

Министерство образования и науки
Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)



Кафедра радиотехнических систем (РТС)



Д.О. Ноздреватых

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ 1

Учебно-методическое пособие по лабораторным работам
для студентов специальности 210601.65 (11.05.01)
«Радиоэлектронные системы и комплексы»

Томск 2014

Министерство образования и науки
Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра радиотехнических систем (РТС)

Утверждаю:
Зав. кафедрой РТС,
профессор, д.т.н.
_____ Г.С. Шарыгин
« » _____ 2014 г.



ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ 1

Учебно-методическое пособие по лабораторным работам
для студентов специальности 210601.65 (11.05.01)
«Радиоэлектронные системы и комплексы»

Разработчик:
Старший преподаватель каф. РТС
_____ Д.О. Ноздревых
« » _____ 2014 г.

АННОТАЦИЯ

Учебно-методическое пособие по лабораторным работам для студентов специальности 210601.65 (11.05.01) «Радиоэлектронные системы и комплексы».

Учебно-методическое пособие по лабораторным работам составлено в соответствии с учетом требований Федерального Государственного образовательного стандарта высшего профессионального образования (ФГОС ВПО) третьего поколения по направлению подготовки (специальности) 210601.65 (11.05.01) «Радиоэлектронные системы и комплексы», утвержденного приказом министерства образования и науки РФ № 2024 от 23 декабря 2010 г.

Содержание

1 Введение.....	8
2 Общие сведения о Mathcad	9
2.1 Интерфейс Mathcad	13
2.1.1 Меню.....	15
2.1.2 Справка по Mathcad.....	15
2.1.3 Панели инструментов	16
3 Основы работы с Mathcad.....	18
3.1 Создания формул и редактирование документа.....	19
3.1.1 Создание формул.....	20
3.1.2 Ввод и редактирование текста.....	25
3.1.3 Правка документа.....	26
3.1.4 Печать и посылка документа по электронной почте.....	27
3.2 Вычисления	27
3.2.1 Числа	27
3.2.2 Переменные	28
3.2.3 Операции.....	29
3.2.3 Операторы присваивания.....	31
3.2.4 Функции.....	32
3.2.5 Функции пользователя.....	33
3.2.6 Простейшие вычисления	34
Лабораторная работа № 1 «Начальные сведения о Mathcad».....	35
3.3 Векторы и матрицы	36
3.3.1 Векторные и матричные операции.....	39
3.3.2 Добавление и удаление столбцов и строк.....	42
3.3.3 Операции над матрицами в аналитической (символьной) форме	43
3.3.4 Основные аксиомы и теоремы теории матриц	45
3.3.5 Применение матриц и векторов для решения систем линейных уравнений	45

Лабораторная работа № 2 «Векторы и матрицы».....	48
3.4 Построение графиков.....	50
3.4.1 Прямоугольная система координат на плоскости	51
3.4.2 Поверхности	57
3.4.3 Полярная система координат на плоскости.....	63
Лабораторная работа № 3 «Построение графиков»	66
3.5 Решение уравнений.....	68
3.5.1 Численное решение нелинейных алгебраических уравнений	68
3.5.2 Численное решение системы линейных алгебраических уравнений.....	71
3.5.3 Решение алгебраических уравнений в аналитической (символьной) форме	74
3.5.4 Решение систем линейных уравнений	74
3.5.5 Символьное решение нелинейных алгебраических уравнений.....	76
Лабораторная работа № 4 «Решение уравнений».....	78
4 Программирование в Mathcad	80
4.1 Введение. О программировании в среде Mathcad	80
4.2 Техника программирования в Mathcad.....	83
4.2.1 Этап 1. «Выработка требований к исходным данным и к форме выдачи результата».....	84
4.2.2 Этап 2. «Разбор способа получения результата. Расчет примеров».....	85
4.2.3 Этап 3. «Составление алгоритма и представление его в схематичной форме»	86
4.2.4 Этап 4. «Расчет контрольных примеров применительно к алгоритму».....	87
4.2.5 Этап 5. «Составление программы, отладка на контрольных примерах».....	88
4.2.6 «Тестирование программы на контрольных примерах»	97
4.2.7 Эксперименты с программой.....	99
Лабораторная работа № 5 «Программирование в Mathcad».....	100
5 Символьные преобразования в Mathcad.....	104
5.1 Введение. Сведения о символьных преобразованиях в Mathcad.....	104

5.1.1	Операции с выделенными выражениями.....	105
5.1.2	Операции с выделенными переменными.....	106
5.1.3	Операции с выделенными матрицами	106
5.1.4	Операции преобразования	106
5.1.5	Стиль эволюции	107
5.2	Осваиваем операции символьной математики	107
5.2.1	Опция Evaluate-Symbolically.....	108
5.2.2	Опция Symbolics-Simplify	110
5.2.3	Опция Symbolics-Expand.....	111
5.2.4	Опция Symbolics-Factor.....	112
5.2.5	Опция Symbolics-Collect	112
5.2.6	Опция Symbolics-Polynomial Coefficients.....	113
5.2.7	Опция Symbolics-Variable	113
5.2.8	Опция Transform-Fourier	117
5.2.9	Опция Transform-Laplace Transform.....	117
5.2.10	Опция Transform-Z	117
	Лабораторная работа № 6 «Символьные преобразования в Mathcad»	118
6	Интерполяция и регрессия, функции сглаживания данных и предсказания	120
6.1	Функции линейной и сплайновой аппроксимации	120
6.1.1	Одномерная линейная аппроксимация	120
6.1.2	Одномерная сплайн-интерполяция и сплайн-аппроксимация.....	121
6.1.3	Двумерная линейная сплайн-интерполяция и сплайн-аппроксимация	123
6.2	Функции для проведения регрессии	125
6.2.1	Функции для линейной регрессии.....	125
6.2.2	Функция для линейной регрессии общего вида	126
6.2.3	Функции для одномерной и многомерной полиномиальной регрессии	127
6.2.4	Функция для нелинейной регрессии общего вида	130

6.3 Функция сглаживания данных	132
6.4 Функция предсказания.....	133
Лабораторная работа № 7 «Интерполяция и регрессия, функции сглаживания данных и предсказания».....	135
7 Учебная задача «Разработка и исследование алгоритма оценки временного положения сигнала при наличии шума»	136
7.1 Полезный сигнал и его параметры	136
7.1.1 Длительность сигнала	137
7.1.2 Энергия сигнала	138
7.2 Случайный шумовой процесс.....	138
7.2.1 Параметры, определяющие шум	138
7.3 Генерация смеси сигнала и шума.....	139
7.4 Оценка временного положения полезного сигнала	140
7.4.1 Алгоритм нахождения временного положения полезного сигнала	140
7.5 Статистическая обработка результатов временного положения сигнала	140
7.5.1 Построение гистограммы.....	140
7.5.2 Расчет среднеквадратичного отклонения (СКО).....	141
7.5.3 Зависимость СКО от отношения сигнал/шум.....	141
Лабораторная работа № 8 «Разработка и исследование алгоритма оценки временного положения сигнала при наличии шума».....	143
Список использованной литературы.....	144
Приложение А	145

1 Введение

Учебно-методическое пособие по лабораторным работам предназначено для быстрого освоения приемов работы в среде пакета программ для проведения математических и инженерных расчетов Mathcad, а также для привития читателю навыков использования Mathcad в текущей учебе. Разработка ориентирована на читателя, имеющего первичные навыки работы с компьютером; знакомство со средой Mathcad не требуется.

Пользователями разработки являются:

- Студенты 1-го курса специальности «Радиоэлектронные системы и комплексы»;

Область использования:

- Проведение лабораторных и организуемых самостоятельных работ в рамках учебной дисциплины «Информационные технологии», предусмотренной учебным планом для специальности «Радиоэлектронные системы и комплексы»;
- Самостоятельное изучение некоторых.

Актуальность темы связана с тем, что пакет Mathcad «встроен» во многие учебные дисциплины как среда для постановки и проведения лабораторных работ, в связи с чем этот пакет должен быть освоен студентами до уровня свободного владения. Преимущество этого пакета перед другими пакетами состоит, прежде всего, в удобном для восприятия, интуитивно ясном представлении материала.

Согласно учебного плана по специальности «Радиоэлектронные системы и комплексы» на лабораторные работы отводится 36 часов.

2 Общие сведения о Mathcad

Mathcad – это мощная и в то же время простая универсальная среда для решения задач в различных отраслях науки и техники, финансов и экономики, физики и астрономии, математики и статистики... Mathcad остается единственной системой, в которой описание решения математических задач задается с помощью привычных математических формул и знаков. Mathcad позволяет выполнять как численные, так и аналитические (символьные) вычисления, имеет чрезвычайно удобный математико-ориентированный интерфейс и прекрасные средства научной графики.

Mathcad является программным средством, предназначенным для выполнения математических расчетов. Вид уравнений на экране Mathcad совпадает с первичной математической записью. Как в электронных таблицах, любое изменение содержимого рабочего документа Mathcad вызывает обновление всех зависимых результатов, в том числе перерисовку графиков. Текстовый процессор Mathcad поддерживается множеством шрифтов.

Система Mathcad существует в нескольких основных вариантах:

- Mathcad Standard – идеальная система для повседневных технических вычислений. Предназначена для массовой аудитории и широкого использования в учебном процессе;
- Mathcad Professional – промышленный стандарт прикладного использования математики в технических приложениях. Программа ориентирована на математиков и научных работников, проводящих сложные и трудоемкие расчеты.
- Mathcad Professional Academic – пакет программ для профессионального использования математического аппарата с электронными учебниками и ресурсами.

Данное пособие ориентировано на использование пакета Mathcad Professional.

Назначения и возможности Mathcad:

- ввод на компьютере разнообразных математических выражений (для дальнейших расчетов или создания документов, презентаций, Web-страниц);
- проведение математических расчетов;
- подготовка графиков с результатами расчетов;
- ввод исходных данных и вывод результатов в текстовые файлы или файлы с базами данных в других форматах;
- подготовка отчетов работы в виде печатных документов;
- подготовка Web-страниц и публикация результатов в Интернете;
- получение различной справочной информации из области математики, и многое другое.

Особенности работы в среде Mathcad:

- в отличие от большинства других математических приложений, построен по принципу «что видите, то и получаете»;
- математические выражения и текст вводятся с помощью формульного редактора Mathcad, который по возможностям и простоте использования не уступает, к примеру, редактору формул, встроенному в Microsoft Word;
- математические расчеты производятся немедленно, в соответствии с введенными формулами;
- графики различных типов (по выбору пользователя) с богатыми возможностями форматирования вставляются непосредственно в документы;
- возможен ввод или вывод данных в файлы различных форматов;
- документы могут быть распечатаны непосредственно в Mathcad в том виде, который пользователь видит на экране компьютера, или сохранены в формате RTF для последующего редактирования в более мощных текстовых редакторах (например, Microsoft Word);
- возможно полноценное сохранение документов Mathcad в формате Web-страниц (генерация вспомогательных графических файлов происходит автоматически);
- имеется опция объединения разрабатываемых вами документов в электронные книги, которые, с одной стороны, позволяют в удобном виде хранить математическую информацию, а с другой – являются полноценными Mathcad-программами, способными осуществлять расчеты;
- символьные вычисления позволяют осуществить аналитические преобразования, а также мгновенно получать разнообразную справочную математическую информацию.

Mathcad имеет свою справочную систему. Обращение к справочной системе: клавиша <F1>; меню HELP; комбинация <Shift>+<F1> – для открытия контекстной, т.е. обозначенной указателем, справки по теме.

Mathcad позволяет решать в одном рабочем документе текст, математические формулы и строить необходимые графики. Основные приемы в Mathcad такие же, как и в любом приложении Windows.

Перечислим основные возможности Среды Mathcad.

Общие возможности

1. Разработка и редактирование документов, содержащих как математические формулы любой сложности, так и все встроенные инструменты Среды Mathcad. Подготовка этих документов к изданию или передаче по сети Internet.
2. Использование общепринятого расширяемого языка разметки XML как универсального способа организации обмена данными с другими приложениями. Это позволяет преобразовывать файлы Mathcad в HTML-страницы и в формат PDF.
3. Возможность вставки в документ широкого спектра объектов
4. Разработка веб-документов и сетевые возможности по их пересылке, получению обновлений и поддержки.

5. Получение документов Mathcad по сети и выполнение расчетов в этих документах.

6. Получение через Internet и подключение новых книг расширения для реализации дополнительных возможностей среды Mathcad.

7. Доступный официальный форум.

8. Использование серверных услуг среды Mathcad (Mathcad Application Server) – удаленное подключение к пакету Mathcad в полной комплектации через стандартный веб-браузер Internet, даже если Mathcad не установлен на компьютере.

9. Импорт документов из MS Access и MS Excel и математическая обработка данных из этих документов.

10. Наличие обширных справочных материалов по математике и основных физических констант, а также большое количество задач с готовыми решениями по многим разделам науки и техники.

11. Выполнение вычислений любой сложности – использование среды Mathcad в качестве сверхмощного научного интеллектуального калькулятора с применением богатой библиотеки встроенных функций (более 680; для сравнения, в MS Excel их около 200), с точностью до 17 значащих цифр (а при использовании специальных операторов – и до 250) и с неограниченными возможностями запоминания промежуточных результатов. При этом имеется возможность вычислений как по введенной в документ формуле целиком, так и по отдельному, выделенному фрагменту формулы.

12. Использование графического редактора для построения двумерных и трехмерных графиков любой сложности, наглядных диаграмм и не только для простого построения, но и для связи графика с формулой, при которой изменение параметра сразу отражается на кривой графика. Имеется также возможность создание объектов движущейся анимации и просмотра импортированных файлов, например, видеофильмов в формате AVI при помощи встроенного в среду Mathcad проигрывателя Playback.

13. Действия с размерностями.

Численные методы вычислений

- Решение уравнений и систем уравнений, как линейных, так и нелинейных. Нахождение корней многочлена.

- Решение неравенств.

- Вычисление определенного интеграла.

- Вычисление несобственных интегралов.

- Вычисление кратных интегралов.

- Численные методы дифференцирования.

- Численное решение обыкновенных дифференциальных уравнений – задача Коши.

- Численное решение обыкновенных дифференциальных уравнений – решение краевой задачи.

- Решение дифференциальных уравнений в частных производных.

- Вычисление суммы и произведения членов ряда.

- Исследование функций и численное определение экстремумов функций одной и нескольких переменных, построение асимптот.
- Решение оптимизационной задачи методом линейного программирования.

Символьные вычисления

1. Выполнение точных вычислений с представлением результатов в традиционной математической форме – с записью ответа в форме радикала и специальных иррациональных чисел π , e , C .

2. Символьные преобразования математических выражений целиком или их фрагментов:

1. разложение выражений в более простые;
2. приведение подобных;
3. разложение на множители;
4. приведение к общему знаменателю;
5. вынесение общего множителя;
6. разложение на элементарные дроби;
7. вычисление коэффициентов полиномов;
8. выполнение подстановок.

3. Аналитическое решение уравнений и систем уравнений.

4. Дифференцирование в символьной форме определение производных любых порядков.

5. Аналитическое определение первообразной.

6. Построение касательной и нормали к плоской кривой и к поверхности.

7. Аналитическое вычисление определенного интеграла.

8. Символьное вычисление кратных интегралов.

9. Решение неравенств.

10. Аналитическое вычисление предела.

11. Аналитическое вычисление суммы ряда конечного или бесконечного.

12. Аналитическое вычисление произведения членов ряда конечного или бесконечного.

13. Аналитическое вычисление суммы/произведения членов ряда конечного или бесконечного, когда пределы и шаг изменения индекса члена ряда задаются (например, сложить четные числа от 10 до $10+k$.)

14. Разложение в ряд Тейлора.

15. Разложение в ряд Фурье.

16. Символьное преобразование Фурье и Лапласа – прямое и обратное.

17. Операции с матрицами в символьной форме: умножение и сложение матриц, поиск обратной матрицы, вычисление определителя, поиск собственных значений и собственных векторов.

Работа с матрицами и матричные вычисления

1. Элементарные матричные действия: создание, импорт, заполнение матриц, задание матриц специального вида, умножение,

сложение, транспонирование и сортировка матрицы в целом или ее фрагмента. Выполнение векторизации – однотипных действий над всеми элементами матрицы.

2. Вычисление определителя, размерности, ранга и следа матрицы, скалярное и векторное умножение векторов, вычисление якобиана, например, для перехода к другим системам координат в тройном интеграле. Вычисление собственных значений и собственных векторов, поиск максимального и минимального элемента матрицы.

3. Матричные преобразования: скалярное и векторное умножение векторов, поиск обратной матрицы и решение системы алгебраических линейных уравнений, всевозможные разложения матрицы на произведение матриц специального вида: двух треугольных – верхней и нижней (LU-преобразование), треугольной и ее же транспонированной (разложение Холецкого), ортогональной и верхней треугольной (QR-разложение), сингулярное разложение.

4. Интегрирование среды Mathcad с матричной математической системой MATLAB и возможность использования ее аппарата открывает удивительные возможности эффективного решения матричных задач неограниченной сложности.

Решение дифференциальных уравнений

Программирование

Составление программ и выполнение расчетов на упрощенном процедурном алгоритмическом языке с возможностью использования всех процедурных конструкций: условных операторов, циклов, массивов, модуль-функций, модуль-процедур.

Комплексные числа

1. Представление комплексных чисел в традиционной форме, возможность выполнения основных арифметических действия с ними.

2. Возможность автоматического получения результатов многих вычислений в виде комплексного числа (например, всех корней многочлена).

3. Возможность задания комплексного аргумента для многих библиотечных функций и получение математически корректного результата.

Обработка данных и финансовые расчеты

Теория вероятностей и математическая статистика

Математическое моделирование

Специальные возможности по прикладным инженерным и научным расчетам

1. Обработка электрических сигналов и расчет электронных устройств.

2. Виртуальная генерация электрических сигналов и их обработка.

2.1 Интерфейс Mathcad

Mathcad работает с *документами*. С точки зрения пользователя, документ – это чистый лист бумаги, на котором можно размещать блоки трех

основных типов: математические выражения, текстовые фрагменты и графические области.

Расположение нетекстовых блоков в документе имеет принципиальное значение – *слева направо и сверху вниз*.

Для создания документа в среде Mathcad необходимо в требуемой папке, например, на рабочем столе в папке «Петров», выбрать из контекстного меню команду *Создать – Mathcad Document*, с клавиатуры ввести название этого документа, например, «Задание к ЛР-1 по Mathcad» и открыть документ, дважды щелкнув на созданный файл. На экране откроется окно (рис. 2.1)

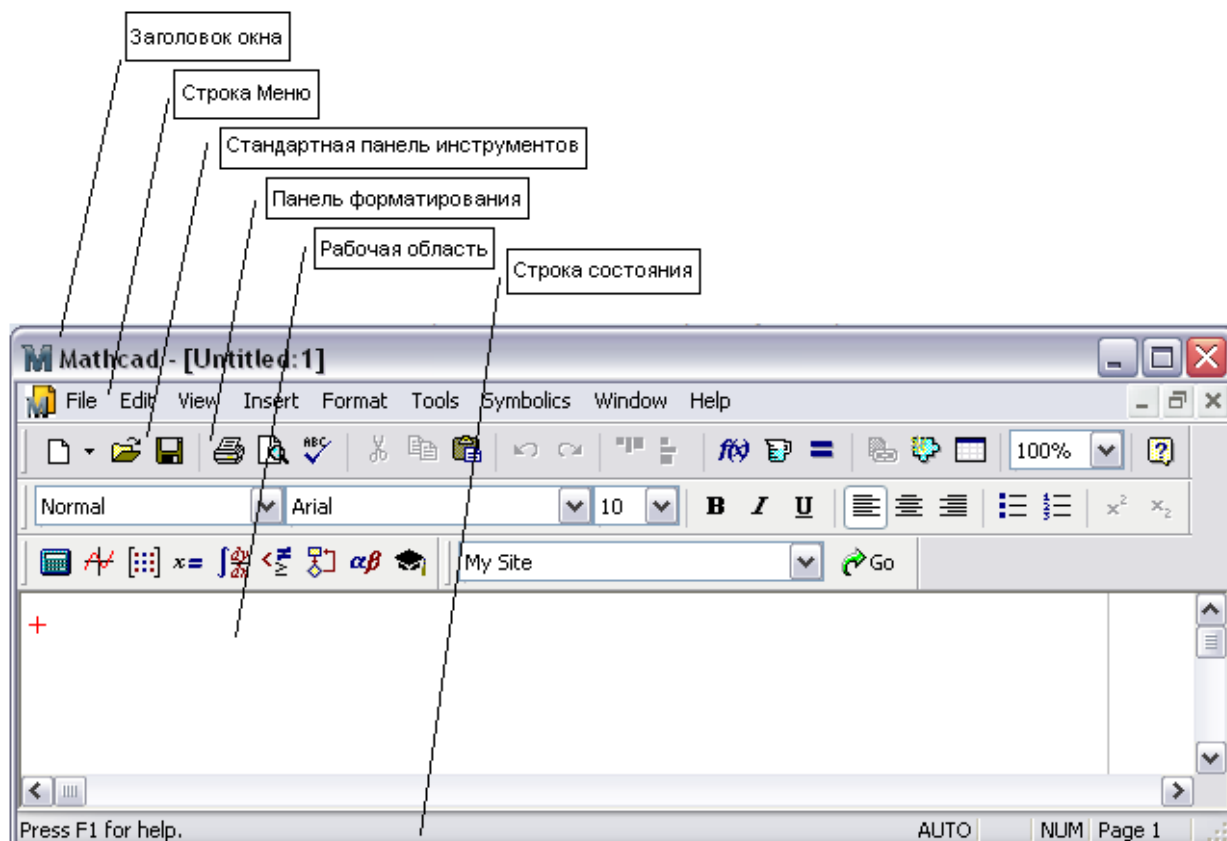


Рис. 2.1 – Созданный документ в Mathcad

Таким образом, оконный интерфейс Mathcad содержит:

1. Заголовок;
2. Строку меню, пункты которого доступны при помощи мыши и клавиатуры (краткое нажатие на клавишу ALT, затем стрелки и ENTER);
3. Стандартной панели инструментов (на них вынесены наиболее часто употребляемые команды). Доступны только при помощи мыши. Активизируются по необходимости из меню инструментов *View-Toolbars*;
4. Линейка горизонтальная (активизируется *View-Rules*). Доступна только при помощи мыши;
5. Горизонтальная и вертикальная полосы прокрутки. Доступны при помощи мыши;
6. Рабочая область;
7. Контекстное меню (активизируется щелчком правой кнопки мыши для конкретного положения курсора);

8. Строка состояния (активизируется *View-Status Bar*). Отображает (слева направо):

- контекстно-зависимая подсказка о готовящемся действии,
- режим вычислений (**AUTO**) или задаваемый вручную (**Calc F9**),
- текущий режим раскладки клавиатуры **NUM**,
- номер страницы, на которой находится курсор.

2.1.1 Меню

Строка меню располагается в самой верхней части окна Mathcad. Она содержит девять заголовков, щелчок мышью на каждом из которых выводит соответствующее меню с перечнем команд:

- *File* (файл) – команды, связанные с созданием, открытием, сохранением, пересылкой по электронной почте и распечаткой на принтере файлов с документами;
- *Edit* (Правка) – команды, относящиеся к правке текста (копирование, вставка, удаление фрагментов и т.п.);
- *View* (Вид) – команды, управляющие внешним видом документа в окне редактора Mathcad, а также команды, создающие файлы анимации;
- *Insert* (Вставка) – команды вставки различных объектов в документы;
- *Format* (Формат) – команды формирования текста, формул и графиков;
- *Tools* (Инструменты) – команды управления расположением окон с различными документами на экране;
- *Symbolics* (Символика) – команды символьных вычислений;
- *Window* (Окно) – команды управления расположением окон с различными документами на экране;
- *Help* (Справка) – команды вызова справочной информации, сведений о версии программы, а также доступа к ресурсам и электронным книгам.

2.1.2 Справка по Mathcad

При необходимости получения справочных сведений во время работы можно использовать один из следующих ресурсов.

- Кнопка <F1> служит для быстрого контекстного доступа к справочной системе;
- *Mathcad Help* (Справка) – система справки или технической поддержки.

Кроме того, в зависимости от версии пакета возможно получение дополнительной справочной информации по *Developer's Reference* (Справке для разработчиков), *Author's Reference* (Справке для авторов) или из *Resource Center* (Центра описания ресурсов).

В случаях, когда назначение команды или кнопки неизвестно или требуется узнать больше о каком-либо параметре в диалоговом окне, нужные сведения можно получить с помощью всплывающих подсказок.

Всплывающие подсказки содержат сведения о различных элементах, отображаемых на экране.

2.1.3 Панели инструментов

Панели инструментов служат для быстрого (за один щелчок мыши) выполнения наиболее часто применяемых команд. Все действия, которые можно выполнить с помощью панелей инструментов, доступны и через верхнее меню. На рисунке 2.2 изображено окно Mathcad, содержащее три основные панели инструментов, расположенные непосредственно под строкой меню. Кнопки в панелях сгруппированы по сходному действию команд:

- *Standard (Стандартные)* – служит для выполнения большинства операций, таких, как действия с файлами, редакторская правка, вставка объектов и доступ к справочным системам;

- *Formatting (Форматирование)* – для формирования (изменения типа и размера шрифта, выравнивания и т.п.) текста и формул;

- *Math (Математика)* – для вставки математических символов и операторов в документы;

Панель *Math* предназначена для вызова на экран еще девяти панелей (рис. 2.2), с помощью которых, собственно, и происходит вставка математических операций в документы. В прежних версиях Mathcad эти математические панели инструментов назывались *палитрами* (palettes) или *наборными панелями*. Чтобы показать какую-либо из них, нужно нажать соответствующую кнопку на панели *Math*. Перечислим назначение математических панелей:

- *Calculator (Арифметика)* – служит для вставки основных математических операций;

- *Graph (Графики)* – для вставки графиков;

- *Matrix (Матрицы)* – для вставки матриц и матричных операторов;

- *Evaluation (Оценка)* – для вставки операторов управления вычислениями;

- *Calculus (Исчисления)* – для вставки операторов интегрирования, дифференцирования, суммирования;

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

- *Programming (Программирование)* – для программирования средствами Mathcad;

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

- *Symbolic (Символы)* – для вставки символьных операторов.

При наведении указателя мыши на многие из кнопок математических панелей появляется всплывающая подсказка, содержащая еще и сочетание «горячих клавиш», нажатие которых приведет к эквивалентному действию.

«Горячие клавиши» приведены в Приложении 3 «Ввод встроенных операторов».

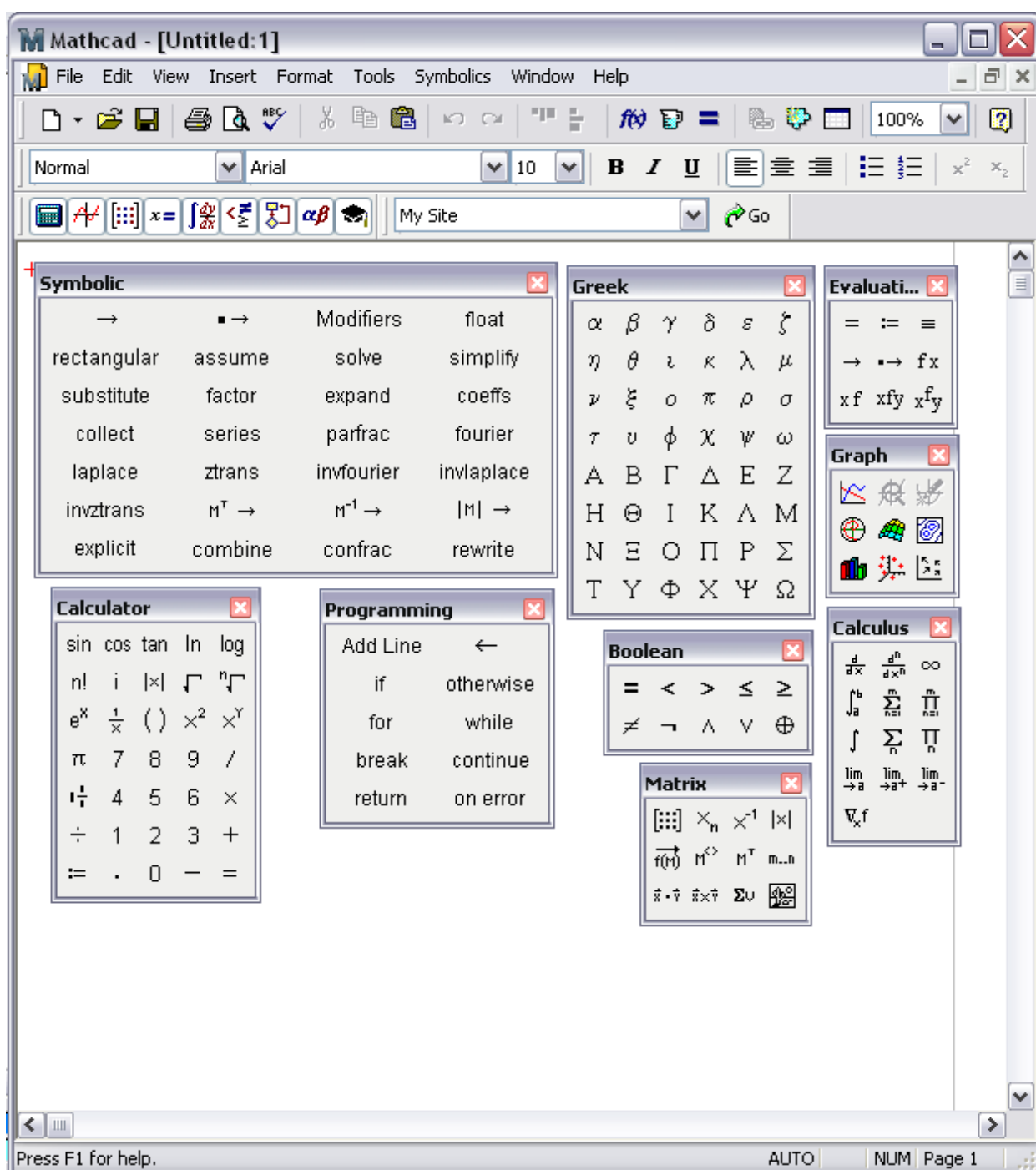


Рис. 2.2 – Панели в пакете MathCAD

Текстовые области можно размещать в любом месте документа командой меню *Insert/Text Region* или вводом с клавиатуры символа ["].

Для ввода математических формул и графиков можно применять панели (палитры) кнопок в клавиатуру. Панели приведены на следующем рисунке 2.2.

Если при запуске нового файла вы не обнаружили общую панель инструментов *Math* (рис. 2.3), то включить ее можно следующим образом (рис. 2.4):

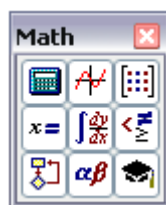


Рис. 2.3 – Общая панель инструментов Math

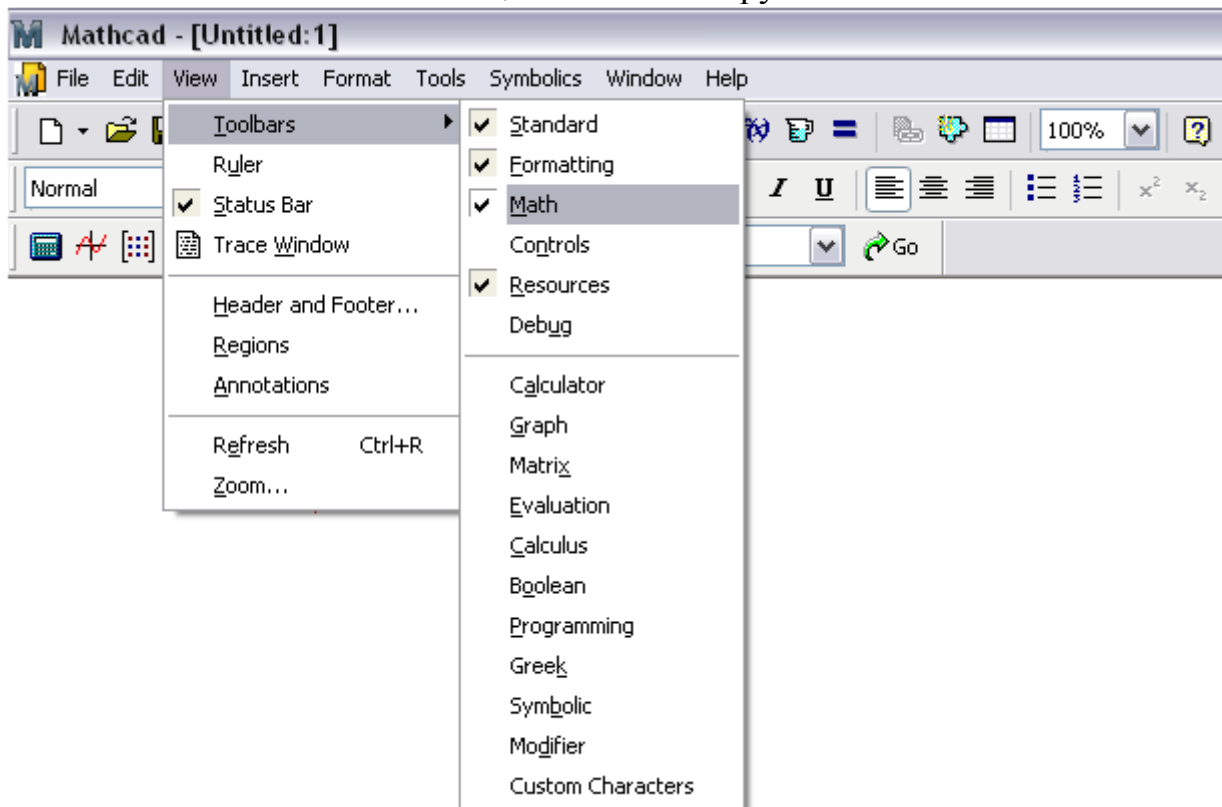


Рис. 2.4 – Включение панели инструментов Math

3 Основы работы с Mathcad

В Mathcad все расчеты организуются на рабочих областях, или «листах» (worksheets), изначально пустых, на которые можно добавлять формулы и текст. Здесь и далее будем называть рабочий лист *документом* Mathcad. Это не совсем точно передает смысл английского термина «worksheet», зато более привычно с точки зрения терминологии Windows-приложений. Каждый Документ представляет собой независимую серию математических расчетов и сохраняется в отдельном файле. Документ является одновременно и листингом Mathcad-программы, и результатом исполнения этой программы, и отчетом, пригодным для распечатки на принтере или публикации в Web.

После запуска пакета в рабочей области окна создается новый пустой документ.

Управление документом осуществляется достаточно просто. Манипулируя меню File можно:

- создать новый документ (New), комбинация клавиш <Ctrl>+<N>;

- открыть для редактирования уже имеющийся документ (Open), комбинация клавиш <Ctrl>+<O>;
- закрыть документ (Close), комбинация клавиш <Ctrl>+<W>;
- сохранить документ (Save), комбинация клавиш <Ctrl>+<S>;
- сохранить документ под новым именем или типом (Save As);
- распечатать документ (Print), комбинация клавиш <Ctrl>+<P>;
- отослать документ по e-mail (Send);
- назначить параметры распечатки страниц документа (Page Setup);
- просмотреть документ (Print Preview);
- закрыть приложение (Exit).

