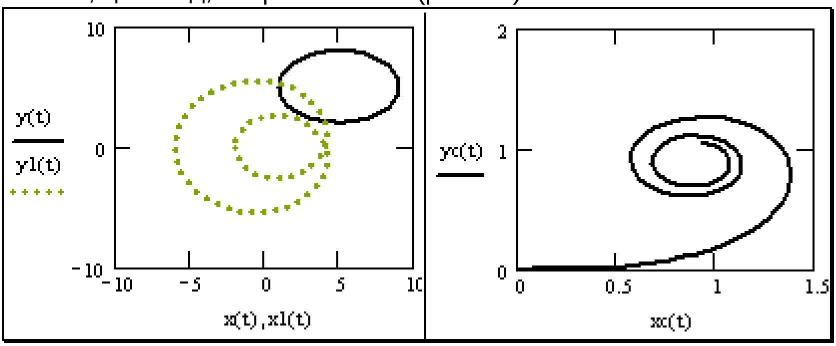
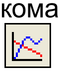


Рис. 15. Примеры графиков параметрически заданных функций

Порядок построения графика параметрически заданной функции

1. Задать изменение параметра функции t на заданном интервале в виде дискретной переменной.
2. Определить две функции, описывающие изменение абсциссы X(t) и ординаты Y(t).
3. Установить курсор правее или ниже объявлений функций и дать команду ***Insert/Graph/X-Y Plot*** или нажать пиктограмму  на панели инструментов “Graph”***.***
4. Заполнить шаблон графика, указывая название функции X(t) по оси OX и функции Y(t) по оси OY (рис. 16).

$$a := 1 \qquad b := 6 \qquad \lambda := 3 \qquad \psi := 0, 0.05 .. 2 \cdot \pi$$

$$x(\psi) := (b + a) \cdot \cos(\psi) - \lambda \cdot a \cdot \cos\left(\psi \cdot \frac{b + a}{a}\right)$$

$$y(\psi) := (b + a) \cdot \sin(\psi) - \lambda \cdot a \cdot \sin\left(\psi \cdot \frac{b + a}{a}\right)$$

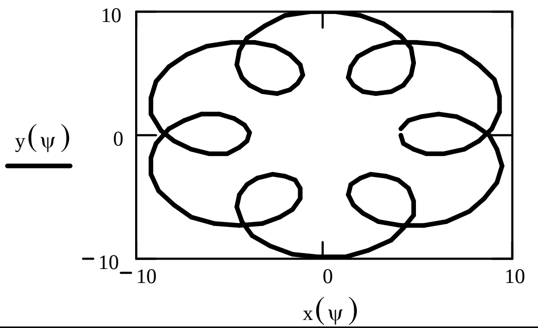


Рис. 16. Построение графика параметрически заданной функции

Форматирование двумерного графика

Если необходимо внести изменения в построенный график, сделать это можно, дав команду ***Format/Graph/X-Y Plot,*** предварительно выделив график.

Диалоговое окно ***”Formatting Currently Selected X-Y Plot”*** («Форматирование текущего выделенного графика») содержит четыре вкладки:

1) Управление опциями осей (***X-Y Axes).***

   Позволяет установить:
   - логарифмический масштаб (*Log Scale*);
   - линии сетки (*Grid Lines*);
   - установить цифровые данные по осям (*Numbered*/Пронумеровать);
   - позволяет выбрать вид координатных осей (*Axes Style*): *Boxed* (рамка), *Crossed* (перпендикулярные оси), *None* (нет осей).

2) Вкладка ***Traces*** служит для управления отображением линий графика. Позволяет установить:
   - имя кривой (*Legend Label*);

- выбрать вид маркера (*Symbol*);
- установить тип линии (сплошная, пунктирная и др.) (*Line*);
- цвет линии (Color);
- тип графика (*Type*) (линией, точками, гистограмма, ступенчатый);
- толщину (*Weight*) линии.

3) Вкладка ***Labels*** (Надписи) позволяет вводить в график дополнительные надписи:
   - заголовок (*Title*);
   - *подписи к осям (Axis* labels);
   - позволяет указать, где поместить заголовок графика: сверху (*Above*), снизу (*Below*). Флажок *Show Ti*tle включает/отключает отображение заголовка.
4) Вкладка ***Defaults*** служит для следующих целей:
   - *Change to Defaults* (вернуть значения по умолчанию) – отображает график с опциями, выбранными по умолчанию;
   - Use *for Defaults* – использует установки текущего чертежа для значений по умолчанию.

Для детального просмотра выделенного графика используется команда ***Format/Graph/Zoom***. На графике с помощью мыши выделяется рамкой тот участок графика, который необходимо рассмотреть более подробно, и нажимается кнопка Zoom.

1.6.2. Построение графика в полярной системе координат

В полярной системе координат каждая точка задается углом ψ и радиусом, длина которого зависит от ψ – $R(\psi)$.

График функции обычно строится при изменении угла ψ в определенных пределах, чаще всего от 0 до π.

Для построения графика в полярной системе координат необходимо задать

1. Изменение углового аргумента функции в виде дискретной переменной ψ в заданных пределах.
2. Определить функцию изменения радиуса $R(\psi)$.
3. Дать команду ***Insert/Graph/ Polar Plot.***
4. Заполнить шаблон графика.

Между декартовыми и полярными координатами существует зависимость (рис. 17)

$$X = R(\Psi)*\cos(\Psi) \quad и \quad Y = R(\Psi)*\sin(\Psi)\,.$$

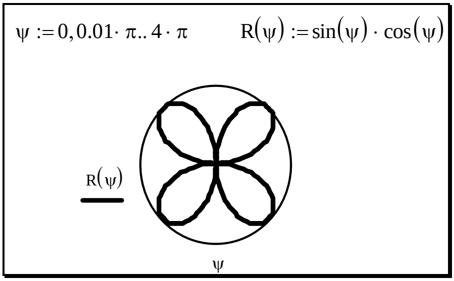


Рис. 17. Построение графика в полярной системе координат

Форматирование графика, заданного в полярной системе координат, аналогично форматированию графика в ДСК.

1.6.3. Масштабирование и трассировка готового графика

Для увеличения масштаба графика необходимо выполнить следующие действия:

1. Поместите курсор в поле графика и вызовите диалоговое окно “X-Y Zoom”, нажав правую кнопку мыши или использовав пиктограмму на панели ***“Graph”*** (рис. 18).

Рис. 18. Панель инструментов для построения графиков

2. .Обведите пунктирной линией при нажатой левой кнопке мыши область графика, которую хотите увеличить.
3. В открытом окне X-Y Zoom выберите Zoom. Выделенная область увеличится до размеров всей рамки графика (рис. 19).
4. Если необходимо вернуться к исходному виду графика, нажмите Unzoom или Full View (Отменить увеличение масштаба). Если надо оставить увеличенное изображение, нажмите ОК.

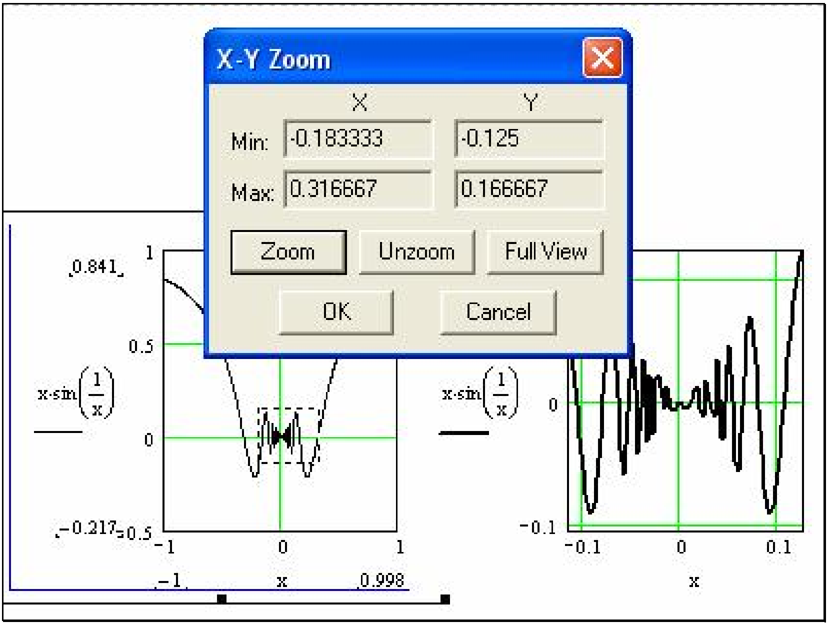


Рис. 19. Масштабирование графика

Для определения координат нужной точки на графике можно воспользоваться пиктограммой **“Trace”**  (трассировка) на панели “Graph” (рис. 20). С помощью кнопок “Copy X” и “Copy Y” можно сохранить полученные значения координат в буфере обмена и присвоить их некоторым переменным для дальнейшего использования.

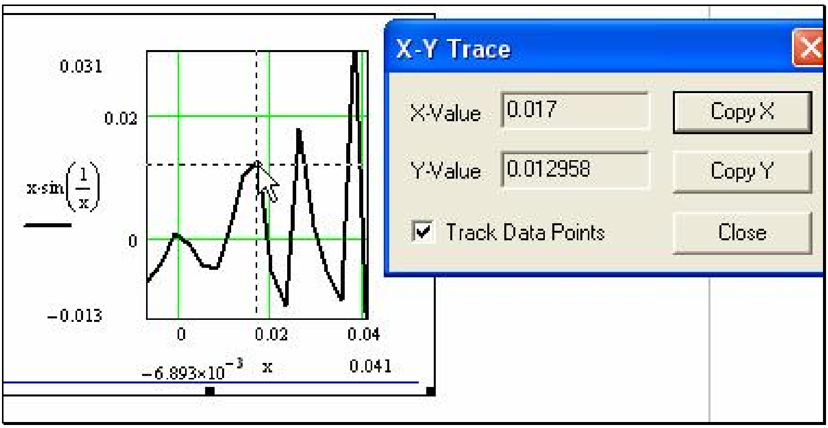


Рис. 20. Определение координат точки с использованием трассировки

1.6.4. Построение поверхностей

Трехмерная поверхность есть функция двух переменных. Для построения одной точки по оси z необходимо знать две координаты: x и y, а так же зависимость z(x,y).

Построение графика поверхности ускоренным путем (заданием функции)

Самым простым способом построения поверхности является задание функции двух аргументов и заполнение шаблона трёхмерного графика вводом имени функции (рис.21).

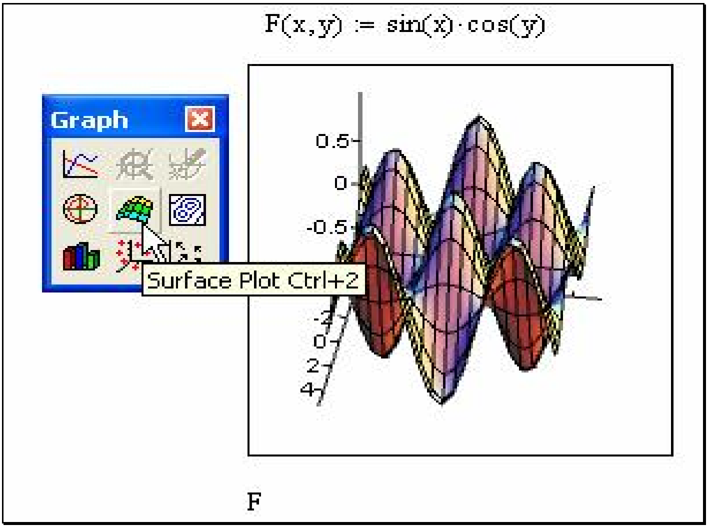


Рис. 21. Построение трёхмерного графика заданием функции

При этом изменение аргументов x и y MathCAD определяет сам. Если необходимо самостоятельно определить количество точек по осям OX и OY и законы расчета аргументов функции, можно воспользоваться следующим алгоритмом.

Построение графика поверхности путем создания массива данных

1. Задаем количество точек по оси OX $i := 0...20$.
2. По оси OY $j := 0..20$.
3. Задаем значение координат x и y в зависимости от их номера. Например:

$x_i := -1.5 + 0.15 * i$, $y_j := -1.5 + 0.15 * j$.

Необходимо учесть, что x и y являются дискретными переменными, поэтому в подстрочнике они имеют индекс, указывающий порядковый номер переменной. Подстрочники можно ввести последовательным нажатием клавиш $x[i$.

4. Задаем вид функции z(x,y).
5. Задаем матрицу ординат M.
6. Задаем команду Insert/Graph/Surface Plot (рис.22, 23).

Например

$$i := 0 .. 20 \qquad\qquad j := 0 .. 20$$

$$x_i := -1.5 + 0.15 \cdot i \qquad\qquad y_j := -1.5 + 0.15 \cdot j$$

$$z(x, y) := \sin\left(x^2 + y^2\right) \qquad\qquad M_{i,j} := z\left(x_i, y_j\right)$$

Рис. 22. Построение трёхмерного графика заданием массива данных

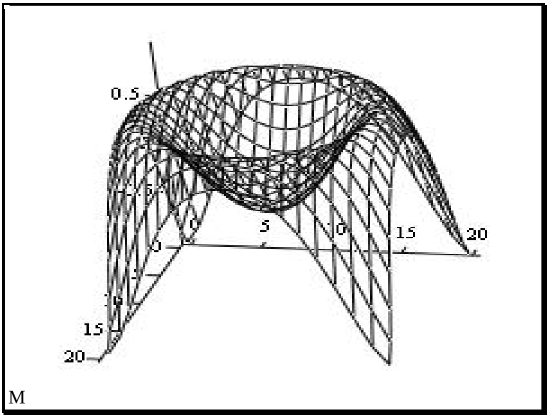


Рис. 23. Построение трёхмерного графика

Размерность матрицы ординат M будет равна произведению m*n, где m – количество точек по оси X, n – количество точек по оси Y

Форматирование трехмерных графиков

Наглядность представления трехмерных поверхностей зависит от множества факторов: масштаба построений, углов поворота фигуры, применения алгоритма удаления невидимых линий, использование функциональной закраски и т.д. Для изменения этих параметров используется диалоговое окно “***3D Plot Format”,*** вызываемое двойным щелчком мыши по графику или командой ***Format/Graph/3D Plot*** при выделенном графике.

В MathCAD Professional 2000 диалоговое окно “***3D Plot Format”*** имеет следующие опции:

1) На вкладке ***General*** можно задать:
   - В группе ***View:***
     - *Rotation* (вращение) –задание угла поворота;
     - *Tilt* (Наклон) –задание угла наклона;
     - *Zoom* – задание масштаба в долях единицы.
   - Группа ***Axes Style*** (стили осей) задает тип отображения осей*: Perimeter* – по периметру; *Corner* – в углу; *None* – без вывода осей.
   - В группе ***Frames*** задается вид рамки: в виде ящика (*Show Box*) или в виде рамки (*Show Border*).
   - Группа ***Display As*** (Отобразить как) служит для задания общего вида 3D графиков:
     - *Surface Plot* – в виде 3D поверхности;
     - *Contour Plot* – в виде линий равного уровня;
     - *Data Points* – в виде отдельных точек в 3D пространстве;
     - *Vector Field Plot* – в виде векторных полей;
     - *Bar Plot* – в виде трехмерных гистограмм;
     - *Patch Plot* – в виде поверхности, состоящей из небольших плоских фигурок, расположенных в узловых точках.
2) На вкладке ***Axes*** (оси) для каждой из трех осей (OX,OY,OZ) можно установить следующие опции:
   - Параметры сетки: *Auto Grid* (автосетка) – автоматический выбор числа линий; цвет (*Line Color*) и толщину (*Line Weight*) линий сетки.
   - Параметры осей: цвет (*Axis Color*), толщину осей (*Axis Weight*) и оцифровку линий (*Show Numbers).*
   - В группе **Axis Limits** можно либо автоматически выявить минимальное и максимальное значение координат (*Auto Scale*), либо установить эти значения самостоятельно (*Minimum Value, Maximum Value*).
3) На вкладке ***BackPlanes*** задаются цвета заднего плана, т.е. плоскостей XOY, XOZ, YOZ: *Fill BackPlans* (заливка заднего плана), *Color* (цвет заднего плана) и рамка каждой из плоскостей (*BackPlane Border*). Можно также задать цвет и толщину вспомогательных линий сетки (*Sub-Grids*).
4) На вкладке ***Title*** задается название графика (*Graph Title*) и место вывода названия: вверху (*Above*), внизу (*Below*), не выводить (*Hide*).

