КИПР
ТУСУР

Кафедра конструирования
и производства радиоаппаратуры

УТВЕРЖДАЮ
Заведующий кафедрой КИПР
_______________**В.Н. ТАТАРИНОВ**
“___” ____________2012 г.

Применение системы автоматизации научно-технических расчетов MathCAD при проектировании РЭС

Методическое пособие для студентов специальностей 211000.62 (бакалавриат) и 162107.65 «Информатика и информационные технологии» (специалитет)

Разработчик:
Доцент кафедры КИПР
_____________**Ю.П. Кобрин**

Томск 2012

СОДЕРЖАНИЕ

1 Введение в MathCAD

1.1 Назначение

Универсальная система автоматизации инженерно-конструкторских расчетов MathCAD [1] [2] [3] [4] [5] [6] [7] [8] позволяет решать широкий спектр разнообразных научных и инженерных расчетов, как простейших - с элементарной арифметикой, так и со сложнейшими реализациями численных методов. MathCAD имеет удобный интуитивный графический интерфейс с пользователем. Входной язык MathCAD прост, предельно приближен к обычному языку описания математических задач, имеет развитую библиотеку встроенных функций и численных методов. Отличительные черты - возможность вычислений в символьной форме, а также наглядность представления результатов (графики всевозможных видов).

Это позволяет студентам, будущим инженерам-радиоэлектроникам, находить с помощью MathCAD решение многих учебных и профессиональных задач:

- ✓ проводить немедленно (как только введена формула) разнообразные математические вычисления, в которых участвуют числовые константы и переменные действительного и комплексного типа;
- ✓ решать разнообразные алгебраические задачи, в том числе с векторами и матрицами;
- ✓ решение уравнений и систем уравнений (неравенств);
- ✓ разлагать функции в ряд Тейлора и Фурье, осуществлять преобразования Фурье и Лапласа;
- ✓ производить дифференцирование и интегрирование функций (аналитическое и численное);
- ✓ выполнять тождественные преобразования аналитических выражений (в том числе упрощение), аналитическое решение уравнений и систем;
- ✓ решать системы дифференциальных уравнений;
- ✓ проводить статистические расчеты и анализ данных;
- ✓ производить линейную и сплайновую аппроксимацию функций заданных по точкам;
- ✓ решать задачи линейного и нелинейного программирования по поиску экстремумов целевой функции в заданном пространстве проектирования (оптимизация);
- ✓ строить двумерные и трехмерные графики с результатами расчетов.

1.2 Интерфейс MathCAD

В среде MathCAD используется технология WYSIWYG (англ. What You See Is What You Gel — как видишь на экране, так и выводится на печать): математические формулы присутствуют в расчете не в виде мало понятного линейного текста с множеством скобок (как, например, в Паскале), а в виде естественного «многоэтажного» математического выражения.

Интерфейс MathCAD по своей структуре похож на интерфейс многих других Windows-приложений (Рис. 1.1).

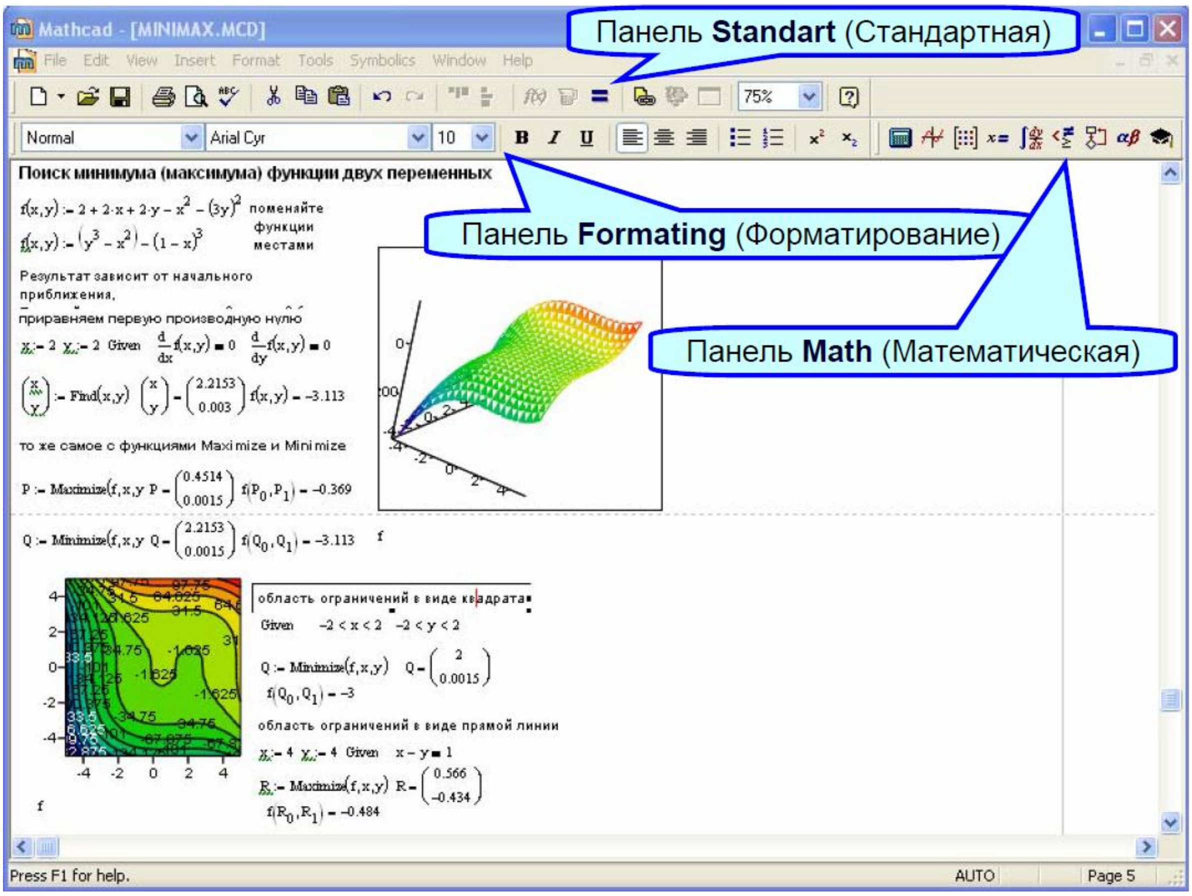


Рис. 1.1 - Вид рабочего окна MathCAD (английская версия)

Сверху вниз располагаются заголовок окна с именем рабочего документа, строка меню и панели инструментов (стандартная и форматирования)[1], которые предназначены для выполнения наиболее часто применяемых команд (Таблица 1.1) и Рабочая область документа (worksheet). Новый

[1] Пользователь может самостоятельно настраивать внешний вид рабочего окна. В целях экономии места на экране каждая панель может либо выведена на экран, либо скрыта с помощью соответствующей команды из меню View (Вид) / Toolbars (Панели).

документ создается автоматически при запуске MathCAD. В самой нижней части окна находится строка состояния.

Пространство документа разбито с помощью тонких серых штриховых линий на прямоугольные участки, соответствующие по размерам формату А4. Маленький красный крестик + (курсор ввода) указывает место, куда будет вставлен любой объект: формула, текст, график или картинка. Переместить его можно простым щелчком мышью на нужном фрагменте листа.

Большинство необходимых действий можно выполнить активизируя команды выпадающих подменю, последовательно открывающихся от щелчка мышью на любом пункте Главного меню (Таблица 1.1).

Таблица 1.1 - Назначение клавиш главного меню

Меню	Назначение
File (Файл)	**команды, связанные с созданием, открытием, сохранением, пересылкой по электронной почте и печатью на принтере файлов с документами**
Edit (Правка)	**команды, относящиеся к правке текста (копирование, вставка, удаление фрагментов и т. д.)**
View (Вид)	**команды, управляющие внешним видом документа в рабочем окне MathCAD, а также команды создания файлов анимации**
Insert (Вставка)	**команды вставки различных объектов в документ**
Format (Формат)	**команды форматирования текста, формул и графиков**
Tools (Инструменты)	**команды управления вычислительным процессом**
Symbolics (Символические вычисления)	**команды символьных вычислений**
Window (Окно)	**команды расположения окон с различными документами на экране**
Help (Помощь)	**команды вызова справочной информации**

Для быстрого выполнения часто применяемых команд предусмотрены девять специальных математических панелей инструментов (Рис. 1.2). При необходимости любую панель инструментов вызывают либо с помощью меню View ⇒ Toolbars (Вид ⇒Панели инструментов) или кнопками управляющей панели Math (Математика) , содержащей инструменты для вставки в документы математических объектов (операторов, графиков, элементов программ и др.). Нажатие любой кнопки вызывает определенную рабочую математическую панель.

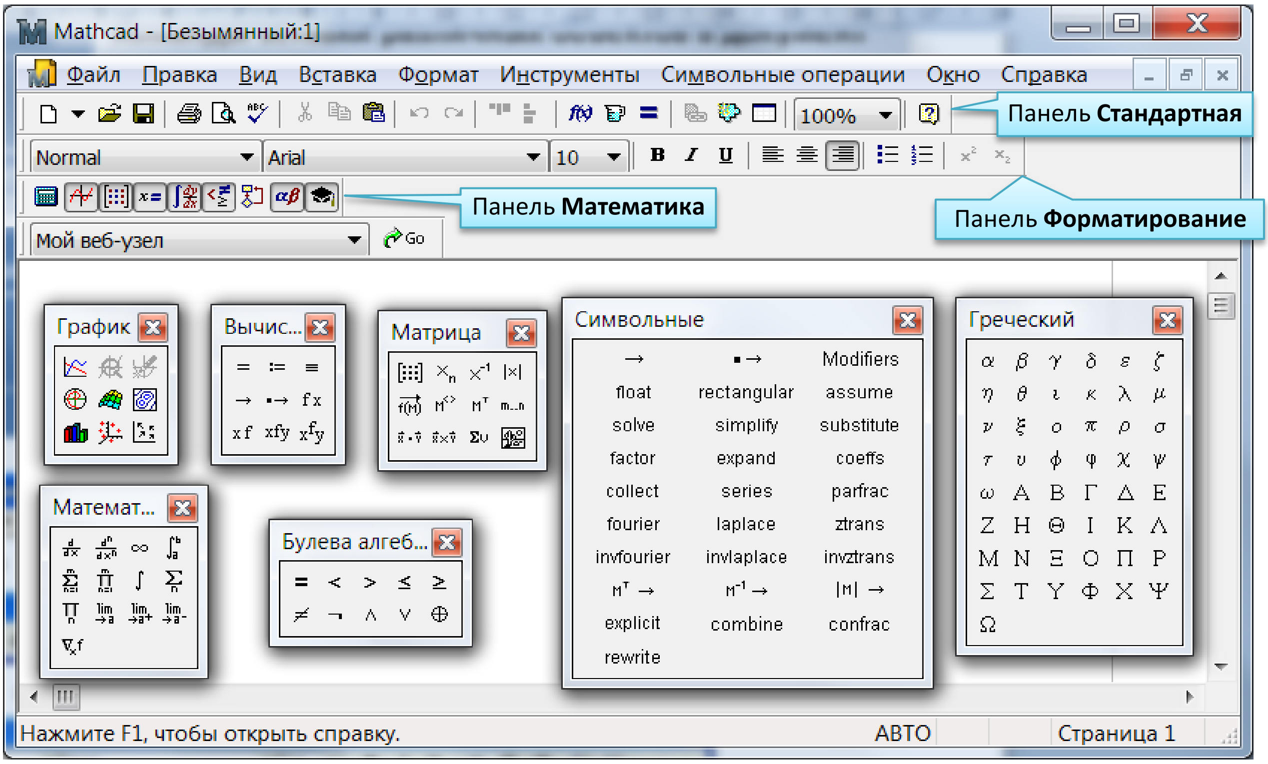


Рис. 1.2 - Главное меню и некоторые панели MathCAD (с руссификацией)

Таблица 1.2 представляет назначение важнейших из этих панелей.

Таблица 1.2 - Назначение важнейших панелей MathCAD

Панель	Назначение
Standard (Стандартная)	**для выполнения действий с файлами, редактирования документов, вставки объектов и т.д.**
Formatting (Форматирование)	**для форматирования текста и формул**
Math (Математическая)	**для вставки математических символов и операторов в документы**
Controls (Контроль)	**для дополнительного контроля над работой** MathCAD **- документа**
Resources (Документация)	**для трассировки выполнения программ**
Graph (График)	**вставка графиков**
Calculator (Калькулятор)	**вставка чисел, основных математических операций, знаков арифметических операций и наиболее употребительных стандартных функций**
Matrix (Матрица)	**вставка шаблонов матриц и матричных операций**
Evaluation (Оценка)	**операторы присвоения значений и вывода результатов расчета**
Greek	**вставка греческих букв**

Панель	Назначение
(Греческие буквы)	
Calculus (Вычисления)	вставка шаблонов дифференцирования, интегрирования, суммирования.
Boolean (Булевы операторы)	вставка логических (булевых) операторов
Symbolics (Символика)	вставка операторов символьных вычислений
Programming (Программирование)	операторы, необходимые для создания программных модулей
Debug	отладки программ и включения (остановки) процесса трассировки (вывода промежуточных результатов расчета на экран).

При наведении курсора на любую кнопку MathCAD появляется всплывающая подсказка с названием операции и сочетанием клавиш, эквивалентным щелчку по данной кнопке. Кроме того, Вы можете перемещать панели в любое место экрана и изменять их форму, а также делать некоторые панели плавающими и наоборот.

2 Основные приемы работы с MathCAD

2.1 Начало работы

Входной язык MathCAD является языком интерпретирующего типа. Когда MathCAD распознает какой-либо объект, он немедленно исполняет указанные в блоке операции.

В системе MathCAD существуют формульные, текстовые и графические объекты. Их можно выделять пунктирными прямоугольниками и перетаскивать (по одному или сразу несколько блоков) с помощью мыши. При этом формульные блоки могут иметь особые признаки - атрибуты, например, активности, пассивности и оптимизации.

Формулы вычисляются с использованием числовых констант, переменных, функций (стандартных и определенных пользователем), а также общепринятых обозначений математических операций. Введенные в документ MathCAD формулы автоматически приводятся к стандартной научно-технической форме записи. Графики, которые автоматически строятся на основе результатов расчетов, также рассматриваются как формулы.

Обратите внимание! В ходе расчетов формулы обрабатываются последовательно построчно: слева направо и сверху вниз.

В текстовых блоках размещаются комментарии, описания и иллюстрации, которые игнорируются при проведении расчетов.

Таблица 2.1 показывает некоторые особенности интерфейса редактора MathCAD при вводе формул.

Таблица 2.1 - Элементы интерфейса MathCAD при вводе формул

	указатель мыши - следует за ее движениями
крестообразный курсор	если ни один объект не выбран, определяет место, куда будут вводиться формулы или текст
уголковый курсор	охватывает в вводимом тексте или формуле определенную часть.
текстовый курсор	появляется при вводе данных в текстовый блок (вертикальная черта).
	местозаполнители (маркеры) внутри незавершенных формул (в местах, которые должны быть заполнены символом или оператором): черный прямоугольник - местозаполнитель фрагмента, в который следует ввести переменную или операнд, черная прямоугольная рамка - местозаполнитель оператора

2.2 Ввод формул

Формула в *MathCAD* очень похожа на оператор присваивания в языке Pascal:

<***Имя*** > ***:=*** <***выражение***>

где:

<***Имя*** > – имя переменной, значение которой изменяется в результате выполнения оператора присваивания; переменные помогают делать расчеты более простыми, понятными и компактными, поэтому без них не обходится решение ни одной более или менее сложной задачи.

: = - символ присваивания;

<***выражение***> - выражение, значение которого присваивается переменной, имя которой указано слева от символа оператора присваивания.

Имя выражения (все, что стоит слева от оператора присваивания) может состоять из латинских, русских, греческих и других букв и цифр, знаков подчеркивания (_), штриха (`), символа процента (%), знака бесконечности (∞), вводимых с клавиатуры.

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

ВНИМАНИЕ! MathCAD воспринимает заглавные и строчные буквы как различные идентификаторы. То же касается букв, изображенных различными шрифтами и разными стилями – это разные имена!

Большинство численных методов, запрограммированных в MathCAD, реализовано в виде встроенных функций. MathCAD не делает различий между именами переменных и функций. И если определить вначале функцию *f(x)*, а затем переменную *f*, то окажется невозможным использовать *f(x)* в последующих расчетах где-либо ниже определения *f*.