Кроме того, открыв меню *File* можно увидеть открытые ранее документы Mathcad.

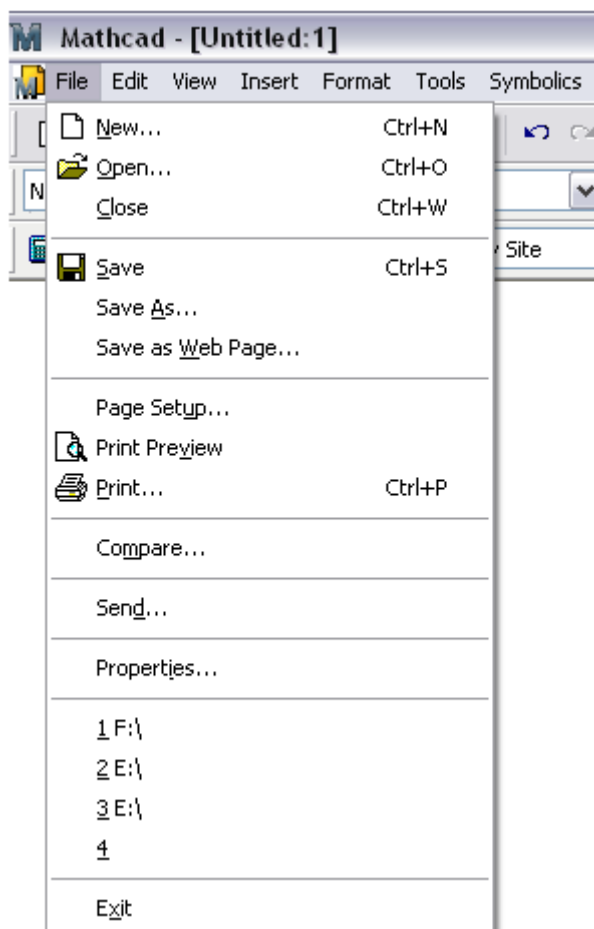


Рис. 3.1 – Вид меню File

3.1 Создания формул и редактирование документа

Язык, на котором изъясняются в Mathcad для изображения констант, переменных величин, операторов, функций, уравнений и иных математических записей, практически полностью совпадает с общепринятым в математике.

Символами этого языка являются: малые и заглавные буквы латинского и греческого алфавита; арабские цифры от 0 до 9; знаки математических операций (+, -, *, /, = и т.д.); имена функций (cos, sin, tan, log, n!, т.д.) и некоторые специальные знаки.

В математике различают константы (целые и вещественные) и переменные величины. Значение константы остается неизменным в процессе выполнения программы, значение переменной – может изменяться.

Пример записи целых констант:

1, -5, 0, 769, -3 и т.д.

Примеры записи вещественных констант:

0.564, -89.439, 7.72 и т.д.

Вещественные константы можно также представить в сжатой форме при большом числе знаков с использованием буквы E в качестве основания 10: aEn , где a – целое или дробное число, E – основание 10, n – целое число, являющееся показателем степени 10.

Большую роль в наборе чисел играет расположение уголка (на экране он синего цвета). Например (рисунок 3.2.a), если уголок расположен, как показано в левой части рисунка, то любые знаки операций (сложение, вычитание и т.п.) будут добавляться к показателю степени, если же уголок расположен, как показано в правой части рисунка, то они будут добавляться ко всему выражению.

3.1.1 Создание формул

Формульный редактор Mathcad позволяет быстро и эффективно вводить и изменять математические выражения. Тем не менее, некоторые аспекты его применения не совсем интуитивны, что связано с необходимостью избежать ошибок при расчетах по этим формулам. Поэтому не пожалейте немного времени на знакомство с особенностями формульного редактора, и впоследствии при реальной работе вы сэкономите гораздо больше.

Элементы интерфейса редактора Mathcad:

- указатель мыши (mouse pointer) – играет обычную для приложений Windows роль, следуя за движениями мыши;
- курсор – обязательно находится внутри документа в одном из трех видов:
 - курсор ввода (crosshair) – крестик красного цвета, который отмечает пустое место в документе, куда можно вводить текст или формулу;
 - линии ввода (editing lines) – горизонтальная (underline) и вертикальная (insertion line) линии синего цвета, выделяющие в тексте или формуле определенную часть (рис. 3.2б);
 - линии ввода текста (text insertion point) – вертикальная красная линия, аналог линий ввода для текстовых областей (рис. 3.3).

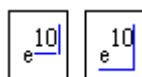




Рис. 3.2 а, б – Линии ввода



Рис. 3.3 – Линия ввода текста

- местозаполнители (placeholders) – появляются внутри незавершенных формул в местах, которые должны быть заполнены символом или оператором:

- местозаполнитель символа – черный прямоугольник;
- местозаполнитель оператора – черная прямоугольная рамка.

Mathcad выполняет расчеты по формулам, которые близки к естественной математической символике. *Формула* – это математическое выражение, которое состоит из операндов, соединенных знаками математических операций. *Операндами* могут быть числа, переменные, встроенные и определенные пользователем функции.

К основным элементам математических выражений Mathcad относятся *типы данных, операторы, функции и управляющие структуры*.

Операторы

Операторы – элементы Mathcad, с помощью которых можно создавать математические выражения. К ним, например, относятся символы арифметических операций, знаки вычисления сумм, произведений, производной и интеграла и т.д.

Оператор определяет:

1. действие, которое должно выполняться при наличии тех или иных значений операндов;
2. сколько, где и какие операнды должны быть введены в оператор.

Операнд – число или выражение, на которое действует оператор. Например, в выражении $5! + 3$ число 3 и выражение $5!$ – операнды оператора $+$ (плюс), а число 5 операнд оператора факториал ($!$). После указания *операндов* операторы становятся исполняемыми по документу блоками.

Типы данных

К *типам данных* относятся числовые константы, обычные и системные переменные, массивы (векторы и матрицы) и данные файлового типа.

Константами называют поименованные объекты, хранящие некоторые значения, которые не могут быть изменены. *Переменные* являются поименованными объектами, имеющими некоторое значение, которое может изменяться по ходу выполнения программы. Тип переменной определяется ее значением; переменные могут быть числовыми, строковыми, символьными и т. д. Имена констант, переменных и иных объектов называют *идентификаторами*. Идентификаторы в Mathcad представляют собой набор латинских или греческих букв и цифр.

Для ввода формул можно использовать пиктограммы всевозможных меню из математической панели инструментов. Пиктограммы позволяют вводить знаки математической символики, греческие буквы, графические и текстовые блоки следующим образом:

- указать позицию ввода: указатель мыши переместить в нужное место экрана и щелкнуть левой кнопкой мыши, в результате это место отметится крестиком;
- выбрать нужную пиктограмму и щелкнуть на ней левой кнопкой мыши (объект, обозначенный на пиктограмме, появится на месте крестика).

Математические выражения имеют строго определенную структуру. Выражения не просто печатаются, а строятся по правилам старшинства операций и другим дополнительным правилам, которые упрощают ввод знаменателей, показателей степени, выражений в радикалах. Скобки вставляются везде, где это необходимо, чтобы указать нужный порядок вычислений. Ключевым шагом в построении и редактировании формул является заключение части выражения в выделяющуюся рамку. При этом то, что заключено в выделяющуюся рамку, становится операндом следующего вводимого оператора. Управлять выделяющей рамкой можно мышью, клавишами управления курсором и клавишей [Space].

При редактировании необходимо следить за правильным размещением выделяющей рамки. Если нужно вставить оператор перед существующим выражением, то его заключают в рамку и нажимают клавишу [Ins].

В MathCAD содержится небольшая группа особых объектов, которые нельзя отнести ни к классу констант, ни к классу переменных, значения которых определены сразу после запуска программы. Их правильнее считать *системными переменными*, имеющими предопределенные системой начальные значения. Изменение значений системных переменных производят во вкладке *Встроенные переменные* диалогового окна *Math Options* команды *Математика – Опции*.

Обычные переменные отличаются от системных тем, что они должны быть предварительно *определены* пользователем, т. е. им необходимо хотя бы однажды *присвоить значение*. В качестве *оператора присваивания* используется знак $:=$, тогда как знак $=$ отведен для *вывода значения* константы или переменной.

Если переменной присваивается начальное значение с помощью оператора $:=$, вызывается нажатием клавиши $:$ (двоеточие) на клавиатуре,

такое присваивание называется *локальным*. До этого присваивания переменная не определена и ее нельзя использовать. Однако, с помощью знака \equiv (клавиша \sim на клавиатуре) можно обеспечить *глобальное* присваивание.

Mathcad прочитывает весь документ дважды слева направо и сверху вниз. При первом проходе выполняются все действия, предписанные локальным оператором присваивания (\equiv), а при втором – производятся действия, предписанные локальным оператором присваивания ($:=$), и отображаются все необходимые результаты вычислений ($=$).

Существуют также жирный знак равенства $=$ (комбинация клавиш $\langle \text{Ctrl} \rangle + \langle \Leftrightarrow \rangle$), который используется, например, как оператор приближенного равенства при решении систем уравнений, и символьный знак равенства \rightarrow (комбинация клавиш $\langle \text{Ctrl} \rangle + \langle + \rangle$).

Дискретные аргументы – особый класс переменных, который в пакете Mathcad зачастую заменяет *управляющие структуры*, называемые циклами (однако полноценной такая замена не является). Эти переменные имеют ряд фиксированных значений, либо целочисленных (1 способ), либо в виде чисел с определенным шагом, меняющихся от начального значения до конечного (2 способ).

1. $Name := Nbegin .. Nend,$

где $Name$ – имя переменной, $Nbegin$ – ее начальное значение, $Nend$ – конечное значение, $..$ – символ, указывающий на изменение переменной в заданных пределах (вводится клавишей $;$). Если $Nbegin < Nend$, то шаг переменной будет равен $+1$, иначе -1 .

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

Здесь $Step$ – заданный шаг изменения переменной (он должен быть положительным, если $Nbegin < Nend$, или отрицательным в обратном случае).

Дискретные аргументы значительно расширяют возможности Mathcad, позволяя выполнять многократные вычисления или циклы с повторяющимися вычислениями, формировать векторы и матрицы.

Ввести математическое выражение можно в любом пустом месте документа Mathcad. Для этого поместите курсор ввода в желаемое место документа, щелкнув в нем мышью, и просто начинайте вводить формулу, нажимая клавиши на клавиатуре. При этом в документе создается *математическая область* (math region), которая предназначена для хранения формул, интерпретируемых процессором Mathcad. Продемонстрируем последовательность действий на примере ввода выражения x^{5+x} .

1. Щелкните мышью, обозначив место ввода.

2. Нажмите клавишу $\langle x \rangle$ – в этом месте вместо курсора ввода появятся регион с формулой, содержащей один символ x , причем он будет выделен линиями ввода.

3. Введите оператор возведения в степень, нажав сочетание клавиш $\langle \text{Shift} \rangle + \langle \wedge \rangle$, либо выбрав кнопку возведения в степень на панели

инструментов Calculator – в формуле появится местозаполнитель для введения значения степени, а линии ввода выделяют этот местозаполнитель.

4. Последовательно введите остальные символы $\langle 5 \rangle$, $\langle + \rangle$, $\langle x \rangle$.

Если же далее в конце выражения поставить знак равенства, а перед самым выражением присвоить переменной x значение 11, то получим вычисленное математическое выражение (рис. 3.4).

$$x := 11$$

$$x^{5+x} = 4.595 \times 10^{16}$$

Рис. 3.4 – Пример вычисления

Таким образом, поместить формулу в документ можно, просто начиная вводить символы, числа и операторы, например $+$ или $/$. Во всех этих случаях на месте курсора ввода создается математическая область, иначе называемая *регионом*, с формулой, содержащей и линии ввода. В последнем случае, если пользователь начинает ввод формулы с оператора, в зависимости от его типа, автоматически появляются и местозаполнители, без заполнения которых формула не будет восприниматься процессором Mathcad.

Чтобы изменить формулу, щелкните на ней мышью, поместив таким образом в ее область линии ввода, и перейдите к месту, которое хотите исправить. Перемещайте линии ввода в пределах формулы одним из двух способов:

- щелкая в нужном месте мышью;
- нажимая на клавиатуре клавиши со стрелками, пробел и $\langle \text{Ins} \rangle$:
 - клавиши со стрелками имеют естественное назначение, переводя линии ввода вверх, вниз, влево и вправо;
 - клавиша $\langle \text{Ins} \rangle$ переводит вертикальную линию ввода с одного конца горизонтальной линии ввода на противоположный;
 - пробел предназначен для выделения различных частей формулы. Если раз за разом нажимать клавишу пробела в формуле, то линии ввода будут циклически изменять свое положение.

В Mathcad возможно вырезание (*Edit-Cut* или $\langle \text{Ctrl} \rangle + \langle X \rangle$), копирование (*Edit-Copy* или $\langle \text{Ctrl} \rangle + \langle C \rangle$) выделенной формулы или ее части с последующей вставкой (*Edit-Paste* или $\langle \text{Ctrl} \rangle + \langle V \rangle$).

И так, математические выражения содержат, как правило, самые различные, в том числе специфические символы, набор которых выполняется в Mathcad не так, как в большинстве текстовых процессоров. При этом для вставки символов доступны следующие инструменты:

- клавиатура;
- панель инструментов Greek для ввода греческих символов;
- панель инструментов Calculator для ввода операторов;
- команда Insert-Function для ввода имен функций (возможен ввод с клавиатуры);

- клавиша <'> (апостроф) для выделения скобками уже введенной части формулы, помещенной между линиями ввода.

Некоторые операторы, например оператор присваивания ($:=$) или знак умножения имеют множественное отображение. Для выбора нужного их отображения в документе следует использовать контекстное меню, вызываемое нажатием правой кнопки мыши после установления курсора на знаке оператора. Выбор отображения таких операторов по умолчанию осуществляется настройками вкладки *Display* (Отображение) диалогового окна *Worksheet Options* (Опции документа), воспользовавшись меню *Tools-Worksheet Options*.

3.1.2 Ввод и редактирование текста

Хотя Mathcad и является математическим редактором, он обладает развитыми средствами по оформлению текста. Текст в Mathcad используется в качестве комментариев к производимым вычислениям, или как элемент оформления документов для создания качественных отчетов. В Mathcad текст можно вводить, редактировать и импортировать.

Для ввода текста необходимо указать программе, что создается не формульный, а текстовый регион: для этого перед вводом первого символа текста нажмите клавишу <'>. При этом курсор принимает вид вертикальной линии красного цвета, который называется *линией ввода текста*.

Для редактирования текста необходимо щелкнуть левой клавишей мыши на область текста, что приведет к выделению текстового региона, в котором размещается данный текст. Далее следует изменить текст, используя курсор и клавиатуру.

Для импортирования текста (например, из Блокнота или MS Word) необходимо:

- находясь в другом приложении, скопировать нужный фрагмент в буфер обмена;
- перейти в окно Mathcad и пометить курсором ввода место вставки;
- ввести текст.

Ввод импортируемого текста возможен двумя способами:

- в текстовом регионе (для этого создается регион ввода и нажимается клавиша <'>, после чего вставляется содержимое буфера обмена нажатием сочетания клавиш <Ctrl>+<V>);

- вставкой импортируемого текста, как объекта OLE (Object Linking and Embedding) нажатием сочетания клавиш <Ctrl>+<V> (при этом для каждого последующего редактирования текста будет вызываться приложение, в котором был создан импортированный фрагмент текста).

Для вставки математических выражений внутри текстовой области необходимо войти внутрь текстового региона и в нужное место текстового региона вставить как обычно математический регион (меню *Insert – Math Region* или сочетание клавиш <Ctrl>+<Shift>+<A>), в котором ввести необходимое математическое выражение. При этом надо помнить, что введенные формулы являются не комментариями, а полноправными

вычислительными элементами документа, переопределяющими введенные ранее (до ввода текстового региона) значения математических переменных.

Для вставки гиперссылок на какое-либо место внутри активного документа, другого документа Mathcad или сайт в Интернет используется команда *Insert-Hyperlink*.

3.1.3 Правка документа

При правке документа используются следующие операции:

- выделение части документа;
- удаление части документа;
- вырезка, копирование, вставка и перемещение части документа;
- выравнивание регионов;
- обновление вида и документа;
- поиск и замена;
- проверка англоязычной орфографии.

Для выделения части документа Mathcad используется стандартная процедура выделения элементов окна принятая в Windows: вне крайнего из них нажимается левая клавиша мыши (определяется курсором начальная точка выделения) и протаскивается ее указатель через все выделяемые регионы. При этом все выделяемые регионы помечаются пунктирными границами. Для снятия выделения достаточно щелкнуть мышью в любой части документа.

Для удаления части документа необходимо сначала выделить эту часть, а затем нажать клавишу или <Ctrl>+.

Текущий регион удаляется нажатием сочетания клавиш <Ctrl>+ или командой *Edit-Delete*.

Для вырезки, копирования выделенных регионов и вставки их из буфера используются меню *Edit*, кнопки правки на панели инструментов *Standard* (Стандартная) или сочетания горячих клавиш:

- <Ctrl> + <X> (вырезать),
- <Ctrl>+<C> (копировать в буфер обмена),
- <Ctrl>+<V> (вставить из буфера обмена).

Для перемещения выделенных регионов в другое место можно использовать прием перетаскивания: если поместить курсор на выделенный регион, он приобретает форму «руки». Далее зафиксировав «руку» нажатой левой клавишей мыши, можно переместить регион в другое место документа.

Для выравнивая регионов документа необходимо воспользоваться командой меню *Format-Align Regions* с опциями *Across* (Горизонтально), *Down* (вертикально).

Для обновления вида документа (при появлении «мусора») необходимо воспользоваться командой обновления документа *View-Refresh*.

Поиск символа, фрагмента или слова документа осуществляется командой *Edit-Find* или сочетанием клавиш <Ctrl>+<F>. Команда имеет окно с кнопками *Find Next* (Искать далее), *Cancel* (Закончить поиск), окно *Find what* (Что искать) и опциями:

- Match whole word only (совпадение слов целиком),
- Match case (учитывать регистр),
- Find in Text Regions (в текстовых областях),
- Find in Math Regions (в математических областях),
- Up (вверх),
- Down (вниз).

Замена символа, фрагмента или слова документа осуществляется командой *Edit-Replace* или сочетанием клавиш <Ctrl>+<H>. Команда имеет окно с кнопками *Find Next* (Искать далее), *Replace* (Заменить), *Replace All* (Заменить все), *Cancel* (закончить поиск) и опциями:

- Match whole word only (совпадение слов целиком),
- Match case (учитывать регистр),
- Find in Text Regions (в текстовых областях),
- Find in Math Regions (в математических областях).

3.1.4 Печать и посылка документа по электронной почте

Готовый документ можно распечатать на активном принтере, воспользовавшись командой *File-Print* (<Ctrl>+<P>). Перед этим можно установить параметры страницы по команде *File-Print Setup* и осуществить предварительный просмотр по команде *File-Print Preview*.

Документ также возможно переслать по электронной почте, воспользовавшись командой *Edit-Send*.

3.2 Вычисления

3.2.1 Числа

Mathcad выполняет вычисления над вещественными и комплексными числами. Числа можно задавать в десятичной, восьмеричной и шестнадцатеричной системах счисления. Признаком восьмеричных чисел является буква «o», шестнадцатеричных чисел – буква «h».

Для обозначения мнимой единицы в комплексных числах используются буквы «i» и «j». Пользователь может выбрать обозначение мнимой единицы по своему усмотрению. Для этого нужно использовать главное меню *Format-Number*. Нельзя использовать для обозначения мнимой единицы буквы «i» и «j» сами по себе. Следует печатать 1i или 1j. Чтобы ввести числа в экспоненциальном представлении, нужно умножить мантиссу на степень десяти.

Пример:

769.07

-22.34578

1.237·10⁴

156o – восьмеричное число

4ABh – шестнадцатеричное число

47.09-0.7i – комплексное число

В вычислениях можно использовать числа, представленных в различных системах счисления. Систему счисления числового результата можно установить командой *Format-Number*.

Пример:

$$156_{\text{o}} + 12 = 122$$

$$14_{\text{h}} + 156_{\text{o}} = 130$$

3.2.2 Переменные

При помощи переменных обозначаются скалярные величины, векторы, матрицы и функции. Имена переменных могут быть любой длины и состоять из латинских, греческих букв, цифр от 0 до 9, символа подчеркивания, символа процента, символа бесконечности (∞). Переменная может быть набрана в любом шрифте, но все символы одного имени должны быть набраны одним шрифтом. Mathcad различает имена, набранные в разных регистрах и в разных шрифтах, например, F, f и курсивное f обозначают различные переменные.

Имя не может начинаться с цифры, символа подчеркивания ($_$), символа процента (%), или символа штриха (\prime); символ бесконечности (∞) может быть только первым символом в имени.

Пример:

A0123 $\alpha\beta$ _120 n% V AB Sum12_k

В именах можно использовать символ (.). Все символы, набранные после нажатия точки (.), представляют нижний индекс. Например, формулу $R_a := 11 + R_b$ следует набрать так: R.a:11+R.b. Заметим, что набранные таким образом переменные не являются элементами массивов.

Некоторые переменные в пакете Mathcad имеют predefined значения. Эти переменные и их значения приведены в таблице 3.1.

Таблица 3.1 – Переменные с predefined значениями

Имена predefined переменных	Значение	Пояснения
π	3.141...	Число π используется с учетом 15 значащих цифр. В символьных вычислениях сохраняется точное значение. Для ввода можно использовать комбинацию <Ctrl>+<P>.
e	2.718...	Основание натуральных логарифмов используется с учетом 15 значащих цифр. В символьных вычислениях сохраняется точное значение.
∞		Бесконечность. В численных расчетах – число 10^{307} . В символьных вычислениях – бесконечность. Для ввода ∞ можно

%	0.01	использовать комбинацию <Ctrl>+<z>. Процент. В выражениях, подобных 10*%, или как масштабный множитель в поле, отводимом для единиц размерности.
TOL	0.001	Допускаемая погрешность для различных приближенных алгоритмов и аппроксимаций.
ORIGIN	0	Начало массива. Определяет индекс первого элемента массива.
PRNCOLWIDTH	8	Ширина столбца, используемая при записи файлов функцией WRITEPRN.
PRNPRECISION	4	Число значащих цифр, используемых при записи файлов функцией WRITEPRN.
FRAME	0	Используется в качестве счетчика при управлении анимациями.

3.2.3 Операции

Вычисления в Mathcad проводятся через вызов оператора или вызов функции. Операторы вводятся нажатием кнопок с их изображением или комбинацией клавиш. Ниже в таблице 3.2 приведены основные операторы и способ их задания.

Таблица 3.2 – Основные операторы и способ их задания

Обозначение	Клавиши	Пояснения Сложение.
X+Y	+	Сложение, если X и Y – скалярные выражения. Сложение элементов массивов, если X и Y – массивы (векторы или матрицы) одинаковой размерности. Если X-массив, а Y-скаляр, то сложение каждого элемента массива X с Y.
X-Y	-	Вычитание. Вычитание, если X и Y – скалярные выражения. Вычитание элементов массивов, если X и Y – массивы (векторы и матрицы) одинаковой размерности. Если X-массив, а Y-скаляр, то вычитание из каждого элемента массива X скаляра Y.
X·Y	*	Умножение. Умножение, если X и Y – скалярные выражения. Скалярное произведение векторов, если X и Y – вектора. Произведение матриц, если X и Y – матрицы соответствующих размерностей. Если X-массив, а Y-скаляр, то умножение каждого элемента массива X на скаляр Y.

			Деление.
$\frac{X}{\bar{Y}}$	/		Если X-скаляр и Y-ненулевой скаляр, то деление X и Y. Если X-массив и Y-ненулевой скаляр, то деление каждого элемента массива X и Y.
			Возведение в степень.
Z^w	^		Возводит Z в степень w, Z и w могут быть вещественными и комплексными.
			Квадратный корень.
\sqrt{Z}	\		При неотрицательном Z возвращает положительное значение квадратного корня; для отрицательных и комплексных – главное значение \sqrt{Z} .
			Корень n-й степени
$\sqrt[n]{Z}$	Ctrl+\		Для натуральных n возвращает значение корня $\sqrt[n]{Z}$
			Абсолютное значение
$ Z $			Для комплексного Z вычисляется $\sqrt{Re(Z)^2 + Im(Z)^2}$
			Факториал
$n!$!		Для целого $n \geq 0$ вычисляется $n \cdot (n-1) \cdot \dots \cdot 2 \cdot 1$
			Производные
$\frac{d}{dx} f(x)$?		Операторы применяются для вычисления производной от функции $f(x)$ в точке. В этом случае нужно задать значение аргумента и, кроме того, все переменные в $f(x)$, должны быть определены. Значение n должно быть целым от 0 до 5.
$\frac{d^n}{dx^n} f(x)$	Ctrl+?		Эти операторы применяются и для символьного дифференцирования, здесь n может быть целым положительным.
			Определенный интеграл
$\int_a^b f(x) dx$	&		Численное интегрирование, вычисляется значение определенного интеграла. Пределы интегрирования a и b и все переменные в функции $f(x)$ должны быть определены.

Mathcad допускает использование следующих операций сравнения. Результаты представлены в таблице 3.3.

Таблица 3.3 – Операции сравнения

Обозначение	Клавиши	Пояснения
$x < y$	<	Результат сравнения равен 1, если $x < y$, в других случаях – 0.
$x > y$	>	Результат сравнения равен 1, если $x > y$, в других случаях – 0.

$x \leq y$	Ctrl+(Результат сравнения равен 1, если $x \leq y$, в других случаях – 0.
$x \geq y$	Ctrl+)	Результат сравнения равен 1, если $x \geq y$, в других случаях – 0.
$z \neq w$	Ctrl+#	Результат сравнения равен 1, если $z \neq w$, в других случаях – 0.
$z = w$	Ctrl+=	Результат сравнения равен 1, если $z = w$, в других случаях – 0.

Для операций больше, меньше, больше или равно и меньше или равно операндами могут быть только действительные числа. В случае операций равно и неравно операндами могут быть вещественные, комплексные числа и массивы. Результат операции сравнения равен 1, если операнды таковы, что для них выполняется операция сравнения, и равен 0 в противном случае.

3.2.3 Операторы присваивания

Для вычисления значений переменной применяются операторы присваивания, имеющие вид:

<имя переменной> <операция присваивания> <выражение>.

Mathcad имеет в своем распоряжении две операции присваивания: ($:=$) и (\equiv). Операция присваивания ($:=$) появляется, если ввести символ двоеточие (:), а вторая после ввода символа «тильда» (~). Кроме того, что для ввода операций присваивания можно применять и пиктограммы из математических панелей инструментов.

Итак, Mathcad имеет два вида оператора присваивания. Разница между ними состоит в следующем. Mathcad все вычисления выполняет за два просмотра. На каждом просмотре документ просматривается слева направо и сверху вниз. Во время первого просмотра Mathcad выполняет вычисления, заданные операторам присваивания (\equiv), а при втором просмотре – вычисления, определенные операторами со знаком ($:=$). Таким образом, вычисления, заданные оператором (\equiv), являются глобальными по отношению к остальным. Это значит, что значения переменных, вычисленных таким оператором, имеют силу на всем документе независимо от его расположения. А значения переменных, вычисленных оператором ($:=$), распространяются только влево и вниз по документу.

Пример:

$$x := 3.175$$

$$z := 8.512 + 5.26i$$

$$J := \int_0^{10} x^2 \cdot \cos(x) dx$$

$$R \equiv \sum_{n=1}^{12} \frac{1}{3 \cdot n + 4}$$

$$A := \begin{pmatrix} 1.1 & 2.2 & 3.3 \\ 4.1 & 5.2 & 6.3 \\ 7.1 & 8.2 & 9.3 \end{pmatrix}$$

3.2.4 Функции

Встроенные функции Mathcad – это основной набор функций, который поставляется вместе с Mathcad. Просмотреть список всех встроенных функций Mathcad можно командой *Insert-Insert Function* или использовать сочетание клавиш <Ctrl+E>. Список всех встроенных функций представлен на рисунке 3.5.

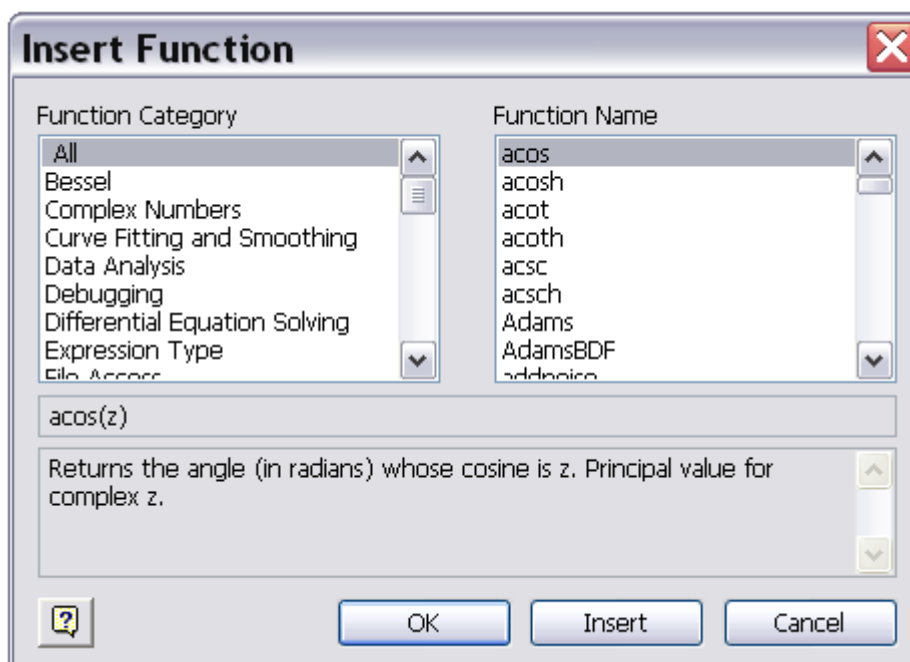


Рис. 3.5 – Список всех встроенных функций

В диалогом окне этой команды выводится имя функции с аргументами, здесь же дано краткое описание выбранной функции. Имена встроенных функций могут вводиться обычным набором на клавиатуре. Имена встроенных функций не чувствительны к шрифту, но чувствительны к регистру – их следует печатать в точности, как они приведены. Поэтому рекомендуется для ввода функций использовать «мастер функций» из меню *Insert-Insert Function*. Набор встроенных функций включает

тригонометрические, гиперболические, логарифмические, экспоненциальную и другие функции, которые записываются в привычном виде. Ниже приведены функции, которые служат для извлечения части своего аргумента, которые приведены в таблице 3.4.

Таблица 3.4 – Функции, которые служат для извлечения части своего аргумента

Функция	Назначения
$\text{Re}(z)$	Вещественная часть комплексного числа z .
$\text{Im}(z)$	Мнимая часть комплексного числа z .
$\text{arg}(z)$	Аргумент комплексного числа z .
$\text{floor}(x)$	Наибольшее целое число $\leq x$ (x – вещественное число).
$\text{ceil}(x)$	Наименьшее целое число $\geq x$ (x – вещественное число).
$\text{mod}(x,y)$	Остаток от деления x на y . Аргументы – вещественные числа.
$\text{angle}(x,y)$	Угол в радианах между положительной полуосью x и вектором (x,y) в плоскости x - y . Аргументы – вещественные числа, результат между 0 до 2π .

Встроенные функции можно предопределить.

3.2.5 Функции пользователя

Определение функции имеет вид:

*<имя функции> (<список аргументов>) <оператор присваивания>
<выражение>*

Аргументами функции могут быть скаляры, векторы, матрицы и функции.

Пример:

Определение

$$f(x,y) := \sqrt{x^2 + y^2}$$

$$g(x,y) := x \cdot y$$

Применение

$$f(4,3) = 5$$

$$g(2.25, 4.11 \cdot e^{-5}) = 13.887$$

$$g(2 - 8i, \sqrt{8 - 4i}) = 0.324 - 24.66i$$

$$g\left[\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \begin{pmatrix} 0 & 1 - 5i \\ 5 - 4i & 2 \end{pmatrix}\right] = \begin{pmatrix} 10 - 8i & 5 - 5i \\ 20 - 16i & 11 - 15i \end{pmatrix}$$

$$I(f, a, b) := \int_a^b f(x) dx$$

$$F(x) := \frac{1}{\sqrt{1 + x + x^2 + x^3}}$$

$$I(F, 3, 5) = 0.224$$

3.2.6 Простейшие вычисления

Для получения численного результата необходимо набрать на экране формулу справа от нее знак (=) или выполнить команду меню *Math-Calculate*. Mathcad вычислит результат и напечатает его справа от знака равенства.

Пример:

$$\sqrt[9]{456.3} = 1.975$$

$$(2.01 + 5.1i)^{-1} = 0.067 - 0.17i$$

$$4^7 - \frac{8774.1}{(2 \cdot \pi - \sqrt{2 - \sin(0.5)^2})} = 1.461 \times 10^4$$

$$\int_0^{\pi} \frac{\sin(x + 0.954 \cdot \pi)}{\cos(x - \pi)} dx = -0.5$$

$$\sum_{n=1}^{12} n! = 5.23 \times 10^8$$

$$\prod_{x=0}^2 (\sin(x) + \cos(x)) = 0.681$$

Лабораторная работа № 1 «Начальные сведения о Mathcad»

Цель работы: познакомиться с программным продуктом Mathcad: интерфейсом, произвести основные вычисления.

Задание на лабораторную работу:

1. Изучить материал, представленный в разделах 2 (пункт 2.1), 3 (3.1, 3.2) соответственно;
2. Произвести вычисления, представленных в примерах соответствующих разделов;
3. Выполнить вычисления по следующим формулам:

$$1. \rightarrow \sqrt[4]{\frac{12,34+3,47i}{|\sin(0,365)-\cos(0,5)|}} + \ln(14,85)$$

$$2. \rightarrow \sum_{k=1}^{12} k \cdot \cos\left(k \cdot \frac{\pi}{6}\right)$$

$$3. \rightarrow \sum_{k=0}^{11} \frac{\sin\left(\frac{7+k}{4+11}\right)}{k! \cdot (5+k)!}$$

$$4. \rightarrow \prod_{n=1}^6 n^3 + \sin\left(n \cdot \frac{\pi}{1,14}\right)$$

$$5. \rightarrow \prod_{n=1}^{12} \left(1 - \frac{0,72}{\sqrt{n}}\right) \cdot e^{\frac{0,72}{\sqrt{n}}}$$

$$6. \rightarrow \sum_{n=1}^{20} \frac{\prod_{n=1}^{20} \ln(n+1)}{\ln(n+1+0,3)}$$

$$7. \rightarrow \int_{2,45}^{9,68} \frac{1+2,43 \cdot x+4,71 \cdot x^2}{\sqrt[5]{1+x}} dx$$

$$8. \rightarrow y = \frac{x \cdot (1+G)}{(1+G) - \frac{23,76}{x^3}}, \text{ где } G = \sum_{n=1}^3 \frac{u^k}{(2k-1) \cdot (2k+1)}, u = \frac{\pi}{5,2}$$

$$9. \rightarrow \sin^{-1} x \frac{2,34^x}{\cos^{-1} 2}$$

$$10. \rightarrow \iint_{-10}^{10} \sin(x) \cdot \cos(y) dx dy$$

3. Введите формулу для вычисления, состоящую из тригонометрической, степенной функции и получите числовое решение.