5) На вкладке ***Appearance*** можно задать цветовое оформление графика: опции заливки (*Fill Options*), опции линий (*Line Options*), опции точек(*Point Options*).
  - В группе **Fill Options** можно переключиться между цветной поверхностью (*Fill Surface*), цветными линиями уровня (*Fill Contours*) и отсутствием заливки (*No Fill*).
  - В группе ***Line Options*** производится переключение между видом линий графика: в виде сетки (*Wire frame*), в виде линий уровня (*Contour Lines*), отсутствием линий (*No Lines*). Здесь же можно изменить толщину линии (*Weight*).
  - В группе ***Point Options*** можно указать, будут ли на графике использоваться цветные точки (*Draw Points*), выбрать вид маркера (*Symbol*) и размер точки (*Size*).
  - В группах **Color Options** и для графика в целом, и для линий, и для точек производится выбор между цветовой картой (*Colormap*) и оформлением единым цветом (*Solid Color*).

6) На вкладке ***Lighting*** выбирается тип освещения графика: указывается с какой стороны и каким цветом освещен график.
7) На вкладке ***Advanced*** можно указать дополнительные настройки, например, изменение размера изображения по вертикали (*Vertical Scale*) или в перспективе (*Perspective*), выбрать цветовую карту (*Choose Colormap*), усилить цвет по одной из осей (*Increasing X,Y,Z*) и т.д.

Вопросы для самопроверки

1. Какая панель инструментов предназначена для добавления в документ MathCAD шаблонов графиков?
2. Какие типы изображений можно получить в документе MathCAD?
3. Укажите порядок построения графика в декартовой системе координат.
4. Как построить на одном изображении несколько графиков функций?
5. Каким образом можно изменить цвет и толщину линий графика? Изменить тип осей?
6. Укажите порядок построения графика в полярной системе координат.
7. Каким образом можно промасштабировать график?
8. Каким образом можно определить координаты определенной точки графика?

9. Укажите порядок построения графика параметрически заданной функции.
10. Укажите способы и порядок построения трёхмерных графиков.
11. Укажите приемы форматирования трёхмерных графиков.

Контрольные задания

Задания I уровня сложности

1. Постройте в ДСК график функции «Локон Аньези» $y = \dfrac{a^3}{(a^2 + x^2)}$, где a – коэффициент, $x \in [-6;\, 6] \quad \Delta x = 0.1$.
2. Постройте график функции в полярной системе координат $\psi(\alpha) = \sin(3\alpha); \quad \alpha \in [0; 2\pi]; \quad \Delta\alpha = \dfrac{\pi}{30}$.
3. Постройте график параметрически заданной функции $\begin{cases} x(t) = t \cdot \sin(t); \\ y(t) = t \cdot \cos(t), \end{cases} \quad t \cup [-\pi;\, \pi] \quad \Delta t = \dfrac{\pi}{24}$.
4. Самостоятельно постройте график поверхности заданием вида функции $f(x, y) = \sin\left(\dfrac{xy}{5}\right)$.
5. На участке $x \in [-3;\, 2], \quad \Delta x = 0.1$ постройте график функции $z(x) = \dfrac{\sin(x^3)}{3}$. Используя трассировку приближенно определите корни уравнения $\dfrac{\sin(x^3)}{3} = 0$.

Задания II уровня сложности

1. Постройте на одном изображении кривые функций, заданных математическим выражением $x \cdot \sin(x)$ и $\dfrac{\cos^3(x)}{3}$ на участке [-5; 5].

2. Постройте график параметрически заданной функции $\begin{cases} x(\beta) = (a+b)\cos\beta - a\cos((a+b)\beta / a); \\ y(\beta) = (a+b)\sin\beta - a\sin((a+b)\beta), \end{cases}$ где a,b – коэффициенты; $\beta \in [-\pi; 2\pi]$; $\Delta\beta = \dfrac{\pi}{36}$.
3. Постройте на одном изображении графики функции $g(x) = x^2 \cdot \sin(x) \cdot \cos\left(\dfrac{x}{2}\right)$, первой $g'(x)$ и второй $g''(x)$ производных.
4. Постройте на участке [–10;0] график функции $f(x) = x^2 \cdot \cos(x)$. Промасштабируйте график на участке [-2; 0]. С помощью трассировки определите координаты локальных минимумов кривой. Сохраните отдельные переменные координаты глобального максимума.
5. На участке [-10; 10] постройте график кусочной функции $$w(t) = \begin{cases} e^t, \textit{ если } t \in [-3;3]; \\ 3 \cdot \sin^2(t) + tg\left(\dfrac{t}{25}\right), \textit{ иначе.} \end{cases}$$
6. Самостоятельно постройте график поверхности функции заданием массива данных $z(x, y) = e^{\sin(x+y)}$ в 30-ти точках по осям OX и OY.

1.7. Массивы, векторы и матрицы

1.7.1. Понятие о массивах

Важным типом данных в системе MathCAD являются массивы.

> ***Массив*** *– это имеющая уникальное имя совокупность конечного числа числовых или символьных элементов, упорядоченных заданным образом и имеющих определенные адреса.*

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

1.7.2. Индексация элементов массивов

Порядковый номер элемента, который является его адресом, называется индексом. Напоминаем, что нижняя граница индексации массивов задается значением системной переменной **ORIGIN,** которая может принимать значение 0 или 1.

Для обращения к элементу массива достаточно в виде подстрочного индекса указать индекс(ы) элемента. Например, если одномерный массив имеет имя V, то его элементами при **ORIGIN=0** будут индексированные переменные

$V_0, V_1, V_{2,...}$

По умолчанию ORIGIN=0, т.е. первый элемент вектора, первая строка и первый столбец матрицы имеют индекс ноль. Чтобы изменить нумерацию индексов, в первой строке документа наберите заглавными буквами ORIGIN:=1. Можно переопределить встроенную переменную ORIGIN, выбрав в главном меню Math Option – Built–in–variable – ORIGIN, набрать индекс первого члена массива, как правило, 1. Переменной ORIGIN можно задавать разные значения, включая отрицательные.

Векторы могут быть двух типов: векторы-строки и векторы-столбцы. Матрица может рассматриваться как совокупность ряда векторов одинаковой длины.

1.7.3. Ввод элементов векторов и матриц

Для задания векторов и матриц можно воспользоваться операцией ***View/Toolbars/Matrix...*** **(Матрицы)**, нажав клавиши ***Ctrl+M*** или введя пиктограмму с изображением шаблона матрицы на панели инструментов ***Math***. Это вызывает появление диалогового окна, в котором надо указать размерность матрицы, т.е. количество строк *m* и столбцов *n.* Для векторов один из этих параметров должен быть равен 1. При m=1 получим вектор-столбец, а при n=1 – вектор-строку.

Матрица является двумерным массивом с числом элементов *mxn.* Элементы векторов и матриц размещаются между большими скобками (рис.24).

Рис. 24. Задание размерности матрицы

Для обращения к элементу вектора-столбца достаточно ***одного*** *индекса.*
Вектор-строка – это матрица, состоящая из одной строки и нескольких столбцов, поэтому для обращения к элементу ***вектора-строки*** *так же, как и для обращения к элементу матрицы, используются* ***два*** *индекса – номер строки и номер столбца (рис. 25).*

Для указания подстрочных индексов после имени переменной вводится знак открывающей квадратной скобки или используется пиктограмма x_n на панели инструментов ***Matrix***.

$$Vect := \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix} \qquad Vect_1 = -1$$

$$S := \begin{pmatrix} 1 & 0.3 & 7 & -9 \end{pmatrix} \qquad S_{0,1} = 0.3$$

$$M := \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \qquad M_{0,1} = 2$$

Рис. 25. Правила обращения к элементам векторов и матриц

Для элементов матрицы подстрочные индексы вводятся с разделением их запятыми. Индексы могут иметь только целочисленные значения.

По умолчанию элементы векторов и матриц нумеруются с нуля (рис. 26).

Эту ситуацию можно изменить, установив другое значение системной переменной **ORIGIN.** Например,

$$ORIGIN := 1$$

$$R := \begin{pmatrix} 2.36 & 8 & 1 \\ -5 & 2 & 0 \\ 3 & 1.2 & -9.4 \end{pmatrix} \qquad R_{2,1} = -5$$

Рис. 26. Задание нижней границы нумерации элементов векторов и матриц

1.7.4. Обработка векторов и матриц

Для работы с матрицами используется панель инструментов Matrix, вызываемая нажатием соответствующей пиктограммы на панели инструментов Math.

На рис. 27 указаны назначения пиктограмм панели инструментов ***Matrix***.

Большинство приведенных операторов достаточно известны из математического аппарата матричных вычислений.

Рис. 27. Описание операторов работы с векторами и матрицами

Под необычным для нашей математической литературы понятием «векторизация» подразумевается одновременное проведение математических операций в их скалярном значении над всеми элементами вектора или матрицы. Это можно понимать и как возможность параллельных вычислений.

Векторизация может изменить смысл математических выражений и даже превратить недопустимое выражение во вполне допустимое. Например, если V – вектор, то выражение **cos(V)** будет недопустимым, поскольку аргументом функции **cos** может быть только скалярная величина или переменная. Однако со знаком векторизации функция **cos(V)** возвращает вектор, каждый элемент которого есть косинус значения соответствующего исходного вектора V.

С помощью оператора векторизации выполняются поэлементные действия с матрицами. Обычно сначала задается некоторая функция, описывающая действия, которые необходимо выполнить с каждым элементом матрицы. Затем выполняется оператор векторизации.

На рис.28 аргумент функции x обозначает отдельный элемент матрицы. Оператор векторизации, использующий функцию z, возводит каждый элемент матрицы в квадрат.

Таким образом, оператор векторизации заметно упрощает запись математических алгоритмов, особенно для обеспечения параллельных вычислений. Впрочем, параллельность относится не к самим вычислениям, а лишь к их алгоритмической записи.

$$C := \begin{pmatrix} 1 & -2 & 4 \\ -2 & -2 & -2 \\ 16 & 4 & 25 \end{pmatrix} \qquad z(x) := x^2 \qquad \overrightarrow{z(C)} = \begin{pmatrix} 1 & 4 & 16 \\ 4 & 4 & 4 \\ 256 & 16 & 625 \end{pmatrix}$$

$$k(x) := \mathrm{if}\left(x < 0, x^2, \sqrt{x}\right) \qquad \overrightarrow{k(C)} = \begin{pmatrix} 1 & 4 & 2 \\ 4 & 4 & 4 \\ 4 & 2 & 5 \end{pmatrix}$$

Рис. 28. Применение оператора векторизации

Функции, обрабатывающие векторы и матрицы, собраны в категории ***Vector and Matrix*** диалогового окна ***Insert function.*** Полный перечень этих функций приведен в приложении 3.

Пример. Заполнить матрицу M случайными числами, а матрицу A по правилу $A_{i,j} = \begin{cases} 1/i, & \text{если } i > 1; \\ j^2, & \text{иначе.} \end{cases}$.

На рис. 29 приведен пример работы функции matrix, которая предназначена для заполнения матрицы указанной размерности по заданному правилу.

$$d(i,j) := \mathrm{rnd}(1) \cdot 10$$

$$M := \mathrm{matrix}(4,4,d) \qquad M = \begin{pmatrix} 8.376 & 4.849 & 7.437 & 4.58 \\ 7.444 & 5.99 & 7.35 & 5.724 \\ 1.516 & 4.252 & 5.171 & 7.515 \\ 1.69 & 4.919 & 6.998 & 1.475 \end{pmatrix}$$

$$f(i,j) := \mathrm{if}\left[i > 1, \left(\frac{1}{i}\right), j^2\right] \qquad A := \mathrm{matrix}(3,3,f)$$

$$A = \begin{pmatrix} 0 & 1 & 4 \\ 0 & 1 & 4 \\ 0.5 & 0.5 & 0.5 \end{pmatrix}$$

Рис. 29. Демонстрация работы функции matrix

Вопросы для самоконтроля

1. Какие средства MathCAD предназначены для создания и обработки векторов и матриц?
2. Чем в MathCAD отличается задание вектора-столбца от задания вектора-строки?
3. Как происходит обращение к отдельному элементу вектора-столбца, вектора-строки и матрицы?
4. Как можно изменить нижнюю границу индексации элементов векторов и матриц?
5. Перечислите способы создания и заполнения матриц.

Контрольные задания

Задания I уровня сложности

1. Создать квадратную матрицу A размерности 4×4 (матрицу заполнить самостоятельно).
2. Найти определитель матрицы A.
3. Транспонировать матрицу A.
4. Присвоить переменным k1 и k2 значения элементов первой строки первого столбца и последней строки последнего столбца.
5. Сохранить в векторах V1 и V2 1 и 3 столбцы матрицы A.
6. Найти сумму элементов каждого вектора и скалярное произведение векторов V1 и V2.

Задания II уровня сложности

1. Применить оператор векторизации к каждому элементу матрицы A. Применяемая функция $z(x) = \sin(x) - 5x^2$, где x – элемент матрицы.
2. Создать матрицу B размерностью 5×7, используя функцию $B_{i,j} = f(i,j) = \begin{cases} (2*i - j, & \text{если } |i - j| > 2; \\ 5j - i^2, & \text{иначе,} \end{cases}$, где i, j – номера строки и столбца вычисляемого элемента. (Применение функции matrix.)
3. Вычленить из матрицы B матрицу C размерностью 4×4 (функция submatrix) и проверить, имеет ли система однородных уравнений с матрицей коэффициентов C решения.
4. Найти произведение матриц A*C.

Раздел 2. ПРИМЕНЕНИЕ СКМ MathCAD В ИНЖЕНЕРНЫХ И ТЕХНИЧЕСКИХ РАСЧЕТАХ

2.1. Решение одиночных уравнений

2.1.1. Решение уравнения графическим способом

Графически решением уравнения вида $f(x)=0$ является точка пересечения графика функции с осью OX. Решением уравнения вида $f(x)=g(x)$ является точка пересечения двух графиков. И в том и другом случае средствами MathCAD с достаточной точностью можно определить корень уравнения. Для этого необходимо построить график функции, а затем, если точка пересечения недостаточно ясно видна, дать команду увеличения изображения ***Format/Graph/Zoom*** (рис. 30).

Для получения координат точки, являющейся корнем уравнения, можно воспользоваться командой трассировки ***Format/Graph/Trace***.