MathCAD содержит огромное число (несколько сотен) встроенных функций. На стандартной панели нажмите кнопку f(x) (Рис. 2.1). В диалоговом окне Вставить функцию (Insert Function) Вы увидите список всех встроенных функций. Просмотрите категории функций (левый список) и списки функций в каждой категории (правый список), чтобы представить себе, какие специальные функции и численные методы можно использовать в расчетах.

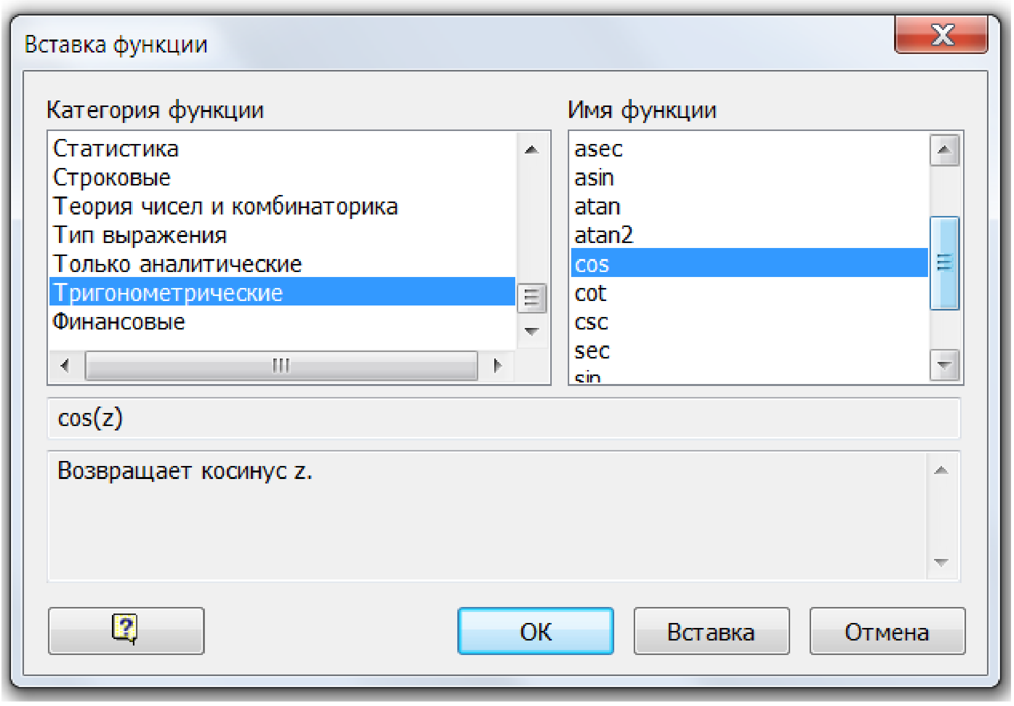


Рис. 2.1 - Диалоговое окно Вставить функцию (Insert Function)

Некоторые имена уже используются MathCAD для встроенных констант, единиц измерения, функций. Их можно переопределить, однако тогда уничтожаются их встроенные значения. Далее этими константами и функциями по первоначальному назначению пользоваться будет нельзя.

Математические выражения набираются частично с клавиатуры, частично путем вставки шаблонов операций с математических панелей.

Команды операций в MathCAD продублированы. Их можно ввести по разному: либо выбрав соответствующий пункт меню, либо нажав соответствующую клавишу или сочетание клавиш на клавиатуре. Так, чтобы ввести пару круглых скобок, просто нажмите **'**.

Рис. 2.2 - Панель Калькулятор (Calculator)

Переменную, которой присвоено значение, можно использовать далее в вычисляемых выражениях.

Оператор присваивания (:=) можно ввести с математической панели Калькулятор (Calculator) (Рис. 2.2).

Проанализируйте, в каких еще случаях целесообразно применять эту панель.

Для еще более быстрого набора оператора присваивания достаточно набрать на клавиатуре «двоеточие» (**:**). Кстати, если ввести «**=**» после еще неопределенной переменной, то также отобразится символ присвоения **:=** .

Чтобы узнать значение переменной (или выражения), следует использовать оператор вычисления - нажмите клавишу (**=**) и получите численный результат.

Чтобы выполнить несложные расчеты по формулам, нужно произвести следующее:

✓ определить место в рабочем окне MathCAD, где должно появиться выражение, щелкнув мышью в соответствующей точке;

✓ ввести левую часть выражения (частный случай выражения - переменная);

✓ ввести знак равенства <**=**>.

При выполнении примера (Рис. 2.3) щелкаем на любом месте экрана левой кнопкой мыши и присваиваем значения переменным x и y.

$$x := 3 \quad y := 2 \quad x + \frac{y}{2.5} = 3.8 \quad \sqrt[3]{x} = 1.442 \quad x^{\frac{1}{3}} = 1.442$$

$$\sin(x - y) = 0.841 \quad \pi\frac{(x+y)}{(x-y)} = 15.708 \quad \int_0^y \cos(x)\,dx = 0.909$$

Рис. 2.3 - Примеры набора и вычисления простейших выражений

Большинство символов, например латинские буквы или цифры, знаки математических операций, набираются на клавиатуре. Названия функций можно вводить из раскрытого диалогового окна Вставка функций (Рис. 2.1), выделив название функции и нажав кнопку Вставка (Insert), или набирая имя функции на клавиатуре в точности так, как оно записано в окне функций.

Греческие буквы, например π, легче всего вставить с помощью панели инструментов Греческий (Greek) (Рис. 2.4). Заметим, что по умолчанию π и e являются математическими константами с начальными значениями:

π = 3.1415926535897931,

e = 2.7182818284590451.

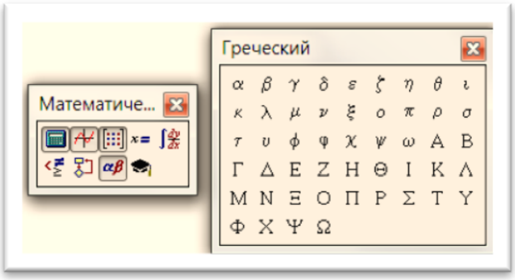
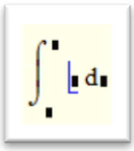


Рис. 2.4 - Вызов панели Греческий (Greek)

Для ввода символа интегрирования вызываем панель Математический анализ (Calculus) нажатием на стандартной панели инструментов кнопки  (Рис. 2.5) и после появления шаблона интеграла заполняем соответствующие местозаполнители.



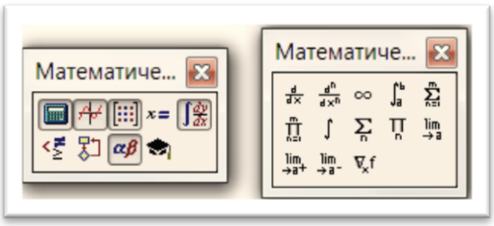


Рис. 2.5 - Вызов панели Математический анализ (Calculus)

В MathCAD существует возможность определять и пользовательские функции одного или нескольких аргументов. Для этого необходимо указать имя функции с аргументами в скобках, набрать символ <:> и определить новую функцию. В примере (Рис. 2.6) вначале присвоены значения параметрам функции a и b, а затем определяется пользовательская функция f(x). Её можно использовать в любых расчетах. В примере показано, как получить результат вычисления функции для аргумента x = 1.

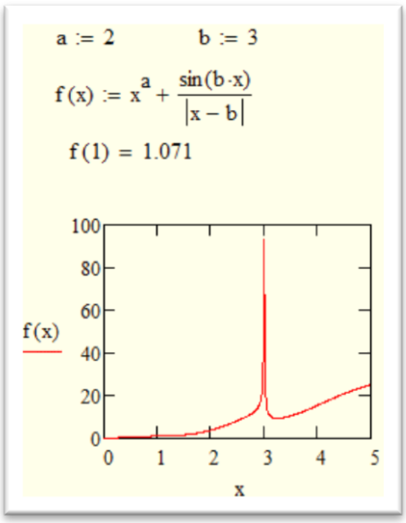


Рис. 2.6 - Определение функции пользователя и вывод ее графика

Для создания графика функции f(x) щелкнем мышью на том свободном месте, где его нужно разместить. На стандартной панели инструментов Математические (Math) нажмем кнопку для вызова панели График (Graph) и на появившейся панели выберем кнопку с нужным типом графика (для построения графика функции одного аргумента нажмём кнопку).

В появившейся пустой заготовке графика с полями ввода для данных (**Ошибка! Источник ссылки не найден.**) достаточно определить значения, которые будут отложены по осям. В нашем случае потребовалось ввести x в местозаполнитель под серединой оси абсцисс и f(x) в поле напротив середины оси ординат. Остальные поля предназначены для ввода границ на осях - максимального и минимального значений, откладываемых на оси. Если оставить местозаполнители диапазона аргумента пустыми, MathCAD автоматически заполнит их при создании графика.

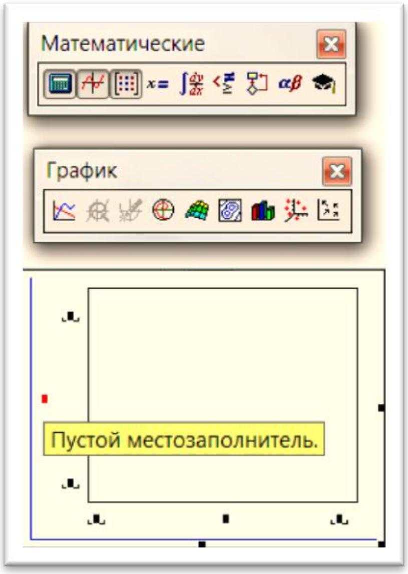


Рис. 2.7 - Определение функции пользователя и вывод ее графика

Если диапазон значений аргумента не задан, по умолчанию график будет построен в диапазоне значений аргумента от –10 до 10 . При необходимости диапазон изменения аргумента x несложно переопределить.

2.3 Редактирование формул

Редактирование введенных выражений производится обычным для всех Windows-приложений способом.

Чтобы изменить формулу, щелкните на ней мышью, поместив в ее область линии ввода, и перейдите к месту, которое хотите исправить.

Перемещайте линии ввода в пределах формулы одним способов:

- ✓ щелкая в нужном месте мышью;
- ✓ нажимая на клавиатуре клавиши со стрелками, которые переводят линии ввода вверх, вниз, влево или вправо;
- ✓ клавиша <**Ins**> переводит вертикальную линию ввода с одного конца горизонтальной линии ввода на противоположный. Например, для вставки оператора отрицания;

✓ пробел предназначен для выделения различных частей формулы.

Нередко поместить линии ввода в нужное место формулы с помощью указателя мыши непросто, поэтому в MathCAD для этого лучше использовать клавиатуру.

Если раз за разом нажимать клавишу пробела в формуле, то линии ввода будут циклически изменять свое положение, как это показано на Рис. 2.8. Если в ситуации, показанной сверху на этом рисунке, нажать стрелку ←, то линии ввода переместятся влево (Рис. 2.9). При нажатии пробела теперь линии ввода будут попеременно выделять одну из двух частей формулы.

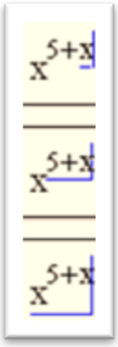


Рис. 2.8 - Изменение положения линий ввода с помощью пробела

Комбинация клавиш со стрелками и пробелом позволяет легко перемещаться внутри формул.

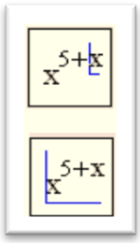


Рис. 2.9 - Изменение положения линий ввода пробелом после сдвига стрелкой ←

Чтобы вставить оператор в формулу, необходимо:

✓ поместить линии ввода на часть формулы, которая должна стать первым операндом;

✓ ввести оператор, нажав кнопку на панели инструментов или сочетание клавиш.

На Рис. 2.10 показано несколько примеров вставки оператора сложения в разные части формулы. В левой колонке приведены возможные размещения линий ввода в формуле, а в правой — результат вставки оператора сложения (т.е. нажатия клавиши <+>). Заметим, что MathCAD при необходимости сам расставляет скобки, чтобы часть формулы, отмеченная линиями ввода, стала первым слагаемым.

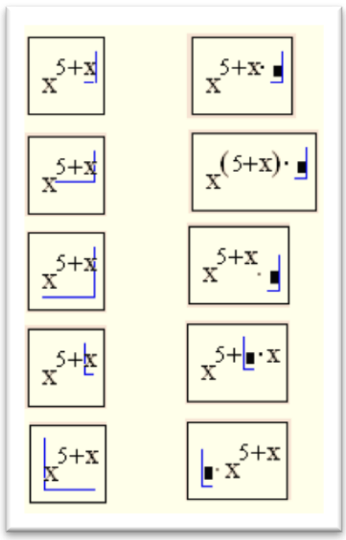


Рис. 2.10 - Вставка оператора в разные части формулы

Для выделения части формулы в некоторой математической области необходимо (Рис. 2.11):

- ✓ поместить ее между линиями ввода, пользуясь, при необходимости, клавишами-стрелками и пробелом;
- ✓ поместите указатель мыши на вертикальную линию ввода, нажать и удерживать левую кнопку мыши;
- ✓ удерживая кнопку мыши, протащить указатель мыши вдоль горизонтальной линии ввода (при этом часть формулы будет выделяться обращением цвета);
- ✓ отпустить кнопку мыши, когда будет выделена нужная часть формулы.

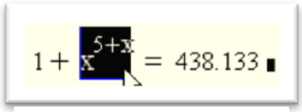


Рис. 2.11 - Выделение части части формулы

Часть формулы можно выделить и без помощи мыши, нажимая клавиши со стрелками при удерживаемой клавише <Shift>.

Выделенную черным фоном область в MathCAD можно использовать для вырезания или копирования части выражения, изменения шрифта, а также для выполнения символьных вычислений с частями выражений. Проще всего для выполнения подобных операций использовать контекстное меню, которое, как обычно, вызывается правой кнопкой мыши. В возникающем контекстном меню появляются самые нужные в данный момент пункты. Например, почти всегда присутствуют пункты Вырезать, Копировать, Вставить, Удалить (Cut, Copy, Paste, Delete).

2.4 Типы данных, используемые в MathCAD

Все значения начинающееся цифрой MathCAD интерпретирует как числа. В MathCAD можно работать со следующими классами *чисел*:

- ✓ обычные вещественные (целые) числа;
- ✓ комплексные числа; для ввода мнимого числа нужно за его модулем ввести символ мнимой единицы i или j. Тем не менее, нельзя символы i, j использовать сами по себе для ввода мнимых чисел. Так, следует печатать 1i или 1j, например, 8 + 5i;
- ✓ двоичные целые числа; двоичное число заканчивается строчной латинской буквой b, например 1011b;

✓ восьмеричные целые числа, восьмеричное число заканчивается строчной латинской буквой o, например, 647o;

✓ шестнадцатеричные целые числа; шестнадцатеричное число должно заканчиваться строчной латинской буквой h, например, 7b2h.

2.5 Ввод текста

Текстовую область (например, комментарии, заголовки и т.п.) можно разместить в любом незанятом месте документа MathCAD. Перед тем как ввести первый символ, следует нажать клавишу <">. В результате на месте курсора ввода появляется новая текстовая область (регион), которая имеет характерное выделение  . При этом курсор принимает вид вертикальной линии красного цвета, которая называется линией ввода текста и аналогична по назначению линиям ввода в формулах.

Набирается текст в MathCAD так же, как и в любом текстовом редакторе. Если Вы примерно знаете, сколько места на листе займет ваш комментарий, то можете сразу растянуть текстовую область до нужных размеров. Чаще же текст просто вводят в область, обрывая строки с помощью Enter, когда они достигают нужной длины. Так приходится поступать, потому что MathCAD автоматически переносов слов не осуществляет.

2.6 Ранжированные (дискретные) переменные

Ранжированные (дискретные) переменные принимают ряд значений (диапазон) при каждом их использовании в качестве аргумента функции. Они определяется начальным и конечным значениями, а также шагом изменения значения. С их помощью MathCAD может выполнять многократные вычисления, т.е. циклы с повторяющимися вычислениями. Без использования дискретных переменных было бы очень непросто построить графики, вывести таблицы результатов расчета.

Для задания ранжированной переменной следует выполнить следующую последовательность действий.

✓ Ввести имя переменной и оператор присваивания.