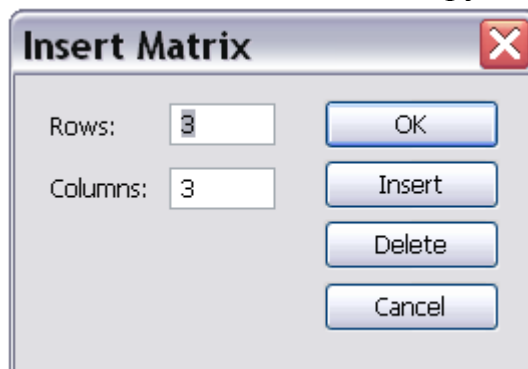
4. Составьте отчет по лабораторной работе №1 (см. Приложение А – Пример оформления отчета). Сделайте соответствующие выводы.

3.3 Векторы и матрицы

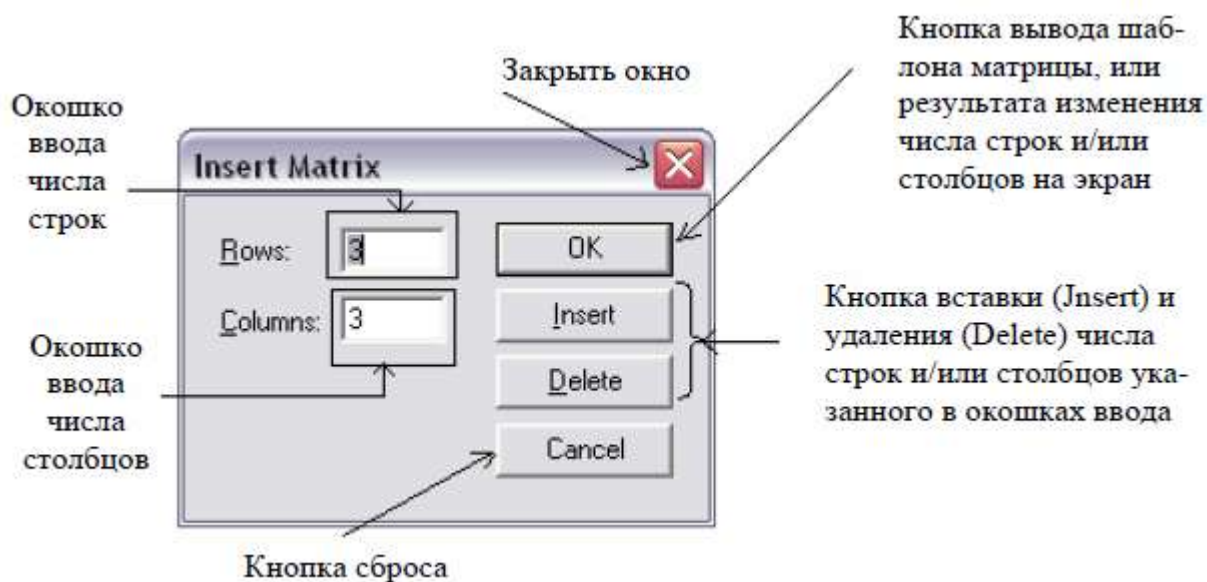
Mathcad выполняет операции над массивами: векторами и матрицами. Создать вектор или матрицу можно тремя способами:

1. Ввод массивов с клавиатуры. Для этого необходимо:

- вызвать диалоговое окно *Insert-Matrix* (рис. 3.6а, б): это можно сделать либо командой *Insert-Matrix* либо клавишами <Ctrl>+<M> либо с помощью кнопки на панели математических инструментов;



а)



б)

Рис. 3.6 а,б – Диалоговое окно для ввода массива с клавиатуры
- задать размерность в диалоговом окне: количество строк (rows), столбцов (columns);

- в появившемся макете заполнить пустые позиции (рис. 3.7) числами или формулами, при этом для перехода к другой позиции можно использовать клавишу *Tab* или клавиши со стрелками.

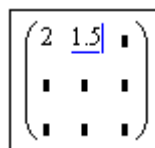


Рис. 3.7 – Пример заполнения массива элементами

Таким способом можно ввести массив, общее число элементов которого не больше 100.

Это же окно используется и для изменения размеров ранее созданного массива: можно вставлять или удалять строки и столбы правее и ниже отмеченного курсором элемента массива. Для изменения размеров массива необходимо:

- отметить курсором элемент массива;
- в полях диалогового окна *Insert-Matrix* нужно указать количество вставляемых (удаляемых) столбцов и строк;
- нажать на кнопку *Insert* или *Delete*.

2. Вектор или матрицу, размеры которых ограничивается лишь доступной памятью компьютера, можно создать, непосредственно вычисляя элементы матрицы по некоторой явной формуле;

3. Для создания массива достаточно также ввести его элементы из файла. При этом нет необходимости делать предварительное описание массива.

Для обращения к элементам массива используются нижние индексы. Чтобы напечатать нижний индекс используют кнопку X_n на панели математических инструментов или клавишу «[». Чтобы ввести оператор верхнего индекса, нужно щелкнуть по кнопке $M\langle \rangle$ или нажать $\langle \text{Ctrl} \rangle + \langle 6 \rangle$ и поместить в поле целое число.

Для присвоения значений элементами массивов применяется дискретный аргумент, т.е. ряд значений, отделяемых одинаковыми шагами. Дискретный аргумент определяется как:

<переменная>:=<первое значение>,<второе значение>..<последнее значение>

Здесь многоточие (..) появляется, если нажать клавишу точка с запятой [;]. Если переменная принимает целочисленное значение с шагом единица, то ***<второе значение>*** можно не писать.

Пример:

1.

$$n := 0..5$$

$$x_n := \cos(10 \cdot n) + n^2$$

$x_n =$

1
0.161
4.408
9.154
15.333
25.965

2.

$$i := 0..3 \quad j := 0..3$$

$$A_{(i,j)} := \frac{i \cdot (j + 7)}{(i + j + 1)}$$

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 3.5 & 2.667 & 2.25 & 2 \\ 4.667 & 4 & 3.6 & 3.333 \\ 5.25 & 4.8 & 4.5 & 4.286 \end{pmatrix}$$

Обращение к элементам матрицы

$$A_{0,0} = 0 \quad A_{1,0} = 3.5$$

$$A_{1,1} = 2.667 \quad A_{1,2} = 2.25$$

Обращение к столбцам матрицы

$$A^{(1)} = \begin{pmatrix} 0 \\ 2.667 \\ 4 \\ 4.8 \end{pmatrix} \quad A^{(2)} = \begin{pmatrix} 0 \\ 2.25 \\ 3.6 \\ 4.5 \end{pmatrix}$$

Обращение к строкам матрицы

$$(A^T)^{(1)} = \begin{pmatrix} 3.5 \\ 2.667 \\ 2.25 \\ 2 \end{pmatrix} \quad (A^T)^{(2)} = \begin{pmatrix} 4.667 \\ 4 \\ 3.6 \\ 3.333 \end{pmatrix}$$

Нижние границы индексов элементов векторов и матриц задает переменная **ORIGIN**. По умолчанию ее значение равно нулю. Другие начальные значения, в том числе и отрицательные, можно установить, заменив значение **ORIGIN** через меню *Math-Options-Built In Variable* или при помощи одного глобального оператора присваивания. Значение переменной **ORIGIN**, установленное этими способами применяется ко всем массивам. Невозможно сделать для разных массивов разные начальные значения индексов.

Возникающие в результате вычислений массивы могут оказаться громоздкими для отображения на экране. Поэтому Mathcad массивы, имеющие большое количество элементов, отображает в виде таблицы с полосами прокрутки. Увидеть элемент можно, используя полосы прокрутки или увеличив границы таблицы с помощью мыши.

Пример:

ORIGIN := 1

$$M := \begin{pmatrix} 1.23 & 2.23 & 3.33 \\ 5.12 & 6.12 & 7.12 \\ 1.23 + 5.12i & 2.23 + 6.12i & 3.33 + 7.12i \end{pmatrix}$$

$$M_{1,1} = 1.23 \quad M_{1,2} = 2.23 \quad M_{1,3} = 3.33$$

$$M_{2,1} = 5.12 \quad M_{2,2} = 6.12 \quad M_{2,3} = 7.12$$

$$M_{3,1} = 1.23 + 5.12i \quad M_{3,3} = 3.33 + 7.12i$$

$$M_{3,2} = 2.23 + 6.12i$$

$$M^{\langle 1 \rangle} = \begin{pmatrix} 1.23 \\ 5.12 \\ 1.23 + 5.12i \end{pmatrix} \quad (M^T)^{\langle 1 \rangle} = \begin{pmatrix} 1.23 \\ 2.23 \\ 3.33 \end{pmatrix}$$

$$M^{\langle 2 \rangle} = \begin{pmatrix} 2.23 \\ 6.12 \\ 2.23 + 6.12i \end{pmatrix} \quad (M^T)^{\langle 2 \rangle} = \begin{pmatrix} 5.12 \\ 6.12 \\ 7.12 \end{pmatrix}$$

$$M^{\langle 3 \rangle} = \begin{pmatrix} 3.33 \\ 7.12 \\ 3.33 + 7.12i \end{pmatrix} \quad (M^T)^{\langle 3 \rangle} = \begin{pmatrix} 1.23 + 5.12i \\ 2.23 + 6.12i \\ 3.33 + 7.12i \end{pmatrix}$$

3.3.1 Векторные и матричные операции

Некоторые из операторов Mathcad имеют особые значения в применении к векторам и матрицам. Например, символ умножения « \cdot » при применении к векторам означает скалярное умножение и умножение матриц – когда применяется к матрицам.

Векторные и матричные операторы доступны из палитры символов или математической панели. Если результатом операции является вектор, то это обязательно вектор-столбец, а не вектор-строка.

Выше, при рассмотрении операторов, обращалось внимание на их особенности при работе с массивами. В следующей таблице 3.5 приведены операторы, имеющие место для массивов и дискретных переменных.

Таблица 3.5 – Операторы, имеющие место для массивов и дискретных переменных

Обозначение	Клавиши	Пояснения
$X \times Y$	Ctrl+*	Векторное произведение трехмерных векторов X и Y
A^n	^	Степень матрицы Для квадратной матрицы A и целого

		положительного n вычисляется n -я степень обратной матрицы A .
$ v $		Длина вектора Для вектора v с вещественными элементами вычисляется длина вектора $\sqrt{v \cdot v}$. Для вектора с комплексными элементами вычисляется $\sqrt{v \cdot \bar{v}}$.
$ A $		Определитель матрицы Для квадратной матрицы A вычисляется определитель.
$\sum_a X$	\$	Суммирование по переменной Для любого выражения X вычисляется сумма по дискретным значениям переменной a .
$\sum v$	Ctrl+\$	Суммирование элементов одномерного массива Вычисляет сумму элементов массива v .
$\sum_{i=n}^m X$	Ctrl+Shift+#	Суммирование Вычисляет сумму значений X для $i=n, n+1, \dots, m$, где X – любое выражение, n, m – целые.
$\prod_a X$	#	Произведение Для любого выражения X вычисляется произведение по дискретным значениям переменной a .
$\prod_{i=n}^m X$	Ctrl+Shift+#	Произведение Вычисляет произведение значений X для $i=n, n+1, \dots, m$, где X – любое выражение, n, m – целые.

Пример:

Транспонирование матрицы

$$\begin{pmatrix} 2.01 & 1.2 & 3.4 & 7.01 & -52 \\ 41 & -2.3 & 1.25 & 4.28 & 7.2 \\ 7 & -8.1 & -4.2 & 12 & 14.2 & 14.8 \\ 0.67 & 9.7 & -5.8 & 4.13 & 0.2 \\ 0 & 4 & 1 & 2 & 3 \end{pmatrix}^T = \begin{pmatrix} 2.01 & 41 & 7 & 0.67 & 0 \\ 1.2 & -2.3 & -12.3 & 9.7 & 4 \\ 3.4 & 1.25 & 12 & -5.8 & 1 \\ 7.01 & 4.28 & 14.2 & 4.13 & 2 \\ -52 & 7.2 & 14.8 & 0.2 & 3 \end{pmatrix}$$

Определитель матрицы и сумма элементов первого столбца матрицы

$$R := \begin{pmatrix} -7 & 9 & -9.7 \\ 4 & 2 & 0 \\ 19 & 67 & 4.25 \end{pmatrix}$$

$$|R| = -2.443 \times 10^3 \quad \sum R^{(1)} = 78$$

Векторные и матричные функции представлены в таблице 3.6.

Таблица 3.6 – Векторные и матричные функции

Имя функции	Результат
rows(A)	Число строк в массиве.
cols(A)	Число столбцов в массиве.
length(v)	Число элементов в векторе.
last(v)	Индекс последнего элемента вектора.
max(A)	Максимальный элемент массива.
min(A)	Минимальный элемент массива.
identity(n)	Единичная матрица размера $n \times n$.
rank(A)	Ранг вещественной матрицы.
augment(A,B)	Массив, сформированный расположением A над B бок о бок. Массивы A и B должны иметь одинаковое количество столбцов.
stak(A,B)	Массив, сформированный расположением A над B. Массивы A и B должны иметь одинаковое количество столбцов.
submatrix(A, ir, jr, ic, jc)	Подматрица, состоящая из всех элементов, содержащихся в строках $ir \leq jr$ и $ic \leq jc$ столбцах.
csort(A,n)	Переставляет строки матрицы A по возрастанию элементов n-го столбца.
rsort(A,n)	Переставляет столбцы матрицы A по возрастанию элементов n-ой строки.
reverse(v) reverse(A)	Обращает порядок элементов вектора или матрицы.

Пример:

Переставить столбцы матрицы $D = \begin{bmatrix} 1 & 5 & 3 \\ -1 & -3 & 8 \\ 32 & 8 & 7 \end{bmatrix}$ по элементам первой строки

$$D := \begin{pmatrix} 1 & 5 & 3 \\ -1 & -3 & 8 \\ 32 & 8 & 7 \end{pmatrix}$$

$$rsort(D,0) = \begin{pmatrix} 1 & 3 & 5 \\ -1 & 8 & -3 \\ 32 & 7 & 8 \end{pmatrix} \quad csort(D,1) = \begin{pmatrix} -1 & -3 & 8 \\ 1 & 5 & 3 \\ 32 & 8 & 7 \end{pmatrix}$$

$$\max(D) = 32$$

$$\min(D) = -3$$

В профессиональной версии Mathcad включен ряд дополнительных матричных функций. Они представлены в таблице 3.7.

Таблица 3.7 – Дополнительные матричные функции

Имя функции	Результат
eigenvals (M)	Возвращает вектор, содержащий собственные значения матрицы M
eigenvec (M,Z)	Для указанной матрицы M и заданного собственного значения Z возвращает принадлежащий этому собственному значению вектор
eigenvecs (M)	Возвращает матрицу, столбцами которой являются собственные векторы матрицы M (порядок расположения собственных векторов соответствует порядку собственных значений, возвращаемых функцией eigenvals)
genvals (M,N)	Возвращает вектор обобщенных собственных значений v, соответствующий решению уравнения $M \cdot x = v_i \cdot N \cdot x$ (матрицы M и N должны быть вещественными)
+ lu (M)	Выполняет треугольное разложение матрицы M: $P \cdot M = L \cdot U$, L и U – соответственно нижняя и верхняя треугольные матрицы. Все четыре матрицы квадратные, одного порядка
+ qr (A)	Дает разложение матрицы A, $A = Q \cdot R$, где Q – ортогональная матрица и R – верхняя треугольная матрица
+ svd (A)	Дает сингулярное разложение матрицы A размером n·m: $A = U \cdot S \cdot V^T$ где U – ортогональные матрицы размером m·m и n·n соответственно, S – диагональная матрица, на диагонали которой расположены сингулярные числа матрицы A
+ svds (A)	Возвращает вектор, содержащий сингулярные числа матрицы A размером m·n, где $m \geq n$
Egeninv (A)	Возвращает матрицу левую обратную к матрице A. $L \cdot A = E$, где E – единичная матрица размером n·n, L – прямоугольная матрица размером n·m, A – прямоугольная матрица размером m·n

3.3.2 Добавление и удаление столбцов и строк

Пусть, например, нужно удалить третий столбец матрицы A, в результате чего должна образоваться матрица размером 3×3. Для этого необходимо установить синий указатель на третьем элементе первой строки матрица A так, чтобы угол указателя был справа и вызвать окно «Матрицы» (<Ctrl>+<M>), далее указать число удаляемых строк (в нашем случае равно 0) и столбцов (равно 1). После этого необходимо нажать клавишу *Delete* окна «Матрицы». Должно получиться так, как показано на рисунке 3.7.

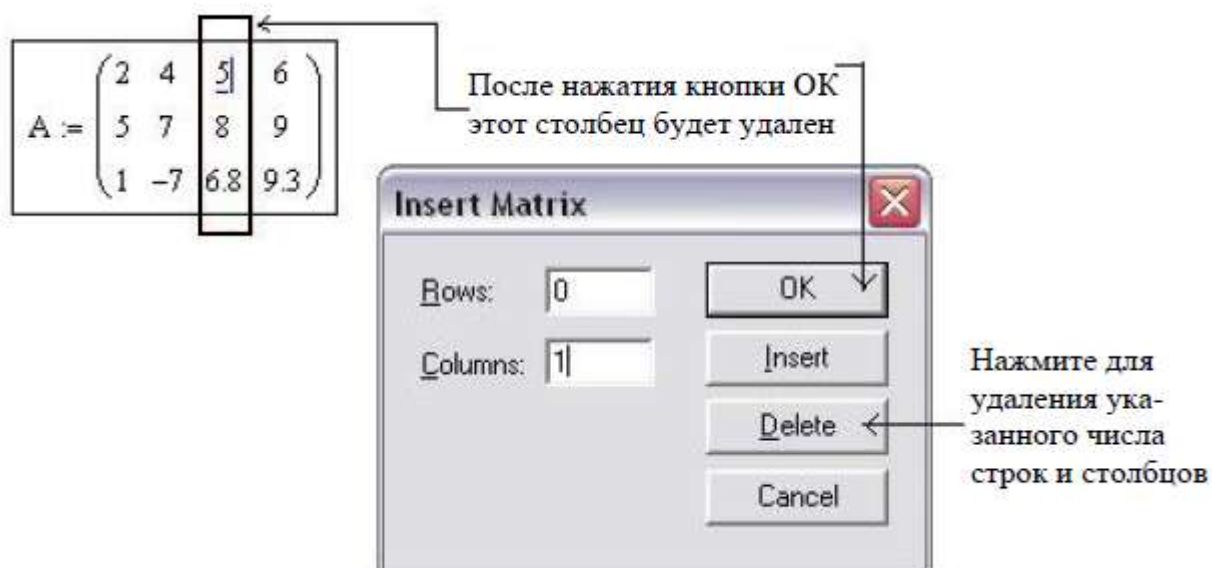


Рис. 3.7 – Пример удаления строк и столбцов матрицы

Для добавления столбцов и строк матрицы необходимо воспользоваться кнопкой *Insert*, показанной на рисунке 3.8.

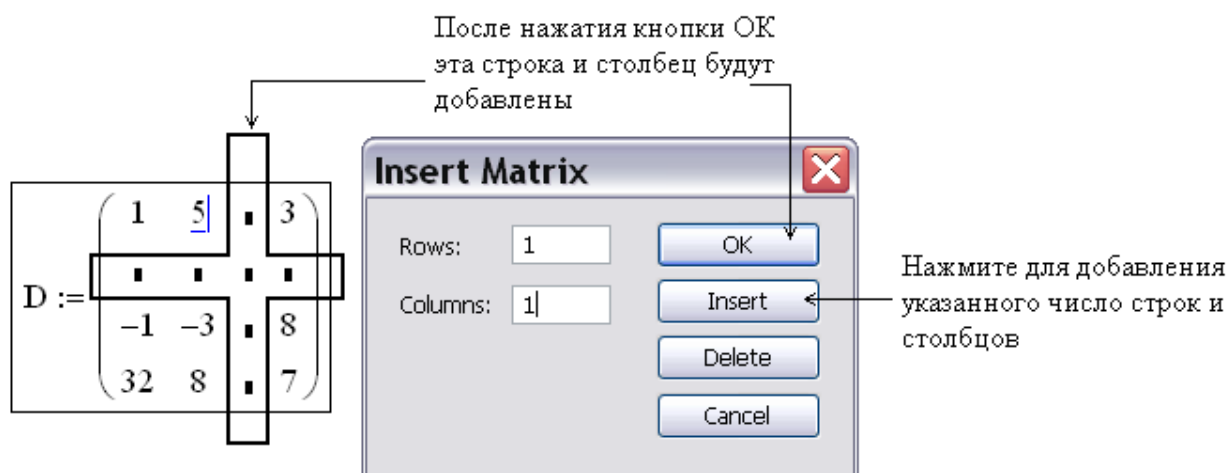


Рис. 3.8 – Пример добавления строк и столбцов матрицы

3.3.3 Операции над матрицами в аналитической (символьной) форме

Системы компьютерной алгебры снабжаются специальным процессором для выполнения аналитических (символьных) вычислений. Его основой является ядро, хранящее всю совокупность формул и формульных преобразований, с помощью которых производятся аналитические вычисления. Чем больше этих формул в ядре, тем надежней работа символьного процессора и тем вероятнее, что поставленная задача будет решена, разумеется, если такое решение существует в принципе (что бывает далеко не всегда).

Ядро символьного процессора Mathcad – несколько упрощенный вариант ядра известной системы символьной математики Maple V фирмы Waterloo Maple Software, у которой MathSoft (разработчик Mathcad)

приобрела лицензию на его применение, благодаря чему Mathcad стал системой символьной математики.

Введение в систему Mathcad символьных вычислений придает ей качественно новые возможности. Символьные вычисления выполняются, в конечном счете, столь же просто для пользователя, как, скажем, вычисление квадрата x .


Операции, относящиеся к работе символьного процессора, содержатся в подменю позиции *Symbolic* (Символика) главного меню.

Чтобы символьные операции выполнялись, процессору необходимо указать, над каким выражением эти операции должны производиться, т.е. надо выделить выражение. Для ряда операций следует не только указать выражение, к которому они относятся, но и наметить переменную, относительно которой выполняется та или иная символьная операция.

Само выражение в таком случае не выделяется, ведь и так ясно, что если маркер ввода выделяет переменную какого-либо выражения, то это выражение уже отмечено наличием в нем выделяемой переменной.

Символьные операции разбиты на пять характерных разделов. Это операции с выражениями, операции с переменными, операции с матрицами, операции преобразования, стиль эволюции. Первыми идут наиболее часто используемые операции. Они могут выполняться с выражениями, содержащими комплексные числа или имеющими решения в комплексном виде.

Символьный процессор системы Mathcad обеспечивает проведение в символьном виде трех наиболее распространенных матричных операций: транспонирования и обращения матриц, а также вычисления их детерминанта.

При символьных вычислениях, прежде всего, следует вызвать панель символьных вычислений нажатием кнопки  на математической панели.

После этого появится панель символьных вычислений, показанная на рисунке 3.9.

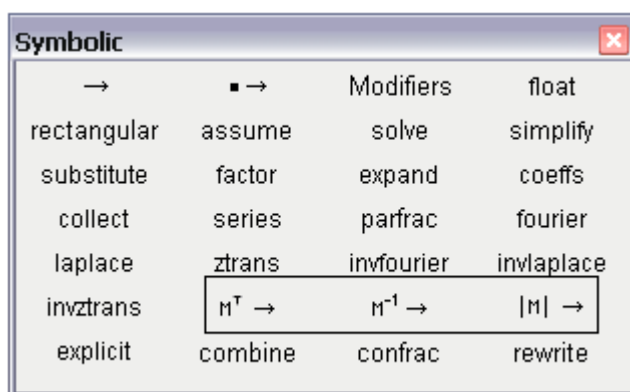


Рис. 3.8 – Панель символьных вычислений

Для символьных операций над матрицами нам понадобится только выделенная часть этого окна, с помощью кнопок которой и производятся транспонирование, обращение матрицы и нахождение ее определителя.

Пример:

Здесь t – некоторый параметр, от которого зависят коэффициенты уравнения.

Для решения системы линейных уравнений формируется квадратная $n \times n$ матрица из коэффициентов левой части и n -мерный вектор из значений $b_i(t)$, $i=1, \dots, n$, входящих в правую часть. Эти матрицы будут иметь вид:

$$A(t) = \begin{bmatrix} a_{11}(t) & a_{12}(t) & \dots & a_{1n}(t) \\ a_{21}(t) & a_{22}(t) & \dots & a_{2n}(t) \\ \dots & \dots & \dots & \dots \\ a_{n1}(t) & a_{n2}(t) & \dots & a_{nn}(t) \end{bmatrix}, B(t) = \begin{bmatrix} b_1(t) \\ b_2(t) \\ \dots \\ b_n(t) \end{bmatrix}$$

Согласно правилам матричной алгебры решение системы линейных уравнений для случая, когда правая часть – не нулевой вектор, возможно тогда и только тогда, когда определитель матрицы не равен 0 ни для одного значения параметра t из диапазона, на котором он задан. В этом случае решение будет n -мерным вектором, зависящим от параметра t :

$$x(t) = [A(t)]^{-1} \cdot B(t).$$

Как видим, для получения решения используются следующие матричные операции:

- обращение квадратной матрицы;
- умножение матрицы на вектор;
- диагностика отличия определителя матрицы от 0 (для установления существования решения).

Пример:

Найдем значения двумерного вектора $x(t)$ при изменении параметра t в диапазоне $0, \dots, 5$ с шагом $0,01$. Пусть каждый элемент матрицы $A(t)$ – экспонента с заданным показателем степени, а каждый элемент вектора $B(t)$ – синусоидальная функция.

Задаем матрицу и вектор коэффициентов.

$$A = \begin{bmatrix} 1 & 1,5 \\ 2 & 4 \end{bmatrix}, B = \begin{bmatrix} 5 \\ 7 \end{bmatrix}$$

Далее проводим векторизацию матрицы A . Каждый элемент ее превратится в экспоненту. Показателем будет соответствующий элемент матрицы A .

$$A(t) = \overrightarrow{e^{A \cdot t}}$$

Термин «векторизация» означает применение одной и той же функции тотально ко всем элементам матрицы. Этот прием часто экономит время при вводе, и при вычислениях за счет параллельных действий над всеми элементами матрицы сразу. Для того чтобы над всеми элементами матрицы выполнялась одна и та же операция, нужно:

- ввести матрицу с постоянными элементами, которые будут использоваться как коэффициенты в формулах для операций с матрицей;
- ввести идентификатор матрицы с указанием ее зависимости от параметра (можно – тот же, что и для матрицы коэффициентов), присвоить ему значение функции, которую мы хотим применить ко всем элементам матрицы;

- результат охватить синим контуром, и нажать на клавиши <Ctrl>+<- (минус)>. Над функцией появится горизонтальная стрелка – символ векторизации.

Для контроля проведем вычисление экспонент непосредственно (должен быть получен тот же результат, что и при векторизации):

$$A1(t) = \begin{bmatrix} e^{1 \cdot t} & e^{1,5 \cdot t} \\ e^{2 \cdot t} & e^{4 \cdot t} \end{bmatrix}$$

Проводим векторизацию вектора В. Каждый его элемент должен превратиться в синус с множителем-элементом вектора В:

$$B(t) = \overrightarrow{\sin(B \cdot t)}$$

Для контроля вводим функцию синуса непосредственно:

$$B1(t) = \begin{bmatrix} \sin(5 \cdot t) \\ \sin(7 \cdot t) \end{bmatrix}$$

Проверяем, действительно ли выполнена векторизация, например, при $t=0,5$.

$$\underline{A} := \begin{pmatrix} 1 & 1,5 \\ 2 & 4 \end{pmatrix} \quad B := \begin{pmatrix} 5 \\ 7 \end{pmatrix}$$

$$A(t) := \overrightarrow{e^{A \cdot t}}$$

$$A1(t) := \begin{pmatrix} e^{1 \cdot t} & e^{1,5 \cdot t} \\ e^{2 \cdot t} & e^{4 \cdot t} \end{pmatrix}$$

$$B(t) := \overrightarrow{\sin(B \cdot t)}$$

$$B1(t) := \begin{pmatrix} \sin(5 \cdot t) \\ \sin(7 \cdot t) \end{pmatrix}$$

$$t := 0.5$$

$$A(t) = \begin{pmatrix} 1.649 & 2.117 \\ 2.718 & 7.389 \end{pmatrix} \quad A1(t) = \begin{pmatrix} 1.649 & 2.117 \\ 2.718 & 7.389 \end{pmatrix}$$

$$B(t) = \begin{pmatrix} 0.598 \\ -0.351 \end{pmatrix} \quad B1(t) = \begin{pmatrix} 0.598 \\ -0.351 \end{pmatrix}$$

Лабораторная работа № 2 «Векторы и матрицы»

Цель работы: получить навыки работы в Mathcad с векторами и матрицами.

Задание на лабораторную работу:

1. Изучить материал, представленный в разделе 3 (пункт 3.3);
2. Произвести вычисления, представленных в примерах соответствующих разделов;
3. Ввести с клавиатуры матрицы A, B и вектор b.

$$A = \begin{bmatrix} 41 & 23 & 11 \\ -17 & 58 & 30 \\ 10 & -23 & 67 \end{bmatrix}, B = \begin{bmatrix} 2 & -1 & 3 \\ 5 & 1 & 2 \\ 7 & -1 & 4 \end{bmatrix}, b = \begin{bmatrix} -1 \\ 2 \\ 3 \end{bmatrix}.$$

Вычислить

$|A|, A^T, B^{-1}, A^3, (2A + B)(A - B^2), A_{1,1}$ и так далее, $B_{1,1}$ и так далее.

Найти максимальный и минимальный элемент матрицы $C = A \cdot B$.

Решить систему линейных алгебраических уравнений $A \cdot x = b$ по формулам Крамера:

$$x_0 = \frac{\Delta_1}{\Delta}, x_1 = \frac{\Delta_2}{\Delta}, x_2 = \frac{\Delta_3}{\Delta},$$

где

$$\Delta = \begin{vmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{vmatrix}, \Delta_1 = \begin{vmatrix} b_0 & a_{01} & a_{02} \\ b_1 & a_{11} & a_{12} \\ b_2 & a_{21} & a_{22} \end{vmatrix},$$

$$\Delta_2 = \begin{vmatrix} a_{00} & b_0 & a_{02} \\ a_{10} & b_1 & a_{12} \\ a_{20} & b_2 & a_{22} \end{vmatrix}, \Delta_3 = \begin{vmatrix} a_{00} & a_{01} & b_0 \\ a_{10} & a_{11} & b_1 \\ a_{20} & a_{21} & b_2 \end{vmatrix}.$$

Решить систему уравнений $A \cdot X = B$ при помощи обратной матрицы.

4. Построить функцию пользователя для вычисления значений характеристического полинома матрицы $A = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix}$, используя

формулу $p(\lambda) = \lambda^3 - p_1\lambda^2 + p_2\lambda - p_3$, где

$$p_1 = \sum_{i=0}^2 a_{ii}, p_2 = \begin{vmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} + \begin{vmatrix} a_{00} & a_{02} \\ a_{10} & a_{22} \end{vmatrix},$$

$$p_3 = \begin{vmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{vmatrix}.$$

Затем вычислить таблицу значений характеристического полинома матрицы B из пункта 1 для $\lambda \in [-4, 4]$ с шагом 0,5.

5. Уравнение окружности, проходящей через три точки $M_1(x_1, y_1), M_2(x_2, y_2), M_3(x_3, y_3)$ определяется уравнением

$$\begin{vmatrix} x^2 + y^2 & x & y & 1 \\ (x_1)^2 + (y_1)^2 & x_1 & y_1 & 1 \\ (x_2)^2 + (y_2)^2 & x_2 & y_2 & 1 \\ (x_3)^2 + (y_3)^2 & x_3 & y_3 & 1 \end{vmatrix} = 0$$

Составить функцию пользователя, которая определяет, лежит ли точка $M(x, y)$ на окружности.

6. Составить примеры самостоятельно таким образом, чтобы они доказывали и подтверждали аксиомы и теоремы теории матриц, приведенных в таблице 3.8.

7. Пояснить на примерах роль переменной ORIGIN.

8. Провести векторизацию. Найти значения двумерного вектора $x(t)$ при изменении параметра t в диапазоне $0, \dots, 2$ с шагом $0,05$. Пусть каждый элемент матрицы $A(t)$ – экспонента с заданным показателем степени, а каждый элемент вектора $B(t)$ – синусоидальная функция.

Задаем матрицу и вектор коэффициентов.

$$A = \begin{bmatrix} 1,3 & 5 \\ 7 & 14 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 7 \end{bmatrix}$$

9. Вычислить следующие функции многочлена:

$$y = 4 \cdot x^2 + 5 \cdot x + 8 \text{ для } x = 1, 2, \dots, 10$$

$$y = 3 \cdot x^4 - 7 \cdot x^3 + 4 \cdot x^2 - 9 \cdot x + 2 \text{ для } x = 0, 0.05, 1, 1.5, \dots, 3$$

10. Вычислить функцию двух переменных:

$$z = 3x^2 + 4y^2 + 8 \text{ для } x = 1, 1.5, 2, \dots, 5; y = 0, 0.5, 1, \dots, 5$$

11. Составьте отчет по лабораторной работе №2 (см. Приложение А – Пример оформления отчета). Сделайте соответствующие выводы.

3.4 Построение графиков

Mathcad позволяет строить графики 7 типов: в прямоугольной и полярных системах координат, графики поверхностей, контурные линии, трехмерные гистограммы, точечные графики и графики векторных полей. Для графиков допустимы следующие операции: создание, удаление, перемещение, изменение размеров графического блока и формата графика.

Панель графиков вызывается нажатием кнопки с изображением графиков на математической панели (рисунок 3.9).