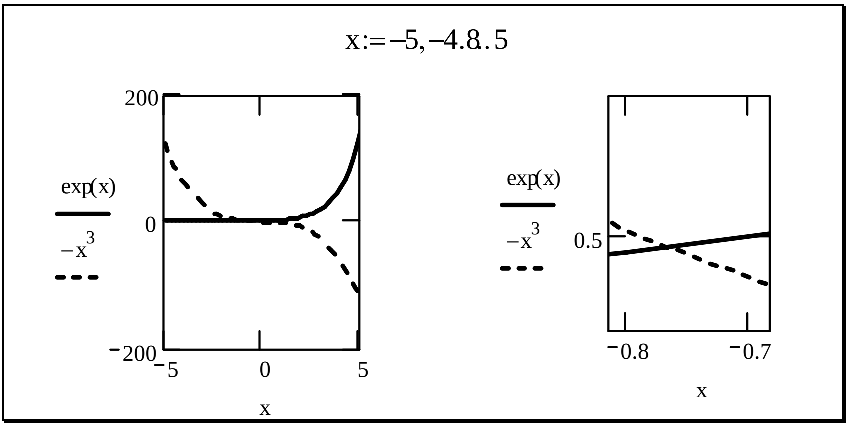


Рис. 30. Нахождение точки пересечения двух графиков

После появления диалогового окна “X-Y Trace” на графике курсором мыши указать нужную точку. При этом в диалоговом окне “X-Y Trace” отразятся значения абсциссы и ординаты указанной точки, которые можно скопировать в буфер обмена (кнопки “Copy X”, “Copy Y”), а затем присвоить каким-либо переменным.

2.1.2. Решение одиночного уравнения. Функции root

Функция **root** используется для решения уравнения с одним неизвестным, представленным в виде $f(x)=0$. Ищется такое значение переменой x, при котором выражение $f(x)$ обращается в 0.

Синтаксис функции **root**

$$\boxed{root(f(x),\quad x,[a,b])},$$

где $f(x)$ – первый аргумент функции root есть либо функция, определенная выше, либо математическое выражение, содержащее левую часть однородного уравнения;

x – имя переменной, относительно которой решается уравнение. Это та переменная, варьируя которую MathCAD будет пытаться обратить выражение в нуль.

$[a, b]$ – интервал поиска корней, локализующий ***один*** корень уравнения. Значения функции на границах заданного интервала должны быть разного знака.

Аргументы a, b можно опустить в том случае, если уравнение имеет один корень. Но тогда переменной уравнения обязательно должно быть присвоено начальное значение, которое MathCAD будет использовать как начальное приближение при поиске корня.

Функция root решает уравнения итерационным методом секущих и поэтому требует перед собой задания начальных значений. Кроме того, функция root, производя вычисления методом спуска, вычисляет и выводит только один корень, ближайший к начальному приближению (рис. 31).

Перед решением уравнения желательно построить график функции. На графике видно пересекает ли кривая нулевую линию, т.е. имеет ли уравнение действительные корни. Если есть точки пересечения кривой с осью, то надо выбирать начальное приближение поближе к значению корня.

Пример. Найти корень уравнения $e^x = x^3$.

Порядок решения уравнения

1. Привести уравнение к однородному виду, т.е. так, чтобы правая часть уравнения содержала нуль $e^x - x^3 = 0$.
2. Определить приближенное значение корня (для этого можно использовать отделение корней графическим способом).

3. Используем функцию root.

$$x := 3$$
$$\mathrm{root}\left(\exp(x) - x^3, x\right) = 1.857$$

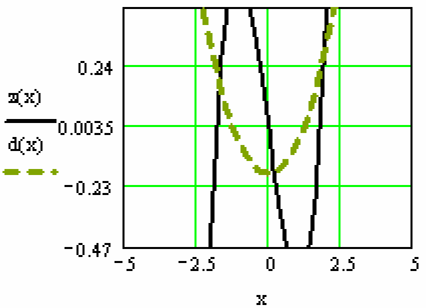
или

$$x := 3 \qquad z(x) := \exp(x) - x^3$$
$$a := \mathrm{root}(z(x), x) \qquad a = 1.857$$

Рис. 31. Нахождение корня с использованием функции root

Если графическое решение уравнения показало наличие нескольких корней (рис. 32), следует использовать функцию root с четырьмя аргументами.

$$x := -5, -4.8 .. 5$$
$$z(x) := 0.254 \cdot x^3 - 0.8269 \cdot x \qquad d(x) := 0.123 \cdot x^2 - 0.1818$$



$$\mathrm{root}(z(x) - d(x), x, -1.9, -1.6) = -1.691$$
$$\mathrm{root}(z(x) - d(x), x, 0.14, 0.3) = 0.216$$
$$\mathrm{root}(z(x) - d(x), x, 1.5, 2) = 1.959$$

Рис. 32. Использование полного синтаксиса функции root для локализации отдельного корня уравнения

MathCAD позволяет находить как комплексные, так и вещественные корни. Для поиска комплексного корня следует взять в качестве начального приближения комплексное число. Если точек пересечения нет, то корни уравнения – мнимые числа. Для их нахождения надо задавать начальное приближение в комплексной форме.

Для ввода мнимой единицы надо на клавиатуре набрать 1i или 1j. При выходе из области мнимого числа единица исчезает, и мы видим комплексное число в обычном виде. В результате расчета корни уравнения появляются в комплексном виде.

Как уже указывалось выше, MathCAD ищет корень последовательными уточнениями, а именно методом секущих.

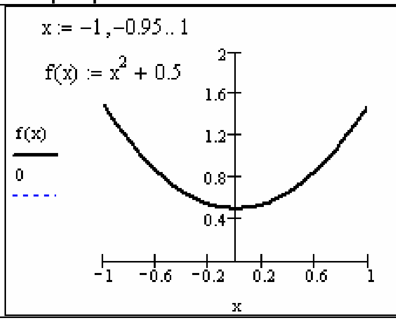
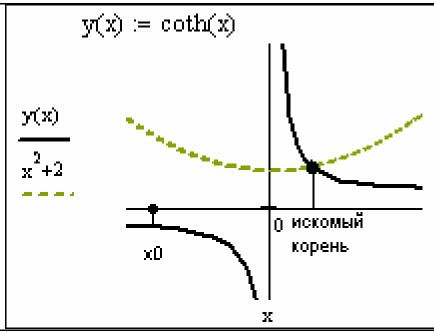
Корень уравнения считается найденным, если при очередном приближении f(x) отличается от нуля на величину, меньшую, чем значение системной переменной TOL, которая равна по умолчанию 0.001.

Решение можно искать с большей или меньшей точностью, изменяя значение TOL либо обычным переприсваиванием, например, TOL:= 0.01, либо изменив значение TOL на вкладке **Built-In Variables** (встроенные переменные) диалогового окна **Math/Options**. В последнем случае значение TOL будет изменено для всего рабочего документа.

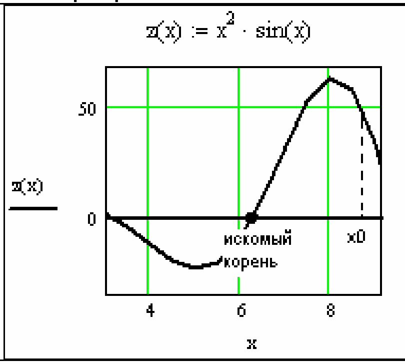
Если после многих приближений функция root не может найти подходящего приближения, то появляется сообщение об ошибке “Cant converge to a solution” – «отсутствует сходимость» (табл.3).

Таблица 3

Причины несходимости

Причины несходимости	Графическое пояснение
▪ Уравнение не имеет корней.	
▪ Корни расположены далеко от начального приближения.	
▪ Функция имеет разрывы между начальным приближением x_0 и корнем.	

Окончание табл. 3

Причины несходимости	Графическое пояснение
▪ Имеются локальные экстремумы между начальным приближением x_0 и корнем.	

Чтобы установить причину ошибки, следует построить график функции.

2.1.3. Нахождение корней степенного полинома с помощью функции polyroots

Для нахождения корней степенного полинома (многочлена) рекомендуется использовать функцию ***polyroots(V),*** которая определяет все корни полинома одновременно.

Здесь V – вектор коэффициентов полинома, начиная со свободного члена, включая все нули.

Если полином имеет N корней, то вектор V включает в себя N+1 коэффициент. Начальное приближение вводить не надо.

Например, решить полиномиальное уравнение вида

$$\frac{x^3}{3} - 0.2513x + 0.12x^2 - 0.016995 = 0 \cdot$$

Графически можно определить, что уравнение имеет три корня (рис. 33). Сохраняем в векторе V коэффициенты уравнения, начиная с нулевой степени перед неизвестным, в векторе Rez – корни.

Для функции polyroots можно выбрать один из двух численных методов решения: метод полиномов Лаггера (он установлен по умолчанию) или метод парной матрицы. Смену метода можно произвести, нажав правую кнопку мыши на слове Polyroot, вызвав контекстное меню (рис. 34).

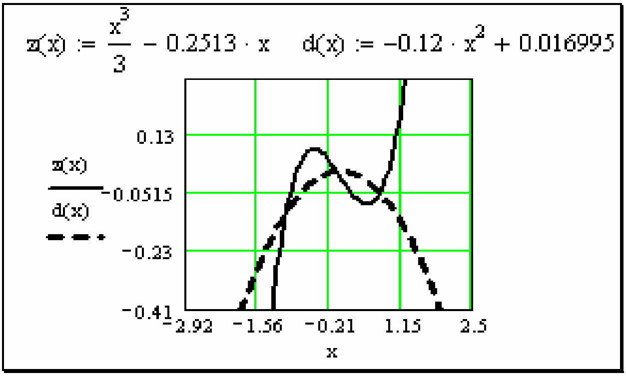


Рис. 33. Отделение корней многочлена

$$V := \begin{pmatrix} -0.016995 \\ -0.2513 \\ 0.12 \\ \frac{1}{3} \end{pmatrix} \qquad Rez := \mathrm{polyroots}(V)$$

$$Rez = \begin{pmatrix} -1.039 \\ -0.066 \\ 0.745 \end{pmatrix}$$

Проверка

$$z(Rez_0) = -0.112 \qquad d(Rez_0) = -0.112$$

Рис. 34. Использование функции polyroots для нахождения корней степенного полинома

Если полином задан неявно, для определения коэффициентов можно воспользоваться математической панелью **Symbolic Toolbar.**

Например, для определения коэффициентов полинома $(x^2 - 2x + 5) \cdot (x^2 - 6)$:

1) ввести k**:= ▪;**
2) выбрать **coeff** на математической панели **Symbolic Toolbar**;
3) заполнить места ввода (рис. 35).

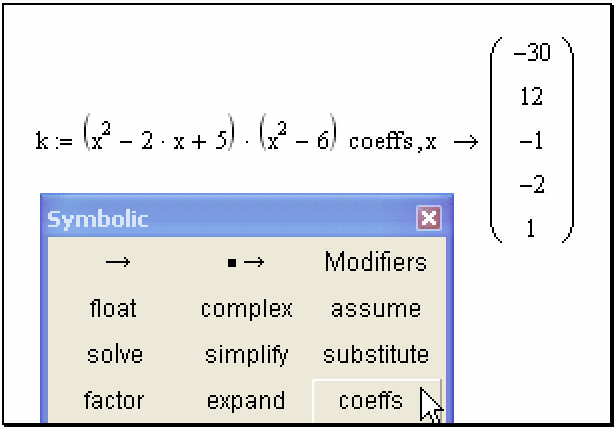


Рис. 35. Определение коэффициентов полинома

Вопросы для самопроверки

1. В каком пункте меню MathCAD находится полный перечень функций, предназначенных для решения уравнений и систем?
2. Какие функции MathCAD предназначены для решения одиночного уравнения?
3. Методом графического отделения корней выявлено, что уравнение имеет три корня. Каким образом можно аналитически найти все решения уравнения?
4. С какой точностью ведется поиск корня при работе функции root? Каким образом можно повлиять на точность решения уравнения?
5. При решении уравнения функция root не находит решения. Перечислите возможные причины.
6. С помощью какой функции удобно искать корни многочлена седьмой степени?
7. Что является результатом работы функции polyroots?

Контрольные задания

Задания I уровня сложности

1. Решить уравнение вида $5^x - 6x - 3 = 0$.
2. Подобрать способ решения уравнения $x^4 - x^3 - 2x^2 + 3x - 3 = 0$. Найти корни уравнения.

3. Графически и аналитически найти корни уравнения $2\lg x - \dfrac{x}{2} + 1 = 0$.

Задания II уровня сложности

1. Подобрать способ решения уравнения $(x^2 - 1)(4x - 3) = 0$. Найти корни уравнения.
2. Найти решение трансцендентного уравнения $4 - \lg\left(\frac{5}{2} x\right) = 3 \cdot \sqrt{\lg\left(\frac{5}{2} x\right)}$.
3. Используя возможности MathCAD определить коэффициенты полиномиального уравнения $\dfrac{x^2 - 1}{x^3 + x^2 - 8x - 12} = 0$. Решить уравнение.
4. Найти все корни многочлена $x^8 - x^7 + 11x^5 - 12x^2 + 88x = 11x^6 - 30x^4 + 58x^3 + 48$.

2.2. Решение систем уравнений

2.2.1. Решение систем линейных неоднородных алгебраических уравнений. Функция Lsolve

Функция Lsolve для решения систем уравнений использует метод Крамера, предназначенный для решения **линейных** (степени неизвестных не выше первой), неоднородных (правая часть ненулевая) уравнений.

Для этого метода систему уравнений вида

$$\begin{cases} a_{11}x1 + a_{12}x2 ... + a_{1n}x_n = b_1, \\ a_{21}x1 + a_{22}x2 ... + a_2 x_n = b_2, \\ .... \\ a_{n1}x1 + a_{n12}x2 ... + a_{nn}x_n = bn \end{cases}$$

представляют в виде $A * x = B$,

где A – матрица коэффициентов;

x – вектор-столбец неизвестных;

B – вектор-столбец свободных членов.

Если определитель матрицы А системы уравнений А*x*=В отличен от нуля, то система имеет единственное решение, которое можно найти по формулам Крамера или использовав функцию Lsolve.

Синтаксис функции

$$\boxed{lsolve(A, B)}.$$

Например, решить систему уравнений:

$$\begin{cases} 1.17x_1 + 2.23x_2 - 0.77x_3 = 1.11; \\ 2.23x_1 - 0.81x_2 + 1.72x_3 = 1.88; \\ -0.77x_1 + 1.72x_2 - 0.65x_3 = 0.57. \end{cases}$$

Вид решения в MathCAD приведен на рис. 36.

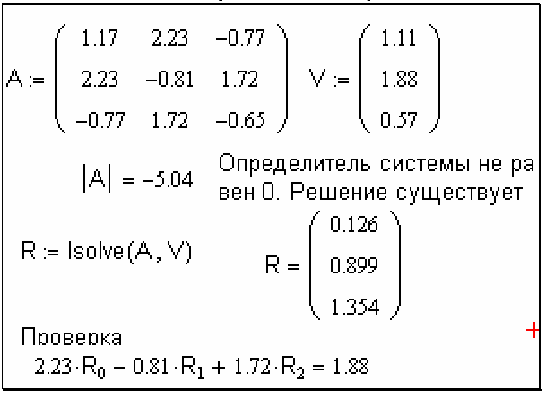


Рис. 36. Использование функции Lsolve для решения системы линейных уравнений

2.2.2. Использование систем линейных уравнений при описании статических математических моделей

Модель – это представление объекта или системы, отражающее их основные свойства.

Современные методы решения инженерных задач предполагают численные эксперименты с математической моделью исследуемого процесса.

Задача математического моделирования − установление связи между входными и выходными переменными в виде математических соотношений и реализация этих моделей на ЭВМ.

Математические модели, рассматриваемые при изучении технических дисциплин, можно разбить на следующие типы:

- статические;
- динамические;
- модели математической физики;
- модели систем массового обслуживания;
- игровые.