✓ Поставить курсор в маркер значения переменной, нажать кнопку Range Variable (переменная-диапазон) панели Матрица (Ма-

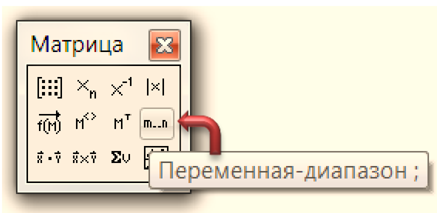


Рис. 2.12 - Определение дискретной переменной

trix) (Рис. 2.12). Появится заготовка в виде двух маркеров, разделенных точками  . Вставить данную заготовку можно также с помощью клавиши «;». Задать оператор ранжированной переменной просто введя с клавиатуры последовательно две точки нельзя.

Чтобы задать ряд значений x = 0, 1 , 2 , 3, 4, 5, в левый маркер заготовки ранжированной переменной введите ее первое значение диапазона, а в правый — последнее $x := 0..5$,.

Обратите внимание, что шаг изменения ранжированной переменной по умолчанию постоянен и равен 1. Однако его можно сделать и произвольным. Для этого нужно, поставив после левой границы интервала запятую, ввести второе значение ранжированной переменной. Разность между вторым и первым ее значением и определит шаг изменения. Размер шага может быть как положительным числом, так и отрицательным.

В частности, если набрать на клавиатуре h: -2,-1.5;2, то на экране появится результат (Рис. 2.13). Размер шага в этом примере 0.5 - разница между -1.5 и -2.

Можно использовать выражения в качестве параметров ранжированной переменной. Однако результаты вычисления этих выражений обязательно должны быть вещественными числами.

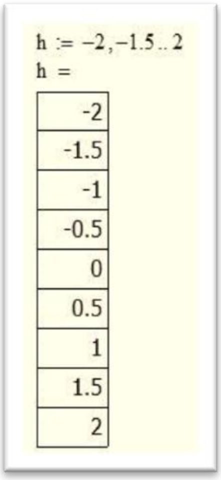


Рис. 2.13 - Задание шага ранжированной переменной

На Рис. 2.14 приведён ещё один пример построения графика функции.

2.7 Массивы

Очень распространенным способом сбора, хранения, обработки и выдачи информации в математике являются массивы.

Массивы – это пронумерованная последовательность величин одинакового типа, обозначаемая одним уникальным именем (идентификатором). В общем случае элементами массивов в MathCAD могут быть числовые (и строковые) константы, переменные и математические выражения.

Массивы можно характеризовать размерностью и размером. Число элементов по каждой размерности характеризует ее размер, а произведение размеров по каждой размерности характеризует общий размер (число элементов) массива. Так, размер матрицы $m \times n$ определяется произведением числа элементов в строке m на число элементов в столбце n.

Элементы массива являются индексированными переменными. Для доступа к данным, хранящимся в определенном элементе массива, необходимо указать имя массива и порядковый номер этого элемента с помощью подстрочного индекса (как это принято и в обычной математике).

Переменную, представляющую собой просто список данных, называют одномерным массивом или вектором. Порядковый номер элемента вектора задается его индексом.

Если возникает необходимость хранения данных в виде таблиц, в фор-

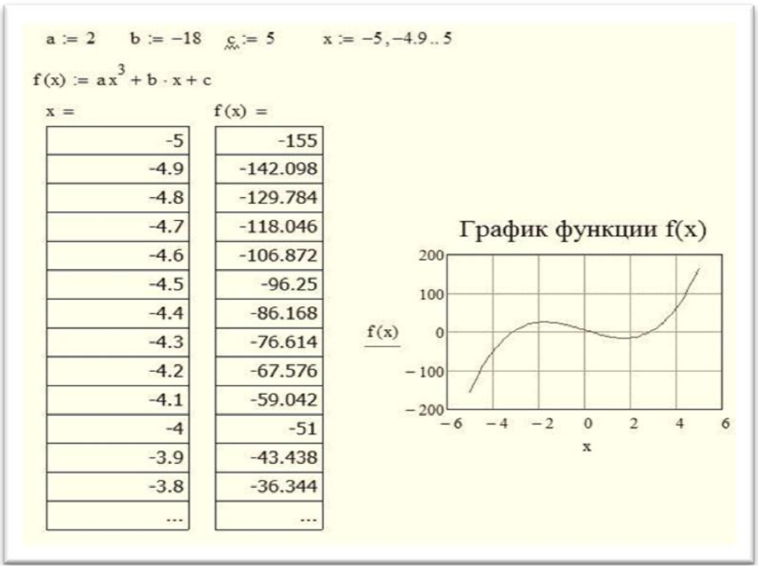


Рис. 2.14 - Пример построения графика функции с использованием ранжированной переменной. Результаты выведены в виде векторов аргумента и функции (столбцов чисел) и графика.

мате строк и столбцов, то необходимо использовать двумерные массивы (матрицы). Для доступа к данным, хранящимся в таком массиве, необходимо указать имя массива и два индекса. Первый индекс должен соответствовать номеру строки, а второй - номеру столбца, в которых хранится необходимый элемент.

Для ввода индекса элемента массива используется символ прямой открывающейся скобки – **[**. Он вводится после имени массива и вызывает шаблон ввода индекса $x_{\blacksquare}$.

Начальное значение каждого индекса в MathCAD определено системной переменной ORIGIN. По умолчанию значение этой переменной равно 0. Однако, если переопределить системную переменную ORIGIN := 1, то индексация будет начинаться с единицы. Индексы могут быть только целыми положительными числами или нулем.

В MathCAD предусмотрен удобный способ задания массивов с помощью их шаблонов, имеющихся в панели инструментов Матрица. Чтобы определить вектор или матрицу, следует:

- ✓ записать имя матрицы, ввести оператор присваивания (:=);
- ✓ на стандартной панели инструментов Математические (Math) нажмем кнопку с изображением матрицы. Откроется панель Матрица (Matrix), на которой вновь выбрать кнопку с изображением матрицы. На этот раз откроется диалоговое окно, в котором надо ввести число строк и число столбцов матрицы и нажать кнопку ОК.
- ✓ В появившемся на экране шаблоне матрицы каждый местозаполнитель заполнить числами или алгебраическими выражениями.

Матрицы можно складывать и вычитать, транспонировать (менять столбцы со строками), обращать, вычислять их определитель и собственные числа и т.п. (проанализируйте внимательно назначение функций категории Вектора и матрицы диалогового окна Вставка функции f(x)).

Рис. 2.15 показывает технологию ввода матрицы, а также примеры применения простейших векторных и матричных операций.

3 Нахождение корней уравнений

3.1 Одно уравнение с одним неизвестным

Для решения одного уравнения с одним неизвестным используется функция root(f(var), var, [a, b]), которая возвращает значение переменной var, для которого функция f(var) принимает нулевое значение. Если указаны необязательные параметры a и b, функция root находит скалярное[2] значение var на этом интервале, иначе до вызова функции root переменной var необходимо присвоить значение начального приближения. Оба аргумента функции root должны быть скалярами.

Для поиска корня функция root использует метод секущей. Когда значение выражения f(x) при очередном приближении становится меньше зна-

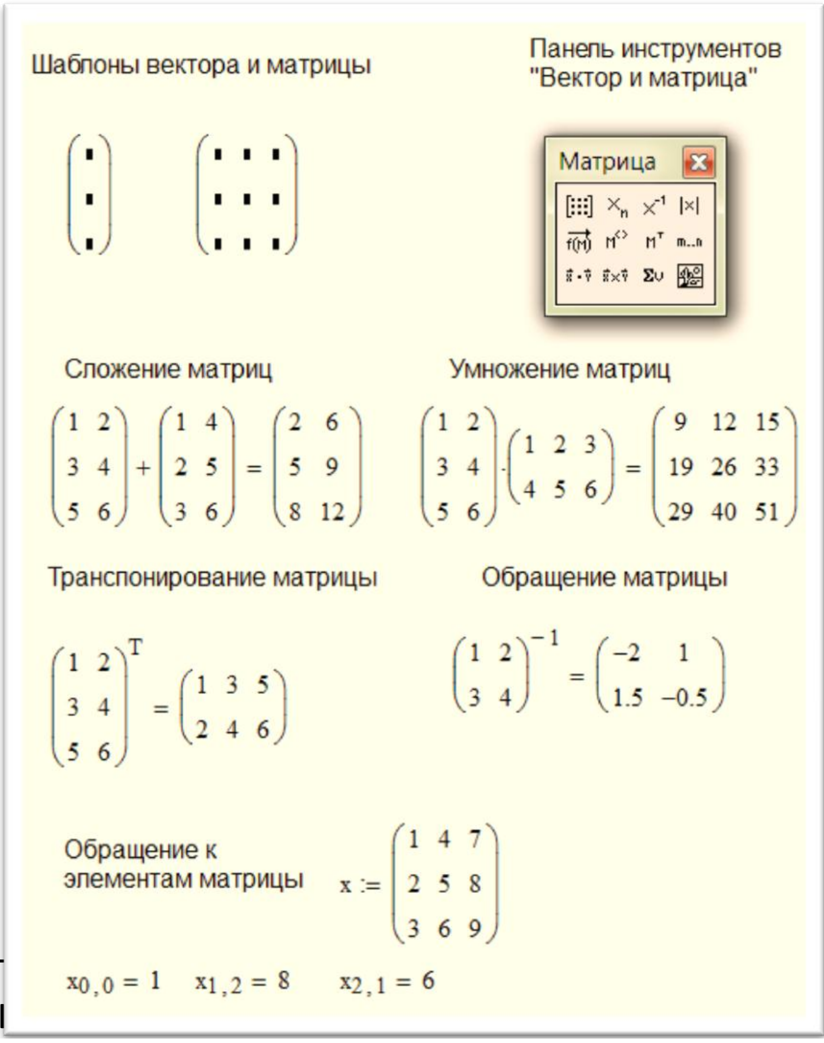


Рис. 2.15 - Примеры простейших операций с матрицами

[2] **Скаля**... можно пере-
мен... кено
одн...

чения встроенной переменной TOL (по умолчанию $TOL = 1 \times 10^{-3}$), корень считается найденным, и функция root возвращает результат.

Рассмотрим пример нахождения корней уравнения $f(x) = ax^3 + bx + c$, график которого показан на Рис. 2.14. Так как функция три раза пересекает ось x – она имеет три корня. Для уточнения этих трех различных корней используем три начальных приближения (Рис. 3.1).

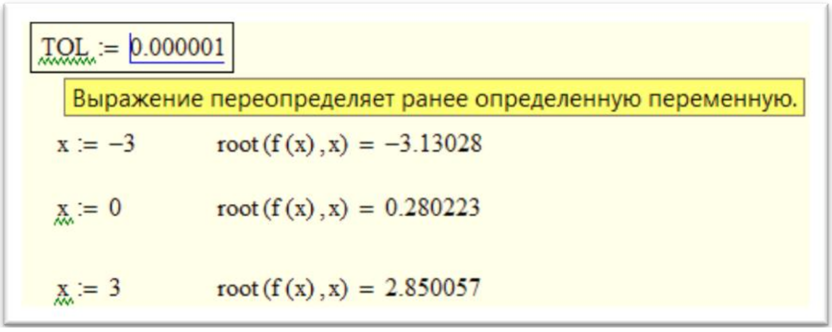


Рис. 3.1 - Нахождение корней уравнения с помощью функции root

Предварительно задаем начальные приближения (guess value) искомым корням (взяты из графика на Рис. 2.14) второму аргументу x функции root .

Чтобы повысить точность расчета, значение погрешности TOL в этом примере переопределено. Результаты выведены с точностью 6 десятичных знаков (по умолчанию выводятся 3 знака после запятой).

3.2 Форматирование результата вычислений

На результат расчета повлиять нельзя, но можно изменить формат вывода чисел. MathCAD вычисляет все выражения с точностью 20 знаков, но выводит на экран не все значащие цифры. Точность и вид выводимых числовых результатов легко настроить с помощью диалогового окна Формат результата (Result Format). Это окно (Рис. 3.2) вызывается командой меню Формат / Результат (Format / Result).

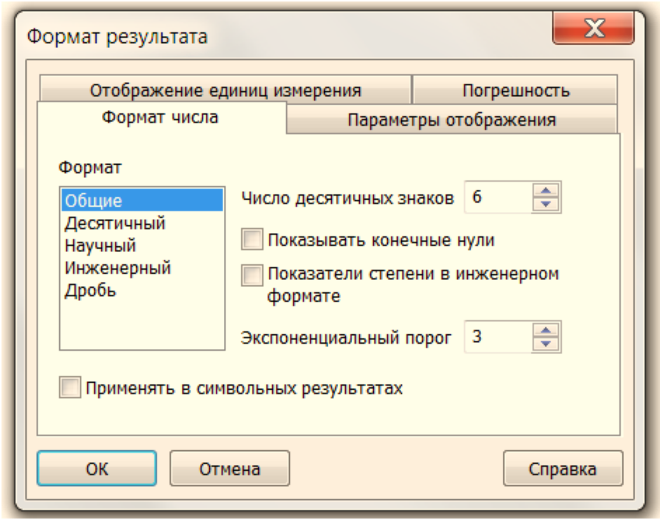


Рис. 3.2 - Диалоговое окно Формат результата

Вызвать это окно также можно двойным щелчком левой кнопкой мыши, предварительно установив указатель мыши на нужном численном результате расчета.

В этом окне можно выбрать следующие форматы.

✓ Общие (General) - формат, выбранный по умолчанию:

- определение числа отображаемых десятичных знаков (decimal places) после точки. Например, число 122.5587 с четырьмя десятичными знаками при отображении с двумя знаками будет выглядеть как 122.56;
- отображение или скрытие незначащих нулей (trailing zeros) — позволяет показывать или скрывать незначащие нули в десятичном представлении числа, т.е. выводить, например, «1.5» вместо «1.500» (даже если установлено количество десятичных знаков, равное 3);
- экспоненциальный порог (exponential threshold), при превышении степени 10 которого число будет показываться с порядком. Например, при пороге 3 число 122.56 будет отображаться как десятичное, а при пороге 2 — уже как «1.23×10^2».

- ✓ Десятичный (Decimal) – числа представляются только в виде десятичной дроби: 122.559. В этом представлении три знака в дробной части устанавливается параметром Число десятичных позиций (Number of decimal places).
- ✓ Научный (Scientific) – числа отображаются только с порядком – так, чтобы целая часть мантиссы состояла из одного символа, например: $10 = 1x10^1$, $122.559 = 1.23\times10^2$. Количество десятичных знаков и отображение незначащих нулей результата определяется пользователем.
- ✓ Инженерный (Engineering) – числа отображаются только с порядком, кратным 3: $1000 = 1x10^3$, $10000 = 10x10^3$, $100000 = 100x10^3$, $1000000 = 1x10^6$.
- ✓ Дробный (Fraction) – числа отображаются в виде правильной или неправильной дроби: $0.5678 = \frac{1386}{2441}$.

3.3 Нахождение корней полинома

Для нахождения корней уравнения, имеющего вид $a_nx^n + a_{n-1}x^{n-1} + + ... + a_1x + a_0$, лучше использовать функцию polyroots(v) вместо root. Она возвращает вектор, содержащий все корни многочлена, коэффициенты которого заданы параметром v. В отличие от функции root, функция polyroots не требует начального приближения и возвращает сразу все корни как вещественные, так и комплексные. Коэффициенты полинома должны быть представлены в порядке возрастания степени x в векторе v длины n + 1.

Найдем корни полинома $f(x) = ax^3 + bx + c$, при $a = 3$, $b = -25$, $c = -80$ с помощью функции polyroots.

Для построения вектора v необходимо использовать панель Матрица (Matrix) (Рис. 3.3) в которой нажмем кнопку . В возникшем окошке определим число строк и столбцов, нажмем Ок, после чего появится шаблон нашего вектора.

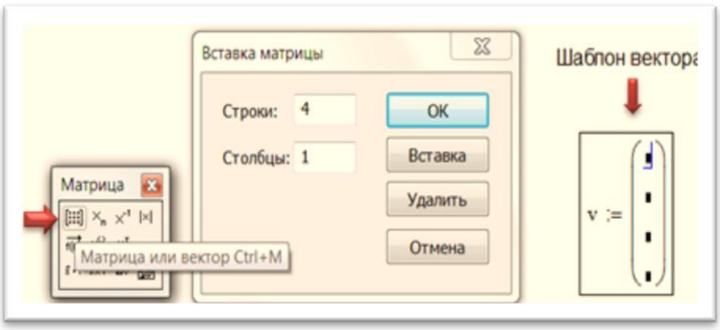


Рис. 3.3 - Вставка вектора или матрицы

Заметим (Рис. 3.4), что в вектор коэффициентов v вошли все коэффициенты, в том числе равные нулю (в общем случае коэффициенты могут быть и комплексными). Первым в векторе записан свободный член полинома, вторым — коэффициент при x^1 и т.д. Соответственно, последним n+1 элементом вектора должен быть коэффициент при старшей степени x^n. Результат – найдены три корня: один действительный и два мнимых.