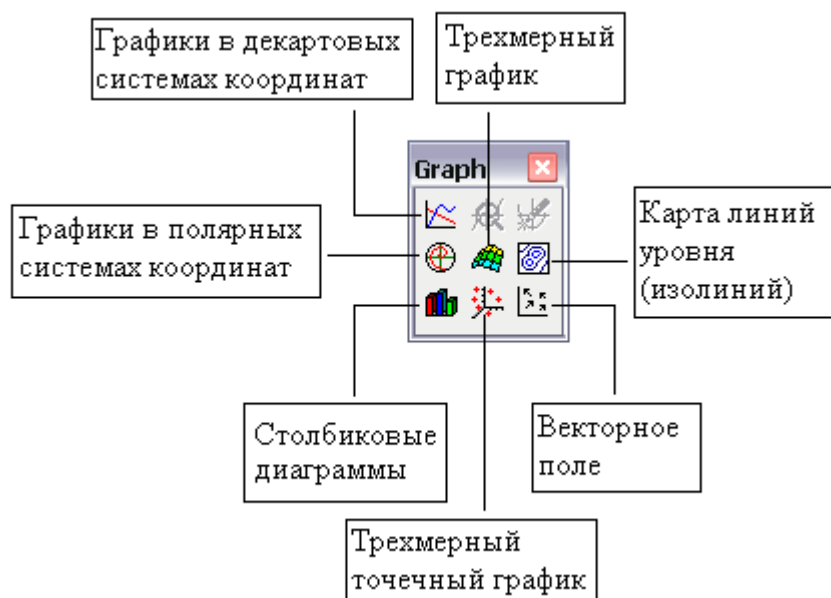


Рис. 3.9 – Панель графиков

На панели графиков расположены девять кнопок с изображением различных типов графиков (название графиков каждой кнопки высвечивается при подводе к ней курсора и ожидании в течение 3-5 секунд): X-Y Plot (<Shift>+<2>) – графики в декартовых координатах, Polar Plot (<Ctrl>+<7>) – графики в полярных координатах, 3D Bar Plot – столбиковые диаграммы, Surface Plot (<Ctrl>+<2>) – трехмерный график, Contour Plot – карта линий уровня (изолиний), Vector Field Plot – векторное поле, 3D Scatter Plot – трехмерный точечный график.

Для удаления, перемещения или формирования графической области необходимо выделить графический блок. Чтобы выделить графический блок, нужно щелкнуть левой кнопкой мыши в области графика.

Для изменения размера графика нужно переместить курсор мыши на границу выделенного блока (при этом он изменится на двунаправленную стрелку) и перемещать мышью, нажав на левую кнопку. Размер блока будет меняться в направлении перемещения курсора. Достигнув необходимого размера, отпустить кнопку мыши.

Перемещать выделенный графический блок можно с помощью мыши или воспользоваться командами меню.

Mathcad позволяет достаточно широко варьировать формат изображения графиков. Можно изменять такие параметры как масштаб по

осям, задавать координатную сетку и значения координат по осям, менять внешний вид графика, цвет и толщину линий, точку зрения и т.д. параметры изменяют через диалоговое окно, содержащее несколько полей. Для вызова диалогового окна, в котором можно форматировать график, следует дважды щелкнуть левой кнопкой мыши на графическом блоке или воспользоваться командой *Format-Graph*.

3.4.1 Прямоугольная система координат на плоскости

Декартова система координат (прямоугольная) хорошо известна. И всё же сформулируем подробнее, каким образом она задаётся на плоскости, и какие величины в результате однозначно определяют положение точки на плоскости. Не будем, однако, слишком углубляться в терминологию, т.к. используемые понятия просты и подробно изучаются в курсе средней школы.

Задать декартову систему координат на плоскости означает зафиксировать, во-первых, точку начала координат, а во-вторых, две перпендикулярные направленные оси (так называемые, оси координат). Причём, эти оси занумерованы. И, конечно, понадобится единичный отрезок, чтобы численно обозначать расстояние между двумя точками.

Таким образом, положение любой точки на плоскости однозначно определено двумя числами: первое число – величина проекции точки на первую ось (взятая с плюсом, если проекция попала на «положительную» часть оси, или с минусом, если на «отрицательную»), а второе – величина проекции на вторую ось. Прямоугольная система координат на плоскости представлена на рисунке 3.10.

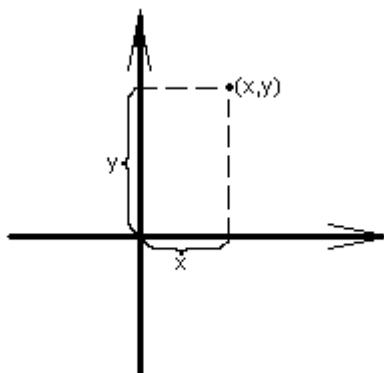


Рисунок 3.10 – Прямоугольная система координат

Стандартным образом декартова система координат обозначается Oxy , оси нумеруются таким образом, что поворот от первой оси ко второй осуществляется против часовой стрелки. Координаты точки – (x, y) .

Для построения графиков в прямоугольной системе координат на плоскости необходимо:

- щелкнуть левой кнопкой мыши в том месте экрана, где нужно построить график;

- создать графический блок, для этого следует выполнить команду меню *Insert-Graph-X/Y Plot* (ввести символ @) или воспользоваться кнопкой из палитры кнопок) или нажать на кнопку с изображением декартовых графиков с панели графиков;

- в появившемся графическом блоке заполнить пустые поля, т.е. задать в среднем поле по горизонтальной оси аргумент, а в среднем поле по вертикальной – функцию. Еще четыре пустых поля можно использовать для указания границ по осям координат.

Пример:

1. Построим график функции $f(x) = x^2 \cdot \cos(x) + \sin(2x) + 5$ на отрезке $[-\pi; \pi]$. Для этого зададим функцию пользователя для вычисления $f(x)$ и область значений переменной x (рис. 3.11).

$$f(x) := x^2 \cdot \cos(x) + \sin(2x) + 5$$

$$x := -\pi, -\pi + 0.1 .. \pi$$

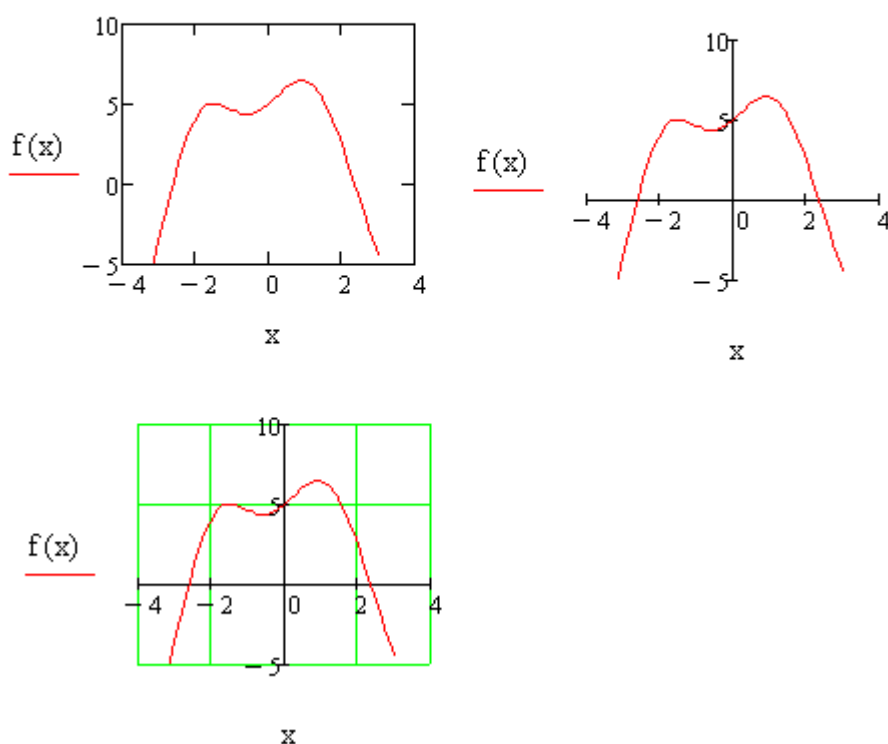


Рисунок 3.11 – Построение графика, с применением различных режимов форматирования

2. Построим графики функций $f(x) = \frac{32}{3+x^2}$ и $g(x) = (5+x) \cdot \sin(x)$, $x \in [-5; 11]$ в одном графическом блоке. В среднем поле вертикальной оси укажем функции, отделив их запятой. Для аргумента укажем границы -5 и 11 и подберем границы для значений функции так, чтобы оба графика рисовались с наименьшим искажением. Пример показан на рисунке 3.12.

$$f(x) := \frac{32}{3 + x^2}$$

$$g(x) := (5 + x) \cdot \sin(x)$$

$$x := -5, -5 + 0.05 .. 11$$

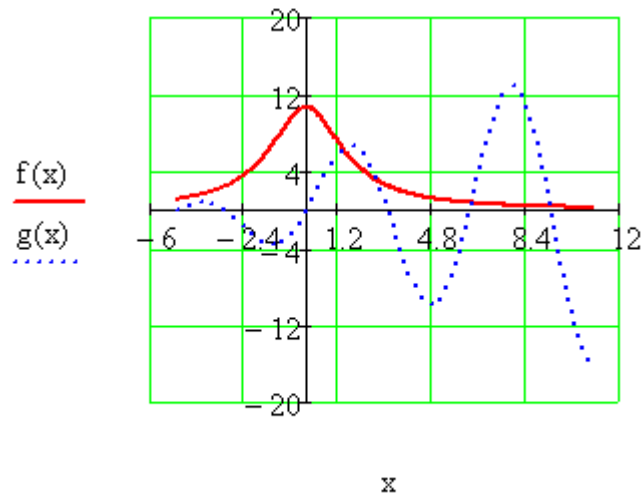


Рисунок 3.12 – Построение двух графиков функций в одном графическом окне

3. Построим графики функций с различными переменными: $f(x) = x^2 + 2 \cdot x + 4$ для $x \in [-10; 8]$ и $g(z) = 3 \cdot \sin(z) + 7$ для $z \in [-7; 10]$, которые изображены на рисунке 3.13.

$$f(x) := x^2 + 2x + 4$$

$$x := -10, -10 + 0.02 .. 8$$

$$g(z) := 3 \sin(z) + 7$$

$$z := -7, -7 + 0.05 .. 10$$

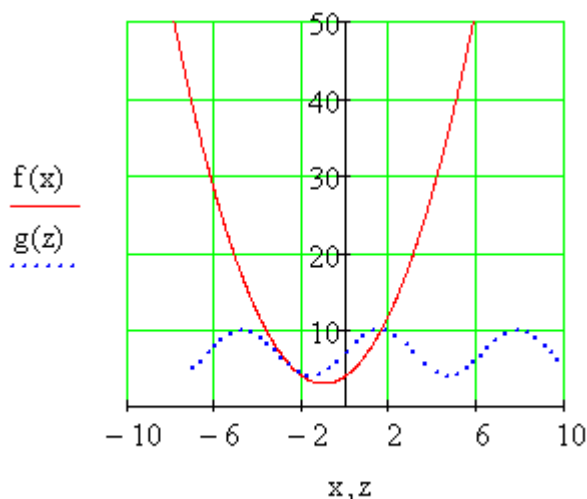
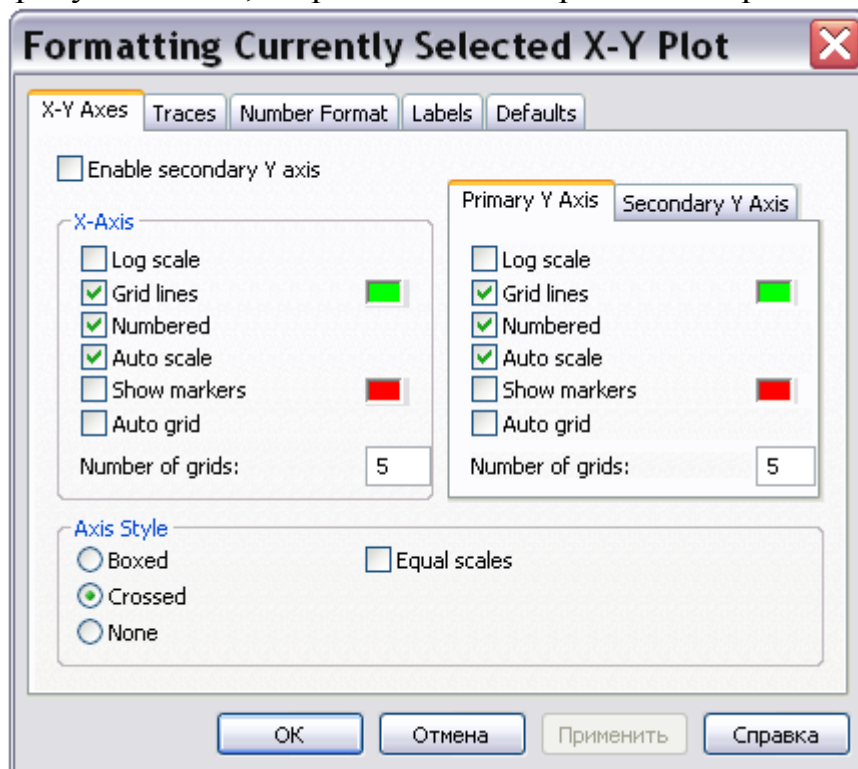
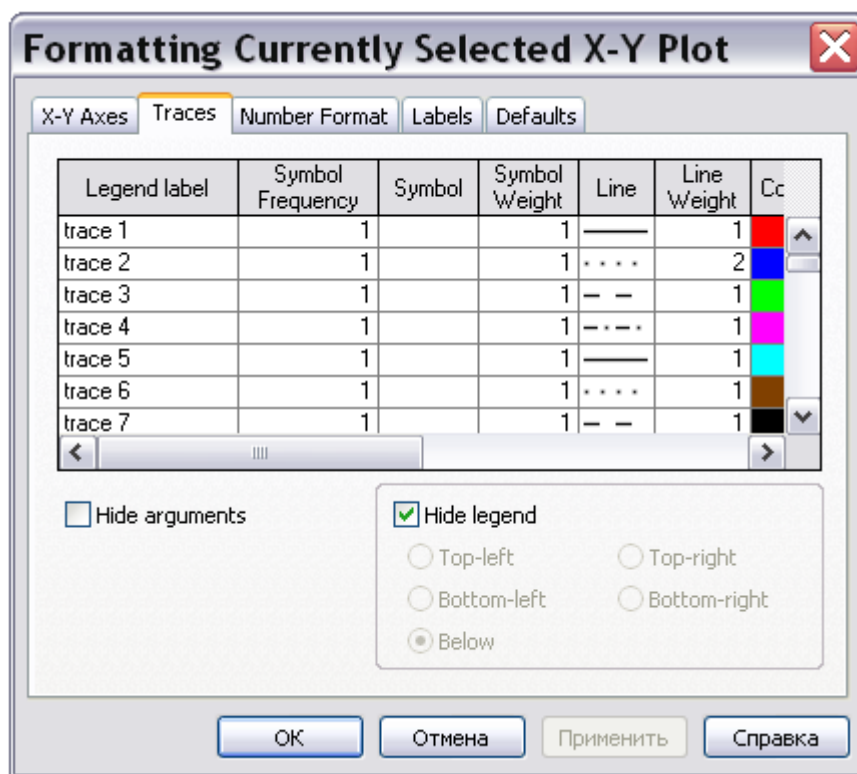


Рисунок 3.13 – Построение двух графиков функций с различными переменными в одном графическом окне

При помещении курсора внутрь графика и щелкнув левой клавишей мыши появится окно, показанное на рисунке 3.14 а, б. Оно состоит из пяти страниц. На рисунке 3.14 а, б представлены первые две страницы.



а)



б)

Рисунок 3.14 – Окна формирования графиков

В левом нижнем углу первой страницы имеются точки Boxed (коробочка), Crossed (оси), None (нет). Нажав на первую точку, введем в график оси координат.

На первой странице имеются строки X-Axes (ось X) и Primary Y-Axes (первая ось Y), а под ними ряд надписей, левая часть которых относится к оси X, а правая – к оси Y: Log Scale (логарифмическая шкала) вводит логарифмический масштаб для соответствующей оси; Grid lines (сетка) – ее нажатие вводит сетку на график; Numbered (оцифровка) – оцифровка сетки; Auto scale (автоматическая оцифровка); Show markers (показ маркеров); Auto grid (автоматическое разбиение сетки).

Наличие надписей Enable secondary Y-Axes (возможность второй оси Y) и Secondary Y-Axes (вторая ось Y) дает возможность формировать графики различного масштаба для различных функций.

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

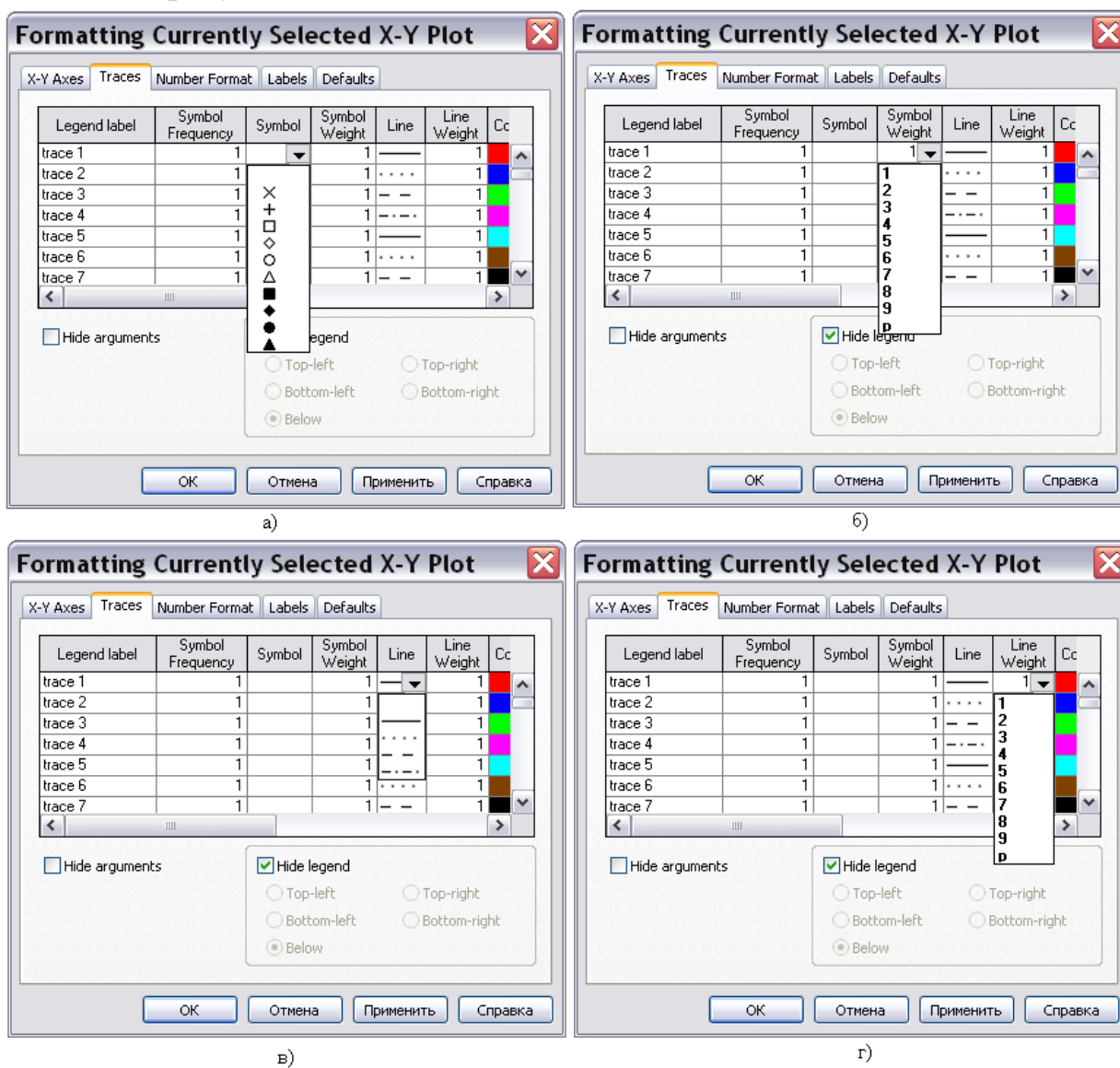
Вводя соответствующие значения в остальные столбцы, можно изменять вид (сплошная линия, пунктир, точки), цвет, толщину и т.д. каждой функции.

На третьей странице окна задается заголовок (Title), место его расположения Above (сверху), Below (снизу), наименования осей (Axis Labels).

Выбрав те или иные требования к графику, нажмем ОК и получим желаемый график.

Чтобы увидеть сделанные изменения, нужно, не закрывая диалоговое окно, нажать кнопку «Применить». Если все в порядке, можно закрывать окно.

Вторая страница окна формирования графиков в полном объеме показана на рисунке 3.15.



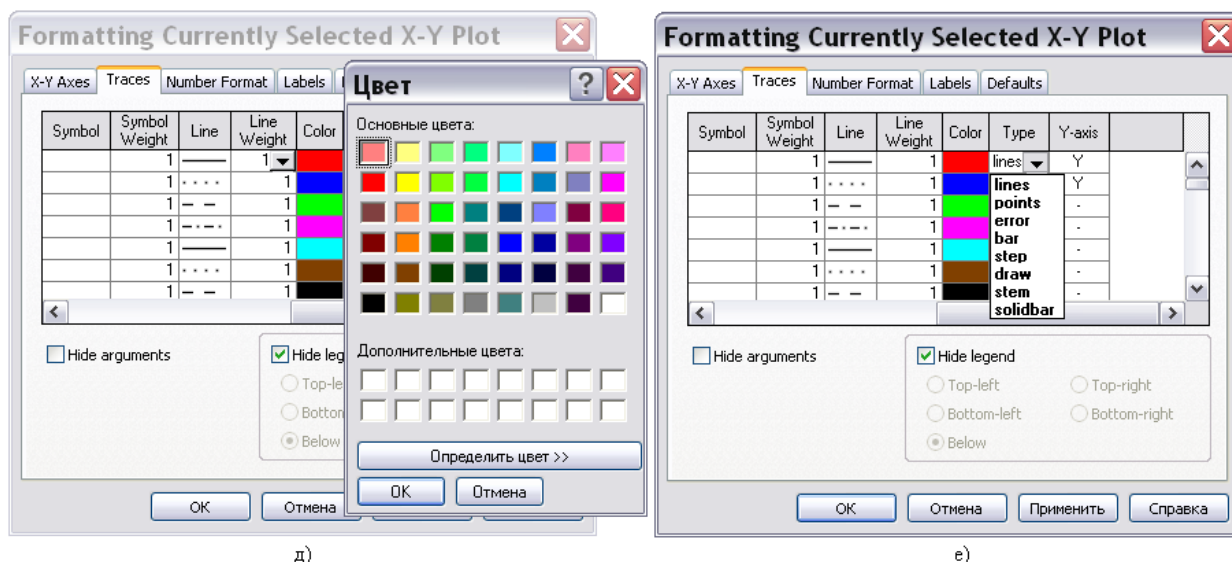


Рисунок 3.15 – Вторая страница окна формирования графиков
 а) «структура» линии графика; б) толщина «структуры» линии графика; в) вид линии; г) толщина линии; д) выбор цвета линии; е) тип линии

3.4.2 Поверхности

Декартовы координаты в пространстве (график поверхности) задаются с помощью точки начала координат и трёх взаимно-перпендикулярных направленных прямых. Прямые занумерованы, задан единичный отрезок. Положение любой точки в пространстве однозначно определено тремя числами: первое число – величина проекции точки на первую ось, второе – величина проекции на вторую ось, третье – на третью. Декартова система координат в пространстве представлена на рисунке 3.16.

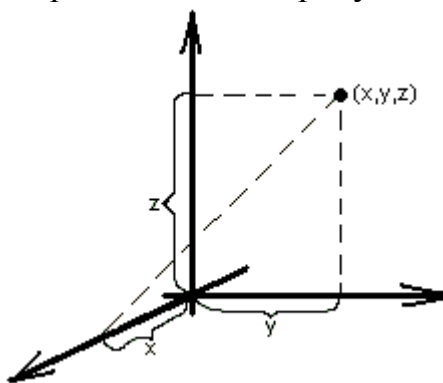


Рисунок 3.16 – Декартова система координат в пространстве

В Mathcad можно строить различные трехмерные графики: поверхности, уровней, столбиковые диаграммы и т.п. мы рассмотрим примеры построения поверхности.

Для построения поверхности необходимо:

- записать исходную функцию;
- определить узлы, в которых будет вычисляться функция;
- установить связь аргументов x и y с узлами;
- вычислить матрицу значений функций двух переменных;

- вызвать графический блок командой меню *Insert-Graph-Surface Plot* или кнопкой из палитры кнопок /;

- в появившемся графическом блоке в единственном пустом поле указать имя матрицы.

Рассмотрим примеры построения графиков поверхностей различными способами.

Пример:

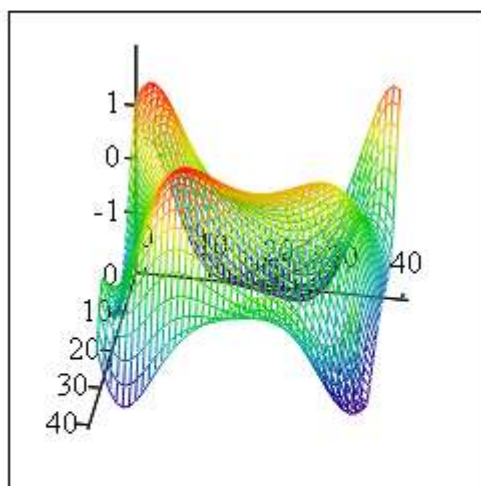
1. Построить график поверхности $f(x, y) = \sin(y^2 + x) + \cos(x^2 + y)$ для x от -2 до 2 , y от -2 до 2

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

$$i := 0..40 \quad j := 0..40$$

$$x_i := -2 + 0.1 \cdot i \quad y_j := -2 + 0.1 \cdot j$$

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



А

Рисунок 3.17 – Построение поверхности в Mathcad

Трёхмерные графики можно изменять. Двойной щелчок левой кнопкой мыши на графическом блоке или команда *Format-Graph-3D Plot* открывает окно (рисунок 3.18), которое имеет девять вкладок, часть из которых описаны ниже:

- Axes – форматирование осей (в виде непосредственно осей, в виде плоскостей, отсутствие осей). На странице Axes (оси) нанесены те же, что и для случая плоских графиков надписи, но теперь уже для трех осей: Grid Lines – нанесение сетки координат, Numbered – оцифровка соответствующей оси, Autoscale – автоматическая разметка осей, Show Markers – показать метки по осям, Autogrid – автоматический показ сетки координат, Numbers of Grids – оцифровка сетки.;

- Title – редактирование заголовка;

- Appearance – изменение цвета и линии поверхности;

- Backplanes – показ заднего плана (показывать (Show), заполнять цветом поверхность (Fill Color) и кромку(Edge Color) и угол, под которым показываются оси координат).

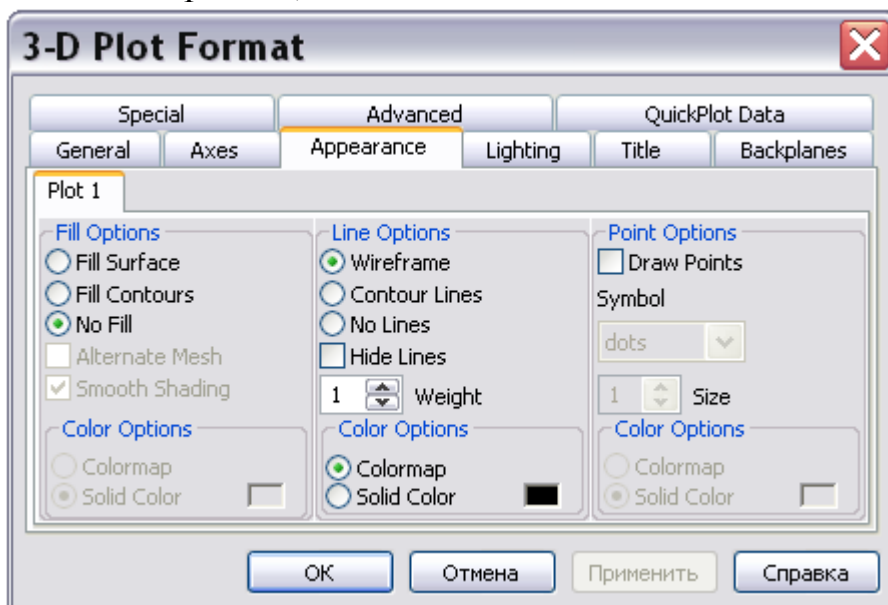


Рисунок 3.18 – Окно для изменения трехмерного графика

2. Рассмотрим другой пример – построение шара.

Построение любой объемной фигуры начинается с того, что мы задаем сетку значений на осях x и y .

- Пусть на каждой оси будет по 25 значений. $N=25$.

- Задаем размещение индексов от 0 до N на осях. Пусть по оси x индекс будет обозначен i , а на оси y – обозначен j .

- Рассчитаем значения углов α_i , α_j к сечению шара плоскостью, параллельной координатной плоскости XOY , и в этом сечении φ_i до пересечения с поверхностью шара – как функции i и j .

- Задаем радиус шара R .

- Рассчитаем координаты точки $x_{i,j}, y_{i,j}, z_{i,j}$ на шаре по соответствующим формулам.

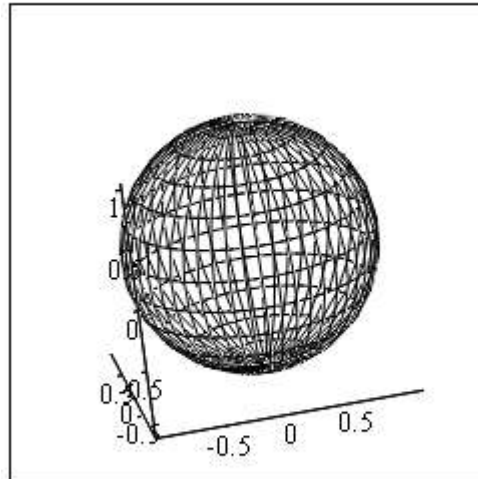
- Выводим графическое поле для построения поверхности и указываем координаты x, y, z в скобках.

Не забываем, что Mathcad различает строчные и прописные литеры, так что если использовались прописные X, Y, Z в формулах, то так и набираются они в слотах.

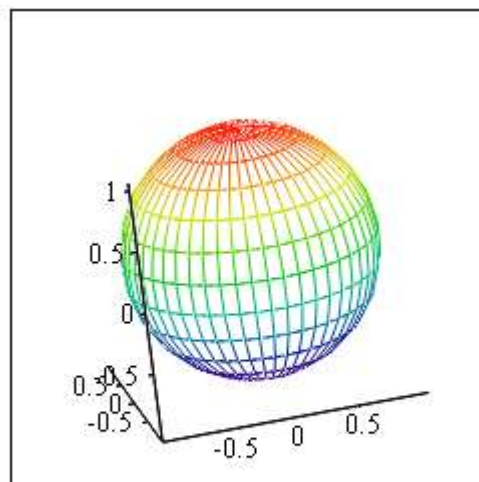
```

N := 25
i := 0..N
j := 0..N
alpha_j := j * 2 * pi / N
phi_i := i * 2 * pi / N
R := 1
x_i,j := R * sin(alpha_j) * cos(phi_i)
y_i,j := R * sin(alpha_j) * sin(phi_i)
z_i,j := R * cos(alpha_j)

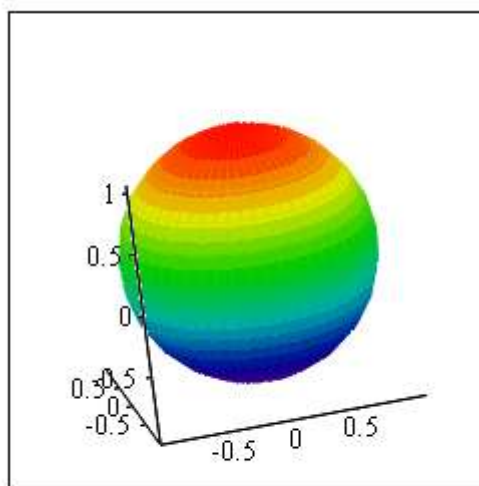
```



(x,y,z)



(x,y,z)



(x, y, z)

Рисунок 3.19 – Трехные графики: а) опции графика – по умолчанию; б) раскраска; опция Appearance – Colormap – Fill Surface; в) раскраска; опция Appearance – Colormap – Fill Contours.

В Mathcad имеется очень удобная функция (CreateMesh), позволяющая задать диапазоны изменения аргументов и частоту нанесения сетки. Рассмотрим действие этой функции на примере построения полусферы.

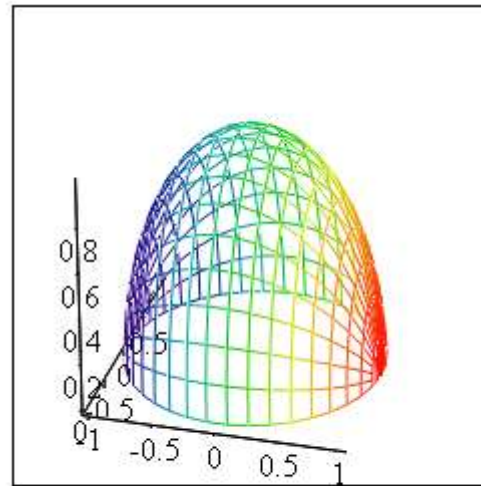
Пример:

Создадим вектор с тремя элементами, в которые впишем координаты точек сферы. Далее используем функцию CreateMesh для показа графика, задавая имя формулы для графика, диапазон изменения аргумента 1, диапазон изменения аргумента 2, число шагов сетки по аргументу 1, число шагов сетки по аргументу 2 соответственно. Далее выводим графическое поле и экспериментируем с пределами изменения аргумента, с шагом сетки, с цветом и другими опциями графика.

$$\underline{R} := 1$$

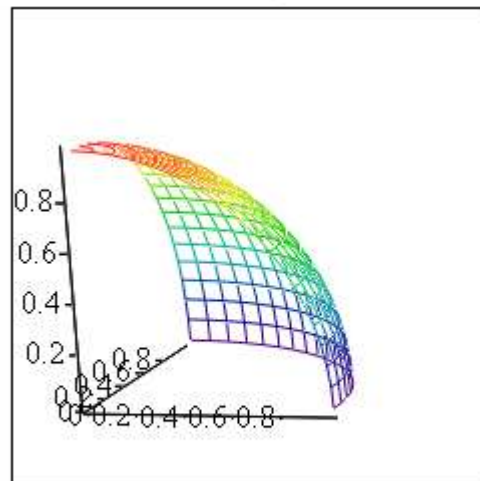
$$\text{Sphere}(\alpha, \phi) := \begin{pmatrix} R \cdot \sin(\alpha) \cdot \cos(\phi) \\ R \cdot \sin(\alpha) \cdot \sin(\phi) \\ R \cdot \cos(\alpha) \end{pmatrix}$$

$$\underline{H} := \text{CreateMesh}(\text{Sphere}, 0, \pi, 0, \pi, 20, 20)$$



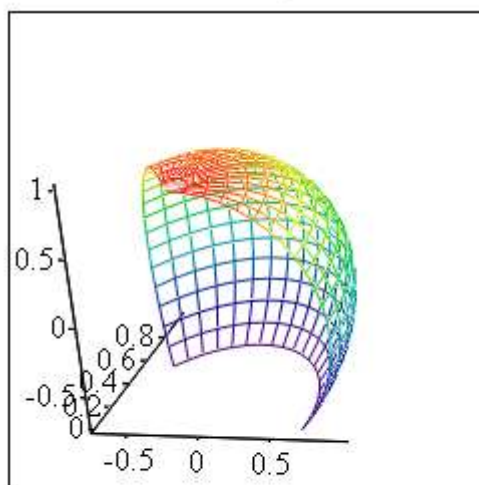
H

H := CreateMesh(Sphere, 0, $\frac{\pi}{2}$, 0, $\frac{\pi}{2}$, 20, 20)



H

H := CreateMesh(Sphere, 0, $\frac{3\pi}{4}$, 0, $\frac{3\pi}{4}$, 20, 20)



Н

Рисунок 3.20 – Примеры графиков, построенных с помощью функции CreateMesh

3.4.3 Полярная система координат на плоскости

Для того, чтобы задать полярную систему координат на плоскости, надо зафиксировать, во-первых, точку начала координат, а во-вторых, луч, выходящий из этой точки. Необходимо также определить единичный отрезок и положительное направление отсчета угла между лучом и отрезком, соединяющим начало координат с какой-либо точкой плоскости.