Статические модели могут быть заданы функцией $F(x, y) = 0$ в неявном виде или функцией $y = f(x)$ в явном виде. Такие модели осуществляют преобразование числа (чисел) – значений входных переменных x в число значение выходной переменной y. Для их реализации на ЭВМ потребуются численные методы решения систем алгебраических уравнений. В СКМ MathCAD для решения систем линейных уравнений используется встроенная функция lsolve(). Аргументами функции являются: первый аргумент – матрица коэффициентов системы уравнений, второй – матрица свободных членов.

Пример. Поведение объекта можно описать следующей системой из трех линейных уравнений:

$$\begin{cases} 2{,}5x1 + 3{,}1x2 \; 1{,}2x3 = 5{,}7; \\ 1{,}4x1 + 1{,}97x2 + 2{,}8x3 = 8{,}65; \\ 5{,}4x1 \; 4{,}1x2 + 3{,}4x3 = 6{,}3. \end{cases}$$

Подготовка данных к решению задачи состоит в задании двух матриц: матрицы коэффициентов *A* и матрицы свободных членов *B* (рис. 37).

Рис. 37. Реализация статической модели с использованием системы линейных уравнений

2.2.3. Решение систем уравнений с помощью функции Find

Find является универсальной функцией, предназначенной для решения сходящихся систем уравнений любого типа и неравенств.

Вычислительный блок MathCAD, использующий функцию Find, позволяет решать системы от 1 до 200 уравнений.

Для решения системы уравнений необходимо:

1. Задать начальные приближения для всех неизвестных, входящих в систему уравнений. MathCAD решает уравнения при помощи итерационных методов. На основе начального приближения строится последовательность, сходящаяся к искомому решению.
2. Создать **вычислительный блок решения** уравнения:
   а) задать слово GIVEN (дано), оно указывает, что далее следует система уравнений или ограничений;
   б) ввести уравнения или неравенства в любом порядке ниже ключевого слова GIVEN:
      - системы уравнений – (при записи уравнений между правыми и левыми частями необходимо использовать оператор отношения (жирный знак равенства, клавиши **Ctrl+=)**;
      - ограничения на поиск решения в виде неравенств, если они есть;

   в) выражение, содержащее функцию Find, с неизвестными в качестве параметров.

   Синтаксис функции Find

$$\boxed{Find(x1, x2, x3..., x_n)},$$

где x_i – искомые неизвестные.

Эта функция возвращает решение системы уравнений. Число аргументов функции Find должно быть равно числу неизвестных в системе уравнений.

Если Find имеет один аргумент, то она возвращает одно значение, если несколько, то функция возвращает ответ в виде вектора.

Например, решить систему уравнений (рис. 38)

$$\begin{cases} 4.5x1^3 + 3.8x2 - 2.2x3 = 8.1; \\ 1.85x1^3 - 2.1x2^2 + 3.45x3 = 4.5; \\ 6.2x1^2 - 4.8x2 + 9.1x3 = 11.1. \end{cases}$$

$x1 := 2 \qquad x2 := 2 \qquad x3 := 3$

GIVEN

$$4.5 \cdot x1^3 + 3.8 \cdot x2 - 2.2 \cdot x3 = 8.1$$

$$1.85 \cdot x1^3 + 2.1 \cdot x2^2 + 3.45 \cdot x3 = 4.5$$

$$6.2 \cdot x1^2 - 4.8 \cdot x2 + 9.1 \cdot x3 = 11.1$$

$$\mathrm{Find}(x1, x2, x3) = \begin{pmatrix} 1.214 \\ 0.199 \\ 0.321 \end{pmatrix}$$

Рис. 38. Пример решения системы уравнений с помощью функции Find

Предупреждение! Внутри блока GIVEN нельзя:

1) Нельзя использовать вложенные блоки решения. Каждый блок решения может иметь только одно ключевое слово GIVEN и имя функции Find.
2) Нельзя использовать оператор присваивания внутри блока решения.
3) Не допускается использование других вычисляемых выражений.
4) Не допускается использование дискретных переменных (x:=0…20).
5) Не допускается использование знака «не равно».

Функция Find реализует градиентные численные методы и предлагает на выбор три метода. Чтобы выбрать метод решения, щелкните правой кнопкой мыши на названии функции Find.В открывшемся контекстном меню и его подменю выберите нужный метод:

Линейный метод (Linear) – метод касательной,

Нелинейный метод (Nonlinear).

Нелинейных методов три:

- метод Сопряженных градиентов (Conjugate Gradient);
- Квази-Ньютоновский метод (Quasi-Newton);
- метод Левенберга (Levenberg-Marquart).

Иногда функция Find не может найти решения. Это может быть вызвано одной из следующих причин:

Причины несходимости

1. Поставленная задача не имеет решения.

2. Возможно поставленная задача не может быть решена с заданной точностью. Попробуйте увеличить значение TOL где-нибудь выше блока решения.
3. Для уравнений, которые не имеют вещественных решений, в качестве начального приближения дано вещественное число. В этом случае начальные приближения должны быть комплексными.
4. В процессе поиска решения последовательность приближений попала в точку локального минимума невязки. Метод поиска решения, который используется MathCAD, не позволяет в этом случае построить следующее приближение, которое бы уменьшило невязку. Для поиска искомого решения попробуйте использовать различные начальные приближения или добавьте ограничения на переменные в виде неравенств, чтобы обойти точку локального минимума. В этом случае полезно построить те или иные графики, связанные с системой. Это поможет выбрать подходящее начальное приближение.

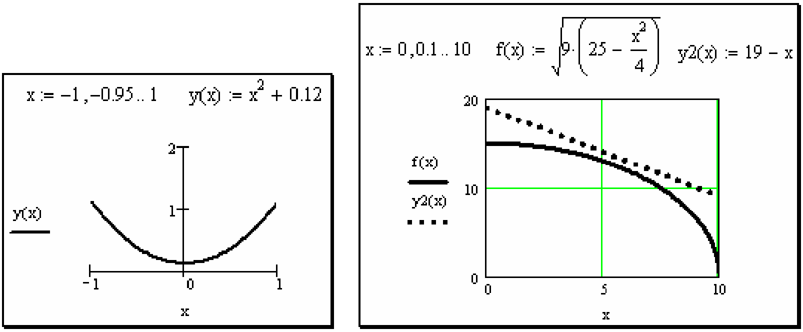
Некоторые системы уравнений не могут быть решены точно. В этом случае используется приближенное решение уравнений и систем.

2.2.4. Приближенное решение систем уравнений. Функция Minerr

Функция Minerr применяется в том случае, когда уравнение или система уравнений не имеют точного решения. Например, график функции не пересекается с осью аргументов – нет вещественных корней (рис. 39, а), или для случая двух уравнений – нет пересечения кривых (рис. 39, б). В таких ситуациях можно найти такое решение, при котором невязка аргумента будет минимальной.

Функция Minerr использует тот же вычислительный алгоритм, что и функция Find. Если решение существует, то замена в вычислительном блоке функции Find на Minerr дает одинаковые результаты.

Различие между функциями Find и Minerr заключается в следующем. Если в результате поиска решения не может быть получено дальнейшее уточнение текущего приближения к решению, то функция Find выдает сообщение «решение не найдено», а Minerr возвращает минимум невязки, т.е. значение аргумента, соответствующее минимальному найденному расхождению. Правила использования функции Minerr такие же, как и функции Find.



а б

Рис. 39.2 Иллюстрация ситуаций, требующих применения функции Minerr для поиска приближенного решения

Синтаксис

$$\boxed{MinErr(x1, x2, x3, \ldots, x_n)}$$

– возвращает приближенное решение системы уравнений. Число аргументов равно числу неизвестных.

Например, решить систему уравнений $\begin{cases} x^2 + 1 = 0; \\ \cos(x - 0.5) = 0. \end{cases}$.

Однако функция Minerr иногда выдает неправдоподобные, далекие от истинного решения, результаты. Поэтому, необходимо включать дополнительную проверку достоверности получаемых результатов. Встроенная переменная ERR дает величину невязки для приближенного решения (рис. 40). Чем меньше эта величина, тем более точным является решение.

x := 0   Начальное приближение корня

Given

$x^2 + 1 = 0$

$\cos(x - 0.5) = 0$

minerr(x) = −0.2

ERR = 1.291   Величина невязки

Рис. 40. Пример использования функции Minerr

При нажатии правой кнопки мыши на имени функции открывается контекстное меню, позволяющее выбрать численный метод вычислений.

2.2.5. Символьное решение уравнений и систем

Символьное решение уравнений удобно в том случае, когда уравнение имеет параметр. Поэтому, вместо того, чтобы решать уравнение для каждого нового значения параметра, можно просто заменять его значения в найденном символьном решении.

Для получения символьного решения можно воспользоваться одним из двух вариантов:

1. *Решение уравнения относительно переменной*

- Напечатайте уравнение, используя для разделения левой и правой частей знак тождественного равенства (жирное равно CTRL + =).
- Выделите переменную, относительно которой нужно решить уравнение.
- Выберите команду **Symbolics/Variable/Solve** (Символика/Переменная/Решить) (рис. 41).
- Используйте найденное решение для определения функции, зависящей от параметров. Например

$$a\cdot x^2 - 4\cdot x = c$$

Символьное решение

$$\begin{bmatrix} \dfrac{1}{(2\cdot a)}\cdot\left[4+2\cdot(4+a\cdot c)^{\left(\frac{1}{2}\right)}\right] \\ \dfrac{1}{(2\cdot a)}\cdot\left[4-2\cdot(4+a\cdot c)^{\left(\frac{1}{2}\right)}\right] \end{bmatrix}$$

Используя копирование, определим x

$$x(a,c) := \frac{1}{(2\cdot a)}\cdot\left[4+2\cdot(4+a\cdot c)^{\left(\frac{1}{2}\right)}\right]$$

Найдем значение корня при определенных кооэффициентах

$$x(1,2) = 4.449$$

Рис. 41. Использование возможностей символьного процессора Simbolic

2. *Использование оператора символьного вывода*

Оператор символьного вывода имеет вид стрелки $\rightarrow$. Вместе с функциями Find и Minerr позволяет найти символьное решение уравнений и систем. В этом случае не надо вводить начальные приближения для неизвестных. Кроме того, в отличие от первого способа символьного решения системы, при изменении исходных данных произойдет обновление результата (рис. 42).

Given

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

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

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

Рис. 42. Использование оператора символьного вывода

2.2.6. Решение систем уравнений с ограничениями

Решить следующую систему:

$$\begin{cases} x^2 = y; \\ 3 - x = y; \\ x > 0. \end{cases}$$

Решением системы будут являться точки пересечения двух графиков.

Если построить графики функций, то можно заметить, что ограничение $x > 0$ отсекает один из возможных корней системы, а именно тот, который расположен левее 0 на числовой прямой (рис.43).

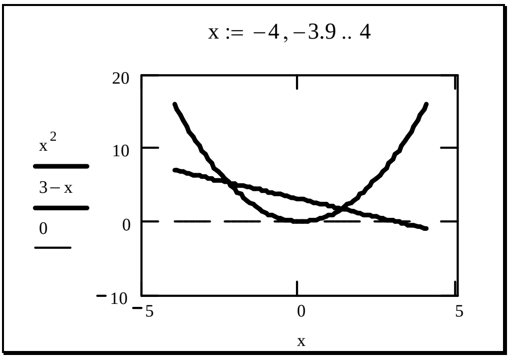


Рис. 43. Отделение корня заданным ограничением

Решение системы уравнений в этом случае будет выглядеть следующим образом (рис. 44).

$x := 0 \quad y := 0$ Начальные приближения корней

Given

$x^2 = y$

$3 - x = y$ Блок решения

$x > 0$

$\mathrm{Find}(x, y) = \begin{pmatrix} 1.303 \\ 1.697 \end{pmatrix}$

Рис. 44. Решение системы уравнений с ограничениями

Вопросы для самопроверки

1. Какая функция предназначена для решения линейных систем уравнений?
2. Что является аргументами функции Lsolve?
3. Перечислите состав блока решения при использовании функции Find.
4. Какие аргументы имеет функция Find?
5. В каких ситуациях следует применять функцию Minerr для решения систем уравнений?
6. Чем отличается работа функций Find и Minerr?
7. Как влияют ограничения на поиск корней системы уравнений?
8. Как найти символьное решение одиночного уравнения, системы уравнений?

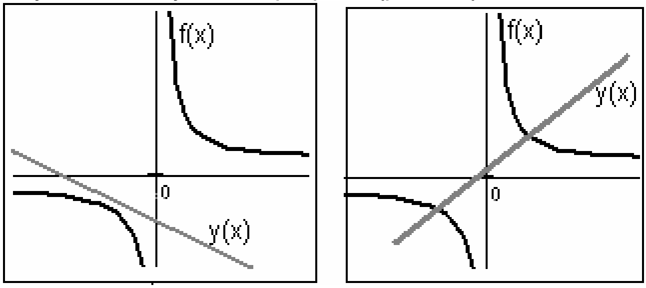
Контрольные задания

Задания I уровня сложности

1. Решить в символьном виде уравнение $a \cdot x^3 = b \cdot x^2 + 1.5$.
2. Найти корни системы уравнений $\begin{cases} 2x_1 + x_2 + 3x_3 = 9; \\ x_1 - 2x_2 + x_3 \quad -2; \\ 3x_1 + 2x_2 + 2x_3 = 7. \end{cases}$
3. Решить систему уравнений $\begin{cases} 2.16x_1 + 1.45x_2 0.89x_3 = 0.61; \\ 1.45x_1 - 2.44x_2 + 1.18x_3 = 1.05; \\ -0.89x_1 + 1.18x_2 - 2.07x_3 \quad -0.83. \end{cases}$

Задания II уровня сложности

1. При графическом поиске корней уравнения $f(x) = y(x)$ получили следующие графики (рис. 45)



а б

Рис. 45. Иллюстрация к контрольному заданию

Какой способ решения уравнения следует выбрать в случае a или b?

2. Решить систему уравнений $\begin{cases} x^3 \cdot \sin(y) = \sqrt{x+1}; \\ y^2 \cdot \cos(x) = 4 - xy. \end{cases}$
3. Найдите приближенное решение уравнения $\ln(|x|) = x^2 + 3$.
4. Найти решение системы уравнений

$$\begin{cases} -40*X1+44*X2-34*X3-43*X4+6*X5 = 8; \\ -25*X1+30*X2+4*X3-44*X4-44*X5 = -45; \\ -11*X1+37*X2+-26*X3-10*X4-6*X5 = -10; \\ -20*X1+29*X2+6*X3+7*X4-42*X5 = 4; \\ 50*X1-48*X2+20*X3+19*X4-24*X5 = 11. \end{cases}$$

5. Найдите приближенное решение системы уравнений

$$\begin{cases} 2a^3 + 5.5b^2 - 24c = 9.6; \\ 4.8a^3 + 9.5b^2 - 49c = 18.6; \\ 6a^3 + 16b^2 - 74c = 34. \end{cases}$$

2.3. Решение обыкновенных дифференциальных уравнений

Решение дифференциальных уравнений широко применяется в практике научно-технических расчетов. Это связано с тем, что дифференциальные уравнения (и системы из них) описывают поведение различных объектов в динамике.

При решении дифференциального уравнения искомой величиной является функция. Для обыкновенных дифференциальных уравнений (ОДУ) неизвестная функция – это функция одной переменной. Дифференциальные уравнения в частных производных – это дифференциальные уравнения, в которых неизвестной является функция двух или большего числа переменных.

MathCAD имеет ряд встроенных функций, предназначенных для решения ОДУ. Каждая из этих функций предназначена для *численного решения* дифференциального уравнения.

В результате решения получается матрица, содержащая значения функции на некотором множестве точек (на некоторой сетке значений).

Существует несколько способов решения дифференциальных уравнений, но при каждом методе требуется, чтобы были заданы, по крайней мере, следующие величины, необходимые для поиска решения:

- начальные условия;
- набор точек, в которых нужно найти решение;
- само дифференциальное уравнение, записанное в некотором специальном виде.