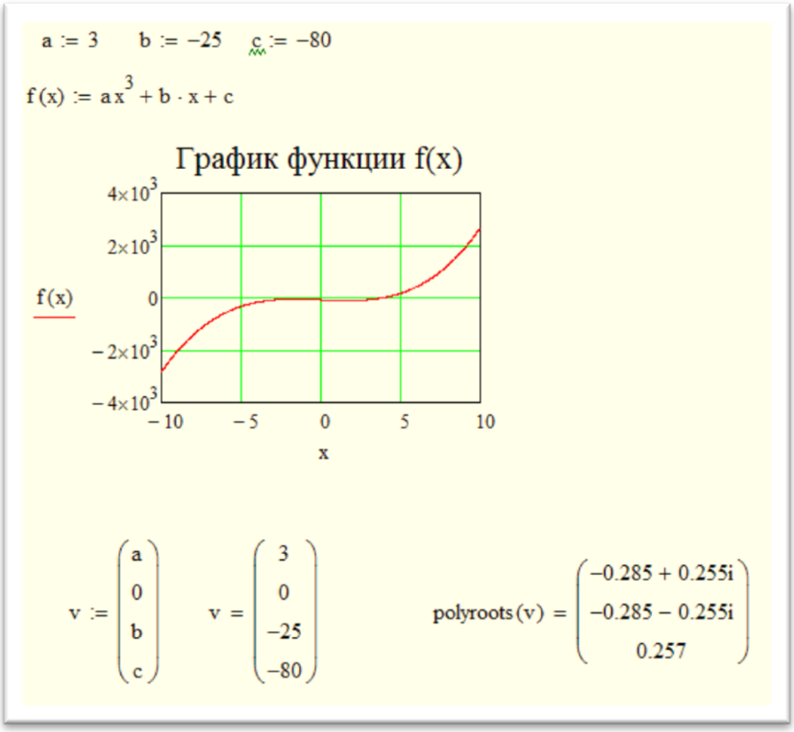


Рис. 3.4 - Нахождение корней уравнения с помощью функции polyroots

Пример нахождения корней полинома с комплексными коэффициентами приведен на Рис. 3.5.

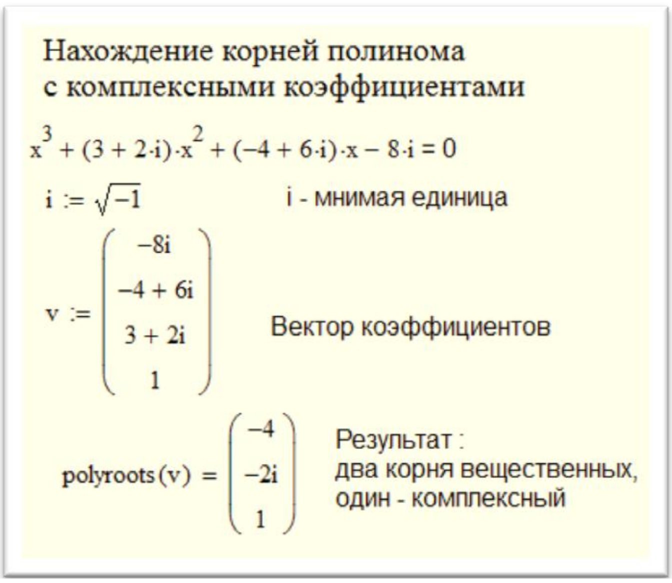


Рис. 3.5 - Нахождение корней полинома с комплексными коэффициентами

3.4 Решение систем линейных уравнений

Многие математические модели подсистем РЭС описываются системами из *n* линейных алгебраических уравнений (ЛАУ) с *n* неизвестными.

$$\begin{aligned} a_{1,1}\cdot x_1 + a_{1,2}\cdot x_2 + \cdots + a_{1,n}\cdot x_n &= b_1 \\ a_{2,1}\cdot x_1 + a_{2,2}\cdot x_2 + \cdots + a_{2,n}\cdot x_n &= b_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ a_{n,1}\cdot x_1 + a_{n,2}\cdot x_2 + \cdots + a_{n,n}\cdot x_n &= b_n \end{aligned}$$

В матричной форме системы линейных алгебраических уравнений запишется так:

A*X = B,

где A – квадратная матрица коэффициентов при неизвестных

$$A = \left[a_{i,j}\right] = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix}$$ - матрица коэффициентов системы;

$$B = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$$ - столбец свободных членов; $$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$ - столбец неизвестных.

Заметим, что у системы с вырожденной матрицей (у вырожденной матрицы определитель det *A* = 0) может либо не быть ни одного решения, либо существовать бесконечное множество. Такие системы MathCAD в общем случае тоже не сможет решить.

Для решения систем линейных алгебраических уравнений в системе ***Math-CAD*** можно предложить несколько методов.

1. Систему линейных уравнений легко решить с помощью обратной матрицы A^{-1}, получаемой путем обращения матрицы коэффициентов при неизвестных A. Вектор неизвестных X может быть получен путем умножения обратной матрицы на вектор-столбец свободных членов B:

$$X = A^{-1}\cdot B.$$

Так, для решения системы линейных уравнений

$$\begin{aligned} x_1 + 3x_2 + 2x_3 &= 10 \\ 2x_1 + 2x_2 + 3x_3 &= 30 \\ 3x_1 + x_2 + x_3 &= 20 \end{aligned}$$

с помощью матриц необходимо, прежде всего, переписать исходную систему в матричном виде. Для этого нужно составить матрицу коэффициентов, вектор неизвестных и вектор правых частей (Рис. 3.6).

$$A := \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 3 \\ 3 & 1 & 1 \end{pmatrix} \qquad B := \begin{pmatrix} 10 \\ 30 \\ 20 \end{pmatrix} \qquad X := A^{-1} \cdot B \qquad X = \begin{pmatrix} 5 \\ -5 \\ 10 \end{pmatrix}$$

Рис. 3.6 - Решение системы линейных алгебраических уравнений с помощью обратной матрицы коэффициентов при неизвестных А

Находить решение системы линейных уравнений через обратную матрицу вполне позволительно, если количество уравнений в системе невелико и расчет должен быть произведен только один раз. Если же система большая или же одновременно нужно решить много систем, то такой подход оптимальным не считается из-за низкой точности.

2. Использование встроенной функции lsolve(…)

Для нахождения корней той же системы уравнений может применяться функция lsolve(A, B). Эта функция возвращает вектор X с решением линейной системы уравнений A·X = B (Рис. 3.7). Функция lsolve анализирует матрицу **A** и выбирает наиболее подходящий метод решения, дающий минимальную погрешность результата. Допустимы и матрицы с комплексными числами.

$$A := \begin{pmatrix} 1 & 3 & 2 \\ 2 & 2 & 3 \\ 3 & 1 & 1 \end{pmatrix} \qquad B := \begin{pmatrix} 10 \\ 30 \\ 20 \end{pmatrix} \qquad \mathrm{lsolve}(A, B) = \begin{pmatrix} 5 \\ -5 \\ 10 \end{pmatrix}$$

Рис. 3.7 - Решение системы линейных алгебраических уравнений с помощью функции lsolve(...)

3. Нахождение корней линейных АУ при помощи блока Given … Find(var1, var2, ...) или Given … Minerr(var1, var2, ...).

Системы линейных (как впрочем, и нелинейных) уравнений в MathCAD могут решаться с помощью вычислительного блока given .. find. Этим вычислительным блоком можно пользоваться, когда решение реально существует (хотя и не является аналитическим). Функция Minerr(var1, var2, ...) пытается найти максимальное приближение даже к несуществующему решению путем минимизации среднеквадратичной погрешности решения методом наименьших квадратов.

Чтобы решить систему алгебраических уравнений с помощью данных функций, нужно:

✓ задать начальные приближения для всех неизвестных, входящих в систему уравнений в виде переменная := значение, т.е. обычным присваиванием переменным заданных значений. Если переменных несколько, то разумно для задания начальных условий использовать векторное представление.

✓ напечатать ключевое слово given (дано)[3], указывающее MathCAD, что далее следует система уравнений;

✓ ввести уравнения и неравенства, входящие в систему, правее и ниже ключевого слова given.

Внимание!! Между левой и правой частями уравнений должен стоять жирный знак равенства **=**. Это не знак присвоения значения, а знак логического равенства! Для его ввода используйте сочетание клавиш Ctrl + = или выберите его в панели Boolean (Булева алгебра).

✓ ввести выражение, которое включает Find(var1, var2, ...) или Given ... Minerr(var1, var2, ...). Эти функции возвращают значения переменных var1, var2, ..., представляющие решение системы уравнений. Если указан только один аргумент, возвращается скаляр, иначе возвращается вектор решений. Функции find и miner могут решать и одно уравнение с одним неизвестным - как частный случай системы уравнений.

На Рис. 3.8 показан пример нахождения корней той же системы уравнений (см. Рис. 3.7) при помощи блока Given ... Find(var1, var2, ...).

---

[3] Убедитесь, что Вы не находитесь в текстовой области. Если нажать клавишу «пробел», то математическое выражение становится текстовой областью и слово given перестает восприниматься как ключевое.

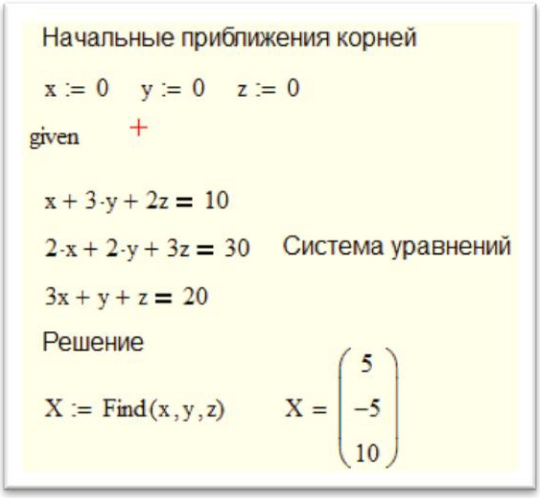


Рис. 3.8 - Решение системы линейных алгебраических уравнений с помощью блока Given ... Find

3.5 Решение систем нелинейных уравнений

Математические модели реальных РЭС, как правило, описываются *системами нелинейных уравнений*.

Необходимость отыскания корней нелинейных алгебраических и трансцендентных[4] уравнений нередко встречается при расчетах электронных схем с элементами, обладающими нелинейными характеристиками (диодами, транзисторами, трансформаторами и т.п.). Нелинейные системы уравнений могут также представлять ММ тепловых режимов РЭС, колебаний амортизаторов, конструкций со многими степенями свободы и т.д.

Под нелинейными уравнениями понимаются алгебраические и трансцендентные уравнения вида

$$\varphi(x) = 0,$$

где x действительное число,

$\varphi(x)$ - нелинейная функция, а под системой нелинейных уравнений - система вида

$$\begin{gathered}
\varphi_1(x_1, x_2, \dots, x_N) = 0, \\
\varphi_2(x_1, x_2, \dots, x_N) = 0, \\
\dots \dots \dots \dots \dots \dots \dots \dots \dots \\
\varphi_N(x_1, x_2, \dots, x_N) = 0,
\end{gathered}$$

---

[4] **Трансцендентным уравнением** называется уравнение, в котором неизвестное входит в аргумент трансцендентных функций (трансцендентные функции – аналитические функции, не являющиеся алгебраическими, например показательные, логарифмические, тригонометрические и т.д.).

не являющаяся системой линейных алгебраических уравнений. Решением этой системы является N-мерный вектор

$$x \equiv (x_1, x_2, \dots, x_N).$$

В отличие от систем линейных уравнений для систем нелинейных уравнений в большинстве случаев не известны прямые методы решения..

Вычисление корней нелинейных уравнений численными методами включает два этапа:

1) отделение корней с целью получения их начальных приближений, т.е. выделение областей изоляции корней [a, b], в пределах которых имеется только один корень;
2) уточнение корней до заданной точности.

Отделить корни можно попробовать аналитически:

✓ если функция f(x) непрерывна на отрезке [a, b] и имеет на его концах разные знаки, т.е. $f(a) \cdot f(b) < 0$, то в этом отрезке есть хотя бы один корень;

✓ если f(x) непрерывна и монотонна (f′(x) не меняет знак) на отрезке [a, b] и при этом $f(a) \cdot f(b) < 0$, то на этом отрезке корень будет единственным.

Учитывая легкость построения графиков функций в MathCAD, для отделения корней нелинейного уравнения разумно использовать графический метод, позволяющий отделить их путем анализа выведенного графика функции. Он заключается в построении графика функции f(x) и приближенной (на глаз) оценке значений x, при которой график f(x) пересекает ось абсцисс.

Для уточнения корней нелинейных систем используются приближенные итерационные вычислительные методы такие, как метод простых итераций, метод Ньютона и многие другие (Таблица 3.1).

Таблица 3.1 - Методы численного решения нелинейных уравнений

Метод:	**Итерационная формула:** $n = 0, 1, 2, \dots$
простых итераций	$x_{n+1} = F(x_n)$
Ньютона	$x_{n+1} = x_n - F(x_n) / F'(x_n)$
модифицированный Ньютона	$x_{n+1} = x_n - \Delta x \cdot F(x_n) / (F(x_n+\Delta x) - F(x_n))$
хорд (на интервале a..b)	$x_{n+1} = x_n - (x_n - a) \cdot F(x_n) / (F(x_n) - F(a))$ при a = const $x_{n+1} = x_n - (x_n - b) \cdot F(x_n) / (F(b) - F(x_n))$ при b = const
комбинированный хорд и касательных	$x_{n+1} = x_n - (x_n - x_{n-1}) \cdot F(x_n) / (F(x_n) - F(x_{n-1}))$

Встроенные функции системы MathCAD, предназначенные для решения нелинейных алгебраических уравнений, направлены на решение второй задачи, т. е. предполагают, что корни уже локализованы и имеются их ориентировочные начальные значения, позволяющие уточнить корни с заданной абсолютной погрешностью TOL (по умолчанию TOL = 0.001).

Системы нелинейных (как и линейных) уравнений в MathCAD могут решаться с помощью вычислительного блока given .. find (см. подраздел 3.4).

Решение ищется методом итераций и при наличии нескольких корней, очевидно, будет найдено лишь ближайшее решение, если оно существует. Следовательно, как и в предыдущем случае, необходимо задание начальной точки, от которой будет происходить поиск.

Решим графически систему нелинейных уравнений (Рис. 3.9).

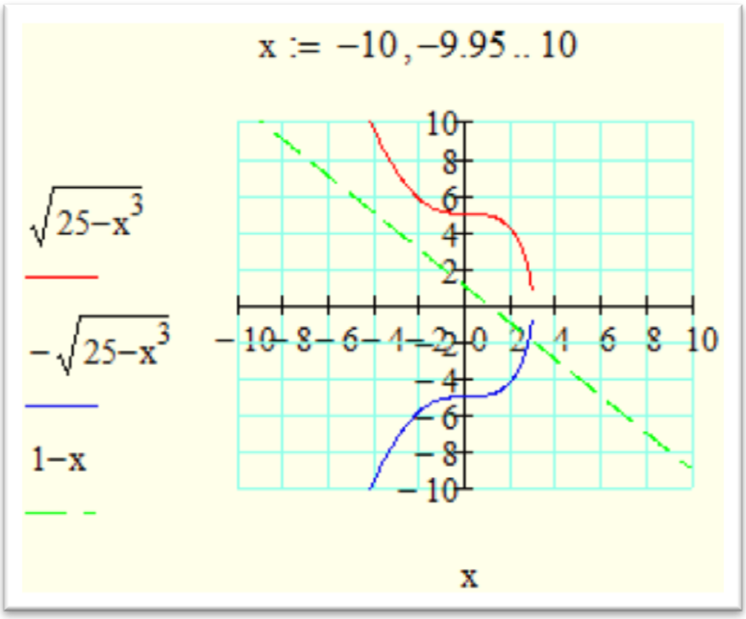


Рис. 3.9 - Графическое решение системы нелинейных уравнений
$x^3 + y^2 = 25,$
$x + y = 1.$

Для этого преобразуем первое и второе уравнения к виду $y = f(x)$:

$y_1 = \sqrt{25 - x^3}$ (верхняя часть графика функции $x^3 + y^2 = 25$),

$y_2 = -\sqrt{25 - x^3}$ (нижняя часть графика функции $x^3 + y^2 = 25$),

$y_3 = 1 - x$.