Положение точки на плоскости задаётся двумя числами. Первое – расстояние от точки до начала координат, а второе – угол между зафиксированным лучом и отрезком, соединяющим точку и начало координат (рис. 3.21).

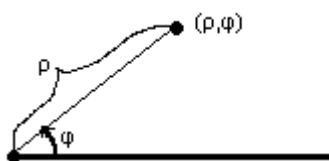


Рисунок 3.21 – Определение положения точки на плоскости

Обычно направление отсчета угла выбирают против часовой стрелки. Стандартное обозначение координат точки в полярной системе – (ρ, φ) . Очевидно, $\rho \geq 0$.

Существуют формулы перехода между заданными стандартным образом декартовой и полярной системами координат. Если они друг другу соответствуют (т.е. должны совпадать начала координат в обеих системах, луч полярной системы координат должен совпадать с «положительной» частью первой оси декартовой системы, должны быть одинаковыми единичные отрезки), то

$$x = \rho \cdot \cos\varphi,$$

$$y = \rho \cdot \sin\varphi.$$

В других случаях формулы зависят от постановки задачи, но получить их легко из геометрических соображений.

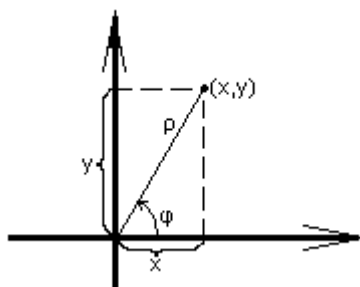


Рисунок 3.22 – Положение точки на плоскости в декартовой системе координат

С помощью этих формул можно осуществлять переход между двумя системами координат, преобразовывать координаты точек, уравнения кривых и т.д.

В полярной системе координат очень просто выглядят уравнения прямых, проходящих через начало координат и окружностей с центром в этой точке. Кроме того, уравнения многих стандартных, часто используемых, кривых принято (с точки зрения простоты) записывать в полярных координатах.

Для построения графиков в полярных системах координат необходимо выбрать на панели графиков – Polar Plot или воспользоваться «горячими» клавишами – $\langle \text{Ctrl} \rangle + \langle 7 \rangle$.

Пример:

Построим графики функций в полярных системах координат:

$y(x) = \sin(x)$; $z(x) = \cos(x)$ для $x \in [0; 2\pi]$.

$x := 0, 0.05 .. 2\pi$

$y(x) := \sin(x)$

$z(x) := \cos(x)$

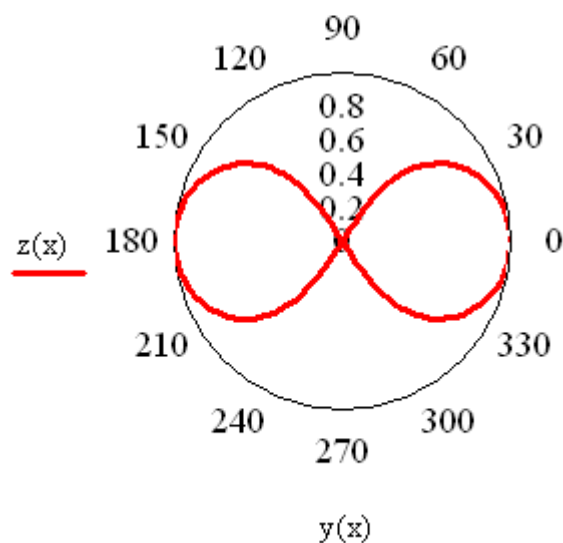


Рисунок 3.23 – Фигура Лисажу в полярной системе координат

Форматирование полярных графиков. Если вы хотите отредактировать график в полярных координатах, необходимо выделить график (щелчком левой кнопки мыши) и выполнить команду *Format-Graph-Polar Plot* или выполнить двойной щелчок на выделенном графике. При этом откроется окно *Formatting Currently Selected Polar Plot* (форматирование полярного графика). Это окно содержит те же вкладки, что и для графиков в декартовой системе координат.

Вкладка *Polar Axes* содержит следующие элементы:

- *Log Scale* (логарифмическая шкала) - используется для создания логарифмической r -оси.
- *Grid Lines* (вспомогательные линии) - отображает сетку линий, соответствующих уравнениям $r=const$ и $q=const$.
- *Numbered* (нумерация) - линии $r=const$ и $q=const$ снабжаются подписями.
- *Show Markers* (показать метки) - при помощи этой опции можно снабдить график двумя дополнительными пунктирными окружностями $r=const$. Для этого надо ввести нужные значения радиуса в появившиеся ячейки. Кроме того, справа от графика указывается минимальный и максимальный радиус; можно увеличить или уменьшить график, введя в эти ячейки собственные значения.
- *Auto Grid* (автосетка) - при установке этой опции число линий сетки определяет Mathcad.

Остальные значения параметров на этой вкладке и на других вкладках те же самые, что и в случае команды *X-Y Plot*.

Лабораторная работа № 3 «Построение графиков»

Цель работы: получить навыки работы в Mathcad при построении графиков.

Задание на лабораторную работу:

Изучить материал, представленный в разделе 3 (пункт 3.4);

Произвести вычисления, представленных в примерах соответствующих разделов.

Выполнить представленное задание:

1. Построить график функции $f(x) = x^2 + 1,2 \cdot x + 3,1$ на отрезке $[-2; 3]$.

2. Построить в одном графическом блоке графики функций: $f(x) = x \cdot \sin(4x)$, $g(x) = \cos(2x) + 0,5$ на отрезке $[-\pi; \pi]$.

3. Построить на одном чертеже графики 3-х различных функций с различными аргументами: $r(b) = 2b - \lg(b) + 1,2$, $r(t) = t - \cos(t)$, $f(x) = x + 2x^2$.

4. Выполнить форматирование построенных графиков: изменить толщину и стиль линий, масштаб, нанести дополнительные линии, ввести надписи.

5. Создать график, путем увеличения некоторой области графика из пункта 3. Прочитать координаты выбранной точки.

6. Построить график функции $z(x, y) = \sin(x^2 + y^2)$ на области $[-1; 1] \times [-1; 1]$.

7. Отформатировать полученный график, изменяя свойства обзора и способы отображения поверхности, добавить заголовок и отформатировать оси.

8. Построить контурный график функции из пункта 6.

9. Построить график функции $z(x, y) = 55x^2 + 25y^2$, область выбрать самостоятельно. Выберите зеленый цвет поверхности, розовый цвет заднего плана, угол представления 45 градусов и оцифрованные оси.

10. Построить график уровней и столбиковую диаграмму для функции из пункта 9.

11. Выведите уравнения (аналогично выводу уравнения шара) для конуса и постройте чертеж половины конуса (рисунок 3.24), высота конуса равна 4, угол при вершине конуса равен $\pi/3$.

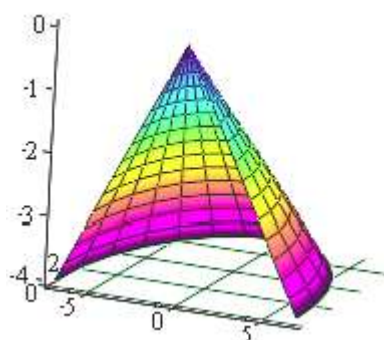


Рисунок 3.24 – Половина конуса

12. Выведите уравнения (аналогично выводу уравнения шара) для цилиндра и постройте чертеж одной третьей части цилиндра, как показано на рисунке 3.25. высота цилиндра равна 4.

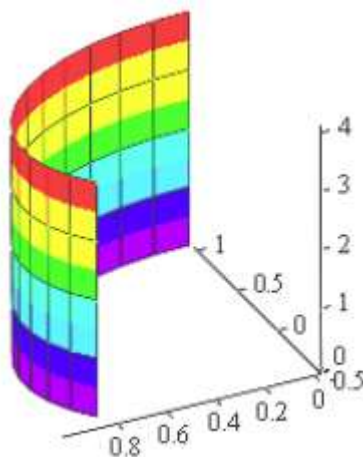


Рисунок 3.25 – Одна третья часть цилиндра

13. Построить графики заданных функций в полярной системе координат

$$r(\theta) = 1 + \sin\left(2 \cdot \theta + 3 \cdot \frac{\pi}{2}\right), \quad d(\theta) = 1 + \frac{\sin(3 \cdot \theta + \pi)}{2}, \quad \text{если } \theta \in [0; 2 \cdot \pi] \text{ с}$$

шагом $N = 15$.

14. Построить кривую на плоскости с заданными параметрами:

$$x_i = \sin\left(2 \cdot \pi \cdot \frac{i}{N}\right), \quad y_i = \sin\left(4 \cdot \pi \cdot \frac{i}{N}\right)$$

15. Построить кривую в пространстве.

$$x_i = \cos\left(\frac{i}{N} \cdot 6 \cdot \pi\right), \quad y_i = \sin\left(\frac{i}{N} \cdot 6 \cdot \pi\right), \quad z_i = \frac{i}{N} \cdot 3$$

16*. Построить векторное и градиентное поле следующей функции:

$$f(x, y) = \sin(x) \cdot \cos(y).$$

17*. Построить график функции $f(x) = x \cdot \sin(x)^2$. Получить вид поверхности путем вращения кривой вокруг оси Ox и Oy (использовать функцию *CreateMesh*).

Составьте отчет по лабораторной работе №3 (см. Приложение А – Пример оформления отчета). Сделайте соответствующие выводы.

3.5 Решение уравнений

Алгебраические уравнения в Mathcad решаются как численными, так и аналитическими методами.

3.5.1 Численное решение нелинейных алгебраических уравнений

В Mathcad корни алгебраических уравнений и систем определяется с помощью встроенных функций.

Для решения одного уравнения с одним неизвестным используется функция $root(f(x),x)$, возвращая значение аргумента x , при котором значение функции $f(x)$ обращается в ноль. Эта функция для поиска корня использует метод секущих. Поэтому перед использованием функции $root(f(x),x)$ необходимо переменной x присвоить начальное значение.

Важно запомнить, что само уравнение в Mathcad не набирается.

Пример:

$$f(x) := x^3 + e^x$$

$$x := 3$$

$$s1 := root(f(x),x)$$

$$s1 = -0.773$$

Графическое решение:

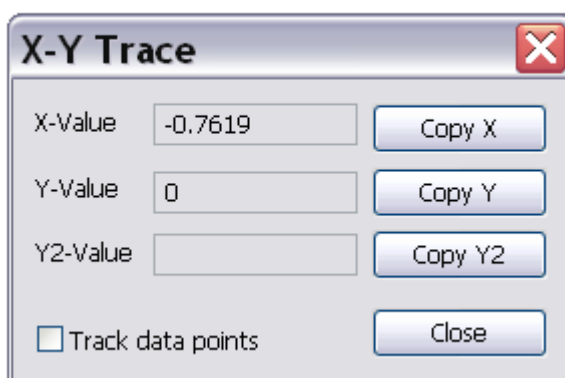
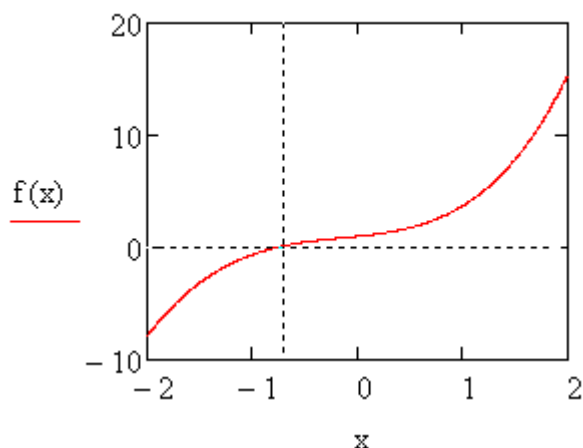


Рисунок 3.26 – Графическое решение уравнения с одной неизвестной функции $f(x) = x^3 + e^x$

Если корней несколько, то потребуется столько же раз вызывать функцию, меняя начальное приближение. Полезно для отделения корней исследовать график функции $f(x)$. Тип найденного решения (вещественный, комплексный) зависит от начального приближения.

Пример:

Найти корни уравнения $x^3 - 10 \cdot x + 2 = 0$

Предварительно построим график:

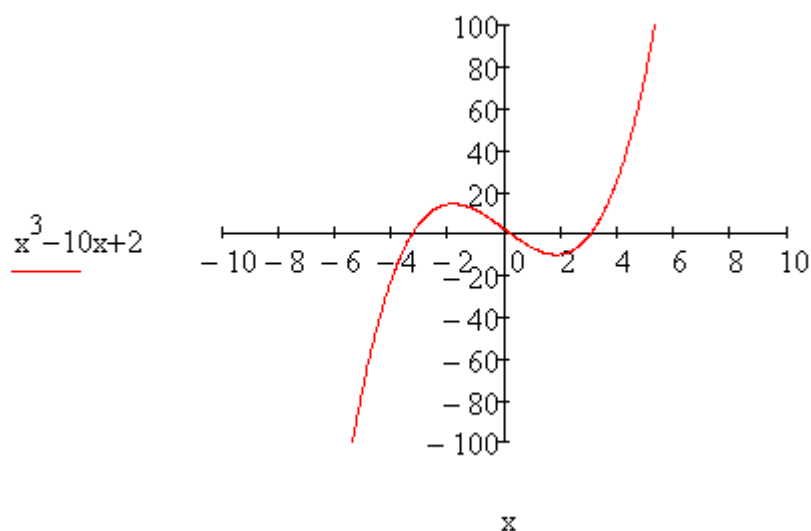


Рисунок 3.27 – Графическое решение уравнения с одной неизвестной функции $f(x) = x^3 - 10 \cdot x + 2$

Зададим функцию пользователя $f(x) = x^3 - 10 \cdot x + 2$. Вычислим корни уравнения, трижды задавая начальное приближение.

$$\underline{\underline{f}}(x) := x^3 - 10x + 2$$

$$x := 0 \quad \text{root}(f(x), x) = 0.201$$

$$\underline{\underline{x}} := -5 \quad \text{root}(f(x), x) = -3.258$$

$$\underline{\underline{x}} := 3 \quad \text{root}(f(x), x) = 3.057$$

Для увеличения или уменьшения точности, с которой ищется корень, можно изменить значение встроенной переменной TOL. При уменьшении значения этой переменной ответ станет более точным, но время вычислений увеличивается.

Если решение с заданной точностью не находится, то на экране появится сообщение «отсутствует сходимость».

Функция *root* может использоваться для решения уравнений с параметром. Для этого определяют данное уравнение как функцию нескольких переменных и решают уравнение, присвоив начальное значение параметрам и переменной.

Для нахождения корней полинома лучше использовать функцию *polyroots(v)*, которая не требует начального приближения и возвращает сразу все корни (вещественные и мнимые).

Решение уравнений с помощью функции *polyroots(v)* можно разбить на несколько этапов:

1. Вводим функцию для исходного уравнения;
2. Составляем вектор столбец v коэффициентов уравнения;
3. Записываем функцию $r = \text{polyroots}(v)$;
4. Выводим результат на экран.

Пример:

Найти решение уравнения $x^3 - 10 \cdot x + 2 = 0$.

Количество решений можно определить по максимальной степени полинома n . В нашем примере количество корней равно 3.

Введем функцию для исходного уравнения и вектор коэффициентов:

Размерность вектора определяется как $n+1$ (к максимальной степени полинома прибавляется единица).

$$1 \cdot x^3 + 0 \cdot x^2 - 10 \cdot x^1 + 2 \cdot x^0 = 0$$

$$\underline{\underline{g}}(x) := x^3 - 10x + 2$$

$$v := \begin{pmatrix} 2 \\ -10 \\ 0 \\ 1 \end{pmatrix}$$

$r := \text{polyroots}(v)$

$$r = \begin{pmatrix} -3.258 \\ 0.201 \\ 3.057 \end{pmatrix}$$

Построим график многочлена $g(x)$ и выведем координаты корней $g(x)$.

$j := 0, 1..2$

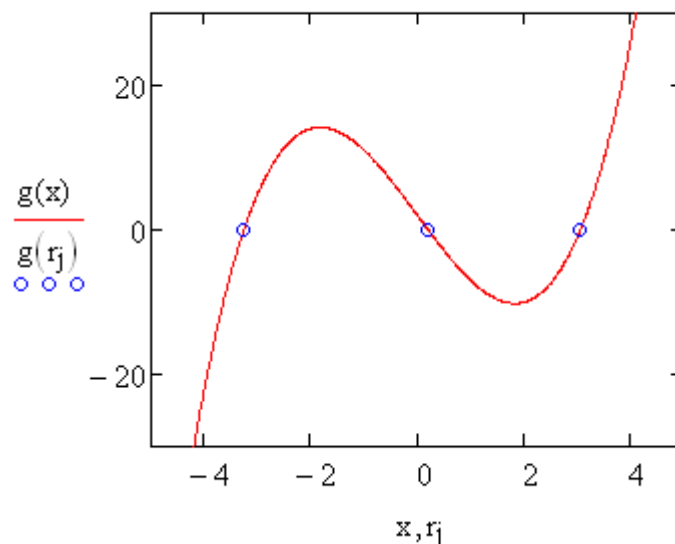


Рисунок 3.28 – Построение графика многочлена и выведение координат корней

Следует обратить внимание, на то, что первый элемент вектора соответствует коэффициенту уравнения при свободном члене.

3.5.2 Численное решение системы линейных алгебраических уравнений

При численном решении систем линейных уравнений используется специальный вычислительный блок, открываемый служебным словом – директивой Given. Блок имеет следующую структуру:

1. Задание начальных приближений;
2. Given;
3. Уравнения;
4. Ограничительные условия выражения с функцией Find.

Mathcad решает системы уравнений при помощи Given, в которых применяются итерационные методы. Для решения системы уравнений нужно выполнить следующее:

1. Задать начальные приближения для всех переменных, значения которых находим. На основе начального приближения строится последовательность, сходящаяся к искомому решению;
2. Напечатать ключевое слово Given;
3. Ниже Given ввести уравнения и неравенства, задающие систему. Левые и правые части уравнений связываются знаками операций

сравнения $>$, \geq , $<$, \leq или символом равно ($=$), который вводится комбинацией клавиш [Ctrl]+[=].

4. Ввести выражение, которое включает функцию Find. В качестве аргументов этой функции нужно указать переменные, значения которых находим. Функция Find возвращает в виде вектора решение системы уравнений.

Пример:

$$x1 := 0 \quad x2 := 0 \quad x3 := 1 \quad x4 := 1$$

Given

$$0.63 \cdot x1 - 9.72 \cdot x2 - 0.62 \cdot x3 + 0.48 \cdot x4 = 1.03$$

$$21.08 \cdot x1 + 0.64 \cdot x2 + 1.95 \cdot x3 + 1.54 \cdot x4 = -9.47$$

$$0.88 \cdot x1 - 0.72 \cdot x2 + 1.36 \cdot x3 + 0.68 \cdot x4 = -0.85$$

$$3.58 \cdot x1 + 1.48 \cdot x2 + 0.82 \cdot x3 + 1.58 \cdot x4 = 1.02$$

$$\begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \end{pmatrix} := \text{Find}(x1, x2, x3, x4)$$

$$\begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \end{pmatrix} = \begin{pmatrix} -0.493 \\ 0.079 \\ -1.497 \\ 2.466 \end{pmatrix}$$

Подобным образом можно решать и нелинейные уравнения. Однако они имеют несколько корней. Задавшись начальными приближениями, мы найдем в лучшем случае один корень, ближайший к начальному приближению. Таким способом имеет смысл искать корни трансцендентных уравнений, имеющие, как известно, бесконечное количество корней.

Пример:

Найти корень трансцендентного уравнения.

$$X \sin(x) + \cos(x) = 25, \text{ ближайший } x=1.$$

Набираем задачу описанным выше способом и находим значение x . Однако получить решение при начальном приближении 10 нам не удастся.

Mathcad позволяет решать системы линейных алгебраических уравнений в матричной форме. Решение можно получить двумя способами.

1 способ.

Как известно, система линейных алгебраических уравнений в матричной форме имеет вид:

$$A \cdot X = B,$$

где A – квадратная матрица коэффициентов,

X – вектор – столбец неизвестных,

B – вектор – столбец свободных членов системы.

Решение системы в матричной форме: $X = A^{-1} \cdot B$.

Решим в матричной форме систему:

$$11 \cdot x + 12 \cdot y + 31 \cdot z = 9$$

$$4 \cdot x + 52 \cdot y + 69 \cdot z = 8$$

$$7 \cdot x + 86 \cdot y + 93 \cdot z = 7$$

1) Наберем `ORIGIN:=1`. Как говорилось выше, это означает, что счет элементов будет производиться не от нуля, а с единицы.

2) Введем матрицу A .

3) Введем вектор – столбец B .

4) Набор выражения для X желательно выполнять, используя соответствующую кнопку матричной панели. После этого наберем `X=` и сразу получим вектор ответа.

$$A := \begin{pmatrix} 1 & 12 & 31 \\ 4 & 52 & 69 \\ 7 & 86 & 93 \end{pmatrix} \quad B := \begin{pmatrix} 9 \\ 8 \\ 7 \end{pmatrix}$$

$$X := A^{-1} \cdot B \quad X = \begin{pmatrix} 7.694 \\ -1.016 \\ 0.435 \end{pmatrix}$$

2 способ

Возможно получения решения матричного уравнения с помощью специальной функции `lsolve`.

`ORIGIN := 1`

$$A := \begin{pmatrix} 1 & 12 & 31 \\ 4 & 52 & 69 \\ 7 & 86 & 93 \end{pmatrix} \quad B := \begin{pmatrix} 9 \\ 8 \\ 7 \end{pmatrix}$$

`X := lsolve(A,B)`

$$X = \begin{pmatrix} 7.694 \\ -1.016 \\ 0.435 \end{pmatrix}$$

Уравнения и неравенства, которые следуют за словом `Given`, называются ограничениями. Ключевое слово `Given`, ограничения, выражение,

содержащее функцию Find, называются блоком решения уравнений. В этом блоке могут появляться выражения строго определенного типа. Нельзя использовать ограничения, содержащие двойное неравенство вида $a < d < c$ или строгое неравенство \neq , операторы присваивания или дискретные аргументы.

Блоки решений не могут быть сложенными друг в друга.

Если в результате решения системы уравнений будет выдано сообщение об ошибке типа «Решение не найдено», это означает, что на каком-то этапе итераций не может быть найдено приемлемое приближение к искомому решению. В этом случае полезно исследовать графики, связанные с системой, для определения области начального приближения. Можно также изменять значение переменной TOL.

Системы уравнений в Mathcad можно также решать, используя функцию Minerr. Эта функция использует тот же алгоритм, что и функция Find. Различие заключается в следующем. Если в случае поиска решения не может быть получено дальнейшее уточнение текущего приближения к решению, Minerr возвращает это приближение, а не сообщение об ошибке. Правила использования функции Minerr такие же, как и для функции Find. Поэтому при использовании функции Minerr необходимо всегда включать дополнительную проверку достоверности получаемых результатов.

Пример:

$$x := 1 \quad y := 0$$

Given

$$\sin(x + 1) - y = 1.2$$

$$2 \cdot x + \cos(y) = 2$$

$$\text{minerr}(x, y) = \begin{pmatrix} 0.51 \\ -0.202 \end{pmatrix}$$

3.5.3 Решение алгебраических уравнений в аналитической (символьной) форме

Mathcad предоставляет возможность решения алгебраических уравнений в символьной (аналитической) форме. Преимуществом символьного решения является возможность решения уравнений с буквенными значениями коэффициентов. Правда, более или менее сложные уравнения символьно в Mathcad не решаются, поэтому приходится обращаться к численным методам.

3.5.4 Решение систем линейных уравнений

Символьное решение линейных систем алгебраических уравнений производится с помощью двух операций: solve (решить) и lsolve. Рассмотрим на примерах различные методы решения.

Пример:

Система линейных алгебраических уравнений задана матрицей M коэффициентов и вектором v правых частей. Найти аналитическое решение.

Сначала вводим матрицу и вектор.

$$M := \begin{pmatrix} 0.3 & 0.2 & 6.6 & -1.1 \\ 4.5 & -1.8 & -0.3 & 6.5 \\ -7.3 & 9.7 & 10.9 & -4.1 \\ 8.1 & -2.7 & 8.7 & 8.9 \end{pmatrix} \quad V := \begin{pmatrix} 1 \\ 0.1 \\ 0.01 \\ 1 \cdot 10^{-3} \end{pmatrix}$$

А) Решение с использованием встроенной функции `lsolve`. Функция набирается с клавиатуры или из окна встроенных функций. Стрелка набирается с панели символьных решений.

$$\text{lsolve}(M, V) \rightarrow \begin{pmatrix} -3.9371585952195628203 \\ -2.9752573457871181164 \\ 0.74590602095089835895 \\ 1.9516188095933060907 \end{pmatrix}$$

Б) Решение с использованием оператора `solve`

$$M \cdot \begin{pmatrix} w \\ x \\ y \\ z \end{pmatrix} = V \quad \left| \begin{array}{l} \text{solve, w, x, y, z} \\ \text{float, 4} \end{array} \right. \rightarrow (-3.937 \ -2.975 \ .7459 \ 1.952)$$

Здесь помимо оператора `lsolve` использован оператор `float` (плавающая точка) и задана точность решения – 4 знака.

Операторы `solve` и `float` набираются последовательно.

В) Решение в скалярной форме

$$\begin{pmatrix} 0.3 \cdot w + 0.2 \cdot x + 6.6 \cdot y - 1.1 \cdot z = 1 \\ 4.5 \cdot w - 1.8 \cdot x - 0.3 \cdot y + 6.5 \cdot z = 0.1 \\ -7.3 \cdot w + 9.7 \cdot x + 10.9 \cdot y - 4.1 \cdot z = 0.01 \\ 8.1 \cdot w - 2.7 \cdot x + 8.7 \cdot y + 8.9 \cdot z = 1 \cdot 10^{-3} \end{pmatrix} \left| \begin{array}{l} \text{solve, w, x, y, z} \\ \text{float, 4} \end{array} \right. \rightarrow (-3.937 \ -2.975 \ .7459 \ 1.952)$$

Г) Решение с созданием решающего блока и директивы `Given`. Директива `Given` и оператор `Find` набираются с клавиатуры.

Given

$$3 \cdot x_1 + 4 \cdot x_2 + 4 \cdot x_3 + 6x_4 + 8x_5 = 21$$

$$5 \cdot x_1 + 7 \cdot x_2 + 8 \cdot x_3 - 7x_4 - 9x_5 = 34$$

$$9 \cdot x_1 + 12 \cdot x_2 + 9 \cdot x_3 + 8x_4 + 4x_5 = 41$$

$$13 \cdot x_1 + 3 \cdot x_2 + 19 \cdot x_3 + 18x_4 + 9x_5 = 141$$

$$23 \cdot x_1 + 13 \cdot x_2 + 9 \cdot x_3 + 8x_4 + 19x_5 = 241$$

$$\text{Find}(x_1, x_2, x_3, x_4, x_5) \rightarrow \begin{pmatrix} \frac{30965}{2539} \\ -58955 \\ \frac{10156}{4366} \\ \frac{2539}{-15189} \\ \frac{5078}{23741} \\ \frac{10156}{10156} \end{pmatrix}$$

Д) Решение системы с буквенными коэффициентами

Given

$$a_1 \cdot x + b_1 \cdot y + c_1 \cdot z = d_1$$

$$a_2 \cdot x + b_2 \cdot y + c_2 \cdot z = d_2$$

$$a_3 \cdot x + b_2 \cdot y + c_2 \cdot z = d_3$$

$$\text{Find}(x, y, z) \rightarrow \begin{bmatrix} \frac{(-d_3) + d_2}{(-a_3) + a_2} \\ \frac{-[(-a_2) \cdot c_2 \cdot d_1 + a_2 \cdot d_3 \cdot c_1 + c_2 \cdot a_1 \cdot d_2 - c_2 \cdot a_1 \cdot d_3 + c_2 \cdot d_1 \cdot a_3 - d_2 \cdot a_3 \cdot c_1]}{[(-a_3) + a_2] \cdot (b_1 \cdot c_2 - c_1 \cdot b_2)} \\ \frac{-[(-a_1) \cdot b_2 \cdot d_2 - b_1 \cdot a_2 \cdot d_3 + b_1 \cdot a_3 \cdot d_2 + a_1 \cdot d_3 \cdot b_2 - d_1 \cdot a_3 \cdot b_2 + d_1 \cdot b_2 \cdot a_2]}{b_1 \cdot a_2 \cdot c_2 - b_1 \cdot a_3 \cdot c_2 + c_1 \cdot a_3 \cdot b_2 - c_1 \cdot b_2 \cdot a_2} \end{bmatrix}$$

3.5.5 Символьное решение нелинейных алгебраических уравнений

Решение уравнения четвертой степени с численными коэффициентами с использованием оператора solve.

Пример:

$$7x^2 - 9x + 1 \text{ solve ,x} \rightarrow \begin{pmatrix} \frac{9}{14} - \frac{\sqrt{53}}{14} \\ \frac{\sqrt{53}}{14} + \frac{9}{14} \end{pmatrix}$$

Решение квадратного уравнения с буквенными коэффициентами.

$$a \cdot x^2 + b \cdot x + c = 0 \text{ solve ,x} \rightarrow \begin{pmatrix} \frac{b - \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a} \\ \frac{b + \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a} \end{pmatrix}$$

Решение квадратного уравнения с буквенными коэффициентами с формированием решающего блока

Given

$$x^2 + y^2 = r^2$$

$$(x - \alpha)^2 + y^2 = r^2$$

$$M(\alpha, r) := \text{Find}(x, y) \rightarrow \begin{pmatrix} \frac{\alpha}{2} & \frac{\alpha}{2} \\ \frac{\sqrt{4 \cdot r^2 - \alpha^2}}{2} & -\frac{\sqrt{4 \cdot r^2 - \alpha^2}}{2} \end{pmatrix}$$

Но уже кубическое уравнение с буквенными коэффициентами не решается!

Лабораторная работа № 4 «Решение уравнений»

Цель работы: получить навыки работы в Mathcad при решении уравнений.

Задание на лабораторную работу:

Изучить материал, представленный в разделе 3 (пункт 3.5);

Произвести вычисления, представленных в примерах соответствующих разделов.

Выполнить представленное задание:

Решить уравнения с помощью функции $\text{root}(f(x), x)$

1. $2^x + 5 \cdot x - 3 = 0$

2. $0.5^x = (x - 2)^2$

3. $(x - 3) \cdot \cos(x) = 1$

4. $\text{arctg}(x) + \frac{1}{3 \cdot x^3} = 0$

5. $\sin(x + \frac{\pi}{3}) - 0.5 \cdot x = 0$

6. $5^x - 3 \cdot x = 0$

7. $(x - 1)^2 \cdot \lg(x + 1) = 0$

Решить уравнения с помощью функции $\text{polyroots}(v)$

8. $5 \cdot x^5 + 6 \cdot x^3 + 8 \cdot x^2 + 2 \cdot x = 0$

9. $x^4 - x^3 - 2 \cdot x^2 + 3 \cdot x - 3 = 0$

10. $3 \cdot x^4 - 8 \cdot x^3 - 8 \cdot x^2 = -2$

11. $2 \cdot x^4 - x^2 - 10 = 0$

12. $5 \cdot x^4 + 8 \cdot x^3 + 3 \cdot x^2 + 9 \cdot x + 8 = 0$

Решить систему уравнений с помощью Find

13. $5x + 6y - 9z + 2v - 7w = 90$

$3x - 4y + 5z - 3v + 4w = 12$

$9x + y + 3z - 2v + 9w = 51$

$7x + 2y - 8z + v + 10w = 32$

$6x + 5y - 4z + 3v - 2w = 87$

14. $4.5x + 7.9y - 2.1v + 6.75w + 7.9u = 43$

$5.6x + 7.2y + 9.8z + 3.9v + 3.4w + 8.3u = 12.54$

$5.6x + 98.5y + 43.7z + 67.85v + 4.9w + 21.5u = 54.98$

$65.75x + 54.32y - 78.32z - 565.9v + 32w + 78.54u = 55.5$

$54.2x + 76.45y + 32.23z + 45.71v + 43.43w + u = 65.21$

$8.9x + 9.8y - 5.6z + 6.5v - 4.5w + 2.1u = 0$

15. $0.46x_1 + 1.72x_2 + 2.53x_3 = 2.44$

$1.53x_1 - 2.32x_2 - 1.83x_3 = 2.83$

$0.75x_1 + 0.86x_2 + 3.73x_3 = 1.06$

16. $4.24x_1 + 2.73x_2 - 1.55x_3 = 1.87$

$2.34x_1 + 1.27x_2 - 3.83x_3 = 2.16$

$3.05x_1 - 1.05x_2 - 0.63x_3 = -1.26$

Решить систему уравнений с помощью Minerr

17. $\sin(x - 6) - y = 1.6$

$$3x - \cos(y) = 0.9$$

18. $x^2 + y^2 = 1$

$$\sin(x+y) - 1.1x = 0.1$$

19. $e^{x+y} - x^2 + y = 2$

$$(x + 0.5)^2 + y^2 = 1$$

20. $\text{tg}(y-x) + x \cdot y = 0.3$

$$x^2 + y^2 = 1.5$$

Составьте отчет по лабораторной работе №4 (см. Приложение А – Пример оформления отчета). Сделайте соответствующие выводы.

4 Программирование в Mathcad

4.1 Введение. О программировании в среде Mathcad

Вплоть до появления 7 версии системы Mathcad возможности программирования в них были крайне ограниченными. Фактически Mathcad позволяла реализовать лишь линейные программы, в основе которых лежит понятие функции. Функция *if* и ранжированные переменные в отдельных случаях могли заменить условные выражения и циклы, но с серьезными ограничениями. Отсутствовала возможность задания завершенных программных модулей.

Отметим, что возможность составлять программы реализована только в версии Professional. Все эти Mathcad-программы с точки зрения программиста представляют собой подпрограммы-функции, которые могут возвращать в качестве результата число, вектор или матрицу. Функции могут вызывать сами себя (рекурсивно определенные функции) или другие подпрограммы-функции, определенные выше в том же Mathcad-документе.

Mathcad содержит собственные (правда, очень скромные) средства программирования; имеется также возможность встраивания программ, написанных на C++.

Перечислим некоторые особенности программирования в Mathcad.