2.3.1. Решение ОДУ с использованием функции rkfixed

Функция **rkfixed** использует для поиска решения метод Рунге-Кутта четвертого порядка.

Рассмотрим пример решения дифференциального уравнения первого порядка – уравнения, которое не содержит производных выше первого порядка от неизвестной функции.

$$\frac{dy}{dx} + 3y = 0$$

с начальными условиями $y(0) = 4$, на участке $[0; 4]$.

Начальное условие – это значение искомой функции в начальной точке интервала.

На рис.46 показан пример решения ОДУ первого порядка.

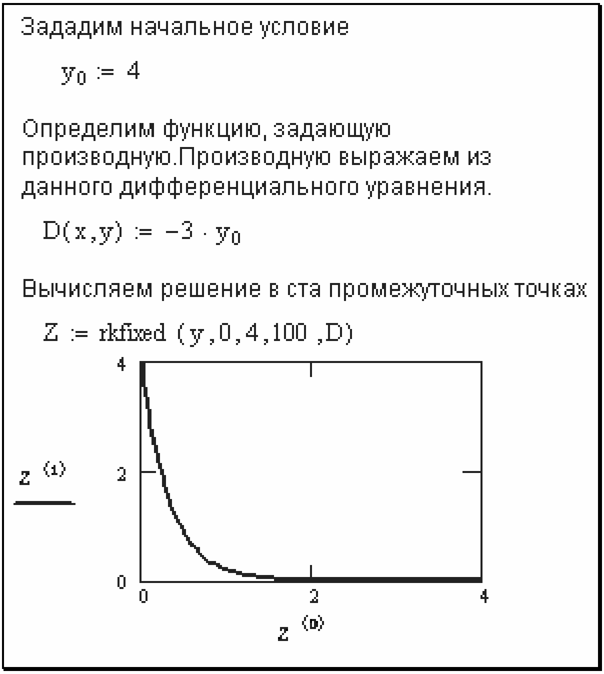


Рис. 46. Решение ОДУ с использованием функции rkfixed

В результате решения получается матрица, имеющая два следующих столбца:

- Первый столбец $Z^{\langle 0\rangle}$ содержит точки, в которых ищется решение дифференциального уравнения.
- Второй столбец $Z^{\langle 1\rangle}$ содержит значения найденного решения в соответствующих точках.

Функция **rkfixed** имеет следующий синтаксис:

$$rkfixed(y, x1, x2, n\_point s, D),$$

где y – вектор начальных условий размерности n; n – порядок дифференциального уравнения или число уравнений в системе (если решается система уравнений).

Для дифференциального уравнения первого порядка вектор начальных условий вырождается в одну точку $y_0 = y(x1)$,

где $x1$ – начальная точка интервала, на котором ищется решение.

Для дифференциального уравнения второго порядка вектор начальных условий будет содержать две компоненты

$y_0 = y(x1)$ и $y_1 = y'(x1)$ – значение первой производной в начальной точке интервала;

$x1, x2$ – граничные точки интервала, на котором ищется решение. Начальные условия, заданные в векторе y, – это значение решения в точке $x1$;

n_points – число точек (не считая начальной), в которых ищется приближенное решение;

D – имя функции двух аргументов $D(x, y)$. Эта функция возвращает значение в виде вектора из n элементов, содержащих производные неизвестных функций. Для дифференциального уравнения первого порядка D содержит одну компоненту; для дифференциального уравнения второго порядка D – две

$$D(x, y) = \begin{pmatrix} y'(x1) \\ y''(x) \end{pmatrix}.$$

Первая компонента – это значение первой производной в начальной точке интервала, вторая компонента D – это значение второй производной, выраженное из дифференциального уравнения.

Например, решить ОДУ $y'' - 2y' = \sin(x) \cdot y$ с начальными условиями

$y(2) = 0; \quad y'(2) = 6$ на участке $[2; 5]$.

Решение дифференциального уравнения представлено на рис. 47.

Задаем начальные условия $\quad y := \begin{pmatrix} 0 \\ 6 \end{pmatrix}$

Задаем значения для первой и второй производных

$$D(x,y) := \begin{pmatrix} y_1 \\ \sin(x) \cdot y_0 + 2 \cdot y_1 \end{pmatrix}$$

$$Z := \mathrm{rkfixed}(y, 2, 5, 100, D)$$

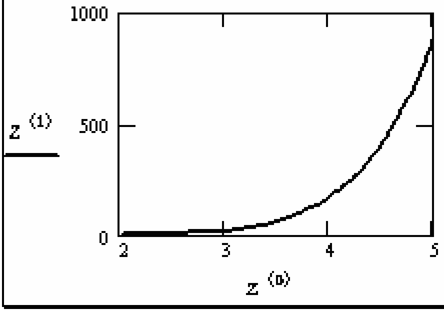


Рис. 47.Решение ОДУ второго порядка с использованием функции rkfixed

Функция rkfixed выдает решение дифференциального уравнения в виде матрицы, в нулевом столбце которой находятся значения аргумента функции, во втором – значения функции, посчитанные в данном случае в ста точках, в третьем – значения первой производной (рис. 48).

Z =

	x	y(x)	y'(x)
	0	1	2
0	2	0	6
1	2.03	0.186	6.374
2	2.06	0.383	6.775
3	2.09	0.592	7.208
4	2.12	0.815	7.672
5	2.15	1.053	8.171
6	2.18	1.306	8.706
7	2.21	1.576	9.2[illegible]
8	2.24	1.86[illegible]	
9	2.27		

Рис. 48. Результат работы функции rkfixed

Поэтому для построения графика на рис. 48 были использованы операторы извлечения столбца матрицы $Z^{\langle 0\rangle}$ – для аргумента функции и $Z^{\langle 1\rangle}$ – для значений функции.

2.3.2. Решение ОДУ более высоких порядков

Методика решения дифференциальных уравнений более высокого порядка является развитием методики, которая применялась для решения уравнений второго порядка. Основное различие состоит в следующем:

- Вектор начальных значений теперь состоит из n элементов, определяющих начальные условия для искомой функции и ее производных $y,\ y',\ y'',\ ...,\ y^{(n-1)}$.
- Функция D является теперь вектором, содержащим n элементов.

$$D(x,y)=\begin{pmatrix} y'(x1)\\ y''(x1)\\ y'''(x1)\\ \ldots\\ y^{(n)}(x)\end{pmatrix}.$$

Матрица, получаемая в результате решения, содержит теперь n столбцов: первый – для значений x, и оставшиеся столбцы – для значений $y(x),\ y'(x),\ y''(x),...,y^{(n)}(x)$.

2.3.3. Решение обыкновенных дифференциальных уравнений с помощью функции odesolve

Основным достоинством использования функции Odesolve является возможность решения обыкновенных дифференциальных уравнений (ОДУ) в стандартном математическом виде, т.е. решить алгебраически относительно производной высшего порядка и записать в виде $y'(x)=f(x)$.

Для численного интегрирования одного ОДУ (равно как и системы ОДУ) можно использовать вычислительный блок Given–Odesolve, появившийся впервые в версии Mathcad 2000 Pro, или использовать встроенные функции, оставшиеся от более ранних версий MathCAD.

Функция Odesolve включает в себя возможности всех ранее созданных функций.

Odesolve возвращает решение обыкновенного дифференциального уравнения в виде функции от аргумента.

Вычислительный блок содержит:

1) ключевое слово Given;
2) дифференциальное уравнение, записанное в обычном виде, с разделением правой и левой частей уравнения знаком “жирного равно”;
3) граничные условия, число которых равно порядку уравнения, функцию Odesolve(*x, xk, n*).

$$\boxed{Odesolve(x, xk, n)},$$

где x – имя переменной, относительно которой решается уравнение;

xk – конец интервала интегрирования (начало интервала интегрирования указывается в начальных условиях),

n – необязательный внутренний параметр, определяющий число шагов интегрирования.

Дифуравнение можно записать либо со штрихом, либо с дифференциалом.

$$y'''(x) + x^2 \cdot y'(x) + x \cdot y(x) = e^x \cdot \cos(x);$$

$$\frac{d^3}{dx^3} y(x) + x^2 \cdot \frac{d}{dx} y(x) + x \cdot y(x) = e^x \cdot \cos(x).$$

Граничные условия записываются только со штрихом (для набора штриха в выражении $y'(0)$ служат клавиши **Ctrl + F7**).

Например, решить задачу Коши на участке [0; 2]

$y''' - x^2 \cdot y' = x \cdot \cos(x)$ при следующих граничных условиях: $y(0) = 1;\ y'(0) = 1;\ y''(0) = -0.5$. Решение приведено на рис. 49. .

Given

$$\frac{d^3}{dx^3} y(x) - x^2 \cdot \frac{d}{dx} y(x) = x \cdot \cos(x)$$

$$y(0) = 1 \qquad y'(0) = 1 \qquad y''(0) = -0.5$$

$$y := \mathrm{Odesolve}(x, 2)$$

Рис. 49. Решение ОДУ с использованием функции odesolve

2.3.4. Использование дифференциальных уравнений при описании динамических математических моделей

Динамические модели реализуют преобразование функции в функцию. Обычно рассматриваются функции времени

$$L(x(t),\, u(t)) = 0 \text{ или } \frac{dx}{dt} = f(x(t),\, u(t))\cdot$$

В этих уравнениях традиционно через $x_i(t),\ i = 1,\, 2,\, ...,\, m$ обозначают выходные переменные (переменные состояния), через $u_j(t),\quad j = 1,\, 2,\, ...,\, n$ – входные переменные. Обычно модель динамической системы задается в виде системы дифференциальных уравнений в нормальной форме

$$\frac{dx}{dt} = f(x,\, u,\, t),\, x \in R^m,\, u \in R^n,\, t \in [0,T],$$

где $f(x,\, u,\, t)$ − вектор-функция правых частей уравнения.

Система дифференциальных уравнений должна быть дополнена m условиями для единственности решения. Обычно это начальные условия, задаваемые в виде

$$x(0) = x^0.$$

Если время t явно входит в функцию $f(x,\, u,\, t),$ то система не автономная; если явно не входит, т.е. $f = f(x,\, u),$ то это нелинейная автономная система; если функция линейная по переменным x и $u,$ то тогда получаем линейную систему

$$\frac{dx}{dt} = Ax + Bu.$$

Для реализации динамических моделей на ЭВМ потребуются численные методы интегрирования дифференциальных уравнений.

В отличие от динамических моделей, ***модели математической физики*** описывают поведение выходной переменной с несколькими независимыми переменными, например, временем t и пространственной переменной x.

Аппарат для построения таких моделей − уравнения в частных производных. Например, одномерное распространение тепла вдоль тонкого стержня описывается следующим уравнением в частных производных:

$$\frac{\partial^2 u(x, t)}{\partial x^2} = k\frac{\partial u(x, t)}{\partial t},$$

где $u(x,t)$ − температура стержня в точке с координатой $x.$ в момент времени t Для выделения единственного решения необходимо задать начальные и граничные условия. Начальными условиями может быть известная температура стержня в начальный момент − функция $u(x, 0) = \varphi(x),$ граничными − известные температуры на концах стержня $u(0, t) = f_1(t), u(l, t) = f_2(t).$

Модели математической физики имеют аналитическое решение в исключительных случаях. Обычные численные методы для их исследования − метод сеток и метод конечных элементов/

Ошибки моделирования

Пусть точное значение выходной переменной y^*, предсказанной по модели $y = f(x),$ равно $y.$ Ошибка моделирования определяется нормой $\left\|y^* - y\right\|$, в качестве которой для вектора $a = (a_1, a_2, ..., a_n)$ наиболее часто используют следующие величины:

$$\|a\|_1 = \max_{1 \le i \le n} |a_i|,$$

$$\|a\|_2 = \sqrt{\sum a_i^{\ 2}}.$$

Относительная ошибка определяется отношением (обычно точное значение y^* не известно)

$$\rho = \frac{\left\|y^* - y\right\|}{\left\|y^*\right\|} \approx \frac{\left\|y^* - y\right\|}{\|y\|}.$$

Можно выделить три источника ошибок моделирования. Первый источник – погрешность моделирования ρ_1, связанная с неточностью самой математической модели. Второй – погрешность численного решения ρ_2 уравнений модели, связанных с применением численных методов, например. метода сеток для решения дифференциальных уравнений. И, наконец, любая реализация численных методов на ЭВМ связана с третьим источником ошибок − ошибками округления ρ_3. Общая ошибка моделирования будет удовлетворять неравенству

$$\rho \le \rho_1 + \rho_2 + \rho_3.$$

Если мы находимся в рамках одной математической модели, то влиять на первый источник ошибок нет возможности, поэтому такую ошибку часто считают неустранимой. На остальные два источника ошибок можно повлиять выбором метода решения и разрядностью представления чисел в ЭВМ. Обычно считают выбор приемлемым, если ρ_2 на порядок меньше ρ_1, а ρ_3 на порядок меньше ρ_2.

Контрольные вопросы

1. Что является решением обыкновенного дифференциального уравнения?
2. Какие функции MathCAD предназначены для решения ОДУ?
3. Перечислите правила использования функции rkfixed.
4. Перечислите правила использования функции odesolve.
5. Какие граничные условия необходимо задать для решения дифференциального уравнения в среде MathCAD?
6. Как построить график функции, являющейся решением ОДУ при использовании функций rkfixed, odesolve?

Контрольные задания

Контрольные задания I уровня сложности

1. Найти частное решение дифференциального уравнения $y'' + 3y' + 2y = 0$ при следующих начальных условиях: $x_0 = 0;\ y(x_0) = 2;\ y'(x_0) = 6$.
2. Найти частное решение дифференциального уравнения $y'' - 9y = 0$ при следующих начальных условиях: $x_0 = 0;\ y(x_0) = -1;\ y'(x_0) = 3$.

Контрольные задания II уровня сложности

1. Ускорение точки дано равенством $j = -\frac{5}{4}S - v$, где v – скорость точки, а S – путь, пройденный ею. Найти уравнение движения точки, если при $t = 0,\ S = v = 0$.
2. Найти частное решение дифференциального уравнения на участке [1;2] $(x^2 - 2y^2)\,dx + 2xy\,dy = 0$ при следующих начальных условиях $x_0 = 1;\ y(x_0) = 0.4;\ y'(x_0) = 1$. Построить график решения.
3. Найти частное решение дифференциального уравнения на участке [0;2] $\frac{dy}{\sqrt{y}} + dx = \frac{dx}{\sqrt{x}}$ при следующих начальных условиях: $x_0 = 0;\ y(x_0) = 0;\ y'(x_0) = 2$. Построить график решения.

2.4. Примеры применения системы MathCAD при решении прикладных и функциональных задач

2.4.1. Решение задач оптимизации

В задачах оптимизации, как правило определяется некоторый критерий или функция, который должен быть оптимизирован (минимизирован или максимизирован).

В MathCAD такие задачи решаются следующим способом:

1. Задаются начальные значения для всех переменных.
2. Определяется целевая функция – функция, которая зависит от переменных и должна быть оптимизирована.
3. В блоке GIVEN задаются все ограничения.
4. Блок GIVEN завершается одной из функций: *maximize* или *minimize*. Синтаксис этих функций следующий:

$Maximize(f, \mathrm{var}1, \mathrm{var}2, \dots, \mathrm{var}_n)$ – ищет такие значения переменных $\mathrm{var}1, \mathrm{var}2, \dots, \mathrm{var}_n$, при которых значение функции f максимально. Возвращает результат в виде вектора, компоненты которого – значения переменных, при которых функция имеет оптимальное значение.