Определим множество значений аргумента функций y на графике с помощью ранжированной переменной x := −10, −9.95 .. 10.

Вызовем шаблон графика одномерной функции (см. Рис. 2.7). Введем x в местозаполнитель под серединой оси абсцисс и первое уравнение $\sqrt{25 - x^3}$ в поле напротив середины оси ординат.

Чтобы отобразить зависимости нескольких функций от одного аргумента на одном графике, необходимо выполнить следующие действия:

✓ Поместить линии ввода так, чтобы они целиком захватывали выражение, стоящее в надписи координатной оси Y $\sqrt{25-x^3}$. Уголок линии ввода при этом обязательно должен находиться в конце имени функции.

✓ Нажать клавишу запятая <**,**>.

✓ Появится новый местозаполнитель (черный квадратик), в который нужно ввести выражение для второй функции $-\sqrt{25-x^3}$.

✓ Щелкнем в любом месте вне этого выражения (на графике или вне его). После этого вторая кривая будет отображена на графике.

✓ Аналогично можно добиться появления третьего местозаполнителя, с помощью которого можно задать ряд данных $1 - \mathrm{x}$ для третьей зависимости.

Заметим, что если две функции имеют разные аргументы, например, $f_1(x)$ и $f_2(y)$, то на оси ординат нужно ввести (через запятую) имена обеих функций, а на оси абсцисс (также через запятую) ввести имена обоих аргументов x и y . Тогда первый график будет построен для первой функции по первому аргументу, второй график для второй функции по второму аргументу.

Прямая и кривые на графике (Рис. 3.9) пересекаются только один раз, поэтому действительный корень один. Решение этой системы уравнений приведено на Рис. 3.10.

Если у нас было бы несколько точек пересечения (несколько действительных корней), то потребовалось бы задать несколько разных начальных приближений и повторить процесс нахождения корней.

Для нахождения мнимых корней надо задавать начальное приближение в виде комплексного числа.

Комплексное число является суммой действительного и мнимого числа, получающегося путем умножения любого действительного числа на мнимую единицу (imaginary unit) i. По определению, $i^2 = -1$.

Чтобы определить мнимое число, например $2i$, введите действительный сомножитель 2 и непосредственно после него введите символ i или j.

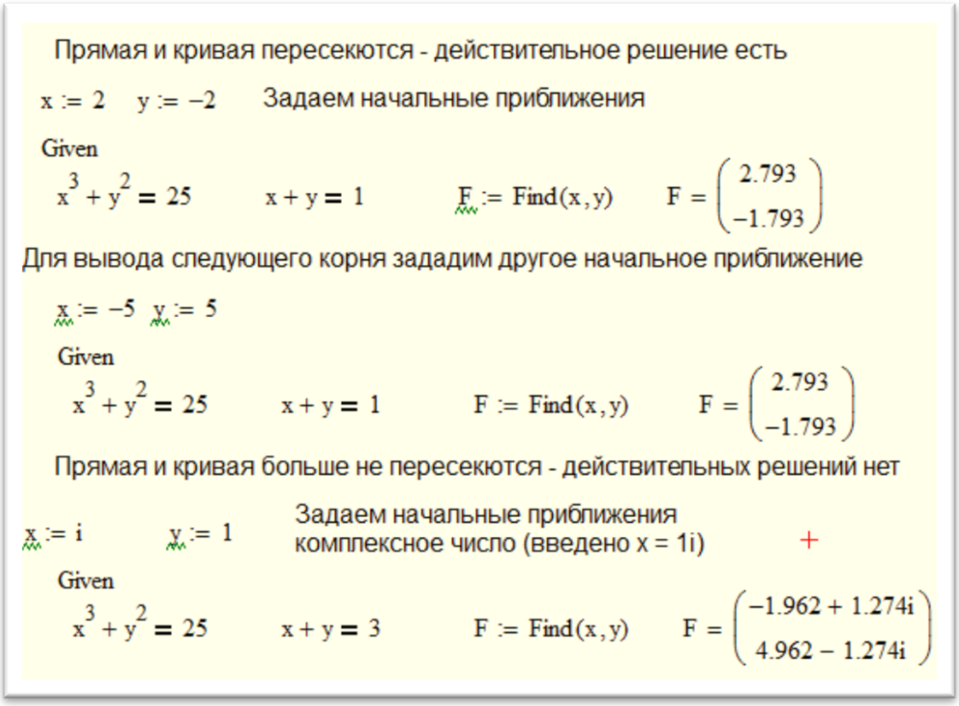


Рис. 3.10 - Решение системы нелинейных уравнений

Для ввода мнимой единицы надо нажать клавиши <1>, <i>. Если просто ввести символ i, то MathCAD интерпретирует его как переменную i. Кроме того, мнимая единица имеет вид $1i$, только когда соответствующая формула выделена. В противном случае мнимая единица отображается просто как i (Рис. 3.11).

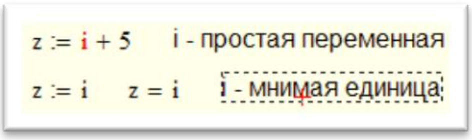


Рис. 3.11 - Ввод мнимой единицы

Между ключевыми словами Given и Find могут быть вставлены неравенства-ограничения.

Заметим, что для решения нелинейных уравнений могут также применяться встроенные функции root(...) и блоки Given .. Minerr(...), Given .. Maximize(...) и Given .. Minimize(...)..

4 Решение обыкновенных дифференциальных уравнений

4.1 Постановка задачи

Большинство реальных научно-технических задач в области радиоэлектроники (особенно относящихся к анализу динамических систем и к их математическому моделированию) базируются на решении систем обыкновенных дифференциальных уравнений (ОДУ).

Обыкновенным дифференциальным уравнением n-го порядка называется уравнение, связывающее между собой значения независимой переменной x (чаще всего это время t), неизвестной функции y = f(x) и её производных (или дифференциалов):

$$F\left(x, \frac{\partial y}{\partial x}, \frac{\partial^2 y}{\partial x^2}, \dots, \frac{\partial^{(n)} y}{\partial x^{(n)}}\right) = 0.$$

Чтобы численно решить в MathCAD систему из ОДУ, ее необходимо представить в форме Коши с указанием начальных условий:

$$\begin{cases} y_1(x_0) = y_{0,1} \\ y_2(x_0) = y_{0,2} \\ \quad \dots \\ y_n(x_0) = y_{0,n} \end{cases} \qquad \begin{cases} y'_1 = f_1(x, y_1, y_2, \dots, y_n) \\ y'_2 = f_2(x, y_1, y_2, \dots, y_n) \\ \quad \dots \\ y'_n = f_n(x, y_1, y_2, \dots, y_n) \end{cases}$$

Первая система здесь задает начальные условия, а вторая представляет систему ОДУ.

Решение системы ОДУ в форме Коши осуществляется аналогично решению одиночного ДУ, но должно быть организовано в *векторной форме*. В векторном виде системы ОДУ можно представить:

$$Y(x_0) = Y_0 \qquad Y' = F(x, Y).$$

При этом добавление очередного уравнения не увеличивает числа уравнений в векторной их записи.

Обыкновенные дифференциальные уравнения n-го порядка:

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

где $y(x_0) = y_{0.0}$, $y'(x_0) = y_{0,1}$ $y''(x_0) = y_{0,2}, \dots,$ $y^{(n-1)}(x_0) = y_{0,n-1},$ несложно свести к системе ОДУ первого порядка, если ввести следующие обозначения:

$$y_1(x) = y(x),\ y_2(x) = y'(x),\ \dots,\ y_n(x) = y^{(n-1)}(x),$$

$$y_{0,0} = y(x_0),\ y_{0,1} = y'(x_0),\ \dots,\ y_{0,n-1} = y^{(n-1)}(x_0).$$

В этом случае можно записать:

$$\begin{cases} y_1(x_0) = y_{0,0} \\ y_2(x_0) = y_{0,1} \\ \quad \dots \\ y_n(x_0) = y_{0,n-1} \end{cases} \qquad \begin{cases} y'_1 = y_2 \\ y'_2 = y_3 \\ \dots \\ y'_{n-1} = y_n \\ y'_n = f(x, y_1, y_2, \dots, y_{n-1}) \end{cases}$$

При использовании любых методов численного интегрирования необходимо, чтобы были заданы, по крайней мере, следующие величины:

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

Все численные методы интегрирования системы ОДУ основаны на последовательном нахождении значений неизвестной функции y начиная с y_0, т.е. y_1, y_2, …, y_m, y_{m+1}, …, y_N. При решении этой задачи на интервале интегрирования выделяется конечное число точек t_i ($i = 0 .. N$), в которых определяются значения y**.** Интервал между соседними точками называется *шагом интегрирования* и обозначается $\Delta t = h_m = t_{m+1} - t_m$ (чаще всего $h_m = h = const$). При этом для $i = 0$ значения t_0 и y_0 должны быть известны как начальные условия - без этого невозможна единственность решения.

Основные характеристики методов интегрирования, от которых зависит их эффективность - точность и устойчивость методов.

Точность интегрирования можно оценить, проанализировав полную ошибку на каждом шаге интегрирования. Полная ошибка зависят от следующих составляющих:

- ✓ ошибки аппроксимации (погрешность метода);
- ✓ ошибки вычислений, связанных с ошибками округления чисел в компьютере;
- ✓ ошибки накопления, равной полной ошибке на предыдущем шаге.

Устойчивость связана с характером изменения накопленной погрешности: если в ходе интегрирования накопленная погрешность не возрастает с увеличением числа шагов, то используемый метод численно устойчив. Устойчивость во многом определяется правильностью выбора метода и шага интегрирования. При увеличении шага увеличивается погрешность вычислений, при уменьшении шага увеличиваются затраты машинного времени.

4.2 Численные методы для ОДУ первого порядка

Рассматриваемые ниже численные методы интегрирования дифференциальных уравнений относятся к явным методам. Для этих методов принципиально необходимо, чтобы решаемое дифференциальное уравнение или система ОДУ были представлены так, чтобы выражения для расчета соответствующих производных в правой части системы ОДУ быть алгебраическими и не содержали никаких производных.

Пусть дано ОДУ первого порядка, приведенное к нормальной форме Коши:

$\frac{dy}{dt} = f(t, y)$ или $y' = f(t,y)$ при начальных условиях $t_0 = 0$ и $y_0 = 1$.

Это уравнение имеет семейство решений $y(t)$. Выбор начального значения $y_0 = y(t_0)$ позволяет выделить одну из кривых семейства.

Рассчитаем функцию $y(t)$ с шагом интегрирования $h = \Delta t = (t_N - t_0) / N$ на интервале $t_0 .. t_N$, где N – число шагов интегрирования, $t_N = T_{пер} = 1$ – практическое время установления переходного процесса.

Вид решения этого уравнения в виде зависимости $y(t)$ зависит от функции $f(t, y)$. Если заменить бесконечно малые dy и dt на малые конечные приращения Δt и Δy, то можно свести дифференциальное уравнение к приближенному *конечно-разностному уравнению* $\Delta y/\Delta t = f(t, y)$.

В методе Эйлера производная берется в начале шага и по ней прогнозируется движение системы на конец шага, считая, что во время шага производная неизменна.

Если известно начальное значение t_0 и соответствующее значение $y_0 = y(t_0)$, то можно вычислить приращение $\Delta y = f(t, y)\cdot\Delta t$ и затем найти $y_1 = y_0 + \Delta y$.

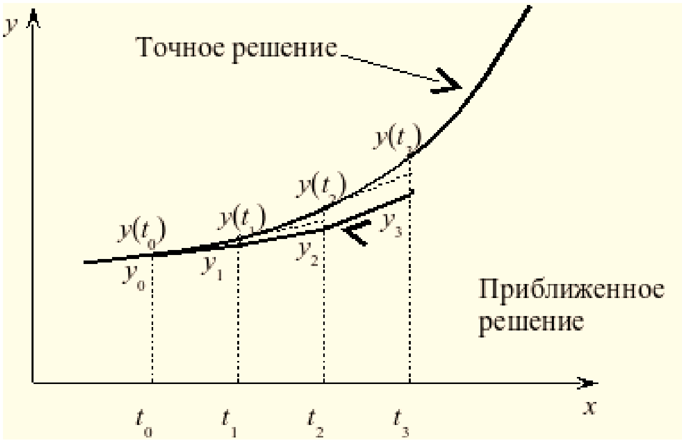


Рис. 4.1 - Геометрическая интерпретация метода Эйлера

Поступая аналогично (Рис. 4.1), мы сможем решить ОДУ численным методом и получить таблицу значений t_i и y_i. Этот простейший метод получил название *метода Эйлера*:

$$t_{i+1} = t_i + h,$$

$$y_{i+1} = y_i + h \cdot f(t_i, y_i).$$

Решим в MathCAD уравнение $\frac{dy}{dt} = t \cdot y$ с помощью явного метода Эйлера. Известно аналитическое решение этого уравнения, которое имеет вид

$y(t) = A \cdot \exp\left(\frac{t^2}{2}\right)$, причем A = 1, t_0 = 0 и y_0 = 1, что позволяет легко оценить погрешность метода, которая, в общем, пропорциональна h^2.

В ходе решения методом Эйлера (Рис. 4.2) в MathCAD сформированы таблицы результатов, а также графики точного и приближенного решения.

Решение обыкновенного дифференциального уравнения y' = t * y методом Эйлера

$h := 0.1$ Шаг интегрирования

$N := 10$ Число шагов интегрирования

$f(t, y) := t \cdot y$ функция, задающая ОДУ y' = f(t, y)

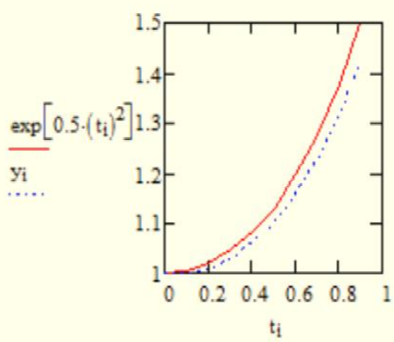
$i := 0..N-1$ задаем цикл интегрирования

$\begin{pmatrix} t_0 \\ y_0 \end{pmatrix} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ задание начальных условий

$\begin{pmatrix} t_{i+1} \\ y_{i+1} \end{pmatrix} := \begin{pmatrix} t_i + h \\ y_i + h \cdot f(t_i, y_i) \end{pmatrix}$ формулы интегрирования методом Эйлера повторяются в цикле

Таблицы с решением

	Значения t (t =)	Значения y (y =)	Точное решение $\exp\left[0.5 \cdot (t_i)^2\right]$	Абсолютная ошибка $y_i - \exp\left[0.5 \cdot (t_i)^2\right]$
0	0	1	1	0
1	0.1	1	1.005	$-5.013 \cdot 10^{-3}$
2	0.2	1.01	1.02	-0.01
3	0.3	1.03	1.046	-0.016
4	0.4	1.061	1.083	-0.022
5	0.5	1.104	1.133	-0.03
6	0.6	1.159	1.197	-0.038
7	0.7	1.228	1.278	-0.049
8	0.8	1.314	1.377	-0.063
9	0.9	1.419	1.499	-0.08
10	1	1.547		



<-- Точное решение

<-- Приближенное решение методом Эйлера

Максимальная абсолютная ошибка
$\varepsilon = -0.08$

График решения:
абсолютная ошибка быстро нарастает

Рис. 4.2 - Решение методом Эйлера

Из рассмотрения графиков решения хорошо заметно, что численное решение при довольно большом шаге *h* заметно отличается от точного аналитического решения. В то же время качественный характер численного решения остается верным.