1. Операторы программы нельзя набирать в ручную. Их вводят щелчком мыши по наименованиям операторов из окна программирования (рис. 4.1). Окно вызывается из меню *View-Toolbars-Programming*.

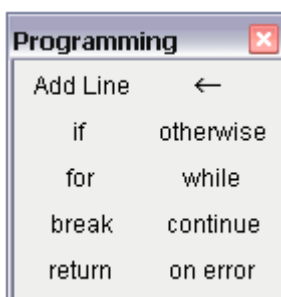

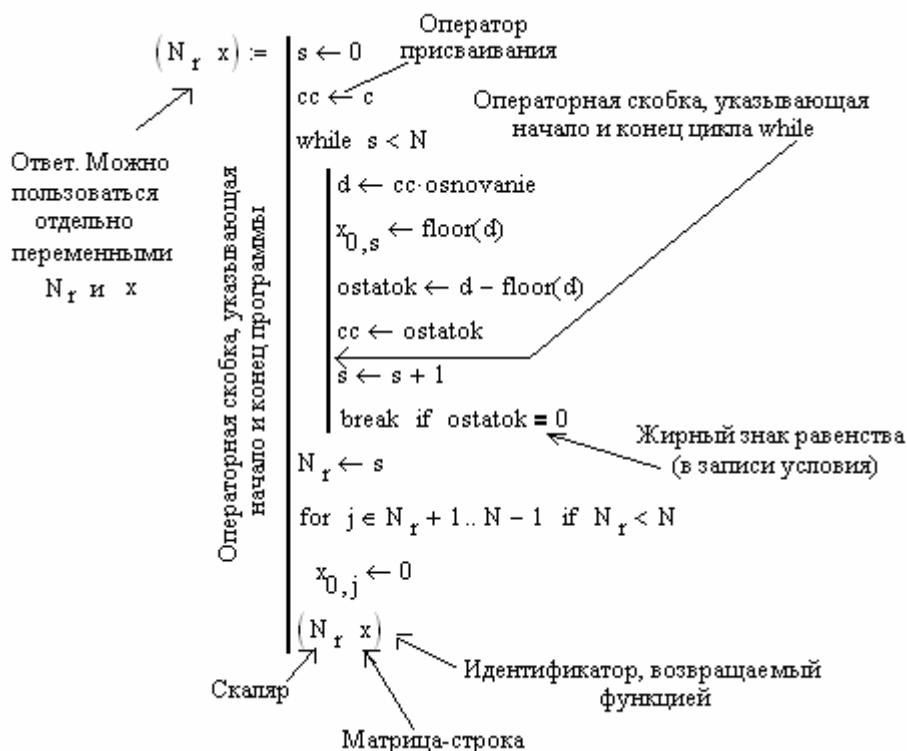


Рисунок 4.1 – Окно программирования в Mathcad

Add Line – добавление линии разделителя операторных скобок (for – конец цикла, if – конец условия и т.п.), break – команда прерывания цикла,
 – символный знак присваивания

2. Все программы являются функциями (не процедурами!).
3. Переменные, объявленные выше и/или левее текста программы, передаются в программу как значения (by value) и не изменяются, если по ходу программы им присваиваются другие значения (часто это обстоятельство забывается и служит источником ошибок).
4. Программы пишутся с указанием начала и конца операторных скобок (for..., while..., if...); такое указание производится с помощью вертикальных линий.

Пример:



5. В арифметических выражениях используется специфический знак присваивания: стрелка справа налево (\leftarrow).
6. В логических выражениях и для организации циклов используются:
 - знаки отношения:
 - а) равно – жирный знак равенства (ввод $\langle \text{Ctrl} \rangle + \langle = \rangle$);
 - б) больше (ввод клавишей $>$);
 - в) меньше (ввод клавишей $<$);
 - г) больше или равно (ввод $\langle \text{Ctrl} \rangle + \langle \text{круглая закрывающая скобка} \rangle$);
 - д) меньше или равно (ввод $\langle \text{Ctrl} \rangle + \langle \text{круглая открывающая скобка} \rangle$);
 - знаки логических действий: условия «И» и «ИЛИ» (ввод знаком умножения $(*)$ и сложения $(+)$).
7. Функция должна заканчиваться идентификатором возвращаемых переменных (простая переменная, матрица или структура – т.е. некоторые из переменных, указанных в идентификаторе, являются простыми переменными, а другие – векторами или матрицами).
8. Программы могут содержать рекурсии.

Ниже в таблице 4.1 приведены операторы, находящиеся на кнопках панели программирования.

Таблица 4.1 – Команды панели Symbolic (символы)

Команда	Функция	Пример
<i>Add Line</i>	Добавляет новую строку под/над (зависит от выделения) текущей строкой.	
\leftarrow	Присваивание	$y \leftarrow 0$

	значения локальной переменной.	
<i>if</i>	Условный оператор (оператор ветвления) <i>if</i> ; условие должно стоять после <i>if</i> , а оператор, который выполняется, если выполнено заданное условие,- перед <i>if</i> .	$f(x) := \begin{cases} -x & \text{if } x < 0 \\ x & \text{otherwise} \end{cases}$ $f(-3) = 3 \quad f(4) = 4$
<i>otherwise</i>	Обозначает оператор, который должен быть исполнен, если условие оператора <i>if</i> не выполняется.	$f(x) := \begin{cases} -x & \text{if } x < 0 \\ x & \text{otherwise} \end{cases}$ $f(-3) = 3 \quad f(4) = 4$
<i>for</i>	Цикл <i>for</i> ; за ключевым словом <i>for</i> следует переменная-счетчик, а после символа принадлежности вводится промежуток изменения этой переменной. Внутренние операторы цикла сдвинуты немного вправо.	$\text{Sum}(n) := \begin{cases} s \leftarrow 0 \\ \text{for } i \in 1..n \\ s \leftarrow s + i \end{cases}$ $\text{Sum}(4) = 10$
<i>while</i>	Цикл <i>while</i> ; внутренние операторы цикла будут исполняться до тех пор, пока будет истинным условие, следующее за ключевым словом <i>while</i> . Пример показывает применение цикла для нахождения нулей функции методом касательных Ньютона.	$N(x, f, f_x) := \text{while } f(x) > 10^{-6}$ $x \leftarrow x - \frac{f(x)}{f_x(x)}$ $N(2, \sin, \cos) = 3.142$
<i>break</i>	Служит для	$\text{break if } i \geq 10$

	преждевременного завершения цикла, чтобы, например, избежать заикливания или слишком продолжительных вычислений.	
<i>continue</i>	Служит для преждевременного завершения текущей итерации цикла; сам цикл при этом продолжается.	<code>continue if $x \geq 10$</code>
<i>return</i>	Преждевременное завершение программы; указанное в ячейке значение будет возвращено.	<code>return y</code>
<i>on error</i>	Если при вычислении выражения <code>expr2</code> возникла ошибка, вычисляется выражение <code>expr1</code> .	<code>expr1 on error expr2</code>

4.2 Техника программирования в Mathcad

При освоении техники программирования (любой, не только в Mathcad) наилучшим способом является решение конкретной задачи и (по его ходу) освоение операторов и приемов. Такой способ имеет особое название DYI («do it yourself» – «делай сам»). Разберем этот способ более подробно.

Для начала рассмотрим этапы, относительно которых будем изучать программирование:

1. Этап 1. Составить требования к входным данным и к форме выдачи результата.
2. Этап 2. Разработать способ получения результата, описанный в литературе, или самому разработать способ получения результата. Рассчитать контрольные примеры, иллюстрирующие разные варианты окончания работы алгоритма.
3. Этап 3. Составить алгоритм и представить его в схематичной форме – блок-схема.
4. Этап 4. Рассчитать контрольные примеры применительно к алгоритму.

5. Этап 5. Составить программу, отладить ее на контрольных примерах.

Необходимо также учесть, что готовую схему или алгоритм вы сразу получите. Не бойтесь делать ошибки, самое главное – во время их обнаружить.

Пример:

Требуется составить программу для перевода правильной десятичной дроби в новую систему счисления и представить результат в слове с заданным числом позиций.

Решение задачи будем разбирать относительно наших этапов.

4.2.1 Этап 1. «Выработка требований к исходным данным и к форме выдачи результата»

Требования к исходным данным. В начале определяем, что в данной задаче представляют собой исходные данные. Согласно заданию в состав исходных данных входят:

1. Основание новой системы счисления, в которую должна переводиться дробь;
2. Число разрядов (если переводится в двоичную систему) или позиций (если система не двоичная), отведенных для представления результата;
3. Правильная десятичная дробь, подлежащая переводу в новую систему счисления. Поскольку программа имеет форму функции, исходные данные укажем в виде абстрактных аргументов (формальных параметров). Примем следующие обозначения:
 - *osnovanie* – основание новой системы счисления;
 - *n_bit* – число разрядов или позиций для представления результата;
 - *fraction* – правильная десятичная дробь, подлежащая переводу в новую систему счисления;
 - *Z* – идентификатор функции, программу для которой предстоит разработать.

В этих обозначениях функция с аргументами будет записана так:

$Z(\textit{osnovanie}, \textit{n_bit}, \textit{fraction})$

«Требования к форме выдачи результата»

Согласно содержательной постановке задачи функция $Z(\textit{osnovanie}, \textit{n_bit}, \textit{fraction})$ должна возвращать массив, j -й элемент которого представляет собой сомножитель, с которым слагаемое $\textit{Разряд}_j \cdot \textit{osnovanie}^{-j}$ входит в представление дроби *fraction* в новой системе счисления.

Для того чтобы прочесть и проверить результат, нужно перевести его снова в десятичную форму:

$$\text{Дес}_{\text{знач}} = [\textit{разряд}]_1 \cdot [\textit{основание}]^{-1} + [\textit{разряд}]_2 \cdot [\textit{основание}]^{-2} + \dots + [\textit{разряд}]_{\textit{n_bit}} \cdot [\textit{основание}]^{-\textit{n_bit}}$$

Например, пусть задано значение основания $osnovanie = 2$ новой системы счисления. Пусть десятичная дробь, подлежащая переводу, равна: $fraction = 0.625$. Пусть число разрядов в слове равно $n_bit = 8$.

Тогда функция должна вернуть массив – строку (1 0 1 0 0 0 0 0). Для того чтобы прочесть это число (т.е. перевести его в десятичную форму), рассчитываем:

$$1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 0 \cdot 2^{-4} + 0 \cdot 2^{-5} + 0 \cdot 2^{-6} + 0 \cdot 2^{-7} + 0 \cdot 2^{-8} = 0.625$$

Важно помнить, что не всякую десятичную дробь можно точно представить в любой системе счисления, если число позиций для выдачи результата ограничено. Поэтому полезно дополнить состав переменных, возвращаемых функцией, сведениями:

– о погрешности представления дроби $fraction$ в новой системе счисления;

– о числе значащих разрядов (если дробь представима точно, то часть разрядов справа будет нулями). В примере, рассмотренном выше, число значащих разрядов равно 3.

Таким образом, определяем перечень переменных, которые должна вернуть функция $Z(osnovanie, n_bit, fraction)$:

- Число значащих разрядов N_r ;
- Массив-строку x из n_bit элементов (столбцов) с результатами перевода дроби в новую систему счисления;
- Погрешность ε от перевода десятичной дроби $fraction$ в новую систему счисления.

Состав возвращаемых переменных неоднороден: часть переменных (N_r, ε) являются простыми переменными, другая часть (x) – массив с индексом $j \in \{0, \dots, n_bit - 1\}$. Как известно, смешанные типы данных называются структурами.

На этом этап первый завершен.

4.2.2 Этап 2. «Разбор способа получения результата. Расчет примеров»

Алгоритм перевода десятичной дроби в новую систему счисления состоит из последовательности умножения дроби на новое основание, выделения целой части (она записывается в очередную позицию слова-ответа), вычисления остатка и повторного умножения его на основание – до тех пор, пока (а) либо не будет получен остаток, равный нулю, либо (б) не будут исчерпаны позиции, отведенные для хранения числа.

Нужно составить примеры для обоих вариантов окончания работы алгоритма.

Пример 1. Пусть число разрядов в слове $n_bit = 8$. В результате вычислений, число 0.78_{10} в двоичной системе представляется не точно: $(0.11000111\dots)_2$. Его дробная часть читается «сверху вниз» в ходе последовательных умножений остатка от вычисления целой части на 2.

Проверка:

$$1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + 0 \cdot \frac{1}{8} + 0 \cdot \frac{1}{16} + 0 \cdot \frac{1}{32} + 1 \cdot \frac{1}{64} + 1 \cdot \frac{1}{128} + 1 \cdot \frac{1}{256} = 0.77734375$$

Погрешность $\varepsilon = (0.78 - 0.77734375) = 0.00265625$

Это же значение можно прочесть и при выводе результата: остаток после последнего шага равен 0.68, номер последнего разряда равен 8, тогда погрешность равна: $\varepsilon = 0.68 \cdot 2^{-8} = 0.00265625$.

Число значащих разрядов получилось равным числу разрядов слова.

Пример2. Перевод правильной десятичной дроби к такой, которая представима в новой системе счисления без погрешности. Переводим 0.625 в двоичную систему. $0.625 \cdot 2 = 1.25$, остаток $0.25 \cdot 2 = 0.5$, остаток $0.5 \cdot 2 = 1$, остаток равен 0. Ответ имеет вид: (0.10100000), число значащих разрядов равно 3, погрешность равна 0.

Проверка:

$$1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{4} + 1 \cdot \frac{1}{8} = 0.625$$

На этом этап второй завершен.

4.2.3 Этап 3. «Составление алгоритма и представление его в схематичной форме»

При разборе алгоритма полезно следить за операциями по примеру (этап 2).

Дано: создать функцию, переводящую заданную десятичную правильную дробь в новую систему счисления с заданным основанием. Число позиций для представления результата задано.

Идентификаторы аргументов:

- *osnovanie* – основание новой системы счисления;
- *n_bit* – число позиций для представления результата;
- *fraction* – правильная десятичная дробь, подлежащая переводу в новую систему счисления.

Получить: функция $Z(\text{osnovanie}, \text{n_bit}, \text{fraction})$ возвращает структуру $\{N_r, x, \varepsilon\}$.

Комментарии:

- N_r – число значащих разрядов (позиций) в слове длиной *n_bit*;
- x – матрица-строка с числом элементов, равным *n_bit*. Элемент $x_j \cdot (\text{osnovanie})^{-(j+1)}$ входит в формулу расчета значения дроби; $j=0, \dots, \text{n_bit} - 1$ (нумерация элементов матрицы – с нуля, как это принято в Mathcad);
- ε – погрешность представления правильной десятичной дроби *fraction* в системе счисления с основанием *osnovanie* в слове с числом позиций *n_bit*.

Для ОБРАЩЕНИЯ функции $Z(\text{osnovanie}, \text{n_bit}, \text{fraction})$
 ЗАДАТЬ числовые значения аргументов (*osnovanie*, *n_bit*, *fraction*);
 АЛГОРИТМ вычисления функции $Z(\text{osnovanie}, \text{n_bit}, \text{fraction})$;
 НАЧАЛО расчета значения функции $Z(\text{osnovanie}, \text{n_bit}, \text{fraction})$.
 ПРИСВОИТЬ начальное значение (ноль) счетчику значащих разрядов N_r .

ПРИСВОИТЬ начальное значение (fraction) переменной «с», обозначающий остаток;

ЦИКЛ ПОКА значение числа значащих разрядов $N_r < n_bit$ ВЫПОЛНЯТЬ
 ВЫЧИСЛИТЬ $c \cdot osnovanie$;

ПРИСВОИТЬ результат вычисления переменной d ;

ПРИСВОИТЬ элементу массива-строки x_0, N_r [целую часть d];

ВЫЧИСЛИТЬ $ostatok$ как РАЗНОСТЬ между d и [целую часть d];

ПРИСВОИТЬ переменной c значение переменной $ostatok$;

УВЕЛИЧИТЬ на 1 значение счетчика значащих разрядов N_r ;

ЕСЛИ $ostatok = 0$ ТО ПРЕРЫВАТЬ ЦИКЛ и ПЕРЕЙТИ к КОМАНДЕ

за скобкой КОНЕЦ ЦИКЛА

ПРИСВОИТЬ новое значение $ostatok$ переменной c ;

КОНЕЦ ЦИКЛА ПОКА...

ЕСЛИ число значащих разрядов N_r меньше заданного числа n_bit разрядов
 ТО

Комментарий: разряды, оставшиеся свободными, заполнить нулями.

ЦИКЛ ДЛЯ КАЖДОГО $j = N_r + 1, \dots, n_bit - 1$ ВЫПОЛНЯТЬ

ПРИСВОИТЬ нуль j – му элементу массива-строки $x_{0,j}$;

КОНЕЦ ЦИКЛА ДЛЯ КАЖДОГО...

КОНЕЦ ЕСЛИ

ПРИСВОИТЬ переменной ε значение $ostatok \cdot osnovanie^{-n_bit}$

ПРИСВОИТЬ возвращаемые значения структуре $\{N_r \cdot \varepsilon\}$

КОНЕЦ АЛГОРИТМА

На этом этап третий завершен.

4.2.4 Этап 4. «Расчет контрольных примеров применительно к алгоритму»

В качестве контрольных примеров используем примеры, рассчитанные на этапе 2.

Пусть число разрядов $n_bit = 8$, основание новой системы счисления $osnovanie = 2$.

Пример 1 (для проверки правильности перевода в новую систему счисления с погрешностью): перевести десятичную правильную дробь 0.78 в двоичную систему счисления.

ФУНКЦИЯ $Z(2,8,0.78)$ должна вернуть:

$Z(2,8,0.78)_{0,0} = 8$ (число значащих разрядов)

$Z(2,8,0.78)_{0,1} = (11000111)$ (дробная часть числа 0.78 в двоичной системе счисления)

$Z(2,8,0.78)_{0,2} = 0.00265625$ (погрешность)

Пример 2 (для проверки правильности перевода в новую систему счисления без погрешности): перевести десятичную правильную дробь 0.65 в двоичную систему счисления.

ФУНКЦИЯ $Z(2,8,0.625)$ должна вернуть:

$Z(2,8,0.625)_{0,0} = 3$ (число значащих разрядов)

$Z(2,8,0.625)_{0,1} = (10100000)$ (дробная часть числа 0.625 в двоичной системе счисления)

$Z(2,8,0.625)_{0,2} = 0$ (погрешность)

На этом этап четвертый завершен.

4.2.5 Этап 5. «Составление программы, отладка на контрольных примерах»

Составление программы будем проводить шаг за шагом. Удобно следить за реализацией алгоритма по этапу 3.

Шаг 1. Вводим идентификатор функции, в скобках перечисляем аргументы, нажимаем знак присваивания (вводится двоеточием). Получается так:

```
Z(osnovanie ,n_bit ,fraction) := ▮
```

Шаг 2. На панели «Программирование» (если ее нет на экране – вызовите из Меню *View – Toolbars - Programming*) найдем иконку *Add Line* («добавить линию») и щелкнем по ней левой кнопкой мыши. Введется операторная скобка «начало программы – конец программы». Получится так:

```
Z(osnovanie ,n_bit ,fraction) := | ▮
                               | ▮
```

Шаг 3. Установим указатель мыши на верхний слот и щелкнем левой кнопкой. Получится так:

```
Z(osnovanie ,n_bit ,fraction) := | ▮
                               | ▮
```

Шаг 4. Щелчком левой кнопки мыши вводим с панели программирования знак присваивания. Получится так:

```
Z(osnovanie ,n_bit ,fraction) := | ▮ ← ▮
                               | ▮
```

Шаг 5. Пользуясь мышью, присваиваем начальное значение (ноль) счетчику числа значащих разрядов N_r (он же будет использоваться и как счетчик циклов). Необходимо учесть, что r – не индекс. Этот параметр вводится косметической точкой (N точка r). Получится так:

```
Z(osnovanie ,n_bit ,fraction) := | N_r ← 0
                               | ▮
```

Шаг 6. Аналогичными действиями присваиваем исходное значение переменной c , которую согласно алгоритму будем в цикле умножать на основание системы счисления. Исходное значение равно той десятичной дроби, которую мы переводим в новую систему счисления. Получится так:

```
Z(osnovanie ,n_bit ,fraction) := | N_r ← 0
                               | c ← fraction
```


Шаг 7. Длина вертикальной линии (операторной скобки), установленная по умолчанию, исчерпана. Нужно ее увеличить. Предварительная операция: охватить синим контуром нижнюю строку программы. Получится так:

$$Z(\text{osnovanie}, n_bit, \text{fraction}) := \left| \begin{array}{l} N_r \leftarrow 0 \\ \underline{c \leftarrow \text{fraction}} \end{array} \right.$$

Шаг 8. Теперь щелчком левой кнопки мыши по иконке *Add Line* на панели «Программирование» удлиним операторную скобку. Получится так:

$$Z(\text{osnovanie}, n_bit, \text{fraction}) := \left| \begin{array}{l} N_r \leftarrow 0 \\ c \leftarrow \text{fraction} \\ \blacksquare \end{array} \right.$$

Шаг 9. Оформим условие цикла: делать вычисления, пока не будут исчерпаны разряды слова (записаны в составе аргументов функции, *n_bit*). Для этого щелкнем левой кнопкой мыши по оператору *while* на панели «Программирование» (НЕЛЬЗЯ набирать от руки!). Получится так:

$$Z(\text{osnovanie}, n_bit, \text{fraction}) := \left| \begin{array}{l} N_r \leftarrow 0 \\ c \leftarrow \text{fraction} \\ \text{while } \blacksquare \\ \blacksquare \end{array} \right.$$

Шаг 10. Введем условие вхождения в цикл. Получится так:

$$Z(\text{osnovanie}, n_bit, \text{fraction}) := \left| \begin{array}{l} N_r \leftarrow 0 \\ c \leftarrow \text{fraction} \\ \text{while } N_r < \underline{n_bit} \\ \blacksquare \end{array} \right.$$

Шаг 11. образуем операторную скобку операторы цикла *while* (щелчком в слоте оператора *while* введем скобку с панели «Программирование», иконка *Add Line*), установим указатель в верхнем слоте внутри операторной скобки цикла. Получится так:

$$Z(\text{osnovanie}, n_bit, \text{fraction}) := \left| \begin{array}{l} N_r \leftarrow 0 \\ c \leftarrow \text{fraction} \\ \text{while } N_r < n_bit \\ \left| \blacksquare \right. \\ \blacksquare \end{array} \right.$$

Шаг 12. Согласно алгоритму перевода десятичной дроби в другую систему счисления присвоим переменной *d* значение произведения дроби на основание, выделим целую часть произведения и присвоим его N_r -му элементу строки *x*, т.е. элементу с индексом $x_{0,s}$ (напоминание: нумерация

элементов массива в Mathcad начинается с нуля, а не с единицы). При первом входе в цикл значение индекса N_r равно 0.

Подготовимся к удлинению операторной скобки (охватим нижнюю строку программы целиком синим контуром). Не забудьте, что r – не индекс (набор через косметическую точку). Получится так:

$$Z(\text{osnovanie}, n_bit, \text{fraction}) := \left| \begin{array}{l} N_r \leftarrow 0 \\ c \leftarrow \text{fraction} \\ \text{while } N_r < n_bit \\ \quad \left| \begin{array}{l} d \leftarrow c \cdot \text{osnovanie} \\ x_{0, N_r} \leftarrow \text{floor}(d) \end{array} \right. \end{array} \right.$$

Шаг 13. Чтобы каждый раз не тратить время на удлинение операторной скобки, «нащелкаем» с помощью иконки *Add Line* (панель «Программирование») сразу несколько слотов. Не бойтесь образовывать лишние слоты – их легко будет удалить. Получится так:

$$Z(\text{osnovanie}, n_bit, \text{fraction}) := \left| \begin{array}{l} N_r \leftarrow 0 \\ c \leftarrow \text{fraction} \\ \text{while } N_r < n_bit \\ \quad \left| \begin{array}{l} d \leftarrow c \cdot \text{osnovanie} \\ x_{0, N_r} \leftarrow \text{floor}(d) \\ \quad \cdot \\ \quad \cdot \\ \quad \cdot \\ \quad \cdot \end{array} \right. \end{array} \right.$$

Шаг 14. Согласно алгоритму следует вычислить остаток (разность между результатом умножения дроби и его целой частью, присваивается переменной *ostatok*) и подготовимся к новому циклу (присвоим цикловой переменной значение $N_r + 1$). Получится так:

```

Z(osnovanie ,n_bit ,fraction) := | Nr ← 0
                                | c ← fraction
                                | while Nr < n_bit
                                |   | d ← c·osnovanie
                                |   | x0,Nr ← floor(d)
                                |   | Nr ← Nr + 1
                                |   | █
                                |   | █
                                |   | █

```

Шаг 15. Оформление досрочного выхода из цикла. Для этого введем щелчком кнопки мыши оператор условия *if* (иконка *if* на панели «Программирование»). Получится так:

```

Z(osnovanie ,n_bit ,fraction) := | Nr ← 0
                                | c ← fraction
                                | while Nr < n_bit
                                |   | d ← c·osnovanie
                                |   | x0,Nr ← floor(d)
                                |   | Nr ← Nr + 1
                                |   | █ if █
                                |   | █
                                |   | █

```

Шаг 16. Введем условие досрочного прекращения цикла: ЕСЛИ *ostatok=0*. Знак равенства носит название жирное равно и ставится при использовании сочетания клавиш <Ctrl>+<=>. Условие вводится справа от оператора *if* – вот так:

```

Z(osnovanie ,n_bit ,fraction) :=
  Nr ← 0
  c ← fraction
  while Nr < n_bit
    d ← c·osnovanie
    x0,Nr ← floor(d)
    Nr ← Nr + 1
    ■ if ostatok = 0
    ■
    ■

```

Шаг 17. Показываем, что должно быть сделано, если $ostatok=0$. В данном случае должен быть приведен цикл (оператор *break*, вводится щелчком по соответствующей иконке на панели «Программирование»). Получится так:

```

Z(osnovanie ,n_bit ,fraction) :=
  Nr ← 0
  c ← fraction
  while Nr < n_bit
    d ← c·osnovanie
    x0,Nr ← floor(d)
    Nr ← Nr + 1
    break if ostatok = 0
    ■
    ■

```

Шаг 18. Если досрочный выход из цикла не произошел, нужно присвоить новое значение остатка *ostatok* переменной *c*. Получится так:

```

Z(osnovanie ,n_bit ,fraction) := | Nr ← 0
                                | c ← fraction
                                | while Nr < n_bit
                                |   | d ← c·osnovanie
                                |   | x0,Nr ← floor(d)
                                |   | Nr ← Nr + 1
                                |   | break if ostatok = 0
                                |   | c ← ostatok
                                |   |

```

Шаг 19. Далее текст программы должен записываться вне операторной скобки *while*. Чтобы выйти из нее, охватите синим контуром часть программы до оператора цикла *while* включительно. Угол контура должен быть справа (иначе слот для ввода продолжения программы появится выше *while*). Переброску угла синего контура удобно делать клавишей <Insert>, а увеличение охвата – клавишей <Пробел>. Получится так:

```

Z(osnovanie ,n_bit ,fraction) := | Nr ← 0
                                | c ← fraction
                                | while Nr < n_bit
                                |   | d ← c·osnovanie
                                |   | x0,Nr ← floor(d)
                                |   | Nr ← Nr + 1
                                |   | break if ostatok = 0
                                |   | c ← ostatok

```

Шаг 20. Теперь, если щелкать левой кнопкой мыши по иконке *Add Line*, то появятся слоты для продолжения программы вне цикла. Заготовим несколько таких слотов. Получится так:

```

Z(osnovanie ,n_bit ,fraction) :=
  Nr ← 0
  c ← fraction
  while Nr < n_bit
    d ← c·osnovanie
    x0,Nr ← floor(d)
    Nr ← Nr + 1
    break if ostatok = 0
    c ← ostatok
  ■
  ■
  ■

```

Шаг 21. Теперь нужно циклически присвоить нули тем разрядам, которые остались незаполненными (если цикл *while* оказался прерванным досрочно, т.е. если для представления дроби *fraction* достаточно меньшего, чем *n_bit* числа разрядов). Вводим условный оператор и справа указываем условие, при котором следует дописывать незаполненные позиции. Получится так:

```

Z(osnovanie ,n_bit ,fraction) :=
  Nr ← 0
  c ← fraction
  while Nr < n_bit
    d ← c·osnovanie
    x0,Nr ← floor(d)
    Nr ← Nr + 1
    break if ostatok = 0
    c ← ostatok
  ■ if Nr < n_bit
  ■
  ■

```

Шаг 22. Показываем, что должно быть сделано, если $N_r < n_bit$. Согласно алгоритму следует циклически присваивать нулевые значения разрядам числа, начиная $N_r + 1$ -го до последнего разряда (его номер $n_bit - 1$, учитывая Mathcad-нумерацию с нуля, а не с единицы). Оператор *for* введем с соответствующей иконки щелчком мыши в слот слева от оператора *if*.

Внутреннюю операторную скобку цикла *for* Mathcad оформит сам. Знак «Две точки» ставится с помощью клавиши точка с запятой. Получится так:

```
Z(osnovanie ,n_bit ,fraction) := | N_r ← 0
                                | c ← fraction
                                | while N_r < n_bit
                                |   | d ← c·osnovanie
                                |   | x_{0,N_r} ← floor(d)
                                |   | N_r ← N_r + 1
                                |   | break if ostatok = 0
                                |   | c ← ostatok
                                | for j ∈ N_r + 1 .. n_bit - 1 if N_r < n_bit
                                |   x_{0,j} ← 0
                                |   ■
                                |   ■
```

В этом случае в цикле – только один оператор, поэтому Mathcad не вводит линию-границу операторной скобки (считает, что и так поймем).

Шаг 23. В ходе вычисления получены данные для определения погрешности (если данная дробь *fraction* не может быть представлена в новой системе точно), см. пример 1. Присваиваем значение погрешности переменной ε , входящей в состав структуры, возвращаемой функцией $Z(\text{osnovanie}, n_bit, \text{fraction})$. Показатель степени вводится клавишами <Shift>+<^>. Получится так:

```

Z(osnovanie ,n_bit ,fraction) :=
  Nr ← 0
  c ← fraction
  while Nr < n_bit
    d ← c·osnovanie
    x0,Nr ← floor(d)
    Nr ← Nr + 1
    break if ostatok = 0
    c ← ostatok
  for j ∈ Nr + 1 .. n_bit - 1 if Nr < n_bit
    x0,j ← 0
  ε ← ostatok·osnovanie-n_bit
  ▮

```

Шаг 24 (последний). Все составляющие структуры (N_r, x, ε) , которые должны возвращаться функцией $Z(\text{osnovanie}, n_bit, \text{fraction})$, получены. После окончания цикла или после его досрочного прерывания значение переменной N_r окажется равным числу значащих разрядов в слове длины n_bit . Значащие разряды – такие, за которыми:

- либо следуют все нули, если заданная дробь *fraction* может быть представлена без потерь точности меньшим, чем n_bit , числом разрядов;
- либо число разрядов окажется исчерпанным с потерей точности представления дроби *fraction*. В этом случае число значащих разрядов равно n_bit .

Рассчитаны также элементы матрицы-строки x с результатом перевода дроби в новую систему счисления и погрешность ε , возникающая при переводе. Осталось записать структуру, возвращаемую функцией.

Оформляем перечень значений, возвращаемых функцией $Z(\text{osnovanie}, n_bit, \text{fraction})$. Согласно заданию функция должна возвращать:

- число значащих разрядов (в обозначениях программы N_r);
- массив значений из 8-ми элементов (матрица-строка x), содержащий перечень значений в разрядах числа;
- погрешность ε от представления заданной дроби *fraction* в новой системе счисления с основанием *osnovanie*.

В данном случае нужно, чтобы функция возвратила разнородную структуру данных (в ней и простые переменные, и массив). Такая структура данных так и называется – структура (в отличие от простой переменной или массива).

Для ввода перечня возвращаемых значений в слот программы используем шаблон матрицы. Можно использовать меню *Insert-Matrix*, но лучше запомнить сочетание клавиш <Ctrl>+<M> (мнемоника M от matrix). Указываем число строк (rows) 1, число столбцов (columns) 3 по числу выводимых элементов структуры. В каждый из слотов матрицы вводим идентификаторы переменных. Если остались лишние слоты – удалим их. Получится так:

$$Z(\text{osnovanie}, n_bit, \text{fraction}) := \left(\begin{array}{l} N_r \leftarrow 0 \\ c \leftarrow \text{fraction} \\ \text{while } N_r < n_bit \\ \quad \left(\begin{array}{l} d \leftarrow c \cdot \text{osnovanie} \\ x_{0, N_r} \leftarrow \text{floor}(d) \\ N_r \leftarrow N_r + 1 \\ \text{break if } \text{ostatok} = 0 \\ c \leftarrow \text{ostatok} \end{array} \right) \\ \text{for } j \in N_r + 1 .. n_bit - 1 \text{ if } N_r < n_bit \\ \quad x_{0, j} \leftarrow 0 \\ \varepsilon \leftarrow \text{ostatok} \cdot \text{osnovanie}^{-n_bit} \\ (N_r \quad x \quad \varepsilon) \end{array} \right)$$

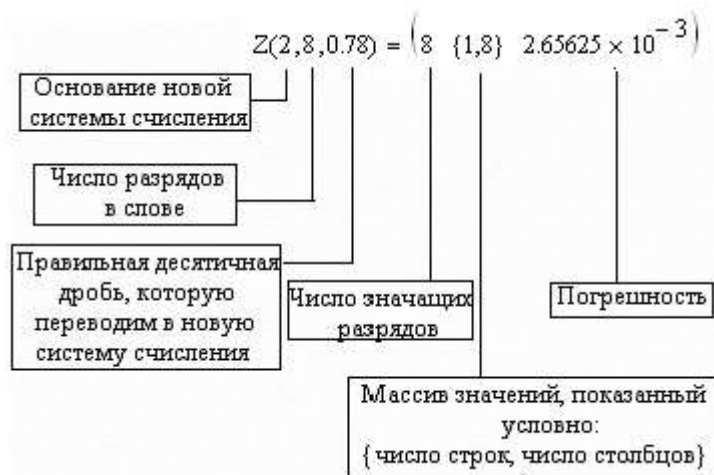
Разработка программы завершена. Мы получили функцию в форме матрицы-строки, первый элемент которой – число значащих разрядов, второй – строка представления заданной дроби в новой системе счисления, третий – погрешность этого представления.

4.2.6 «Тестирование программы на контрольных примерах»

Сначала посмотрим, что возвратит функция при обращении к ней. Набираем:

$$Z(2, 8, 0.78) = \blacksquare$$

Выведется структура:



Чтобы прочесть позиции в массиве $\{1,8\}$, нужно обратиться к соответствующему элементу структуры. Сделаем это в ходе тестирования программы, для чего используем заранее рассчитанные примеры. Для удобства чтения скопируем результаты, которые должна показать программа.