Функция *minimize* работает аналогично: $Minimize(f, \mathrm{var}1, \mathrm{var}2, \ldots, \mathrm{var}_n)$.

Пример. Найти такое значение аргумента x, при котором функция $y(x) = \sin(x^2) - 2$ достигает наибольшего значения на участке (-1;2).

Необходимо заметить, что построение графика функции позволило правильно выбрать начальное приближение для переменной x. Если между начальным приближением и искомым значением переменной находится какой-либо экстремум, MathCAD может не найти решения (рис. 50).

$y(x) := \sin\left(x^2\right) - 2$

$x := -1, -0.98 .. 2$

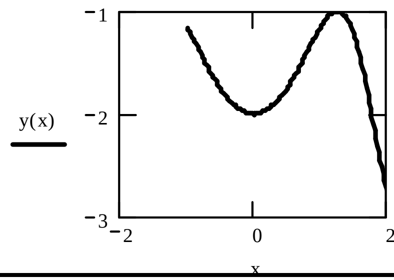


Рис. 50. Построение графика функции для визуализации экстремумов

Решение задачи представлено на рис. 51.

1. Зададим начальное приближенное значение для x $x := 1$

2. Определим функцию $y(x) := \sin\left(x^2\right) - 2$

3. Зададим ограничения

Given

$x > -1$

$x < 2$

$\mathrm{Maximize}(y, x) = 1.253$

Рис. 51. Поиск экстремума с помощью функции Maximize

Вышепоставленную задачу можно решить другим способом. Этот способ основан на том, что в точке экстремума первая производная от функции равна 0 (рис. 52).

1. Зададим начальное приближенное значение для x $x := 1$

$y(x) := \sin\left(x^2\right) - 2$

2. Решим уравнение

Given

$\frac{d}{dx} y(x) = 0$

$\mathrm{Find}(x) = 1.253$

Рис. 52.Решение дифференциального уравнения для поиска экстремума

Если экстремумов у функции несколько, начальное приближение переменной позволяет выбрать нужный экстремум.

Пример. Дана задача линейного программирования. Необходимо найти максимум функции $f = -x1 - x2 + 3x3 - 2x4$ при следующих ограничениях:

$$\begin{cases} x1 + 3x2 - x3 + 2x4 \le 7; \\ -2x2 + 4x3 - 2x4 \le 12; \\ 4x1 - 4x2 + 3x3 - 2x4 \le 14; \\ x_i \ge 0 \quad (i = 1 \ldots 4). \end{cases}$$

Поставленная задача решается следующим образом (рис. 53):

1. Задаем начальные (приближенные) значения для всех переменных

$x1 := 1 \qquad x2 := 2 \qquad x3 := 2 \qquad x4 := 1$

2. Определяем целевую функцию

$f(x1, x2, x3, x4) := -x1 - x2 + 3 \cdot x3 - 2 \cdot x4$

3. Задаем ограничения в блоке GIVEN

GIVEN

$x1 + 3 \cdot x2 - x3 + 2 \cdot x4 \le 7$

$-2 \cdot x2 + 4 \cdot x3 - 2 \cdot x4 \le 12$

$4 \cdot x1 - 4 \cdot x2 + 3 \cdot x3 - 2 \cdot x4 \le 14$

$x1 \ge 0 \qquad x2 \ge 0 \qquad x3 \ge 0 \qquad x4 \ge 0$

Рис. 53. Решение в MathCAD задачи линейного программирования

4. Определим переменную p, которая воспримет результат выполнения функции maximize

$$p := \text{Maximize}(f, x1, x2, x3, x4)$$

$$p = \begin{pmatrix} 0 \\ 4 \\ 5 \\ 0 \end{pmatrix}$$

5. Находим наибольшее значение функции

$$f(p_0, p_1, p_2, p_3) = 11$$

Рис. 53. Окончание (см. также с. 71)

Пример. Транспортная задача.

Имеется m пунктов производства однородного груза $A_1, A_2, ..., A_m$ и n пунктов потребления груза: $B_1, B_2, ..., B_n$;

$\alpha_i(\textit{тонн})$ – количество груза в пункте A_i;

$\beta_i(\textit{тонн})$ – потребность в грузе пункта B_j.

Известна матрица затрат – стоимость перевозки одной тонны из пункта производства A_i в пункт потребления B_j.

	B_1	B_2	B_3
A_{1i}	12	3	6
A_2	2	7	5

Составить оптимальный план перевозок, если известно:

Количество грузов в пунктах производства $\alpha_1 = 120$ $\alpha_2 = 80$.

Потребность в грузах: $\beta_1 = 85$ $\beta_2 = 55$ $\beta_3 = 60$.

Решение в MathCAD (рис. 54).

Общие затраты на перевозку можно определить как сумму произведений затрат на перевозку одной тонны, умноженных на количество перевозимых тонн. Для этого соответствующий элемент матрицы Zat_1t нужно умножить на соответствующий элемент матрицы K. Специальной матричной операции, позволяющей выполнить данное действие, в MathCAD нет, поэтому определим собственную функцию, позволяющую создать нужную матрицу.

$$f(i, j) := Zat\_1t_{i,j} \cdot K_{i,j}$$

$\alpha 1 := 120$ $\alpha 2 := 80$

$\beta 1 := 85$ $\beta 2 := 55$ $\beta 3 := 60$

Определим матрицу затрат на перевозку одной тонны

$$Zat\_1t := \begin{pmatrix} 12 & 3 & 6 \\ 2 & 7 & 5 \end{pmatrix}$$

Определим приближенную матрицу перевозок- количеств перевозимых грузов из пункта Ai в пунк Bj

$$K := \begin{pmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \end{pmatrix}$$

Рис. 54. Задание исходных данных в транспортной задаче

С помощью встроенной функции matrix сформируем матрицу из 2 строк и 3 столбцов, каждый элемент которой вычисляется с помощью функции f(i,j).

$$Zat := \mathrm{matrix}(2, 3, f)$$

$$Zat = \begin{pmatrix} 120 & 30 & 60 \\ 20 & 70 & 50 \end{pmatrix}$$

Теперь необходимо найти суммарные затраты на перевозку всех грузов. Именно эта величина должна быть сведена к минимуму.

Специальной матричной операции, находящей сумму всех элементов матрицы, нет, поэтому определим суммарные затраты следующим образом.

$$Sum\_Zat(K) := \left(\sum_{j=0}^{3} \sum Zat^{\langle j \rangle} \right)$$

Эта запись анналогична следующей.

$$Sum\_Zat(K) := \sum Zat^{\langle 0 \rangle} + \sum Zat^{\langle 1 \rangle} + \left(\sum Zat \right)^{\langle 2 \rangle}$$

Другими словами, находим сумму элементов каждого столбца матрицы Zat и затем все складываем.

Sum_Zat(K) – является функцией, зависящей от количества перевозимых тонн.

Теперь опишем ограничения, заданные по условию задачи.

Given

$$\sum K^{\langle 0 \rangle} = \beta 1$$

Суммы по столбцам дают потребность в грузах

$$\sum K^{\langle 1 \rangle} = \beta 2$$

$$\sum K^{\langle 2 \rangle} = \beta 2$$

$$\sum_{j=0}^{2} K_{0,j} = \alpha 1$$

Суммы по строкам дают количество грузов в пунктах производства

$$\sum_{j=0}^{2} K_{1,j} = \alpha 2$$

Минимизируем суммарные затраты, варьируя элементы матрицы К (рис. 55).

$$Kol := \mathrm{Minimize}(Sum\_Zat, K)$$

$$Kol = \begin{pmatrix} 36.389 & 41.696 & 41.915 \\ 48.611 & 13.304 & 18.085 \end{pmatrix}$$

Рис. 55. Результат решения транспортной задачи

Матрица Kol содержит такое количество тонн перевозимых грузов из пункта в пункт, при котором суммарные затраты на перевозку будут минимальными.

Пример. Раскрой ящика из железного листа.

а) Раскрой под заданный объем (рис. 56).

б) Раскрой ящика из железного листа с максимально возможным объемом (рис. 57).

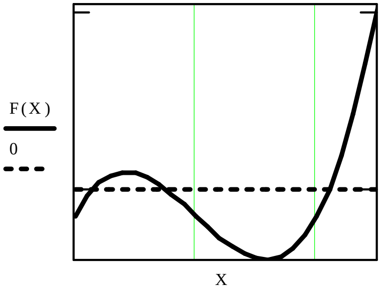
1. Раскрой под заданный объем $V0 := 7.5$

Ширина и длина листа: $W := 4$ $L := 8$

Объем ящика как функция от отгиба X:

$$V(X) := (L - 2 \cdot X) \cdot (W - 2 \cdot X) \cdot X$$

$$F(X) := V(X) - V0 \qquad X := 0, 0.2 .. 5$$



Графическое решение задачи - функция F(X)=V(X) - V0 пересекает линию нуля в трех точках, что дает три решения данной задачи

Решение задачи с применением функции root:

Начальное X	Численное решение	Комментарий
$X := 0$	$\text{root}\,(V(X) - V0, X) = 0.297$	Плоский ящик
$X := 1$	$\text{root}\,(V(X) - V0, X) = 1.5$	Глубокий ящик
$X := 4$	$\text{root}\,(V(X) - V0, X) = 4.203$	Физически нереальное решение (X<W/2)

Рис. 54.Поиск решения для задачи «Раскрой ящика под заданный объем»

Ширина и длина листа: $W := 4$ $L := 8$

Объем ящика как функция от отгиба X:

$$V(X) := (L - 2 \cdot X) \cdot (W - 2 \cdot X) \cdot X$$

2. Вычисление отгиба X для получения ящика максимального объем

$X := 1$ Начальное значение X

Given Открытие блока решения

Оптимизируемая функция V(X)

$V(X) = 100$ Заведомо большой объем

$XM := \text{Minerr}(X)$ Поиск X=XM для V(X)=max

$XM = 0.845$ Найденное значение XM

$V(X) = 12$ Максимальный объем ящика

Рис. 55. Вычисление высоты ящика для получения максимального объема

2.4.2. Построение модели по экспериментальным данным

Задача интерполяции

Часто теоретических знаний оказывается недостаточно для построения математической модели. В этих условиях можно воспользоваться данными, полученными экспериментально.

Пусть известна математическая модель $y = f(x, c)$ с точностью до параметров $c = (c_0, c_1, ..., c_m)$. Необходимо найти функцию, проходящую через точки $((x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$, где n – число экспериментов).

Для этого должно выполняться условие Лагранжа

$$\begin{cases} f(x_0, c) = y_0, \\ f(x_1, c) = y_1, \\ \quad . \quad . \quad . \quad . \quad . \quad , \\ f(x_n, c) = y_n. \end{cases} \tag{1}$$

Если число параметров функции $c_i, \ i = 0, 1, ..., m$ совпадает с числом узлов $x_j, \ j = 0, 1, ..., n,$ то система (1) может иметь решение. В этом случае говорят о *задаче интерполяции.*

Если $m > n,$ то число неизвестных в системе (1) больше числа уравнений, система недоопределена и для получения единственного решения необходимо задание дополнительных условий. Тогда можно поставить *задачу сплайн-интерполяции*.

Если же $m < n,$ то система (1) переопределена и точного выполнения условий Лагранжа добиться невозможно. В этом случае возникает задача аппроксимации.

Для интерполяции обычно используется полином

$$f(x, c) = P_n(x) = c_0 + c_1 x + ... + c_n x^n = \sum c_i x^i.$$

Условие Лагранжа дает следующую систему линейных алгебраических уравнений для определения коэффициентов интерполяции c:

$$\begin{cases} c_0 + c_1 x_0 + ... + c_n x_0^n = y_0, \\ . \quad . \quad . \quad . \quad . \ , \\ c_0 + c_1 x_n + ... + c_n x_n^n = y_n \end{cases} \tag{2}$$

или

$$Ax = y,$$

где x – вектор узлов интерполяции $x = (x_0, x_1, ..., x_n)$; y – вектор значений функции в узлах интерполяции $y = (y_0, y_1, ..., y_n)$;

$$A = \begin{bmatrix} 1 & x_0 & x_0^2 & ... & x_0^n \\ 1 & x_1 & x_1^2 & ... & x_1^n \\ . & . & . & . & . \\ 1 & x_n & x_n^2 & ... & x_n^n \end{bmatrix} = \left\{x_i^j\right\} i, j = 1, 2, ..., n. \qquad (3)$$

Решение системы (2) дает значение коэффициентов c.

$$c = A^{-1}y.$$

Полученное уравнение кривой проходит точно через заданные точки. Вне узлов интерполяции математическая модель может иметь значительную погрешность. Ниже приведен документ MathCAD с программой решения задачи интерполяции (рис. 58).

При большом числе узлов интерполяции приходится решать систему уравнений такой же размерности. Это может привести не только к эффекту волнистости, как на нижнем графике, но и к существенным ошибкам вычислений. В этом случае целесообразно использовать сплайн-интерполяцию.

Сплайн-интерполяция

Этот метод является формализацией чертежного опыта, в котором для проведения кривых линий используется гибкая рейка. Уравнение гибкой рейки

$$\frac{d^4y}{dx^4} = 0.$$

Решение этого дифференциального уравнения с точностью до констант.

$$y(x) = a + bx + cx^2 + dx^3.$$

Опишем поведение интерполируемой функции на каждом отрезке $[x_{i-1}, x_i]$, $i = \overline{1, n}$ сплайном

$$\varphi_i(x) = a_i + b_i x + c_i x^2 + d_i x^3.$$

Задача интерполяции

$x0 := 0 \quad xk := 1 \quad n := 6$ начало интервала, конец интервала и число узлов

$a := 1 \quad b := 2 \quad f(x) := \sin(a \cdot x) \cdot e^{-(b \cdot x)^{\frac{3}{2}}}$ исходная функция

$h := \dfrac{xk - x0}{n - 1}$ шаг интерполяции

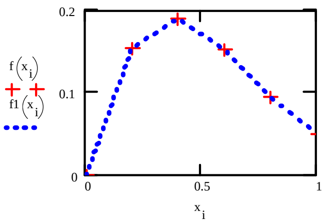
$i := 0 .. n - 1 \quad j := 0 .. n - 1$

$x_i := x0 + i \cdot h \quad g_i := f(x_i) \quad a_{i,j} := (x_i)^j$ формирование матрицы и вектора системы линейных уравнений

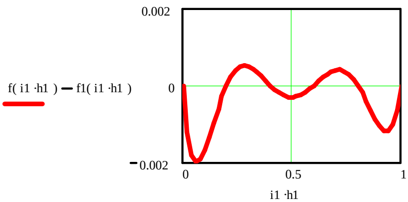
$c := a^{-1} \cdot g \quad c^T = (0 \quad 1.082 \quad -1.444 \quad -0.988 \quad 2.448 \quad -1.049)$ коэффициенты интерполяционного полинома

$f1(x) := \sum_i c_i \cdot x^i$ интерполяционный полином

$h1 := \dfrac{h}{10} \quad i1 := 0 .. (n - 1) \cdot 10$



сравнение исходной и интерполирующей кривой, x_i - аргумент функции $f(x_i)$ $f1(x_i)$



ошибка интерполяции

Рис. 56. Решение задачи интерполяции

Нам нужно определить $4n$ коэффициентов у n сплайнов, а условий Лагранжа только $2n$.

$$\varphi_i(x_{i-1}) = y_{i-1}, \quad \varphi_i(x_i) = y_i .$$

Недостающие условия можно получить, наложив требования на равенство первых и вторых производных для сплайнов во внутренних узлах интерполяции

$$\varphi_i'(x_i) = \varphi_{i+1}'(x_i), \quad \varphi_i''(x_i) = \varphi_{i+1}''(x_i), \quad i = \overline{1, n-1}$$

и равенство нулю вторых производных в начальном и конечном узлах (не закрепленная рейка ведет себя как прямая линия)

$$\varphi_1''(x_0) = 0, \quad \varphi_n''(x_n) = 0.$$

На рис. 59 приведен пример сплайн-интерполяции данных.