Для уменьшения погрешности решения при том же шаге интегрирования следует применять методы более высокого порядка. Например, алгоритм модифицированного (уточненного) метода Эйлера (Рис. 4.3) состоит в том, что производную вычисляют не в *i*-ой точке, а между двумя соседними точками: *i* и *i + 1*. Модифицированный метод Эйлера реализуется на каждом шаге вычислений следующими итерационными формулами:

$$t_{i+1} = t_i + h,$$

$$y_{i+1} = y_i + h \cdot f(t_i + h/2,\ y_i + h \cdot f(t_i, y_i) / 2).$$

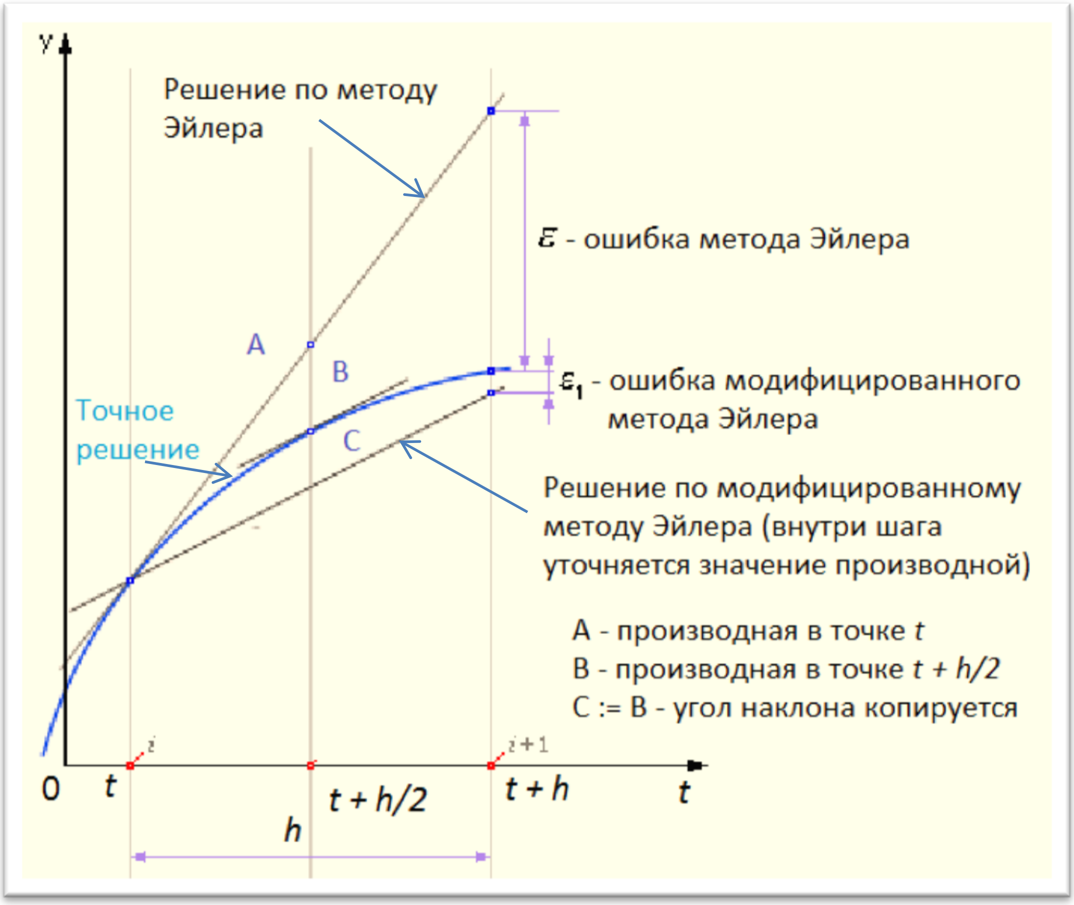


Рис. 4.3 - Геометрическая интерпретация модифицированного метода Эйлера

Решим в MathCAD с помощью модифицированного метода Эйлера уравнение из предыдущего примера (Рис. 4.4).

Решение обыкновенного дифференциального уравнения $y' = t * y$ модифицированным методом Эйлера

$h := 0.1$ Шаг интегрирования

$N := 10$ Число шагов интегрирования

$f(t, y) := t \cdot y$ функция, задающая ОДУ $y' = f(t, y)$

$i := 0..N-1$ задаем цикл интегрирования

$\begin{pmatrix} t_0 \\ y_0 \end{pmatrix} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ задание начальных условий

$\begin{pmatrix} t_{i+1} \\ y_{i+1} \end{pmatrix} := \begin{pmatrix} t_i + h \\ y_i + h \cdot f\left(t_i + \frac{h}{2}, y_i + h \cdot \frac{f(t_i, y_i)}{2}\right) \end{pmatrix}$ формула интегрирования модифицированным методом Эйлера повторяется в цикле

Таблицы с решением

Значения t

$t =$

	0
0	0
1	0.1
2	0.2
3	0.3
4	0.4
5	0.5
6	0.6
7	0.7
8	0.8
9	0.9
10	1

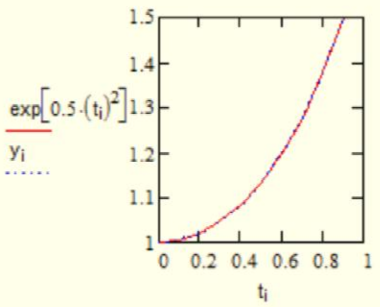
Значения y

$y =$

	0
0	1
1	1.005
2	1.02
3	1.046
4	1.083
5	1.133
6	1.197
7	1.277
8	1.376
9	1.497
10	1.646

Точное решение / Абсолютная ошибка

$\exp\left[0.5 \cdot (t_i)^2\right]$	$y_i - \exp\left[0.5 \cdot (t_i)^2\right] =$
1	0
1.005	$-1.252 \cdot 10^{-5}$
1.02	$-5.097 \cdot 10^{-5}$
1.046	$-1.187 \cdot 10^{-4}$
1.083	$-2.22 \cdot 10^{-4}$
1.133	$-3.707 \cdot 10^{-4}$
1.197	$-5.792 \cdot 10^{-4}$
1.278	$-8.683 \cdot 10^{-4}$
1.377	$-1.267 \cdot 10^{-3}$
1.499	$-1.815 \cdot 10^{-3}$



<-- Точное решение

<-- Приближенное решение

Максимальная абсолютная ошибка
$\varepsilon = -0.001815$

График решения:
точное и приближенное решения практически совпадают

Рис. 4.4 - Решение модифицированным методом Эйлера

Результаты показывают, что несмотря на довольно большой шаг *h*, заметного отличия приближенного решения от точного нет.

Тем не менее, методы Эйлера обычно используются только для быстрых грубых расчетов. При более высоких требованиях к точности решения целесообразно воспользоваться одним из самых употребляемых методов - Рунге-Кутты четвертого порядка, так как он не требует вычисления производных высших порядков. При этом погрешность расчета пропорциональна h^4.

Решим с помощью явного метода Рунге-Кутты четвертого порядка на MathCAD то же самое уравнение из предыдущего примера.

Приведенное ниже решение (Рис. 4.5) иллюстрирует возможности данного метода.

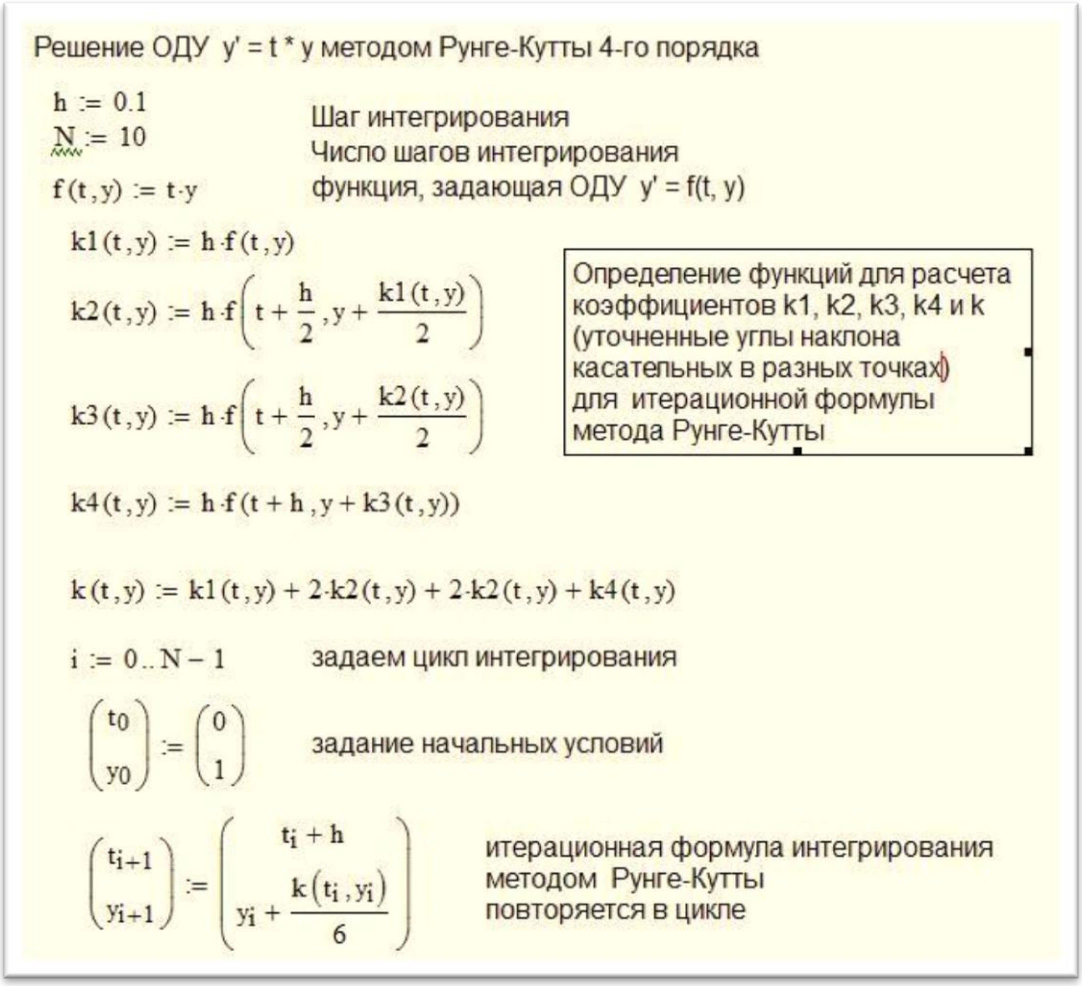


Рис. 4.5 – Решение ОДУ методом Рунге-Кутты четвертого порядка

Если проанализировать формулы метода Рунге-Кутты четвертого порядка (см. Рис. 4.5), то можно заметить, что каждый шаг расчета представляет собой шаг по методу Эйлера с соответствующим уточнением углов наклона касательных (геометрических представлений производных) в разных точках расчета.

Результаты расчета приведены в соответствующих таблицах и на графике (Рис. 4.6). Анализ результатов показывает, что точность решения методом Рунге-Кутта контрольного примера настолько высока, что на графике погрешность обнаружить невозможно. Небольшие отличия можно выявить лишь внимательно проанализировав табличные данные.

Таблицы с решением

Значения t

$t =$

	0
0	0
1	0.1
2	0.2
3	0.3
4	0.4
5	0.5
6	0.6
7	0.7
8	0.8
9	0.9
10	1

Значения y

$y =$

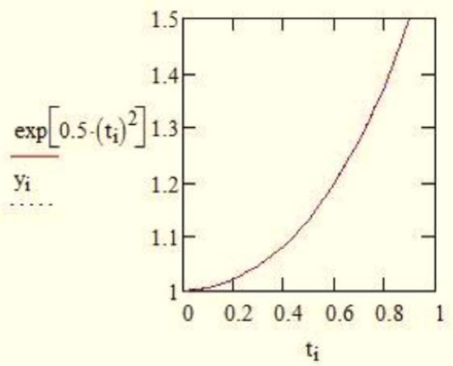
	0
0	1
1	1.005
2	1.02
3	1.046
4	1.083
5	1.133
6	1.197
7	1.277
8	1.377
9	1.499
10	1.648

Точное решение

$\exp\left[0.5\cdot(t_i)^2\right]$
1
1.005
1.02
1.046
1.083
1.133
1.197
1.278
1.377
1.499

Абсолютная ошибка

$y_i - \exp\left[0.5\cdot(t_i)^2\right]$
0
$-4.167\cdot10^{-6}$
$-1.698\cdot10^{-5}$
$-3.973\cdot10^{-5}$
$-7.486\cdot10^{-5}$
$-1.262\cdot10^{-4}$
$-1.996\cdot10^{-4}$
$-3.031\cdot10^{-4}$
$-4.485\cdot10^{-4}$
$-6.522\cdot10^{-4}$



<-- Точное решение

<-- Приближенное решение методом Рунге-Кутты

Максимальная абсолютная ошибка
$\varepsilon = -0.0006522$

График решения:
точное и приближенное решения визуально совпадают

Рис. 4.6 - Результаты решения ОДУ методом Рунге-Кутты четвертого порядка

4.3 Встроенные функции MathCAD для численного интегрирования одного ОДУ

Встроенная функция odesolve позволяет записывать уравнение ОДУ в блоке решения в привычном виде. Вычислительный блок для решения ОДУ состоит из следующих частей:

- *Given* (дано) - ключевое слово;
- ОДУ и начальное условие, записанное с помощью *логических* операторов, или система ОДУ и условия к ней, причем дифференциальное уравнение (в том числе входящее в систему) должно быть строго линейным (то есть высшая производная в нем не должна иметь никаких сомножителей или степенных показателей). Начальные условия должны быть в форме $y(t_o) = b$;

- *odesolve(t, tk, N)*[5] - встроенная функция для решения ОДУ, где:
  - ✓ *t* — имя переменной, относительно которой решается уравнение;
  - ✓ *tk* — конец интервала интегрирования. Начало интервала интегрирования указано выше в начальных условиях;
  - ✓ *N* — необязательный внутренний параметр, определяющий число шагов интегрирования, используемых для получения решения дифференциального уравнения. Чем больше N, тем с большей точностью будет получен результат, но времени будет затрачено на его поиск больше. Обычно подбором этого параметра можно заметно (в несколько раз) ускорить расчеты без существенного ухудшения их точности. Параметр N можно удалить, предоставив MathCAD возможность самому выбирать число шагов интегрирования.

Внимание! Появление других математических выражений в вычислительном блоке между словами given и odesolve недопустимо. Текстовую область внутри вычислительного блока размещать можно. Граничные условия можно задавать лишь в двух точках, одна из которых — начало интервала интегрирования.

Решим в MathCAD контрольное уравнение с помощью вычислительного блока *Given/Odesolve* (Рис. 4.7).

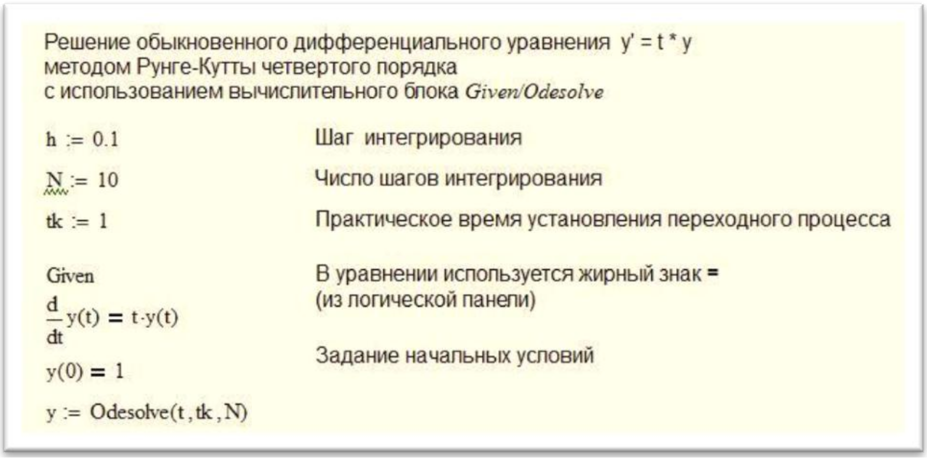


Рис. 4.7 – Алгоритм решения уравнения с помощью вычислительного блока *Given/Odesolve*

---

[5] Допустимо, и даже часто предпочтительнее, задание функции *Odesolve*(*t*, *tk*, *step*) с тремя параметрами, где *step* - внутренний параметр численного метода, определяющий количество шагов, в которых метод Рунге-Кутта, будет рассчитывать решение дифференциального уравнения.

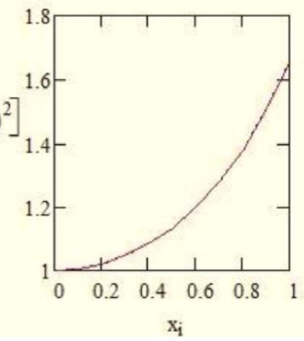
Не забывайте о том, что вставлять логические операторы следует при помощи панели инструментов Булева алгебра (Boolean). Символ производной можно ввести средствами панели Вычисления (Calculus) или Математический анализ (Math). Выбирайте тот или иной способ представления производной из соображений наглядности представления результатов - на ход расчетов он не влияет.

Имеется возможность выбирать между численными методами. Для смены метода необходимо нажатием правой кнопки мыши на области функции *odesolve* вызвать контекстное меню и выбрать в нем один из пунктов (метод Адамса, с постоянным шагом, адаптивный метод, метод RADAUS). По умолчанию применяется первый из них, т.е. метод Адамса. Результаты интегрирования ОДУ приведены на Рис. 4.8.