Пример 1 (для проверки правильности перевода в новую систему счисления с погрешностью): перевести десятичную правильную дробь 0.78 в двоичную систему счисления:

ФУНКЦИЯ $Z(2,8,0.78)_{0,0} = 8$ (число значащих разрядов)

$Z(2,8,0.78)_{0,1} = (11000111)$ (дробная часть числа 0.78 в двоичной системе счисления)

$Z(2,8,0.78)_{0,2} = (0.00265625)$ (погрешность)

Тестируем программу. Набираем:

$Z(2,8,0.78)_{0,0} =$

$Z(2,8,0.78)_{0,1} =$ Если MathCAD выведет: $(1\ 1\ 0\ 0\ 0\ 1\ 1\ 1)$

$Z(2,8,0.78)_{0,2} =$ 2.65625×10^{-3}

то тест по примеру 1 успешно пройден.
В противном случае ищите ошибки.

Пример 2 (для проверки правильности перевода в новую систему счисления без погрешности): перевести десятичную правильную дробь 0.625 в двоичную систему счисления.

ФУНКЦИЯ $Z(2,8,0.625)$ должна вернуть:

$Z(2,8,0.625)_{0,0} = 3$ (число значащих разрядов)

$Z(2,8,0.625)_{0,1} = (10100000)$ (дробная часть числа 0.65 в двоичной системе счисления)

$Z(2,8,0.625)_{0,2} = 0$ (погрешность)

Тестируем программу. Набираем:

$$\begin{array}{l}
 Z(2,8,0.625)_{0,0} = \\
 Z(2,8,0.625)_{0,1} = \text{Если MathCAD выведет: } \begin{array}{l} \nearrow 3 \\ \text{---} (1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0) \\ \searrow 0 \end{array} \\
 Z(2,8,0.625)_{0,2} =
 \end{array}$$

то тест по примеру 2 успешно пройден.
В противном случае ищите ошибки.

После успешного прохождения тестов этап 5 завершен.
Разработка программы закончена.

4.2.7 Эксперименты с программой

Любопытно, что программа позволяет переводить десятичные дроби в системы счисления с нецелыми основаниями (не только положительными, но и отрицательными). И сама правильная должна десятичная дробь не обязательно должна быть положительной. Почти так всегда и бывает: с помощью программы почти всегда удается сделать больше, чем предусмотрено заданием.

Например, переведем десятичную дробь 0.625 в систему счисления с основанием π . Получим:

$$Z(\pi, 8, 0.625)_{0,1} = (1 \ 3 \ 0 \ 0 \ 0 \ 2 \ 1 \ 2)$$

Проверка:

$$1 \cdot \frac{1}{\pi} + 3 \cdot \frac{1}{\pi^2} + 0 \cdot \frac{1}{\pi^3} + 0 \cdot \frac{1}{\pi^4} + 0 \cdot \frac{1}{\pi^5} + 2 \cdot \frac{1}{\pi^6} + 1 \cdot \frac{1}{\pi^7} + 2 \cdot \frac{1}{\pi^8} = 0.6249$$

Можно записать проверку компактнее:

$$\sum_{j=1}^8 \pi^{-j} \cdot [(Z(\pi, 8, 0.625)_{0,1})^{(j-1)}]_0 = 0.6249 \quad \text{Почти } 0.625$$

Так же можно проверить и провести еще какие-либо эксперименты, например: попытаться перевести отрицательную дробь, неправильную дробь, использовать отрицательное основание системы счисления.

Лабораторная работа № 5 «Программирование в Mathcad»

Цель работы: получить навыки работы в Mathcad при решении уравнений.

Задание на лабораторную работу:

Изучить материал, представленный в разделе 4 (пункт 4.1, 4.2) соответственно.

1. Создайте программный блок, вычисляющего факториал при помощи цикла *for*.

2. При помощи алгоритма Евклида определить наибольший общий делитель. Для реализации алгоритма использовать цикл с ключевым словом *while*.

3. Составьте программу-функцию, вычисляющую n коэффициентов разложения функции $f(x)$, заданной на отрезке $[-L, L]$, в ряд Фурье – т.е. параметрами являются f , n , L , причем в результате должна выдаваться матрица значений: нулевая по счету строка с коэффициентами A_n , а первая – с коэффициентами B_n . Для выделения этих коэффициентов используйте транспонированную матрицу и из нее будет выбираться нулевой столбец для коэффициентов A_n и первый – для B_n .

4. Разработайте функцию, генерирующую импульсы прямоугольной формы (рис. 4.2). Прямоугольный импульс характеризуется четверкой параметров: {продолжительность импульса; верхний уровень (максимум); нижний уровень (минимум); начальное значение}. Эти параметры, показанные на рисунке 4.2, должны принимать значения, указанные пользователем. Им же задается максимальная продолжительность серии импульсов (на рисунке 4.2. – максимальное число тактов отсчета дискретного времени).

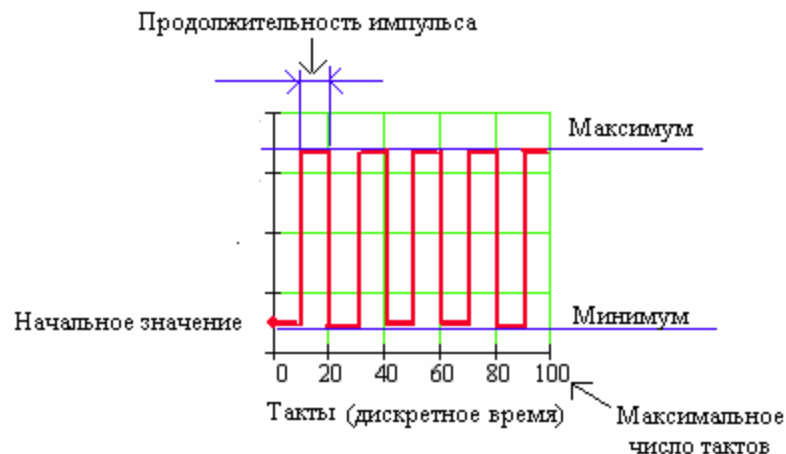


Рисунок 4.2 – Иллюстрация к содержательной постановке задачи ДАНО:

X_{\max} – максимальное значение импульса (действительное число);

X_{\min} – минимальное значение импульса (действительное число);

Δs – продолжительность импульса (число тактов дискретного времени, в течение которого значение импульса не изменяется (целое число));

N – число тактов дискретного времени, для которого должна быть получена серия импульсов;

X_{start} – начальное состояние (стартовое значение), задается для первого импульса серии.

ПОЛУЧИТЬ: функция $X(X_{max}, X_{min}, \Delta s, N, X_{start})$, возвращает вектор X с элементами X_s , показывающими значение импульса в s -м такте, $s = 0, \dots, N$. Значение элемента вектора X_0 должно быть равно заданному аргументу функции X_{start} .

Помощь при выполнении задания.

Позиция 1. «Нахождение идеи алгоритма»

Зададим себе вопрос – чем отличается номер такта дискретного времени, в который происходит «переброска» импульса с минимума на максимум и наоборот, от других номеров тактов, в которых импульс не изменяет своего значения? Пусть, например, заданная пользователем продолжительность импульса составляет 11 тактов, а номер текущего такта 28. Ясно, что число 28 не удастся представить как целое число продолжительностей. Поэтому в текущем такте импульс будет иметь то же значение, что и в предшествующем, 27-м.

Если номер текущего такта равен, например, 44, то (поскольку в 44 такта укладывается целое число 11-тактных продолжительностей) нужно изменять импульс с выполнением следующего правила:

1. если на предшествующем такте импульс имел максимальное значение, то его значение нужно заменить на минимальное;
2. если же на предшествующем такте импульс имел минимальное значение, то его значение нужно заменить на максимальное.

Отсюда следует, что можно использовать такой «ключ» переброски импульса с минимума на максимум и наоборот:

Пусть имеет место s -й такт дискретного времени. Тогда, если $\frac{s}{\Delta s}$ – целое число, значение импульса нужно «перебросить», а в противном случае – сохранить прежнее значение.

Позиция 1 закончена.

Позиция 2. «Подбор встроенных Mathcad-функций».

Полный список функций Mathcad можно просмотреть с помощью иконки $f(x)$ на строке *Standard* (если эта строка на мониторе не показана – включите ее из меню *View-Toolbars-Standard*). Подходящие функции – округление до целого числа сверху $ceil()$ («до потолка») или снизу $floor()$ («от пола»). Обе они могут быть использованы для определения момента переброски значения импульса. Остановимся на функции округления снизу $floor()$. Ее действие легко понять на примере:

$$\text{floor}\left(\frac{18}{11}\right) = 1 \quad \text{floor}\left(\frac{43}{11}\right) = 3 \quad \text{floor}\left(\frac{44}{11}\right) = 4 \quad \text{floor}\left(\frac{4}{11}\right) = 0$$

Мы видим, что если номер такта кратен Δs , то $\frac{s}{\Delta s} = \text{floor}\left(\frac{s}{\Delta s}\right)$, например, $\text{floor}\left(\frac{44}{11}\right) = 4$. Для номеров тактов, не кратных Δs , $\text{floor}\left(\frac{s}{\Delta s}\right) < \frac{s}{\Delta s}$, например, $\text{floor}\left(\frac{43}{11}\right) = \frac{44}{11} = 3 < \frac{43}{11}$.

Это свойство можно использовать как ключ для изменения импульса с минимума на максимум и наоборот.

Позиция 2 закончена.

Позиция 3. «К разработке алгоритма»

ЕСЛИ $\frac{s}{\Delta s} = \text{floor}\left(\frac{s}{\Delta s}\right)$ И значение импульса X_{s-1} в $(s-1)$ -м такте было меньше или равно (\leq) минимальному значению X_{min} ТО присвоить импульсу X_s в s -м такте максимальное значение;

ЕСЛИ $\frac{s}{\Delta s} = \text{floor}\left(\frac{s}{\Delta s}\right)$ И значение импульса X_{s-1} в $(s-1)$ -м такте было больше или равно (\geq) максимальному значению X_{max} ТО присвоить импульсу X_s в s -м такте минимальное значение;

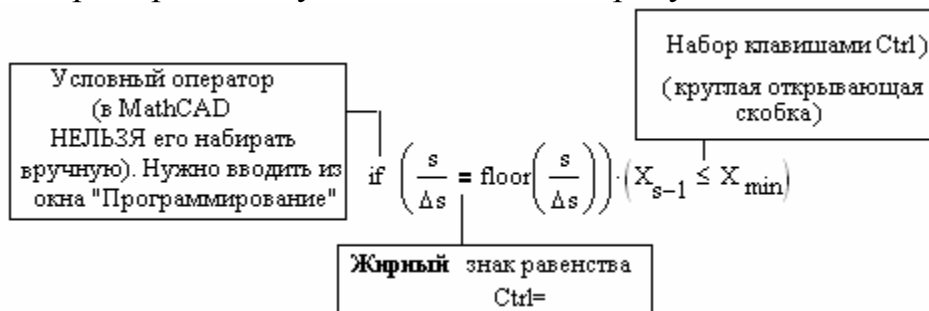
В ОСТАЛЬНЫХ СЛУЧАЯХ присвоить импульсу X_s в s -м такте то же значение X_{s-1} , что и в предыдущем, $(s-1)$ -м такте.

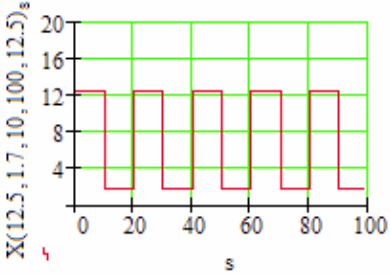
КОНЕЦ ЕСЛИ

Позиция 3 закончена.

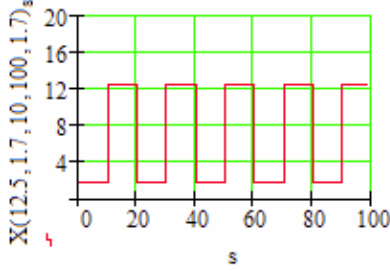
Позиция 4. «Некоторые детали»

1. Запись условий с логическим оператором И делается с использованием знака умножения (*).
2. В записи условий нужно использовать жирный знак равенства.
3. Пример записи условия показан на рисунке 4.3.

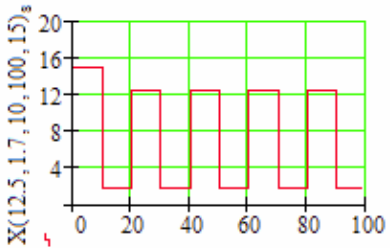




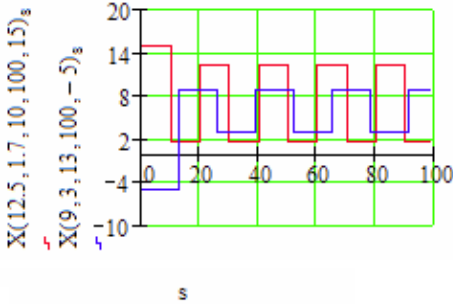
а) Импульсы стартуют с максимального значения



б) Импульсы стартуют с минимального значения



в) Импульсы стартуют со значения, несовпадающего ни с минимальным, ни с максимальным



г) Две серии, показанные на одном графике. Параметры импульсов различны

Рисунок 4.4 – Вывод результата работы программы на график
Позиция 5 закончена.
Помощь при выполнении задания закончена.

Составьте отчет по лабораторной работе №5 (см. Приложение А –
Пример оформления отчета). Сделайте соответствующие выводы.

5 Символьные преобразования в Mathcad

5.1 Введение. Сведения о символьных преобразованиях в Mathcad

Mathcad содержит замечательные средства работы не только с числами, но и с формулами (т.е. может оказать помощь в упрощении сложных выражений, приведении подобных слагаемых, нахождении пределов, интегрировании, дифференцировании, решении уравнений в виде формулы зависимости от коэффициентов и во многих других вопросах...). Символьная математика, встроенная в Mathcad, заимствована из известного пакета программ для символьных вычислений Maple.

Операции, относящиеся к работе символьного процессора, содержатся в подменю *Symbolics* (Символика) главного меню (рис. 5.1).

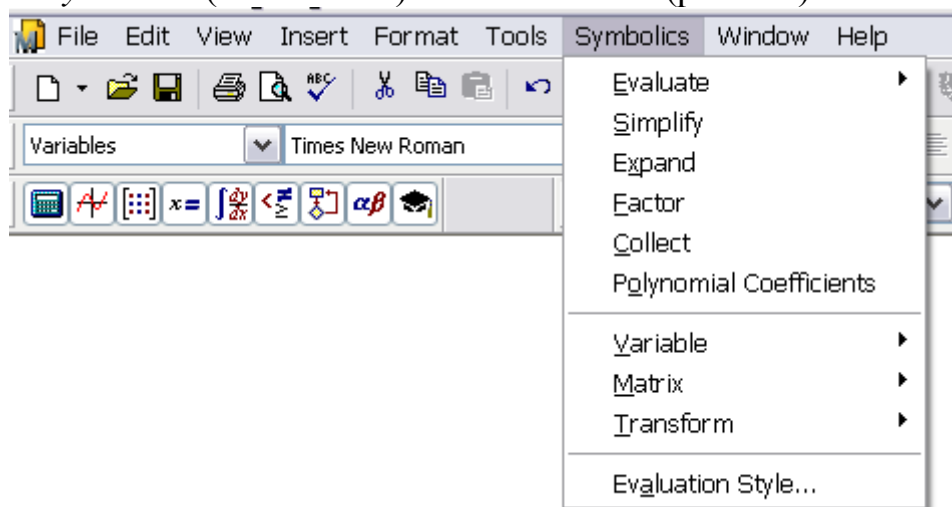


Рисунок 5.1 – Символьный процессор подменю *Symbolics* главного меню

Чтобы символьные операции выполнялись, процессору необходимо указать, над каким выражением эти операции должны производиться, т.е. надо выделить выражение. Для ряда операций следует не только указать выражение, к которому они относятся, но и наметить переменную, относительно которой выполняется та или иная символьная операция.

Само выражение в таком случае не выделяется: ведь и так ясно, что если маркер ввода выделяет переменную какого-либо выражения, то это выражение уже отмечено наличием в нем выделяемой переменной.

Следует отметить некоторые особенности при работе с командами меню *Symbolics*.

Для символьных вычислений выражения необходимо указывать явно. Например, недопустимо вводить некоторую функцию пользователя $f(x)$ и пытаться найти ее производные или интеграл. Это существенное ограничение, о котором надо всегда помнить. Однако оно преодолимо при выполнении вычислений с помощью функций системы *SmartMath*, которая описывается в дальнейшем; главное в том, что для вывода символьных вычислений в этом случае используется оператор \rightarrow .

Иногда результат вычислений содержит встроенные в систему специальные математические функции; в этом случае результат помещается

в буфер обмена. Используя команду *Paste* или клавишу *F4*, можно вставить содержимое буфера обмена в документ как ТЕКСТ и проанализировать полученный результат.

К недостаткам работы с командами меню *Symbolics* следует отнести то, что это ручная работа, одношаговая. При дальнейшем использовании результатов символьных вычислений необходимо с помощью операций *Copy* и *Past* присвоить этот результат некоторой переменной или функции. Кроме того, при изменении формулы, которая подвергается символьному преобразованию, результат (даже при установленном Автоматическом режиме вычислений) не пересчитывается.

Если операция невыполнима – система выводит сообщение об ошибке или просто повторяет выделенное выражение (без изменений).

В качестве минуса при работе с командами меню *Symbolics* следует отметить не очень наглядную визуализацию процесса вычислений. Хотя при установке определенных параметров стиля эволюции выражений можно вывести некоторые комментарии.

Символьные операции разбиты на четыре характерных раздела. Первыми идут наиболее часто используемые операции. Они могут выполняться с выражениями, содержащими комплексные числа или имеющими решения в комплексном виде.

5.1.1 Операции с выделенными выражениями

К операциям с выделенными выражениями относятся следующие (табл. 5.1):

Таблица 5.1 – Операции с выделенными выражениями

Вид операции	Назначение операции
<i>Evaluate</i> (Вычислить)	Преобразовать выражение с выбором вида преобразований из подменю
<i>Simplify</i> (Упростить)	Упростить выделенное выражение с выполнением таких операций, как сокращение подобных слагаемых, приведение к общему знаменателю, использование основных тригонометрических тождеств и т. д.
<i>Expand</i> (Разложить по степеням)	Раскрыть выражение [например, для $(X+Y) \cdot (X-Y)$ получаем $X^2 - Y^2$]
<i>Factor</i> (Разложить на множители)	Разложить число или выражение на множители [например, $X^2 - Y^2$ даст $(X+Y)(X-Y)$]
<i>Collect</i> (Разложить по подвыражению)	Собрать слагаемые, подобные выделенному выражению, которое может быть отдельной переменной или функцией со своим аргументом (результатом будет выражение, полиномиальное относительно выбранного выражения)
<i>Polynomial Coefficients</i> (Полиномиальные коэффициенты)	Найти коэффициенты полинома по заданной переменной, приближающего выражение, в котором эта переменная использована

5.1.2 Операции с выделенными переменными

Следующая группа символьных операций выполняется с выражениями, требующими указания переменной, по отношению к которой выполняется операция. Для этого достаточно установить на переменной курсор ввода. Само выражение при этом не указывается отдельно, поскольку указание в нем на переменную является одновременно и указанием на само выражение. Если выражение содержит другие переменные, то они рассматриваются как константы.

К числу операций с выделенными переменными относятся (табл. 5.2):

Таблица 5.2 – Операции с выделенными переменными

Вид операции	Назначение операции
<i>Solve</i> (Решить относительно переменной)	Найти значения выделенной переменной, при которых содержащее ее выражение становится равным нулю (решить уравнение или неравенство относительно выделенной переменной)
<i>Substitute</i> (Заменить переменную)	Заменить указанную переменную содержимым буфера обмена
<i>Differentiate</i> (Дифференцировать переменную)	Дифференцировать все выражение, содержащее выделенную переменную, по отношению по к этой переменной (остальные переменные рассматриваются как константы)
<i>Integrate</i> (Интегрировать по переменной)	Интегрировать все выражение, содержащее выделенную переменную, по этой переменной
<i>Expand to Series...</i> (Разложить в ряд)	Найти несколько членов разложения выражения в ряд Тейлора относительно выделенной переменной
<i>Convert to Partial Fraction</i> (Разложить на элементарные дроби)	Разложить на элементарные дроби выражение, которое рассматривается как рациональная дробь относительно выделенной переменной

5.1.3 Операции с выделенными матрицами

Операции с выделенными матрицами представлены позицией подменю *Matrix* (матричные операции), которая имеет свое подменю со следующими операциями:

Transpose (транспонировать) – получить транспонированную матрицу;

Invert (обратить) – создать обратную матрицу;

Determinant (определитель) – вычислить детерминант (определитель) матрицы.

5.1.4 Операции преобразования

В позиции *Transform* содержится раздел операций преобразования, создающий подменю со следующими возможностями:

Fourier Transform (Преобразование Фурье) – выполнить прямое преобразование Фурье относительно выделенной переменной;

Inverse Fourier Transform (Обратное преобразование Фурье) – выполнить обратное преобразование Фурье относительно выделенной переменной;

Laplace Transform (Преобразование Лапласа) – выполнить прямое преобразование Лапласа относительно выделенной переменной (результат – функция от переменной s),

Inverse Laplace Transform (Обратное преобразование Лапласа) – выполнить обратное преобразование Лапласа относительно выделенной переменной (результат – функция от переменной t);

Z (Z-преобразование) – выполнить прямое Z-преобразование выражения относительно выделенной переменной (результат – функция от переменной z);

Inverse Z (Обратное Z-преобразование) – выполнить обратное Z-преобразование относительно выделенной переменной (результат – функция от переменной n).

5.1.5 Стил ь эволюции

В меню *Transform* находится команда *Evaluation Style...* (рис. 5.2), при исполнении которой на экран выводится диалоговое окно, позволяющее установить вид вывода результатов при работе с символьным процессором Mathcad.

В группе переключателей *Show evaluation steps* (стиль эволюции) – можно задать вывод результата символьной операции под основным выражением (со вставкой между исходным выражением и результатом пустой строки и без нее) или рядом с ним. Кроме того можно установить опции *Show Comments* для вставки комментариев и задать вывод результирующего выражения вместо исходного *Evaluate in Place*.

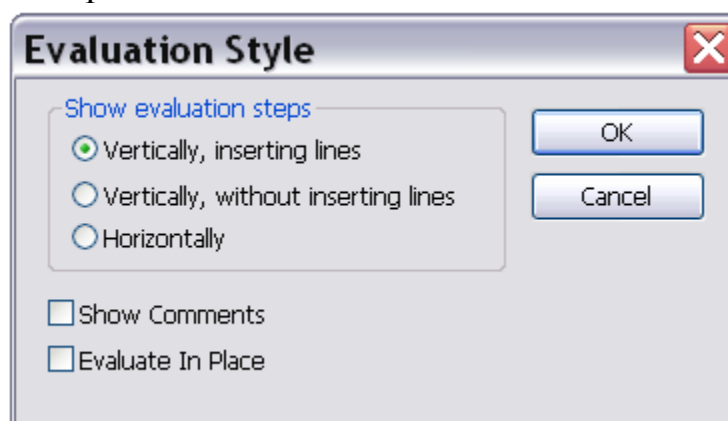


Рисунок 5.2 – Команда *Evaluation Style...* в меню *Transform*

5.2 Осваиваем операции символьной математики

Операции символьной математики проводятся из Меню *Symbolics* (рис. 5.3). Освоим ряд его позиций.

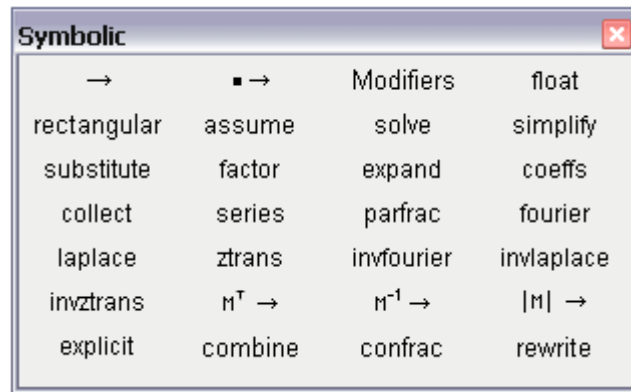


Рисунок 5.3 – Окно символьной математики в Mathcad

5.2.1 Опция Evaluate-Symbolically

Эта операция содержит подменю со следующими командами:

- *Evaluate Symbolically* <Shift>+<F9> (вычислить в символах) – выполнить символьное вычисление выражения;
- *Floating Point Evaluation...* (с плавающей точкой) – выполнить арифметические операции в выражении с результатом в форме числа с плавающей точкой;
- *Complex Evaluation* (в комплексном виде) – выполнить преобразование с представлением в комплексном виде.

Опция *Evaluate-Symbolically* позволяет сделать с формулой то, что Вы сами обычно бы сделали для ее приведения к «красивой форме». Обратите внимание: понятие «красивая формула» – совершенно неформальное. Но, поскольку символьную математику разрабатывали профессиональные научные работники, в понятие «красота формулы» они вложили свои, привычные представления. Например: все подобные слагаемые должны быть приведены; формула должна быть как можно короче; если есть слагаемые с дробными коэффициентами, не представимыми точно десятичной дробью, то ответ нужно представить с коэффициентами в форме дробей и т.п. Поэтому чаще всего (но не всегда!) результат использования опции *Evaluate-Symbolically* будет таким же, как Вы сами бы его получили (если бы умели считать быстро и без ошибок).

Команда *Evaluate-Symbolically* тут наиболее важная. Назначение других команд очевидно: они нужны, если результат требуется получить в форме комплексного или действительного числа. К примеру, если вы хотите вместо числа p получить $3.141\dots$, используйте команду *Floating Point Evaluation*. В режиме символьных вычислений результат может превосходить машинную бесконечность системы. При этом число точных значащих цифр результата практически не ограничено (или, точнее говоря, зависит от емкости ОЗУ).

Символьная операция *Evaluate Symbolically* <Shift>+<F9> (Вычислить) обеспечивает работу с математическими выражениями, содержащими встроенные в систему функции и представленными в различном виде: полиномиальном, дробно-рациональном, в виде сумм и произведений, производных и интегралов и т. д. Операция стремится произвести все

возможные численные вычисления и представить выражение в наиболее простом виде. Она возможна над матрицами с символьными элементами. Производные и определенные интегралы, символьные значения которых вычисляются, должны быть представлены в своей естественной форме.

Особо следует отметить возможность выполнения численных вычислений с повышенной точностью – 20 знаков после запятой. Для перехода в такой режим вычислений нужно числовые константы в вычисляемых объектах задавать с обязательным указанием десятичной точки, например 10.0 или 3.0, а не 10 или 3. Этот признак является указанием на проведение вычислений такого типа.

Пример:

Вычислить выражение:

$$4 \cdot x + 15 \cdot y^2 + \frac{4}{7} \cdot x + 19 \cdot x + 99 \cdot x^2 - \frac{1}{19} \cdot x$$

Для этого необходимо выбрать опцию *Evaluate-Symbolically* из меню *Symbolics* (или нажать <Shift>+<F9>). Получается:

$$99 \cdot x^2 + \frac{3128 \cdot x}{133} + 15 \cdot y^2$$

Это же выражение можно преобразовать и другим способом: необходимо ввести знак символьного равенства (\rightarrow) или сочетание клавиш: <Ctrl>+<Shift>+<точка> (клавиша точка в американском варианте называется period).

$$4 \cdot x + 15 \cdot y^2 + \frac{4}{7} \cdot x + 19 \cdot x + 99 \cdot x^2 - \frac{1}{19} \cdot x \rightarrow 99 \cdot x^2 + \frac{3128 \cdot x}{133} + 15 \cdot y^2$$

Преобразование таким образом позволяет добавлять необходимые элементы в вводимое выражение и Mathcad будет проделывать вычисления автоматически.

$$16 \cdot x^2 + 4 \cdot x + 15 \cdot y^2 + \frac{4}{7} \cdot x + 19 \cdot x + 99 \cdot x^2 - \frac{1}{19} \cdot x \rightarrow 115 \cdot x^2 + \frac{3128 \cdot x}{133} + 15 \cdot y^2$$

Знак символьного преобразования носит названия режима *live symbolics*.

Пример:

Пользуясь режимом *live symbolics*, найдите предел, к которому стремится выражение:

$$\left(1 + \frac{1}{x}\right)^x, \text{ если } x \rightarrow 0$$

Для ввода предела используем панель *Calculus* из меню *View-Toolbars*, на появившемся панели найдем кнопку пределов (*lim*) со слотом для ввода имени переменной, по которой ищется предел.

$$\lim_{x \rightarrow 0} \left(1 + \frac{1}{x}\right)^x \rightarrow 1$$

5.2.2 Опция *Symbolics-Simplify*

Символьная операция *Simplify* (упростить) – одна из самых важных. Эта операция позволяет упрощать математические выражения, содержащие алгебраические и тригонометрические функции, а также выражения со степенными многочленами (полиномами).

Опция *Symbolics-Simplify* позволяет упрощать и приводить к «красивой форме» более сложные выражения, чем алгебраические. В такие выражения могут входить дроби, для сокращения которых нужно искать корни уравнения, элементарные функции (тригонометрические, логарифмические ...) и т.п.

Упрощение означает замену более сложных фрагментов выражений на более простые. Приоритет тут отдается простоте функций. К примеру, функция $\tan(x)$ считается более сложной, чем функции $\sin(x)$ и $\cos(x)$. Поэтому $\tan(x)$ упрощается так, что получает представление через соотношение этих функций, что несколько неожиданно, так как в некоторых пакетах символьной математики, например *Derive*, ситуация иная: они заменяют отношение $\sin(x)/\cos(x)$ функцией $\tan(x)$.

Эта команда открывает широкие возможности для упрощения сложных и плохо упорядоченных алгебраических выражений.

Система *Mathcad* содержит встроенную функцию для вычисления значений определенных интегралов приближенным численным методом. Ею целесообразно пользоваться, когда нужно просто получить значение определенного интеграла в виде числа. Однако команда *Simplify* применительно к вычислениям определенных интегралов делает гораздо больше – она ищет аналитическое выражение для интеграла. Более того, она способна делать это и при вычислении кратных интегралов, пределы которых – функции.

Операцию *Simplify* можно использовать и для вычисления сумм и произведений символьных последовательностей. Результат операции, как и следовало ожидать, получается в символьной форме (если она существует).

Приведенные примеры могут создать впечатление, что *Mathcad* лихо справляется со всеми производными, интегралами, суммами и произведениями с помощью операции *Simplify*. К сожалению, это далеко не так. Нередко система не справляется с кажущимися простыми справочными примерами. Надо помнить, что символьный процессор системы *Mathcad* обладает заметно урезанной библиотекой функций и преобразований (в сравнении с библиотекой системы *Maple V*). Поэтому часто система не находит решение в замкнутом виде, хотя оно и приводится в справочнике. Тогда система повторяет введенное выражение или сообщает об ошибке.

В результате преобразований могут появляться специальные функции – как встроенные в систему (функции Бесселя, гамма-функция, интеграл вероятности и др.), так и ряд функций, дополнительно определенных при загрузке символьного процессора (интегральные синус и косинус, интегралы

Френеля, эллиптические интегралы и др.). Последние нельзя использовать при создании математических выражений.

Пример:

Вычислить выражение:

$$2^{\log(x,2)}$$

$$2^{\log(x,2)}$$

x

Вычисления проводятся с помощью опции *Simplify* из меню *Symbolics*. *Simplify* переводится как упрощай.

$$2^{\log(x,2)} \text{ simplify } \rightarrow x$$

5.2.3 Опция *Symbolics-Expand*

Действие операции *Expand* (разложить по степеням) в известном смысле противоположно действию операции *Simplify*. Подвергаемое преобразованию выражение расширяется с использованием известных (и введенных в символьное ядро) соотношений, например алгебраических разложений многочленов, произведений углов и т. д. Разумеется, расширение происходит только в том случае, когда его результат однозначно возможен. Иначе нельзя считать, что действие этой операции противоположно действию операции *Simplify*. К примеру, операция *Simplify* преобразует сумму квадратов синуса и косинуса в 1, тогда как обратное преобразование многозначно и потому в общем виде невыполнимо.

Опция *Symbolics-Expand* позволяет раскрывать скобки в сложных выражениях.

При преобразовании выражений операция *Expand Expression* старается более простые функции представить через более сложные, свести алгебраические выражения, представленные в сжатом виде, к выражениям в развернутом виде и т. д.

Пример:

$$(x^2 + a \cdot x + b)^3$$

$$a^3 \cdot x^3 + 3 \cdot a^2 \cdot b \cdot x^2 + 3 \cdot a^2 \cdot x^4 + 3 \cdot a \cdot b^2 \cdot x + 6 \cdot a \cdot b \cdot x^3 + 3 \cdot a \cdot x^5 + b^3 + 3 \cdot b^2 \cdot x^2 + 3 \cdot b \cdot x^4 + x^6$$

Для использования режима *live symbolic* Mathcad нужна информация о том, какие из переменных относятся к коэффициентам, а какие обозначают переменные. Поэтому следует при наборе использовать для переменных прописные литеры.

Пример:

$$(X^2 + a \cdot X + c)^2 \text{ expand } \rightarrow X^4 + 2 \cdot X^3 \cdot a + X^2 \cdot a^2 + 2 \cdot X^2 \cdot c + 2 \cdot X \cdot a \cdot c + c^2$$

Если заменить показатель степени на 7, то получится моментальный результат («вручную» подобный пример займет очень много времени).

5.2.4 Опция Symbolics-Factor

Операция *Factor Expression* (разложить на множители) используется для факторизации – разложения выражений или чисел на простые множители. Она способствует выявлению математической сущности выражений; к примеру, наглядно выявляет представление полинома через его действительные корни, а в том случае, когда разложение части полинома содержит комплексно-сопряженные корни, порождающее их выражение представляется квадратичным трехчленом.

В большинстве случаев (но не всегда) операция факторизации ведет к упрощению выражений. Термин *факторизация* не является общепризнанным в отечественной математической литературе, но мы его оставляем в связи с созвучностью с англоязычным именем этой операции.

Пример: разложить многочлен на множители

$$x^3 - 6 \cdot x^2 + 11 \cdot x - 6$$

$$(x - 3) \cdot (x - 1) \cdot (x - 2)$$

Пример: разложить число на множители

$$123456$$

$$2^6 \cdot 3 \cdot 643$$

5.2.5 Опция Symbolics-Collect

Операция *Collect* (разложить по подвыражению) обеспечивает замену указанного выражения выражением, скомплектованным по базису указанной переменной, если такое представление возможно. В противном случае появляется окно с сообщением о невозможности комплектования по указанному базису.

Эта команда особенно удобна, когда заданное выражение есть функция ряда переменных и нужно представить его в виде функции заданной переменной имеющей вид степенного многочлена. При этом другие переменные входят в сомножители указанной переменной, представленной в порядке уменьшения ее степени.

В том случае, когда комплектование по базису указанной переменной невозможно, система выдает сообщение об этом. Оно выводится в отдельном небольшом информационном окошке.