$x0 := 0 \quad xk := 1 \quad n := 6$ начало итервала, конец интервала и число узлов

$a := 1 \quad b := 2 \quad f(x) := \sin(a \cdot x) \cdot e^{-(b \cdot x)^{\frac{3}{2}}}$ исходная функция

$h := \dfrac{xk - x0}{n - 1}$ шаг интерполяции

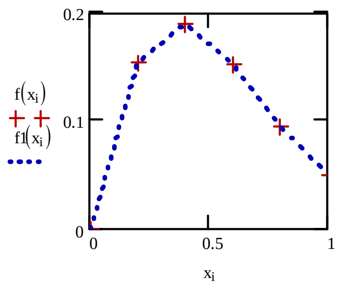
$i := 0 .. n - 1$

$x_i := x0 + i \cdot h \quad g_i := f(x_i)$ формирование исходных данных

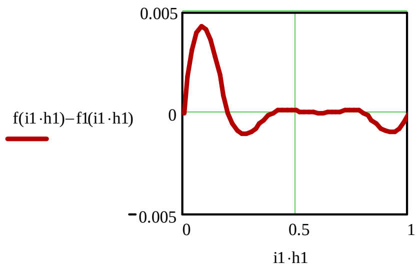
$v := \mathrm{lspline}(x, g)$ определение коэффициентов сплайна

$f1(z) := \mathrm{interp}(v, x, g, z)$ вычисление сплайна

$h1 := \dfrac{h}{10} \quad i1 := 0 .. (n - 1) \cdot 10$



сравнение исходной $f(x_i)$ и интерполирующей кривой $f1(x_i)$



ошибка сплайн-интерполяции

Рис. 57. Пример сплайн-интерполяции

2.4.3. Аппроксимация функций

При большом числе измерений и при наличии помех задача интерполяции становится лишенной практического смысла, так как число параметров в интерполяционной функции равно числу измерений, а присутствие ошибок измерений может привести к разным значениям функции при одном и том же значении аргумента функции. Далее будем предполагать известной математическую модель с точностью до констант c.

$$y = f(x_1, ..., x_n, c_1, ..., c_m).$$

Пусть имеются данные эксперимента для входных переменных x_k^i, $k = 1, 2, ..., n$, $i = 1, 2, ..., N$, где n – размерность вектора x; N – число экспериментов и соответствующие значения входных переменных x^i и выходной переменной y^i в эксперименте i. Подставляя эти числовые данные в уравнение для модели, получим систему уравнений

$$\begin{cases} y^1 = f(x_1^1, ..., x_n^1, c_1, ..., c_m), \\ y^2 = f(x_1^2, ..., x_n^2, c_1, ..., c_m), \\ . \quad . \quad . \quad . \quad . \quad . \quad ., \\ y^N = f(x_1^N, ..., x_n^N, c_1, ..., c_m). \end{cases} \tag{4}$$

При $m = N$ имеем замкнутую систему уравнений для определения параметров c в задаче интерполяции, модель будет проходить точно через экспериментальные точки. Однако для проверки соответствия модели реальности никакой информации не будет. Если $m < N$, то число условий избыточно и выбор m параметров вектора c удовлетворить всем N условиям (4) не может. Функцию невязки (меру уклонения) между экспериментальными данными и предсказанными по модели (3) можно представить в виде нормы.

$$Q(c) = \| y - f(x, c_1, ..., c_m) \|.$$

Наиболее часто используют квадратичную норму

$$Q^2(c) = \sum_{i=1}^{N} (y^i - f(x^i, c_1, ..., c_m))^2$$

и минимизация этой ошибки приводит к решению уравнения методом наименьших квадратов. В общем случае минимизация *Q* может быть осуществлена лишь численно.

Рассмотрим простой пример аппроксимации многомерной (двумерной) функции. Пусть в результате экспериментов получена таблица данных (табл. 4) и мы хотим получить модель в виде функции $y(x_1, x_2) = c_0 + c_1 x_1 + c_2\sqrt{x_2}$ (рис. 60).

Аппроксимация полиномиальной функцией

$x0 := 0 \quad xk := 1 \quad n := 10$ начало интервала, конец интервала и число узлов

$a := 1 \quad b := 2 \quad f(x) := \sin(a \cdot x) \cdot e^{-(b \cdot x)^{\frac{3}{2}}}$ исходная функция

$h := \dfrac{xk - x0}{n - 1} \quad i := 0..\, n - 1 \quad x_i := x0 + i \cdot h \quad g_i := f(x_i)$

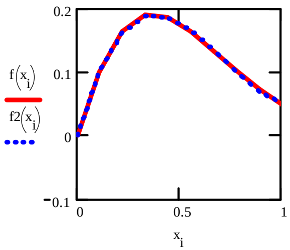
$m := 3$ порядок аппроксимирующего полинома

$p := 0..\, m \quad q := 0..\, m$

$\phi_{p,q} := \sum_i (x_i)^p \cdot (x_i)^q \quad b_p := \sum_i (x_i)^p \cdot f(x_i)$ формирование матрицы и вектора системы линейных уравнений

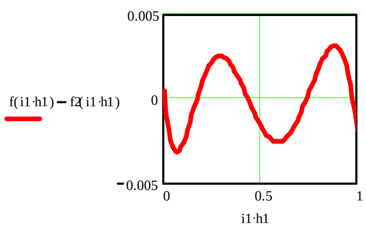
$d := \phi^{-1} \cdot b \quad d^T = \left[-4.355 \cdot 10^{-4} \;\; 1.138 \;\; -2.037 \;\; 0.952\right]$ коэффициенты аппроксимирующего полинома

$f2(x) := \sum_p d_p \cdot x^p$ апроксимирующий полином



сравнение исходной $f(x_i)$ и аппроксимирующей кривой $f2(x_i)$

$h1 := \dfrac{h}{10} \quad i1 := 0..\, (n - 1) \cdot 10$



ошибка аппроксимации

Рис. 58.Аппроксимация полиноминальной функции

Таблица 4

Экспериментальные данные

Номер экспери-мента i	Значе-ния x_1^i	Значения x_2^i	Значения y^i	Значения по модели	Ошибка ап-проксимации
1	0	0	0,5	0,391	0,109
2	0	1	1,2	1,318	–0,118
3	1	0	1,6	1,527	0,073
4	1	1	2,4	2,455	–0,055
5	2	1	3,4	3,591	–0,191
6	2	4	4,7	4,518	0,182

2.4.4. Адекватность математической модели

При решении технических задач с использованием математических моделей всегда необходимо рассматривать вопрос о соответствии или несоответствии математической модели объективной реальности, т.е. осуществлять проверку адекватности выбранной математической модели.

Несоответствие может быть вызвано двумя причинами: либо сама функция $f(x)$ не соответствует реальности, либо случайные факторы исказили результаты экспериментов, вследствие чего коэффициенты в модели определены с ошибками, что сказывается на точности моделирования.

Пусть по данным N экспериментов определяется модель с m параметрами. Если $N = m,$ то мы имеем задачу интерполяции, в которой модель точно проходит через экспериментальные точки, однако, мы не имеем ни одного дополнительного измерения для того, чтобы оценить адекватность модели. Если $N > m,$ то данные $N - m$ экспериментов (число степеней свободы) можно использовать для оценки адекватности модели при помощи остаточной дисперсии

$$S = \frac{\sum_{i=1}^{N}(y^i - f(x^i))^2}{N - m}.$$

Эта величина может быть признана неудовлетворительной, что приводит к необходимости увеличить число экспериментов (увеличив тем самым величину знаменателя) или уменьшить числитель, взяв другую функцию f и введя, например, поправочный член. Для того, чтобы выбрать решение, необходимо оценить, чем вызвана неадекватность модели: неудачной функцией *f* или ошибками в измерениях. Рассмотрим случай, когда выходная величина y измеряется с ошибками. Пусть ошибки измерений e подчиняются нормальному закону с нулевым математическим ожиданием $Me = 0$ (систематической ошибки нет) и дисперсией σ_y^2. Тогда необходимо оценить гипотезу о том, что ошибки моделирования с дисперсией S и измерений σ_y^2 взяты из одной выборки, тогда различия между этими величинами несущественны. Этот вопрос может быть разрешен с помощью отношения Фишера $F = \dfrac{S}{\sigma_y^2}$. Если $F > F_T(p, n, m)$, где *p* – уровень доверительной вероятности; n – число степеней свободы для большей дисперсии S; m – число степеней свободы (обычно равное бесконечности) для σ_y^2, то данная гипотеза должна быть отвергнута и ошибку моделирования можно уменьшить за счет изменения функции модели. В противном случае ошибка моделирования с вероятностью P может быть обусловлена ошибками измерений, и никакого улучшения моделирования за счет функции f добиться не удастся. Тогда нужно улучшать условия эксперимента. Табличное значение критерия Фишера $F_T(p, n, m)$ можно найти в книгах по статистике.

В Mathcad значение критерия вычисляется с помощью встроенной функции pF. Ниже приведен документ MathCAD, позволяющий аппроксимировать набор данных с помощью полинома и определить степень адекватности по критерию Фишера (рис. 61).

При $n = 4$ отношение Фишера *F* = 4,09 еще больше его табличного значения *F* = 2,67. В приведенном рисунке (рис. 61) соотношение вычисленного значения *F* = 2,62 и табличного *F* = 2,74 позволяет сделать вывод о том, что остаточная дисперсия сравнима с шумом измерений и дальнейшее улучшение модели за счет изменения функции бессмысленно.

? Вопросы для самопроверки

1. Какие бывают виды математических моделей?
2. Какой математический аппарат используется для описания статических моделей?

3. Какие результаты можно получить с помощью интерполяции?
4. В каких случаях целесообразно использовать сплайн-интерполяцию?
5. Какой метод используется в задачах аппроксимации?
6. Как проверить адекватность математической модели?

Проверка адекватности по критерию Фишера

$x0 := 0$ $xk := 1$ $n := 15$ начало интервала, конец интервала и число узлов

$a := 10$ $b := 2$ $f(x) := 5\cdot\sin(a\cdot x)\cdot e^{-(b\cdot x)^{\frac{3}{2}}}$

$h := \frac{xk - x0}{n - 1}$

$i := 0..n-1$ $D := 0.08$

$x_i := x0 + i\cdot h$ $g_i := f(x_i)$ $d1_i := g_i\cdot\left[1 + \left(rnorm(n, 0, D)_i\right)\right]$

искажение ошибками измерений с нулевым математическим ожиданием и дисперсией D

Аппроксимация

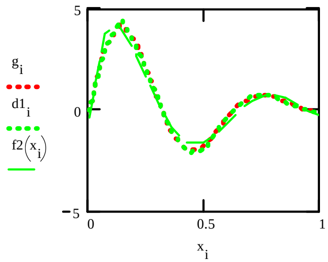
$m := 5$ порядок аппроксимирующего полинома

$p := 0..m$ $q := 0..m$

$\phi_{p,q} := \sum_i (x_i)^p\cdot(x_i)^q$ $b_p := \sum_i (x_i)^p\cdot d1_i$

коэффициенты аппроксимирующего полинома

$d := \phi^{-1}\cdot b$ $d^T = [-0.351\ \ 98.727\ \ -658.679\ \ 1.511\cdot10^3\ \ -1.444\cdot10^3\ \ 493.056]$ $f2(x) := \sum_p d_p\cdot x^p$



исходная g_i, искаженная $d1_i$ и аппроксимирующая $f2(x_i)$ функции

остаточная дисперсия критерий Фишера

$S := \frac{\sum_i (d1_i - f2(x_i))^2}{n - m - 1}$ $S = 0.21$ $F := \frac{S}{D}$ $F = 2.625$

$a := 1$ Given $0.995 = pF(a, n - m, 100)$ $FT := Find(a)$ $FT = 2.744$

Рис. 59. Проверка адекватности по критерию Фишера

Контрольные задания

1. Определить интерполяционный полином, считая, что измерения $y^* = (2{,}1;\ 6{,}8;\ 9{,}3;\ 7{,}2;\ 4.2)$ сделаны в узлах интерполяции $x = (0{,}1;\ 0{,}2;\ 0{,}3;\ 0{,}4;\ 0{,}5)$.

   Ответ. Коэффициенты интерполяционного полинома

$$c^T = \left[3.7\ \ -91.083\ \ 1.02\cdot10^3\ \ -2.942\cdot10^3\ \ 2.542\cdot10^3\right].$$

2.4.5. Примеры физических расчетов

Пример. Преодоление самолетом звукового барьера.

В физике известна функция *M (u, h),* определяющая так называемое число Маха, зависящее от скорости самолета *u* и высоты полета *h*. В момент преодоления звукового барьера значение *M (u, h)* становится равным 1. Скорость, на которой преодолевается этот барьер, падает с увеличением высоты полета.

Рис. 62 содержит определение функции *M (u, h).* График на рис. 63 содержит зависимости *M (u, h)* от скорости самолета *u* для четырех значений высоты полета *h,* а также уровень единицы. Звуковой барьер преодолевается в момент, когда соответствующая зависимости *M (u, h)* кривая пересекает уровень единицы.

Самолет преодолевает звуковой барьер, если M(u,h) достигает 1. Здесь u(км.час)-скорость полета, h(м)- высота полета. Функция M(u,h) вычисляется по формуле:

$a := 1222.5 \quad b := 6.875\cdot10^{-6} \quad c := 0.3048 \quad d := -5.2656 \quad e := 0.286$

$$M(u,h) := \sqrt{5\cdot\left[\left[\left[\left[\left[1+0.2\cdot\left(\frac{u}{a}\right)^2\right]^{3.5}-1\right]\cdot\left(1-b\cdot\frac{h}{c}\right)^d\right]+1\right]^e-1\right]}$$

Примеры вычислений

$u := 650$	$h := 7000$	$M(u,h) = 0.801$
$u := 900$	$h := 9000$	$M(u,h) = 1.203$

Рис. 60. Расчет скорости преодоления самолетом звукового барьера

Графики M(u,h) для 4 значений высоты h

$h1 := 1000 \quad h2 := 5000 \quad h3 := 10000 \quad h4 := 20000$

$u := 200, 250 .. 900$

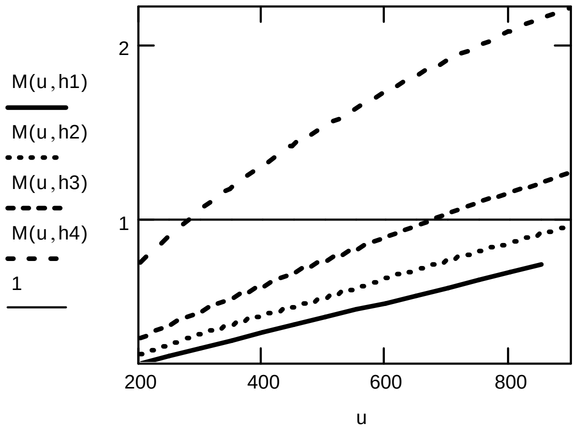


Рис. 61. Графики зависимостей M(u,h)

Более полезно не просто знать саму по себе зависимость *M (u, h),* а из условия *M (u, h) =1* получить зависимость скорости преодоления звукового барьера *u (h)* от высоты *h.* Для этого можно воспользоваться функцией поиска корня указанного условия (рис.64).