$i := 0..N \quad x_0 := 0 \quad x_{i+1} := x_i + h$ Задаем параметры цикла

Таблицы с решением

	Значения x, $x =$	Значения y, $y(x_i) =$	Точное решение, $\exp\left[0.5\cdot(x_i)^2\right]$	Абсолютная ошибка, $y(x_i) - \exp\left[0.5\cdot(x_i)^2\right]$
0	0	1	1	0
1	0.1	1.005	1.005	$1.502\cdot10^{-5}$
2	0.2	1.02	1.02	$2.937\cdot10^{-5}$
3	0.3	1.046	1.046	$4.644\cdot10^{-5}$
4	0.4	1.083	1.083	$6.219\cdot10^{-5}$
5	0.5	1.133	1.133	$6.434\cdot10^{-5}$
6	0.6	1.197	1.197	$7.384\cdot10^{-5}$
7	0.7	1.278	1.278	$8.154\cdot10^{-5}$
8	0.8	1.377	1.377	$8.76\cdot10^{-5}$
9	0.9	1.499	1.499	$9.952\cdot10^{-5}$
10	1	1.649	1.649	$1.152\cdot10^{-4}$
11	1.1			



$\exp\left[0.5\cdot(x_i)^2\right]$

$y(x_i)$

<-- Точное решение

<-- Приближенное решение

Максимальная абсолютная ошибка
$\varepsilon = -0.0001152$

График решения:
точное и приближенное решения практически совпадают

Рис. 4.8 - Результаты решения с помощью вычислительного блока *Given/odesolve*

4.4 Встроенные функции для решения систем ОДУ

Для численного решения систем ОДУ в MathCAD введен ряд функций. Остановимся на некоторых функциях, дающих решения для систем обыкновенных дифференциальных уравнений, представленных в нормальной форме Коши различными численными методами:

- rkfixed(y_0, *to*, *tk*, *intvls*, D) - метод Рунге-Кутты с постоянным шагом. Эта функция подходит для качественного и быстрого решения подавляющего большинства систем ОДУ.

- Rkadapt(y_0, *to*, *tk*, *intvls*, D) - метод Рунге-Кутты с переменным шагом. Использует не постоянный, а зависящий от скорости изменения функций решения шаг дискретизации переменной. Применяется в случае более жестких[6] и сложных систем уравнений, что позволяет повысить точность расчета или сэкономить время по сравнению с методом Рунге-Кутты с постоянным шагом

- - *Bulstoer(y_0, to, tk, intvls, D)* - метод Булирша-Штера. Позволяет получать более точные решения, чем rkfixed, затрачивая на это меньше времени, если функции решения системы достаточно гладкие и плавно изменяющиеся. При выполнении этого условия функция Bulstoer

---

[6] Жёсткими называют системы дифференциальных уравнений, решение которых содержит одновременно очень быстрые и очень медленные составляющие переходных процессов. Типичным видом таких систем являются дифференциальные уравнения, описывающие поведение радиоэлектронных схем с многократно различающимися постоянными времени, обусловленными значениями реактивных элементов. И действительно, в одной и той же схеме одновременно могут быть конденсаторы как с емкостями в несколько пикофарад, так и в несколько тысяч микрофарад, катушки индуктивности с величиной как в несколько генри, так и в несколько микрогенри. Возникает проблема выбора шага интегрирования. Чтобы учесть вклад в переходный процесс «быстрых» составляющих необходим исключительно маленький шаг интегрирования, выбираемый исходя из наивысшей скорости изменения значений переменных. Но это чрезвычайно замедляет вычислительный процесс. Большой шаг интегрирования резко снижает точность решения и часто приводит к его неустойчивости. Поэтому для интегрирования жестких систем необходимо применять специально разработанные методы (например, методы Адамса, Булирша-Штера, Розенброка и др.).

- BDF(y_0, *to*, *tk*, intvls, D, [J], [tol]) – (Backward Differentiation Formula) неявный многошаговый метод численного интегрирования для решения жестких систем ОДУ;

- AdamsBDF(y_0, *to*, *tk*, intvls, D, [J], [tol]) - использование методов BDF для жестких систем и методов Адамса для нежестких систем;

- Stiffr(y_0, *to*, *tk*, intvls, D, AJ) – использует метод Розенброка с расширенной функцией Якоби AJ для жестких систем.

В этих функциях

N – порядок системы уравнений;

y_0 - вектор начальных значений в точке *to* размера $N\times 1$;

t_0 - начальная точка расчета;

tk - конечная точка расчета;

intvls - число шагов, на которых численный метод находит решение (число строк матрицы результатов).

D - векторная функция размера $N\times 1$ двух аргументов - скалярного *t* и векторного *y*. При этом *y* - искомая векторная функция аргумента *t* того же размера $N\times 1$.

AJ(t, y) - матрица размером N x (N + 1). Первый столбец содержит производные $\frac{\partial D}{\partial t}$. Остальные строки и столбцы представляют собой матрицу Якоби $\frac{\partial D}{\partial y_k}$ системы ОДУ. Например, если

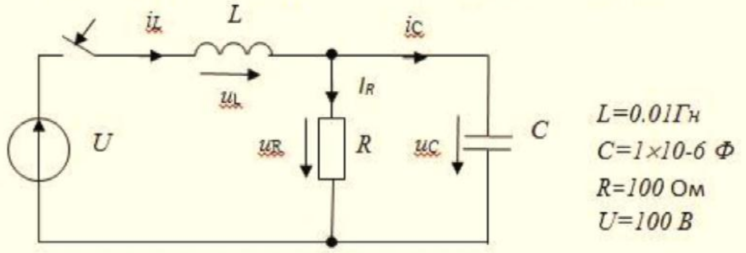
$$D(t,y)=\begin{pmatrix} ty_1 \\ -2y_1y_0 \end{pmatrix}, \text{то } AJ(t,y)=\begin{pmatrix} y_1 & 0 & t \\ 0 & -2y_1 & -2y_0 \end{pmatrix}.$$

Соблюдайте регистр первой буквы рассматриваемых функций, поскольку это влияет на выбор алгоритма счета.

Каждая из приведенных функций выдает решение в виде матрицы размера (*intvls +i*) $\times$ (*N+i*). В ее левом столбце находятся значения аргумента t, делящие интервал на равномерные шаги, а в остальных *N* столбцах - значения искомых функций $y_0(t), y_1(t),\ldots, y_{N-i}(t)$, рассчитанные для этих значений аргумента. Поскольку всего точек (помимо начальной) *intvls*, то строк в матрице решения будет всего *intvls +1*.

Решим систему уравнений методом Рунге Кутты с переменным выбором шага (Рис. 4.9).

Рассчитаем переходный процесс в электрической схеме



Система дифференциальных уравнений, описывающих эту схему, имеет вид:

$$\frac{du_C}{dt} = \frac{1}{C}\left(i_L - \frac{u_C}{R}\right);$$

$$\frac{di_L}{dt} = \frac{1}{L}\left(U - u_C\right).$$

Используем метод Рунге-Кутты с переменным шагом

Исходные данные:

$N := 100$ число строк в матрице результатов

$L := 0.01$ Гн $\quad C := 1\cdot 10^{-6}$ Ф $\quad R := 100$ Ом $\quad U := 100$ В

$$D(t,y) := \begin{bmatrix} \dfrac{\left(y_1 - \dfrac{y_0}{R}\right)}{C} \\ \dfrac{\left(U - y_0\right)}{L} \end{bmatrix}$$

Определяем функцию для расчета производных системы дифференциальных уравнений
Здесь i_L обозначим y_0, а u_C это y_1

$y0 := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ Задаем нулевые начальные условия (схема включается)

$t0 := 0$ время включения ключа (начала переходного процесса)

$tk := 1\cdot 10^{-3}$ время практического установления переходного процесса

$u := \mathrm{Rkadapt}(y0, t0, tk, N, D)$ решаем систему ОДУ

Рис. 4.9 - Программа решения системы уравнений методом Рунге Кутты с переменным выбором шага

Самая значимой здесь является строка листинга, в которой, собственно, определяется система ОДУ. Сравните рассматриваемую систему, записанную в стандартной форме с формальной ее записью в MathCAD, чтобы не делать впоследствии ошибок.

Не забывайте, что векторную функцию *D(t, y)* следует определять через компоненты вектора *y* нажатием клавиши <[>.

Результаты расчета приведены на Рис. 4.10

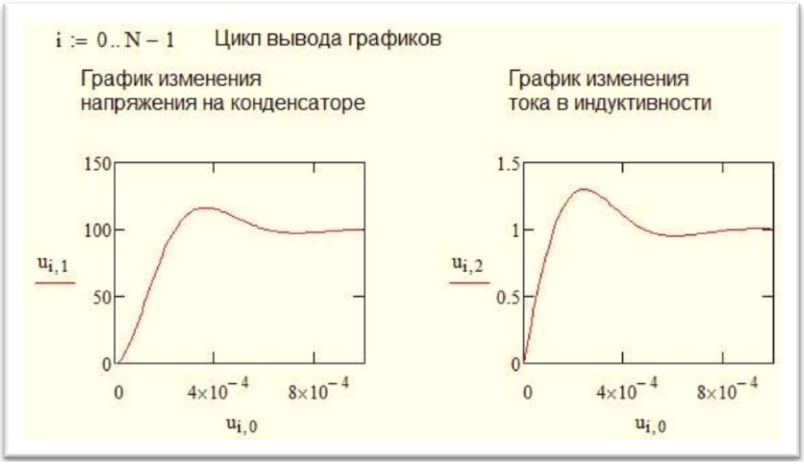


Рис. 4.10 - График решения системы ОДУ методом Рунге-Кутты с переменным выбором шага

В заключение следует сказать несколько слов об особенностях различных численных методов. Все они основаны на аппроксимации дифференциальных уравнений разностными аналогами. В зависимости от конкретной формы аппроксимации, получаются алгоритмы различной точности и быстродействия. В MathCAD использован наиболее популярный алгоритм Рунге-Кутты четвертого порядка, описанный в большинстве книг по методам вычислений. Он обеспечивает малую погрешность для широкого класса систем ОДУ за исключением жестких систем. Поэтому в большинстве случаев стоит применять функцию *rkfixed*. Если по различным причинам время расчетов становится критичным или точность и устойчивость неудовлетворительны, стоит попробовать вместо *rkfixed* другие функции. Сделать это несложно благодаря одинаковому набору параметров - достаточно только поменять имя функции в программе.

Функция *Rkadapt* может быть полезна в случае, когда известно, что решение на рассматриваемом интервале меняется слабо, либо существуют участки медленных и быстрых его изменений. Метод Рунге-Кутты с переменным шагом разбивает интервал не на равномерные шаги, а более эффективным способом. Там, где решение меняется слабо, шаги выбираются более редкими, а в областях его сильных изменений - частыми. В результате для достижения одинаковой точности требуется меньшее число шагов, чем для *rkfixed*.

Остальные методы рекомендуются для решения жёстких систем ОДУ.

5 Поиск экстремума функции в MathCAD

5.1 Поиск локального экстремума функции одной переменной без ограничений

В MathCAD с помощью встроенных функций решается только задача поиска локального экстремума. Чтобы найти глобальный максимум (или минимум), требуется либо сначала вычислить все их локальные значения и потом выбрать из них наибольший (наименьший), либо предварительно просканировать с некоторым шагом рассматриваемую область, чтобы выделить из нее подобласть наибольших (наименьших) значений функции и осуществить поиск глобального экстремума, уже находясь в его окрестности.

Построим график функции f(x) = x4 + 5x3 - 10x на интервале (-5, 2). Как видно из Рис. 5.1, она имеет глобальный максимум на левой границе интервала, глобальный минимум, локальный максимум, локальный минимум и локальный максимум на правой границе интервала (в порядке слева направо).

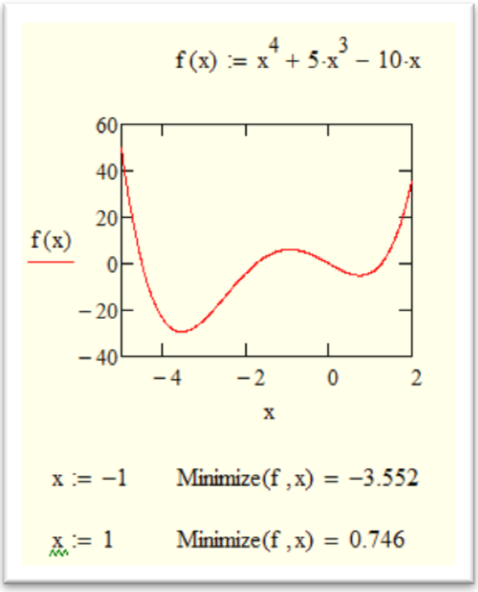


Рис. 5.1 – Нахождение локальных минимумов функции f(x) = $x^4 + 5x^3 - 10x$

Для решения задач поиска максимума и минимума в MathCAD имеются встроенные функции Minerr, Minimize и Maximize. Все они используют те же градиентные численные методы, что и функция Find для решения уравнений.

- ✓ Minimize(f, var1, var2, ...) - Возвращает вектор значений переменных var1, var2, ..., при которых функция f достигает минимума, при условии выполнения ограничений в блоке решения. Возвращает скаляр, если указана одна переменная, иначе возвращает вектор решений (см. Рис. 5.1).
- ✓ Maximize(f, var1, var2, ...) —Возвращает вектор значений переменных var1, var2, ..., при которых функция f достигает максимума, при условии выполнения ограничений в блоке решения (Рис. 5.2).

Всем аргументам функции f предварительно следует присвоить начальные значения, причем они будут восприниматься как начальные приближения для градиентного метода.

Как видно из рисунков, существенное влияние на результат оказывает выбор начального приближения, в зависимости от чего в качестве ответа выдаются различные локальные экстремумы. В последнем случае численный метод вообще не справляется с задачей, поскольку начальное приближение $x = 1$ выбрано далеко от области локального максимума, и поиск решения уходит в сторону увеличения $f(x)$.

5.2 Поиск экстремума функции одной переменной при ограничениях

В задачах на условный экстремум функции минимизации и максимизации должны быть включены в вычислительный блок, т.е. им должно предшествовать ключевое слово Given. В промежутке между Given и функцией поиска экстремума с помощью булевых операторов записываются логические выражения (неравенства, уравнения), задающие ограничения на значения аргументов минимизируемой функции. В листинге Рис. 5.3 показаны примеры поиска условного экстремума на различных интервалах, определенных неравенствами. Сравните результаты работы этого листинга с двумя предыдущими.

$$f(x) := x^4 + 5\cdot x^3 - 10\cdot x$$
$$x := -1$$
$$\text{Given} \quad -5 < x < -2 \quad \text{Minimize}(f, x) = -3.552$$
$$x := 1$$
$$\text{Given} \quad x > 0 \quad \text{Minimize}(f, x) = 0.746$$
$$x := -10$$
$$\text{Given} \quad -3 < x < 0 \quad \text{Maximize}(f, x) = -0.944$$

Рис. 5.3 - Примеры поиска условного экстремума функции

Не забывайте о важности выбора правильного начального приближения и в случае задач на условный экстремум.

$$f(x) := x^4 + 5\cdot x^3 - 10\cdot x$$
$$x := -1 \quad \text{Maximize}(f, x) = -0.944$$
$$x := 1 \quad \text{Maximize}(f, x) = 5.369 \times 10^7$$

Рис. 5.2 – Нахождение локального максимума функции $f(x) = x^4 + 5x^3 - 10x$

5.3 Нахождение экстремума функции многих переменных

Вычисление экстремума функции многих переменных не несет принципиальных особенностей по сравнению с функциями одной переменной. Поэтому ограничимся примером (Рис. 5.4) нахождения максимума и минимума функции, показанной в виде графиков трехмерной поверхности и линий уровня. Обратите внимание, как с помощью неравенств, введенных логиче-

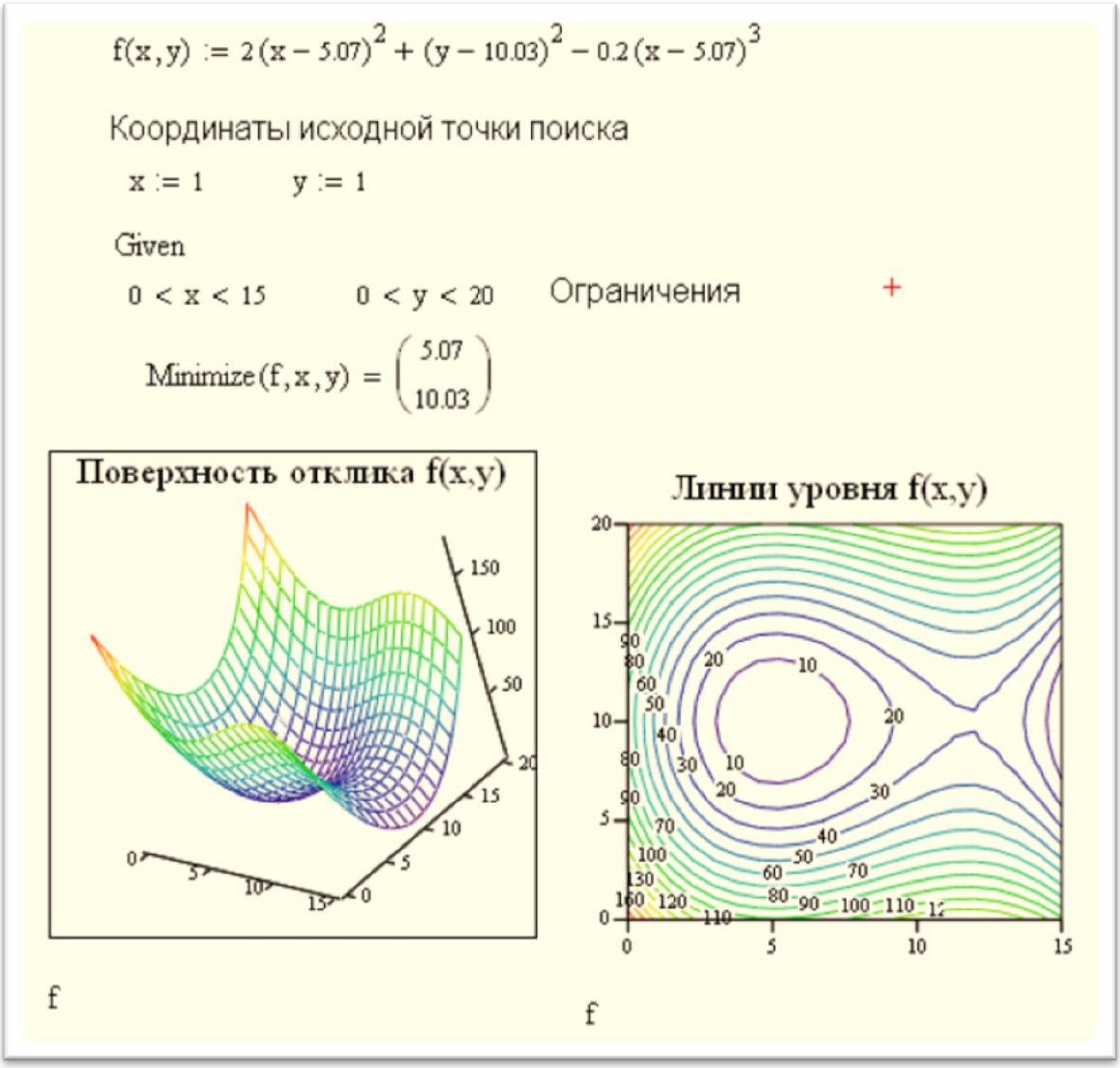
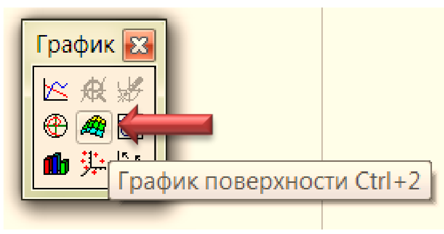


Рис. 5.4 - Экстремум функции двух переменных

скими операторами, задается пространство проектирования (область определения целевой функции) на плоскости (x, y).

Для построения трехмерного графика (графика поверхности) выполните следующие действия:

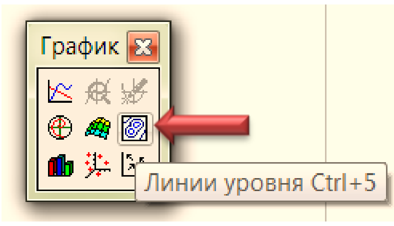
- ✓ Наберите имя функции двух переменных, знак присвоения значения :=, выражение функции.
- ✓ Установите курсор в то место, где вы хотите построить график.
- ✓ В математической панели щелкните мышью на кнопке Панель графиков (Graph Toolbar), изображающей График поверхности (Surface Plot). На месте курсора появится шаблон трехмерного графика.



✓ В единственном поле ввода шаблона графика введите имя функции (без параметров).

✓ Щелкните мышью вне области шаблона. График построен (см. Рис. 5.4 слева).

Аналогично можно построить и график линий уровня. Для этого в математической панели щелкните мышью на кнопке Панель графиков, изображающей график Линий уровня (Contour plot).



Панель форматирования трехмерных графиков, приведенная на Рис. 5.1, содержит 9 вкладок, открывающих огромные возможности форматирования графиков.

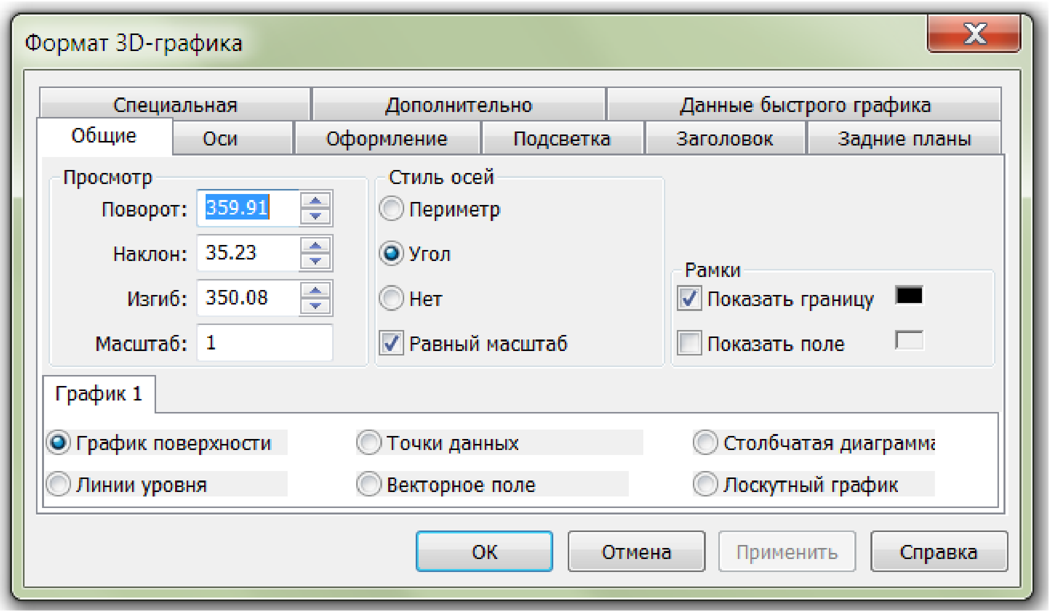


Рис. 5.1 - Панель форматирования трехмерных графиков

6 Символьные вычисления

Кроме числовых расчетов, MathCAD может производить некоторые вычисления и в символьном виде. Если при вычислении выражения в числовом виде MathCAD возвращает одно или несколько чисел, то результатом аналитического вычисления выражения обычно становится другое выражение.

Заметим, что выполняются символьные операции намного сложнее числовых. Многие функции, например интегралы или корни, представить в формульном виде нельзя.

Есть две возможности символьных вычислений:

✓ с использованием меню Символьные операции (Symbolics) из главного меню MathCAD (если необходимо быстро получить однократный аналитический результат для одного выделенного выражения, не сохраняя сам ход вычислений);

✓ с использованием панели Символьные (Symbolic) математической панели и с помощью оператора символьного знака равенства →, ключевых слов и модификаторов символьного процессора и обычных формул. Этот способ более популярен, так как позволяет записывать выражения в традиционной математической форме и сохранять символьные вычисления в документах MathCAD.

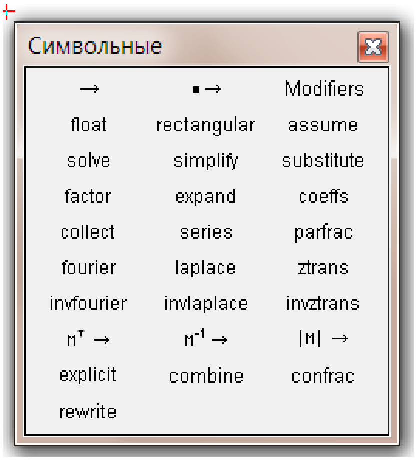


Рис. 6.2 - Панель Символьные операции (Symbolics)

6.1 Преобразование выражений к простому виду

Чтобы преобразовать символическое выражение к более простой форме MathCAD использует различные арифметические формулы, приведение подобных слагаемых, тригонометрические тождества, пересчет обратных функций и др. Чтобы упростить выражение с помощью меню необходимо:

✓ ввести выражение;

✓ выделить мышью или с помощью клавиатуры, как это принято в Windows, выражение целиком или его часть, которую нужно упростить;

✓ выполнить команду меню Символьные операции / Упростить (Symbolics / Simplify). На экране появится результат - упрощенный вариант исходного выражения (Рис. 6.3). Если выражение больше невозможно упростить, MathCAD просто повторяет его.

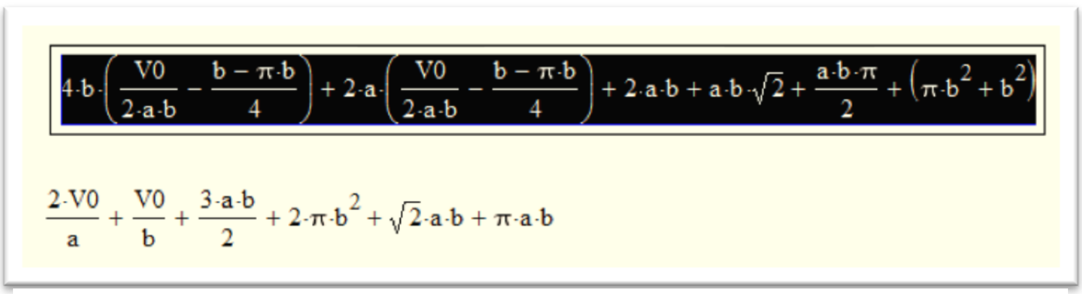


Рис. 6.3 - Использование меню для упрощения выражения

Для упрощения выражения при помощи оператора символьного вывода используйте ключевое слово simplify (упростить) (Рис. 6.4) из панели Символьные операции.

$$(a+2\cdot b)\cdot c-c^2\cdot(a+5\cdot b)+c \text{ simplify } \rightarrow c\cdot(a+2\cdot b-a\cdot c-5\cdot b\cdot c+1)$$

Рис. 6.4 - Использование ключевого слова simplify

Обратите внимание, что если некоторым переменным, входящим в выражение, ранее были присвоены некоторые значения, то они будут подставлены в него при выполнении символьного вывода (Рис. 6.5).

$$a := 10 \qquad b := 2$$

$$(a+2\cdot b)\cdot c-c^2\cdot(a+5\cdot b)+c \text{ simplify } \rightarrow -5\cdot c\cdot(4\cdot c-3)$$

Рис. 6.5 - Упрощение выражения с подстановкой значения переменных

6.2 Расширение выражений (Expand) и разложение на простые множители (Factor)

В ходе символьного разложения, выполняемого с использованием оператора символьного вывода и ключевого слова expand (расширить), раскрываются все суммы и произведения, а сложные тригонометрические зависимости разлагаются с помощью тригонометрических тождеств.

Разложение выражений на простые множители, выполняемое с использованием оператора символьного ввода и ключевого слова factor (разлагать на множители) позволяет разложить полиномы на произведение более простых полиномов, а целые числа — на простые сомножители.

Примеры применения рассмотренных операций разложения приведены на (Рис. 6.6).

$$(a+b)^3 \text{ expand } \rightarrow a^3+3\cdot a^2\cdot b+3\cdot a\cdot b^2+b^3 \quad \text{операция символьного разложения}$$

$$x^4-16 \text{ factor } \rightarrow (x-2)\cdot(x+2)\cdot\left(x^2+4\right) \quad \text{разложение на простые множители}$$

Рис. 6.6 - Разложение выражений (Expand) и на простые множители (Factor)

6.3 Вычисление интегралов и производных

Для вычисления определенного или неопределенного интеграла аналитически следует:

✓ щелкнуть значок $\int$ или $\int_a^b$ панели инструментов Математический анализ (Calculus) (Рис. 2.5), чтобы вставить оператор неопределенного или определенного интеграла;

✓ заполнить местозаполнитель для подынтегрального функцию (предварительно эту функцию можно определить как пользовательскую) и местозаполнители для пределов интегрирования (если они есть);

✓ поместить переменную интегрирования x в местозаполнитель рядом с «d», введем символьный знак равенства «→» и нажмем клавишу Ввод.

✓ для аналитического вычисления производной воспользуемся оператором производной MathCAD $\frac{d}{dx}$ и динамическим символьным знаком равенства «→».

Примеры применения проанализированных операций интегрирования и дифференцирования приведены на Рис. 6.7.

$f(x) := \sin(a \cdot x)\cdot\cos(b \cdot x)$ определяем пользовательскую функцию

$\int f(x)\,dx \rightarrow -\frac{\cos[x\cdot(a-b)]}{2\cdot a - 2\cdot b} - \frac{\cos[x\cdot(a+b)]}{2\cdot a + 2\cdot b}$ вычисляем интеграл

$\frac{d}{dx}f(x) \rightarrow a\cdot\cos(a\cdot x)\cdot\cos(b\cdot x) - b\cdot\sin(a\cdot x)\cdot\sin(b\cdot x)$ берем производную

Рис. 6.7 - Вычисление интегралов и производных

6.4 Приведение подобных слагаемых (Collect)

Для приведения подобных слагаемых полинома с помощью оператора символьного вывода следует:

✓ ввести выражение и нажать кнопку Collect (собирать) на панели Символические (Symbolic);

✓ ввести в местозаполнитель после вставленного ключевого слова collect имя переменной, относительно которой требуется привести подобные слагаемые (в первой строке примера на Рис. 6.8 это переменная a, во второй - b, в третьей - c);

✓ ввести оператор символьного вывода «→» и нажать клавишу Ввод.

$$(a+2\cdot b)\cdot c-c^2\cdot b\cdot(a+3\cdot b)+c \text{ collect}, a \rightarrow \left(c-b\cdot c^2\right)\cdot a+2\cdot b\cdot c-3\cdot b^2\cdot c^2+c$$

$$(a+2\cdot b)\cdot c-c^2\cdot b\cdot(a+3\cdot b)+c \text{ collect}, b \rightarrow -3\cdot c^2\cdot b^2+\left(2\cdot c-a\cdot c^2\right)\cdot b+c+a\cdot c$$

$$(a+2\cdot b)\cdot c-c^2\cdot b\cdot(a+3\cdot b)+c \text{ collect}, c \rightarrow [-b\cdot(a+3\cdot b)]\cdot c^2+(a+2\cdot b+1)\cdot c$$

Рис. 6.8 - Приведение подобных слагаемых по разным переменным

Список литературы

1. **Очков , В.Ф.** *Mathcad 14 дпя студентов, инженеров и конструкторов. .* — СПб. : БХВ-Петербург, 2007. — 368 с.
2. **Каганов, В.И.** *Радиотехника + компьютер + Mathcad.: .* - М. : Горячая линия - Телеком , 2001. - 416 с.
3. **Поршнев С.В., Беленкова И.В.** *Численные методы на базе MathCAD.* - СПб. : БХВ-Петербург, 2005. - 464 с.
4. **Панферов А.И., Лопарев А.В., Пономарев В.К.** *Применение Mathcad в инженерных расчетах: Учеб. пособие.* - СПб. : СПбГУАП, 2004. - 88 с.
5. **Гурский Д.А., Турбина Е.С.** *Вычисления в Mathcad 12. .* — СПб. : Питер, 2006. — 544 с.
6. **Васильев, А.Н.** *Mathcad 13 на примерах.* — СПб. : БХВ-Петербург, 2006. - 528 с.
7. **В.П. Дьяконов, И.В. Абраменкова, А.А. Пеньков.** *Новые информационные технологии: Учебное пособие. Часть 3. Основы математики и математическое моделирование.* - Смоленск : СГПУ, 2003. - 192 с.
8. **Фриск, В.В.** *Основы теории цепей. Расчеты и моделирование с помощью пакета компьютерной математики MathCAD.* — М. : СОЛОН-Пресс, 2006. — 88 с.