$i := 0 .. 10 \quad h_i := 2500 \cdot i u := 100 \qquad u_i := \mathrm{root}\left(M\left(u, h_i\right) - 1, u\right)$

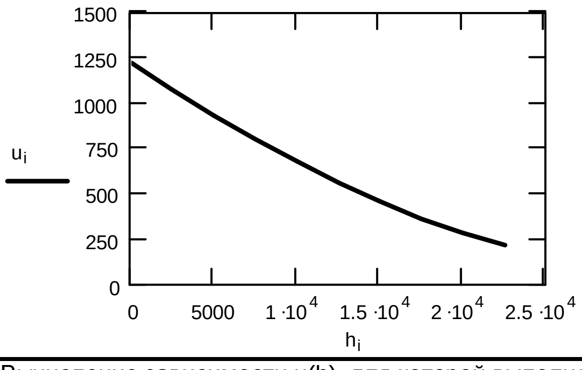


Рис. 62. Вычисление зависимости u(h), для которой выполняется условие M(u,h)-1=0

Этот пример дает наглядное представление о решении простых физических задач, базирующихся на уравнениях теоретической физики, и демонстрирует возможность находить с помощью MathCAD корень нелинейных уравнений.

2.4.6. Электро- и радиотехнические расчеты

Электро- и радиотехнические расчеты – весьма благодатная почва для приложения MathCAD. Некоторые из таких расчетов (например, вычисление емкости или индуктивности различных компонентов) достаточно просты и сводятся (за редкими исключениями) к вычислениям по готовым формулам.

В электро- и радиотехнике широко применяются комплексные числа и величины. MathCAD обладает всеми возможностями для работы с ними, начиная от вычислений импеданса (полного сопротивления) простых цепей и кончая работой с матрицами, имеющими комплексные элементы.

Пример. Вычисление характеристик системы по ее операторной функции.

Если схема описана операторной функцией, то по ней можно вычислить важнейшие характеристики линейной системы: АЧХ (амплитудно-частотными), ФЧХ (фазочастотными) и переходную характеристику (т.е. реакцию системы во времени на единичный скачок напряжения или тока на ее входе). Для вычисления АЧХ и ФЧХ достаточно заменить оператор р на величину *iw,* тогда модуль полученного выражения будет описывать АЧХ, а аргумент – ФЧХ.

Построение АЧХ и ФЧХ для системы второго порядка, описанной операторной функцией, дано на рис. 65.

Представлено также построение по операторной функции передаточной характеристики системы. Для каждого значения времени *t* эта характеристика определяется интегралом с бесконечным пределом интегрирования:

$$h(t) = \frac{2}{\pi}\int_0^\infty (\mathrm{Re}(y(j\omega)\sin(\omega t)/\omega)d\omega\,.$$

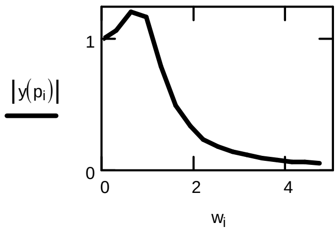
На рис. 66 показано построение переходной характеристики.

$$y(p) := \frac{1}{p^2 + 0.9 \cdot p + 1}$$

Для этого операторного выражения вычисляются АЧХ, ФЧХ и переходная характеристика линейной системы

$j := \sqrt{-1}$ $\qquad i := 0 .. 20$ $\qquad w_i := 0.314 \cdot i$ $\qquad p_i := w_i \cdot j$

АЧХ системы



ФЧХ системы

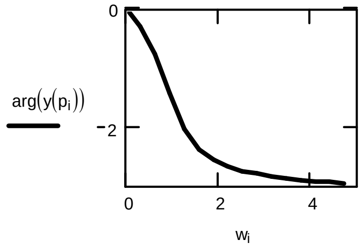


Рис. 63. Построение АЧХ и ФЧХ линейной системы по ее операторной характеристике

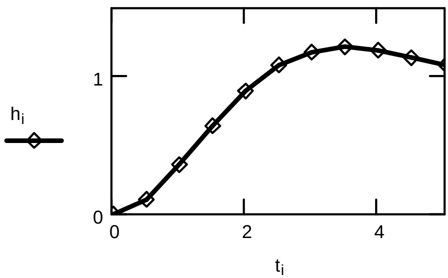
Построение переходной характеристики

$TOL := 0.01$

$$h(t) := 0.637 \cdot \int_0^5 Re(y(w \cdot j)) \cdot \frac{\sin(w \cdot t)}{w}\, dw$$

$i := 0 .. 10$ $\qquad t_i := 0.5 \cdot i$ $\qquad h_i := h(t_i)$

Переходная характеристика системы



$t_i =$	$h_i =$
0	0
0.5	0.105
1	0.349
1.5	0.636
2	0.885
2.5	1.073
3	1.173
3.5	1.206
4	1.186
4.5	1.135
5	1.079

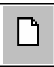
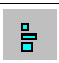
Рис. 64.Построение переходной характеристики

У всех реальных систем АЧХ на высоких частотах имеет спад и при больших *w>wmax* принимает близкие к нулю значения. В нашем примере это происходит уже при *w>5,* что позволяет заменить бесконечный предел интегрирования на конечный.

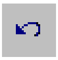
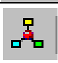
Вычисление *h (t)* на рис. 66 выполнено по этому алгоритму. Рядом с графиком *h(t)* выведена таблица времени *t* и значений *h(t).* Анализируемая цепь является цепью 2-го порядка и имеет соответствующие признаки заметно демпфированной цепи 2-го порядка – характерный подъем АЧХ на некоторой частоте и переходную характеристику с небольшим выбросом.

Приложение 1

Перечень основных пиктограмм панелей инструментов «Стандартная» и «Форматирование»

Вид пиктограммы	Описание
Операции с файлами	
	**New Worksheet** (Создать) – создание нового документа с очисткой окна редактирования.
	**Open Worksheet** (Открыть) – загрузка ранее созданного документа из диалогового окна.
	**Save Worksheet** (Сохранить) – запись текущего документа с его именем. Ниже мы рассмотрим эти операции более подробно.
Печать и контроль документов	
	**Print Worksheet** (Печать) – распечатка документа на принтере. **Print Preview** (Просмотр) – предварительный просмотр документа.
	**Check Spelling** (Проверка) – проверка орфографии документа. Назначение пиктограммы очевидно. Отметим лишь, что проверка орфографии действует только для англоязычных документов.
	Управление масштабом.
Кнопки операций редактирования	
	**Cut** (Вырезать) – перенос выделенной части документа в буфер обмена (Clipboard) с очисткой этой части документа.
	**Copy** (Копировать) – копирование выделенной части документа в буфер обмена с сохранением выделенной части документа.
	**Paste** (Вставить) – перенос содержимого буфера обмена в окно редактирования на место, указанное курсором мыши.
Кнопки размещения блоков	
	**Align Across** (Выровнять по горизонтали) – блоки выравниваются по горизонтали.
	**Align Down** (Выровнять вниз) – блоки выравниваются по вертикали, располагаясь сверху вниз.

Окончание прил. 1

Вид пиктограммы	Описание
Кнопки операций с выражениями	
	**Insert Function –** вставка функции из списка, появляющегося (Вставить функции) в диалоговом окне. MathCAD имеет множество встроенных функций, от элементарных до сложных статистических и специальных математических. Синтаксис их записи порой легко забывается. Поэтому возможность вставки функции с помощью кнопки **Insert Function** очень удобна.
	**Insert Unit** (Вставить единицы) – вставка единиц измерения. Эта кнопка позволяет вставить нужную единицу измерения.
	**Calculate** (Пересчитать) – пересчет выделенного выражения. Этой кнопкой можно воспользоваться в одном из двух случаев: если отключен режим автоматического вычисления, и если документы большие, то при их изменении не всегда выгодно запускать вычисления с самого начала. Операция **Calculate** (Пересчитать) позволяет запускать вычисления для выделенных блоков, что может уменьшить время вычислений.
	**Undo** (Отменить) – отмена предшествующей операции редактирования.
Пиктограммы работы со справочной базой данных	
	Быстрый вызов шпаргалок – примеров применения системы.
	Вызов справочной системы MathCAD.
	Включение гиперссылки.
	Создание проекта, объединяющего данные из нескольких приложений.
	Вставка дополнительных компонентов.

Приложение 2

Некоторые функции, реализованные в системе MathCad

Тригонометрические функции	
Название функции	**Описание**
sin(z)	Синус
cos(z)	Косинус
tan(z)	Тангенс
sec(z)	Секанс
csc(z)	Косеканс
cot(z)	Котангенс
Гиперболические функции	
sinh(z)	Гиперболический синус
cosh(z)	Гиперболический косинус
tanh(z)	Гиперболический тангенс
sech(z)	Гиперболический секанс
csch(z)	Гиперболический косеканс
coth(z)	Гиперболический котангенс
Обратные тригонометрические функции	
asin(z)	Обратный тригонометрический синус
acos(z)	Обратный тригонометрический косинус
atan(z)	Обратный тригонометрический тангенс
Обратные гиперболические функции	
asinh(z)	Обратный гиперболический синус
acosh(z)	Обратный гиперболический косинус
atanh(z)	Обратный гиперболический тангенс
Показательные и логарифмические функции	
exp(z)	Экспоненциальная функция
ln(z)	Натуральный логарифм (по основанию *e)*
log(z)	Десятичный логарифм (по основанию 10)
Функции комплексного аргумента	
Re(z)	Выделение действительной части комплексного числа
Im(z)	Выделение мнимой части комплексного числа
arg(z)	Вычисление аргумента (фазы)

Окончание прил.2

К важнейшим встроенным специальным математическим функциям принадлежат ***функции Бесселя***, являющиеся решениями дифференциального уравнения второго порядка. Функции Бесселя описывают колебательные процессы и широко используются в физике и электро- и радиотехнике.

Функции Бесселя	
J0(x)	Функция Бесселя первого рода нулевого порядка
I0(x)	Модифицированная функция Бесселя первого рода нулевого порядка
Y0(x)	Функция Бесселя второго рода нулевого порядка
K0(x)	Модифицированная функция Бесселя второго рода нулевого порядка
J1(x)	Функция Бесселя первого рода первого порядка
I1(x)	Модифицированная функция Бесселя первого рода первого порядка
Y1(x)	Функция Бесселя второго рода первого порядка
K1(x)	Модифицированная функция Бесселя второго рода первого порядка
Jn(n,x)	Функция Бесселя первого рода n-го порядка
In(n,к)	Модифицированная функция Бесселя первого рода n-го порядка
Yn(n,c)	Функция Бесселя второго рода n-го порядка
Kn(n,x)	Модифицированная функция Бесселя второго рода n-го порядка.

Гамма-функция

Другой, широко распространенной специальной функцией, вычисление которой (причем как при действительном, так и комплексном аргументе *z)* предусмотрено в системе MathCAD, является гамма-функция ***Gamma(a,z).*** Она широко применяется и в статистических расчетах. В них используется также функция ошибок ***erf(x),*** называемая еще интегралом вероятности.

Приложение 3

Функции работы с матрицами

Название	Описание
Векторные функции	
length(V)	Возвращает длину вектора
last(V)	Возвращает индекс последнего элемента
max(V)	Возвращает максимальный по значению элемент
min(V)	Возвращает минимальный по значению элемент
Re(V)	Возвращает вектор действительных частей вектора с комплексными элементами
Im(V)	Возвращает вектор мнимых частей вектора с комплексными элементами.
Матричные функции	
augment(A,B)	«Сливает две матрицы», т.е. формирует матрицу, в первых столбцах которой содержится матрица A, в последних матрица B. Матрицы A и B должны содержать одинаковое количество строк
identity(n)	Создает единичную квадратную матрицу размером *n*n*
stack(A,B)	Формирует матрицу, в первых строках которой содержится матрица A, в последних B. Матрицы A и B должны иметь одинаковое количество столбцов
submatrix (A,ir,jr,ic,jc)	Формирует матрицу, которая является подблоком матрицы A, расположенным в строках с ***ir*** по ***jr*** и в столбцах с ***ic*** по ***jc***
diag(V)	Создает диагональную матрицу, элемент главной диагонали которой вектор V
matrix(m,n,f)	Создает и заполняет матрицу размерностью m*n. Элемент матрицы, расположенный в i-й строке и j-м столбце равен значению f(i,,j)
Re(M)	Возвращает матрицу действительных частей матрицы M с комплексными элементами
Im(M)	Возвращает матрицу мнимых частей матрицы M с комплексными элементами

Название	Описание
Функции, возвращающие специальные характеристики матриц	
ols(M)	Возвращает число столбцов матрицы M
rows(M)	Возвращает число строк матрицы M
rank(M)	Возвращает ранг матрицы M
tr(M)	Возвращает след (сумму диагональных элементов) квадратной матрицы M
mean(M)	Возвращает среднее значение элементов массива M
median(M)	Возвращает медиану элементов массива M
cond1(M)	Возвращает число обусловленности матрицы, вычисленное в норме L1
cond2(M)	Возвращает число обусловленности матрицы, вычисленное в норме L2
conde(M)	Возвращает число обусловленности матрицы, вычисленное в норме евклидова пространства
condi(M)	Возвращает число обусловленности матрицы, основанное на равномерной норме
norm1(M)	Возвращает L1, норму матрицы M. $\|x\| = \sum_{i=1}^{n} \lvert x_i \rvert$
norm2(M)	Возвращает L2, норму матрицы M. $\|x\| = \sqrt{\sum_{i=1}^{n} \lvert x_i \rvert^2}$
пorme(M)	Возвращает евклидову норму матрицы M
normi(M)	Возвращает неопределенную норму матрицы M
lsolve(A,B)	Функция, возвращающая решение системы линейных неоднородных алгебраических уравнений вида $Ax = B$

Приложение 4

Перечень основных ошибок в системе MathCAD

Сообщение об ошибке	**Комментарий**
Array size mismatch	Несоответствие размера массива
Cannot be defined	Не может быть определено
Cannot take subscript	Не содержит верхних (нижних) индексов
Did not find solution	Решение не найдено
Dimension to non real power	Размерность массива – не целое число
Duplicate	Дублирование
Equation too large	Слишком большое выражение
Error in constant	Ошибка в константе
Error in solve block	Ошибка в блоке решения
Illegal array operation	Неверная операция с массивом
Illegal function name	Неверное имя функции
Illegal range	Неправильный диапазон
Index out of bounds	Индекс вне границ
Misplaced comma	Неуместная запятая
Missing operand	Пропущенный операнд
Must be integer	Должно быть целым
Must be non zero	Должно быть ненулевым
Must be positive	Должно быть положительным
Must be range	Должен быть диапазон
No matching Given	Нет соответствующего Given
No scalar value	Нескалярная величина
Overflow	Переполнение
Singularity	Деление на ноль
Too few arguments	Слишком мало аргументов
Too few constraints	Слишком мало ограничений
Too few subscripts	Мало нижних индексов
Too many arguments	Слишком много аргументов
Undefined	Не определено
Unmatched parenthesis	Несоответствие скобок

Библиографический список

1. Дьяконов В.П., Абраменкова И.В. MathCad 7.0 в математике, физике и в Internet. – М.: Нолидж, 1998. – 352 с., ил.
2. Дьяконов В.П. Справочник по. MathCAD PLUS 6.0 PRO. – М.: СК Пресс, 1997. – 336 с., ил.
3. MathCAD 6.0 PLUS. Руководство пользователя / пер.с англ. М.: Филин, 1996. – 712 с., ил.
4. Очков В.Ф. MathCAD 6.0 PLUS для студентов и инженеров. М.: СК Пресс, 1996. – 238 с., ил.
5. Макаров Е.Г. Инженерные расчеты в Mathcad. – С-Пб.: Питер, 2002. – 503 с., ил.
6. Макаров Е.Г. Инженерные расчеты в Mathcad 14. – С-Пб.: Питер, 2007. – 405 с., ил.

СОДЕРЖАНИЕ