Пример:

$$(a + b + c)^2$$

$$c^2 + (2 \cdot a + 2 \cdot b) \cdot c + (a + b)^2 \quad \text{по } a$$

$$b^2 + (2 \cdot a + 2 \cdot c) \cdot b + (a + c)^2 \quad \text{по } b$$

$$a^2 + (2 \cdot b + 2 \cdot c) \cdot a + (b + c)^2 \quad \text{по } c$$

5.2.6 Опция *Symbolics-Polynomial Coefficients*

Операция *Polynomial Coefficients* (полиномиальные коэффициенты) в ранних версиях Mathcad отсутствующая, служит для вычисления коэффициентов полинома.

Операция применяется, если заданное выражение – полином (степенной многочлен) или может быть представлено таковым относительно выделенной переменной. Результатом операции является вектор с коэффициентами полинома. Операция полезна при решении задач полиномиальной аппроксимации и регрессии.

Пример:

$$a \cdot x^3 + b \cdot x^2 + c \cdot x + d$$

$$\begin{pmatrix} d \\ c \\ b \\ a \end{pmatrix}$$

5.2.7 Опция *Symbolics-Variable*

Опция *Symbolics-Variable* содержит весьма полезные позиции: символьное решение уравнений; символьное дифференцирование; символьное интегрирование и др.

Опция *Symbolics-Variable-Solve* служит для нахождения решений уравнений в символьной форме. Как и в описанных выше позициях, возможно однократное («мертвое») решение путем выделения искомой переменной и использования позиций меню, а также возможен режим *live symbolic*, в котором искомые переменные обозначаются прописными литерами.

Если задано некоторое выражение $f(x)$ и отмечена переменная x , то операция *Solve* (Решить) возвращает символьные значения указанной переменной x , при которых $f(x)=0$. Это очень удобно для решения алгебраических уравнений, например квадратных и кубических, или для вычисления корней полинома.

Ранее отмечалось, что усложнение уравнения, например переход от квадратного уравнения к кубическому, может вызвать и существенное усложнение результата. Тогда система представляет решение в более компактном виде (но без общепринятой математической символики) и предлагает занести его в буфер обмена.

С помощью операции *Paste* (вставить) в позиции *Edit* (правка) главного меню можно перенести решение в основное окно системы, но оно имеет уже тип текстового комментария, а не математического выражения, пригодного для дальнейших преобразований. Впрочем, часть его можно (опять-таки с помощью буфера обмена) ввести в формульные блоки для последующих преобразований и вычислений.

Более того, форма представления результата в таком случае отличается от принятой в системе Mathcad (например, в качестве знака деления используется косая черта и т. д.). Это сделано ради компактности представления результатов вычислений.

В случаях, описанных выше, пользователю надо реально оценить свои силы в упрощении решения. Это придется сделать вручную. При технических расчетах специалист нередко знает, какие из параметров решения несущественны и может отбросить их. Однако для строгих математических расчетов это не всегда возможно, поэтому даже громоздкий результат может быть весьма полезным с познавательной точки зрения.

Пример: решение уравнения

$$ax^2 + bx + c = 0 \text{ solve, } x \rightarrow \left(\begin{array}{c} \frac{b - \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a} \\ \frac{b + \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a} \end{array} \right)$$

Пример: решение системы уравнений

$$\left(\begin{array}{l} 2 \cdot x^2 + 3 \cdot y = 10 \\ 7 \cdot x - 19 \cdot y = 14 \end{array} \right) \text{ solve, } \left(\begin{array}{l} x \\ y \end{array} \right) \rightarrow \left(\begin{array}{cc} \frac{\sqrt{35705}}{76} - \frac{21}{76} & \frac{7 \cdot \sqrt{35705}}{1444} - \frac{1211}{1444} \\ -\frac{\sqrt{35705}}{76} - \frac{21}{76} & -\frac{7 \cdot \sqrt{35705}}{1444} - \frac{1211}{1444} \end{array} \right)$$

Опция *Symbolics-Variable-Substitute* возвращает новое выражение, полученное путем постановки на место указанной переменной некоторого другого выражения. Последнее должно быть подготовлено и скопировано (операциями *Cut* или *Copy*) в буфер обмена. Наряду с получением результата в символьном виде эта команда позволяет найти и числовые значения функции некоторой переменной путем замены ее на числовое значение. Операция *Substitute* переводится как «Заменить переменную».

Подстановки и замены переменных довольно часто встречаются в математических расчетах, что делает эту операцию весьма полезной. Кроме того, она дает возможность перейти от символьного представления результата к числовому.

Пример:

$y - a$ в буфер обмена

$x^3 + 2 \cdot x^2 + 1$ исходное выражение

$3 \cdot a^2 \cdot y - a^3 + 2 \cdot a^2 - 3 \cdot a \cdot y^2 - 4 \cdot a \cdot y + y^3 + 2 \cdot y^2 + 1$ результат

$x^3 + 2 \cdot x^2 + 1$ исходное выражение

2 в буфер обмена

17 результат

Опция *Symbolics-Variable-Differentiate* служит для дифференцирования в символьной форме. Как и в описанных выше позициях, возможно однократное («мертвое») дифференцирование путем выделения переменной, по которой ищется производная, и использования позиций меню, а также возможен режим *live symbolic*, в котором переменная дифференцирования обозначается прописными литерами.

Нахождение символьного значения производной – одна из самых распространенных задач в аналитических вычислениях. Операция *Differentiate* (дифференцировать по переменной) возвращает символьное значение производной выражения по той переменной, которая указана курсором. Для вычисления производных высшего порядка (свыше 1) нужно повторить вычисление необходимое число раз.

Пример: нахождение корней производных

$$\frac{d}{dX} a^X \cdot \sin(X)^2$$

$$\frac{d}{dX} a^X \cdot \sin(X)^2 \rightarrow a^X \cdot \sin(X)^2 \cdot \ln(a)$$

Пример: нахождение производных высших порядков

$$\frac{d^3}{dX^3} a^X \cdot \sin(X) \rightarrow a^X \cdot \sin(X) \cdot \ln(a)^3$$

Опция *Symbolics-Variable-Integrate* служит для интегрирования в символьной форме. Как и в описанных выше позициях, возможно однократное («мертвое») интегрирование путем выделения переменной, по которой ищется интеграл, и использование позиций меню, а также возможен режим *live symbolic*.

Другая не менее важная операция при символьных вычислениях – вычисление интегралов (или нахождение первообразных) для аналитически

заданной функции. Для этого используется операция *Integrate* (интегрировать по переменной). Она возвращает символьное значение неопределенного интеграла по указанной курсором ввода переменной. Выражение, в состав которого входит переменная, является подынтегральной функцией.

Как и для операции дифференцирования, в состав исходных выражений и результатов символьного интегрирования могут входить встроенные в систему специальные математические функции.

Пример: нахождение неопределенного интеграла

$$e^{-\alpha \cdot x} \cdot \cos(\beta \cdot x)$$

$$e^{-\alpha \cdot x} \cdot \cos(\beta \cdot x)$$

$$\frac{e^{-\alpha \cdot x} \cdot (\alpha \cdot \cos(\beta \cdot x) - \beta \cdot \sin(\beta \cdot x))}{\beta^2 + \alpha^2}$$

Пример: нахождение определенного интеграла в режиме *live symbolic*

$$\int_0^{\infty} e^{\frac{-x^2}{2}} dx$$

$$\int_0^{\infty} e^{\frac{-x^2}{2}} dx \rightarrow \frac{\sqrt{2} \cdot \sqrt{\pi}}{2}$$

Опция *Symbolics-Variable-Expand to Series...* служит для разложения в ряд Тейлора. Операция *Expand to Series...* (разложить в ряд) возвращает разложение в ряд Тейлора выражения относительно выделенной переменной с заданным по запросу числом членов ряда n (число определяется по степеням ряда). По умолчанию задано $n=6$ $\langle n \rangle$.

Разложение возможно для функции заданной переменной. В разложении указывается остаточная погрешность разложения. На рисунке представлено применение этой операции для разложения функции $\sin(x)/x$. Минимальная погрешность получается при малых x .

Символьные операции нередко можно комбинировать для решения сложных задач.

Пример:

$$e^z$$

$$1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} + \frac{z^6}{720} + \frac{z^7}{5040} \quad n=8$$

$$1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} \quad n=6$$

Опция *Symbolics-Variable-Convert to Partial Fraction* (разложить на элементарные дроби) возвращает символьное разложение выражения, представленное относительно заданной переменной в виде суммы правильных целых дробей.

Пример:

$$\frac{x + a}{x + x^2 + b \cdot x}$$

$$\frac{x}{x + x^2 + b \cdot x} + \frac{a}{x + x^2 + b \cdot x}$$

по переменной a

$$\frac{a}{x \cdot (b + 1)} - \frac{\frac{a}{b + 1} - 1}{b + x + 1}$$

по переменной x

5.2.8 Опция Transform-Fourier

Пример:

$$a \cdot t$$

$$-2i \cdot \pi \cdot t \cdot \Delta(1, \omega)$$

$$-2i \cdot \pi \cdot a \cdot \Delta(1, \omega)$$

5.2.9 Опция Transform-Laplace Transform

Пример:

$$\frac{-t}{1 - e^r}$$

$$\frac{1}{s \cdot (r \cdot s + 1)}$$

5.2.10 Опция Transform-Z

Пример:

$$a \cdot t$$

$$\frac{a \cdot z}{(z - 1)^2}$$

Лабораторная работа № 6 «Символьные преобразования в Mathcad»

Цель работы: получить навыки работы в Mathcad при использовании символьных вычислений.

Задание на лабораторную работу:

Изучить материал, представленный в разделе 5 (пункт 5.1, 5.2) соответственно.

1. Найдите предел функции:

$$\frac{\sin(2x)}{\sin(3x)}, \text{ если } x \rightarrow 0$$

2. Упростите выражение:

$$\frac{\sin(x)}{\operatorname{tg}} \cdot \cos(x) + \sin(x) \cdot \operatorname{tg}(x) \cdot \cos(x) \cdot \frac{1+4 \cdot x+4 \cdot x^2}{1+2 \cdot x}, \lim_{x \rightarrow \infty} \left[\frac{x+1}{x+3} \right]^{2 \cdot x+1}$$

3. Раскрыть скобки:

$$(x^3 + 7 \cdot x^2 + 3 \cdot x - 4)^4, (a \cdot x^2 + b \cdot x + c)^2$$

4. Решить уравнение:

$$4 \cdot x^2 - 5 \cdot x + 6 = 0, 3 \cdot x^2 + 2 \cdot x - 1 = 0$$

5. Решить систему уравнений:

$$\begin{cases} 4 \cdot x^3 - 5 \cdot y = 11 \\ 2 \cdot x + 7 \cdot y^2 = -4 \end{cases}, \begin{cases} -x + y^2 + 5 \cdot z = -3 \\ 2 \cdot x + 3 \cdot y - z = 2 \\ x^2 - y + 2 \cdot z = -1 \end{cases}$$

6. Найти корни производных:

$$\frac{d^3}{d(x)^3} \cos(x), \frac{d}{dx} \cos(x)$$

7. Используя интегральное преобразование Лапласа (и обратное преобразование Лапласа) решите в символьном виде дифференциальное уравнение:

$$\frac{d}{dt} y(t) - k \cdot y(t) = \sin(t)$$

8. Используя палитру символьных преобразований, найдите точку, в которой функция двух переменных имеет экстремум

$$\frac{d}{dt} y(t) - k \cdot y(t) = \sin(t)$$

9. Вычислить в символьном виде интеграл:

$$x \cdot \int_0^{\infty} \exp(-c \cdot t) dt$$

10. Вычислите коэффициенты полинома по степеням x и y :

$$x^2 \cdot y^2 + 5 \cdot x^2 \cdot y - 6 \cdot x \cdot y^2 + 7 \cdot x \cdot y - 8 \cdot x + 9 \cdot y - 10$$

11. Исследуйте аналитически функцию $f(x)$: определите точки максимума и минимума, точки перегиба, постройте графики $f(x), f''(x), f'''(x)$

$$f(x) = \frac{2 \cdot x}{x^2 + 1}$$

12. Разложить на множители:

$$x^3 - 6 \cdot x^2 + 21 \cdot x - 5, 10244785$$

13. Разложить по подвыражению:

$$(y + z) \cdot (z + y) \text{ по } y, \text{ по } z$$

14. Разложить полином относительно с

$$a \cdot x^3 + b \cdot x^2 + c \cdot x + d$$

15. Разложить в ряд Тейлора (n=6)

$$\frac{\sin(x)}{x}$$

16. Найти преобразование Фурье, обратное преобразование Фурье, преобразование Лапласа, обратное преобразование Лапласа от выражения: $\sin(x)$

17. Найти Z преобразование и обратное Z преобразование от $\frac{1}{x}$.

Составьте отчет по лабораторной работе №6 (см. Приложение А – Пример оформления отчета). Сделайте соответствующие выводы.

6 Интерполяция и регрессия, функции сглаживания данных и предсказания

6.1 Функции линейной и сплайновой аппроксимации

К функциям линейной и сплайновой аппроксимации относятся:

- Одномерная линейная аппроксимация;
- Одномерная сплайн-интерполяция и сплайн-аппроксимация
- Двумерная линейная сплайн-интерполяция и сплайн-аппроксимация

6.1.1 Одномерная линейная аппроксимация

При проведении научно-технических расчетов часто используются зависимости вида $y(x)$, причем число точек этих зависимостей ограничено. Неизбежно возникает задача получения приемлемой представительности функций в промежутках между узловыми точками (интерполяция) и за их пределами (экстраполяция). Эта задача решается аппроксимацией исходной зависимости, т.е. ее подменой какой-либо достаточно простой функцией. Система Mathcad предоставляет возможность аппроксимации двух типов: кусочно-линейной и сплайновой.

При кусочно-линейной интерполяции, или аппроксимации, вычисления дополнительных точек выполняются по линейной зависимости. Графически это означает просто соединение узловых точек отрезками прямых, для чего используется следующая функция: $linterp(VX, VY, x)$.

Для заданных векторов VX и VY узловых точек и заданного аргумента x эта функция возвращает значение функции при ее линейной аппроксимации. При экстраполяции используются отрезки прямых с наклоном, соответствующим наклону крайних отрезков при линейной интерполяции.

Пример:

$$m := \begin{pmatrix} 0 & 0 \\ 4 & 4.5 \\ 5 & 7 \\ 7 & 6.5 \\ 10 & 9 \\ 11 & 11 \\ 1.5 & 5 \\ 8.5 & 9.5 \\ 12.5 & 9 \end{pmatrix}$$

```

m := csort(m,0)
X := m<0>  Y := m<1>
f(x) := linterp(X,Y,x)
scale := 100  j := 0..scale

x_j := min(X) + j * (max(X) - min(X)) / scale
i := 0..length(X) - 1

```

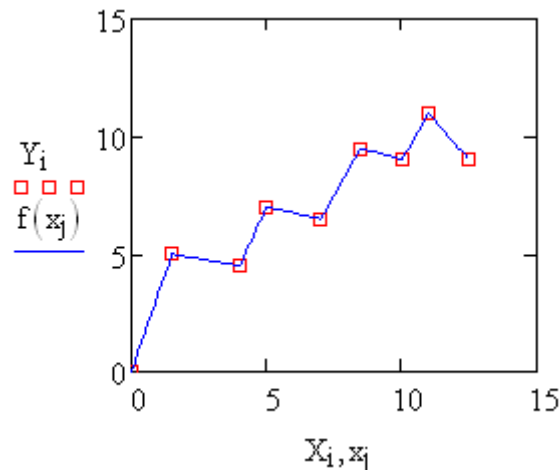


Рисунок 6.1 – Вид кусочно-линейной интерполяции

6.1.2 Одномерная сплайн-интерполяция и сплайн-аппроксимация

При небольшом числе узловых точек (менее 10) линейная интерполяция оказывается довольно грубой. Для целей экстраполяции функция *linterp* (*VX*, *VY*, *x*) не предназначена и за пределами области определения может вести себя непредсказуемо.

Гораздо лучшие результаты дает сплайн-аппроксимация. При ней исходная функция заменяется отрезками кубических полиномов, проходящих через три смежные узловых точки. Коэффициенты полиномов рассчитываются так, чтобы непрерывными были первая и вторая производные.

Для осуществления сплайновой аппроксимации система Mathcad предлагает четыре встроенные функции. Три из них служат для получения векторов вторых производных сплайн-функций при различном виде интерполяции:

- *cspline*(*VX*, *VY*) – возвращает вектор *VS* вторых производных при

- приближении в опорных точках к кубическому полиному;
- $pspline(VX, VY)$ – возвращает вектор VS вторых производных при приближении к опорным точкам параболической кривой;
 - $lspline(VX, VY)$ – возвращает вектор VS вторых производных при приближении к опорным точкам прямой.
 - $interp(VS, VX, VY, x)$ – возвращает значение $y(x)$ для заданных векторов VS, VX, VY и заданного значения x .

Таким образом, сплайн-аппроксимация проводится в два этапа. На первом с помощью функций $cspline$, $pspline$ или $lspline$ отыскивается вектор вторых производных функции $y(x)$, заданной векторами VX и VY ее значений (абсцисс и ординат). Затем, на втором этапе для каждой искомой точки вычисляется значение $y(x)$ спомощью функции $interp$.

Пример:

$$\begin{array}{l}
 \underline{m} := \begin{pmatrix} 0 & 0 \\ 4 & 4.5 \\ 5 & 7 \\ 7 & 6.5 \\ 10 & 9 \\ 11 & 11 \\ 1.5 & 5 \\ 8.5 & 9.5 \\ 12.5 & 9 \end{pmatrix} \quad \begin{array}{l} m := \text{csort}(m, 0) \\ X := m^{\langle 0 \rangle} \quad Y := m^{\langle 1 \rangle} \end{array}
 \end{array}$$

$$S1 := \text{cspline}(X, Y) \quad f1(x) := \text{interp}(S1, X, Y, x)$$

$$S2 := \text{pspline}(X, Y) \quad f2(x) := \text{interp}(S2, X, Y, x)$$

$$S3 := \text{lspline}(X, Y) \quad f3(x) := \text{interp}(S3, X, Y, x)$$

Построение графиков: $scale := 100 \quad j := 0..scale$

$$x_j := \min(X) + j \cdot \frac{\max(X) - \min(X)}{scale} \quad i := 0..length(X) - 1$$

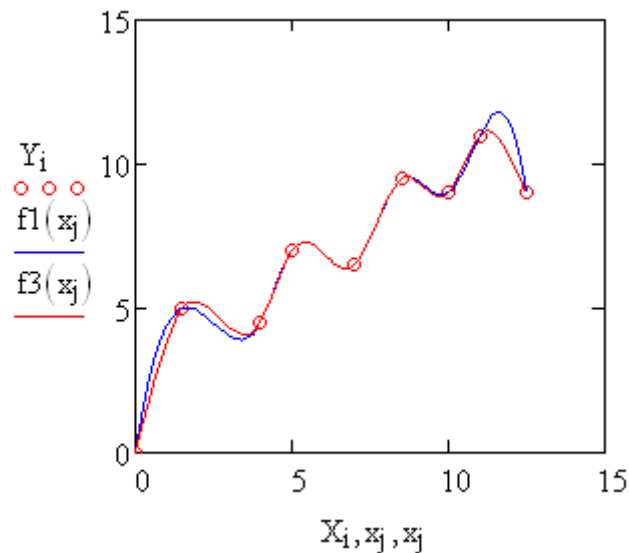


Рисунок 6.2 – Вид одномерной сплайн-интерполяции и сплайн-аппроксимации

6.1.3 Двумерная линейная сплайн-интерполяция и сплайн-аппроксимация

Для повышения качества построения 3D-графиков имеется возможность осуществления двумерной сплайн-интерполяции. Это позволяет существенно повысить представительность сложных графиков SD-функций, в том числе контурных.

Интерполяция функции 2-х переменных проводится также в два этапа:

1. Вычисляется вектор VS вторых производных в опорных точках с помощью функций:

- $cspline(Mxy, Mz)$,
- $pspline(Mxy, Mz)$,
- $lspline(Mxy, Mz)$

Здесь Mxy – матрица размера $n*2$, строки которой определяют по диагонали (x, y) координаты прямоугольной сетки,

Mz – матрица размера $n*n$ значений функции в узлах вышеопределенной сетки.

2. Вычисление с помощью функции $interp(VS, Mxy, Mz, V)$. Здесь V – вектор координат (x, y) .

На рисунке 6.3 справа показан график функции 2-х переменных после проведения двумерной сплайн-интерполяции, а слева – без нее.

Пример:

$$\mathbf{Mz} := \begin{pmatrix} 0.18 & 0.14 & -0.14 & -0.51 & -0.29 & 0.33 \\ 0.93 & 0.17 & -0.76 & -0.99 & -0.31 & -0.83 \\ -0.65 & -0.33 & -0.66 & 0.24 & 0 & 0 \\ -0.55 & -0.22 & 0.47 & 0.74 & -0.11 & 0 \\ -0.98 & 0.17 & 0.37 & 0.81 & 0.39 & -0.87 \\ -0.71 & 0.13 & 0.76 & 0.31 & 0.3 & -0.16 \end{pmatrix}$$

$n := \text{rows}(\mathbf{Mz})$

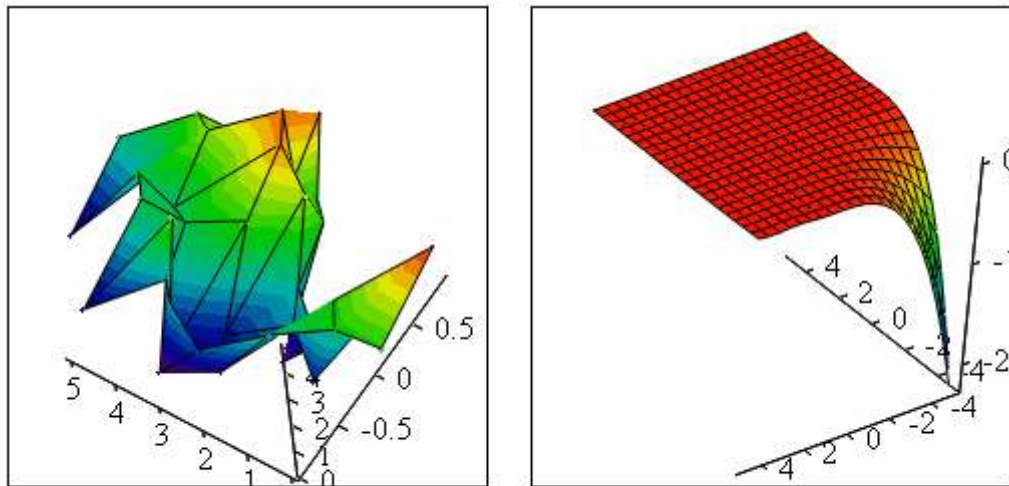
$$\mathbf{X} := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} \quad \mathbf{Y} := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$

$\mathbf{Mxy} := \text{augment}(\text{sort}(\mathbf{X}), \text{sort}(\mathbf{Y}))$

$$\mathbf{Mxy} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 2 & 2 \\ 3 & 3 \\ 4 & 4 \\ 5 & 5 \end{pmatrix}$$

$\mathbf{S} := \text{cspline}(\mathbf{Mxy}, \mathbf{Mz})$

$f(x,y) := \text{interp}\left[\mathbf{S}, \mathbf{Mxy}, \mathbf{Mz}, \begin{pmatrix} x \\ y \end{pmatrix}\right]$



Mz

f

Рисунок 6.3 – Вид двумерной линейной сплайн-интерполяции и сплайн-аппроксимации

6.2 Функции для проведения регрессии

К функциям для проведения регрессии относятся:

- Функция для линейной регрессии;
- Функция для линейной регрессии общего вида;
- Функции для одномерной и многомерной полиномиальной регрессии;
- Функция для нелинейной регрессии общего вида

6.2.1 Функции для линейной регрессии

Другой широко распространенной задачей обработки данных является представление их совокупности некоторой функцией $y(x)$. Задача регрессии заключается в получении параметров этой функции такими, чтобы функция приближала облако исходных точек (заданных векторами VX и VY) с наименьшей среднеквадратичной погрешностью.

Чаще всего используется линейная регрессия, при которой функция $y(x)$ имеет вид: $y(x) = a + b \cdot x$ и описывает отрезок прямой. К линейной регрессии можно свести многие виды нелинейной регрессии при двухпараметрических зависимостях $y(x)$.

Для проведения линейной регрессии в систему встроен ряд приведенных ниже функций:

- $\text{corr}(VX, VY)$ – возвращает скаляр – коэффициент корреляции Пирсона;
- $\text{intercprt}(VX, VY)$ – возвращает значение параметра a (смещение линии регрессии по вертикали);
- $\text{slope}(VX, VY)$ – возвращает значение параметра b (наклона линии регрессии).

Пример:

$$m := \begin{pmatrix} 0 & 0 \\ 4 & 4.5 \\ 5 & 7 \\ 7 & 6.5 \\ 10 & 9 \\ 11 & 11 \\ 1.5 & 5 \\ 8.5 & 9.5 \\ 12.5 & 9 \end{pmatrix}$$

$$m := \text{csort}(m, 0)$$

$$X := m^{(0)}$$

$$Y := m^{(1)}$$

$$n := 0.. \text{rows}(m) - 1$$

$$a := \text{intercept}(X, Y)$$

$$b := \text{slope}(X, Y)$$

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

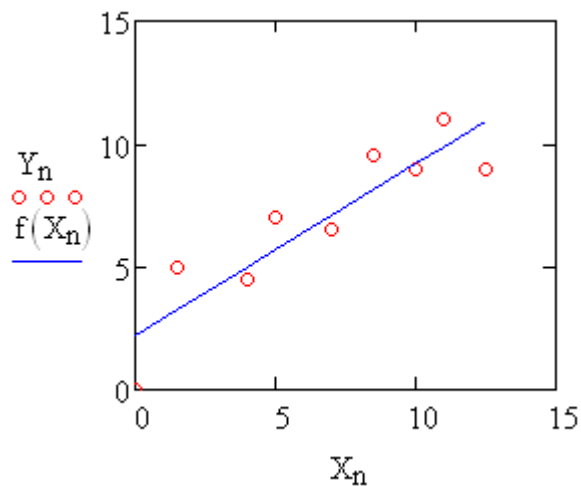


Рисунок 6.4 – Вид функции для линейной регрессии

6.2.2 Функция для линейной регрессии общего вида

В Mathcad реализована возможность выполнения линейной регрессии общего вида. При ней заданная совокупность точек приближается функцией вида:

$$F(x, K1, K2, \dots, Kn) = K1 \cdot F1(x) + K2 \cdot F2(x) + \dots + Kn \cdot Fn(x).$$

Таким образом, функция регрессии является линейной комбинацией функций $F1(x)$, $F2(x)$, ..., $Fn(x)$, причем сами эти функции могут быть

нелинейными, что резко расширяет возможности такой аппроксимации и распространяет ее на нелинейные функции.

Для реализации линейной регрессии общего вида используется функция $\text{linfit}(VX, VY, F)$. Эта функция возвращает вектор коэффициентов линейной регрессии общего вида K , при котором среднеквадратичная погрешность приближения облака исходных точек, если их координаты хранятся в векторах VX и VY , оказывается минимальной. Вектор F должен содержать функции $F1(x), F2(x), \dots, Fn(x)$, записанные в символьном виде.

Расположение координат точек исходного массива может быть любым, но вектор VX должен содержать координаты, упорядоченные в порядке их возрастания, а вектор VY – ординаты, соответствующие абсциссам в векторе VX .

Пример:

$$\underline{F}(x) := \begin{pmatrix} 1 \\ x \\ 2 \cdot x^2 - 1 \end{pmatrix} \quad \underline{S2} := \text{linfit}(X, Y, F) \quad S2 = \begin{pmatrix} 1.028 \\ 1.347 \\ -0.026 \end{pmatrix}$$

$$g2(t) := F(t) \cdot S2$$

$$t := 0, 0.1 \dots 12.5$$

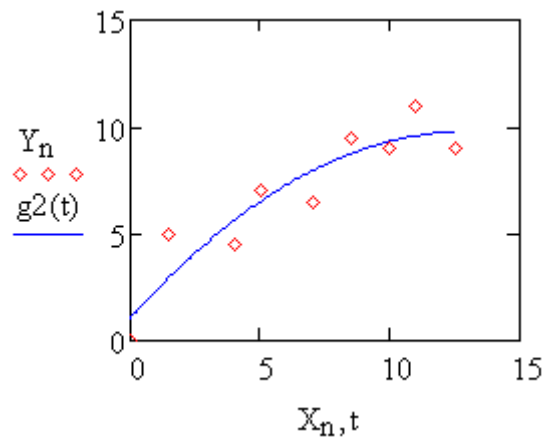


Рисунок 6.5 – Вид функции для линейной регрессии общего вида

6.2.3 Функции для одномерной и многомерной полиномиальной регрессии

Введена в новую версию Mathcad и функция для обеспечения полиномиальной регрессии при произвольной степени полинома регрессии:

$$\text{regress}(VX, VY, n)$$

Она возвращает вектор VS , запрашиваемый функцией $\text{interp}(VS, VX, VY, x)$, содержащий коэффициенты многочлена n -й степени, который наилучшим образом приближает «облако» точек с координатами, хранящимися в векторах VX и VY .

Для вычисления коэффициентов полинома регрессии используется функция submatrix .

На практике не рекомендуется делать степень аппроксимирующего полинома выше четвертой – шестой, поскольку погрешности реализации регрессии сильно возрастают.

Функция *regress* создает единственный приближающий полином, коэффициенты которого вычисляются по всей совокупности заданных точек, т.е. глобально. Иногда полезна другая функция полиномиальной регрессии, дающая локальные приближения отрезками полиномов второй степени, – *loess(VX, VY, span)*. Эта функция возвращает используемый функцией *interp(VS, VX, VY, r)* вектор *VS*, дающий наилучшее приближение данных (с координатами точек в векторах *VX* и *VY*) отрезками полиномов второй степени. Аргумент *span > 0* указывает размер локальной области приближаемых данных (рекомендуемое начальное значение – 0,75).

Чем больше *span*, тем сильнее сказывается сглаживание данных. При больших *span <* эта функция приближается к *regress(VX, VY, 2)*.

Пример:

```
vs1 := regress(X, Y, 2)
```

```
f1(x) := interp(vs1, X, Y, x)
```

```
vs2 := loess(X, Y, 0.75)
```

```
f2(x) := interp(vs2, X, Y, x)
```

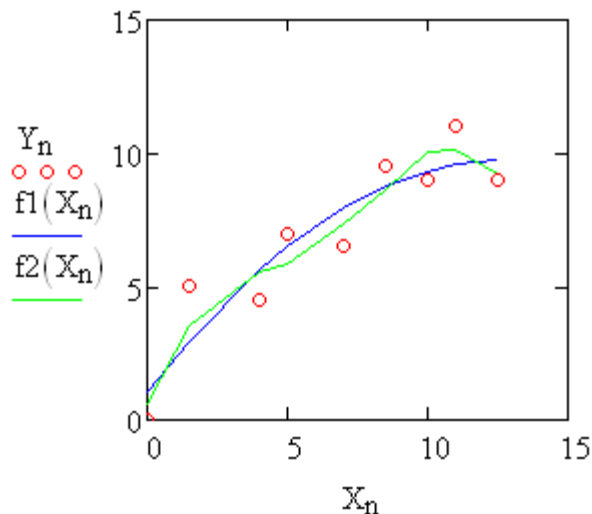


Рисунок 6.6 – Вид функции для одномерной полиномиальной регрессии

Mathcad позволяет выполнять также многомерную регрессию, самый типичный случай которой – приближение трехмерных поверхностей. Их можно характеризовать массивом значений высот *z*, соответствующих двумерному массиву *Mxy* координат точек (x, y) на горизонтальной плоскости.

Новых функций для этого не задано. Используются уже описанные функции в несколько иной форме:

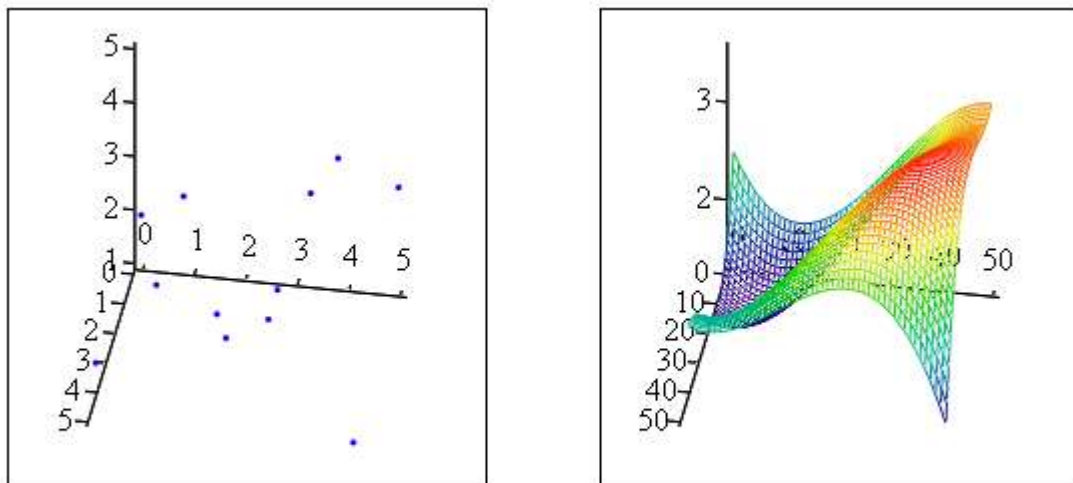
- *Regress(Mxy, Vz, n)* – возвращает вектор, запрашиваемый функцией *interp(VS, Mxy, Vz, V)* для вычисления многочлена *n*-й степени, который наилучшим образом приближает точки

множества Mxy и Vz . Mxy – матрица $m \times 2$, содержащая координаты x и y .

- Vz – m -мерный вектор, содержащий z -координат, соответствующих m точкам, указанным в Mxy ;
- $loess(Mxy, Vz, span)$ – аналогична $loes(VX, VY, span)$, но в многомерном случае;
- $interp(VS, Mxy, Vz, V)$ – возвращает значение z по заданным векторам VS (создается функциями $regress$ или $loess$) и Mxy , Vz и V (вектор координат x и y заданной точки, для которой находится z).

Пример:

$$\begin{array}{l}
 Mxy := \begin{pmatrix} 2 & 2 \\ 2 & 3 \\ 3 & 2 \\ 3 & 3 \\ 1 & 1 \\ 1 & 4 \\ 4 & 1 \\ 4 & 4 \\ 0 & 0 \\ 0 & 5 \\ 5 & 0 \\ 5 & 5 \end{pmatrix} \quad
 Vz := \begin{pmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 3 \\ 4 \\ 3 \\ 5 \\ 2 \\ 3 \\ 2 \\ 1 \end{pmatrix} \quad
 VS := regress(Mxy, Vz, 3) \\
 i := 0..50 \quad j := 0..50 \\
 x_i := \frac{i}{10} \quad y_j := \frac{j}{10} \\
 Z_{i,j} := interp \left[VS, Mxy, Vz, \begin{pmatrix} x_i \\ y_j \end{pmatrix} \right] \\
 X := Mxy^{(0)} \quad Y := Mxy^{(1)} \quad Z := Vz
 \end{array}$$



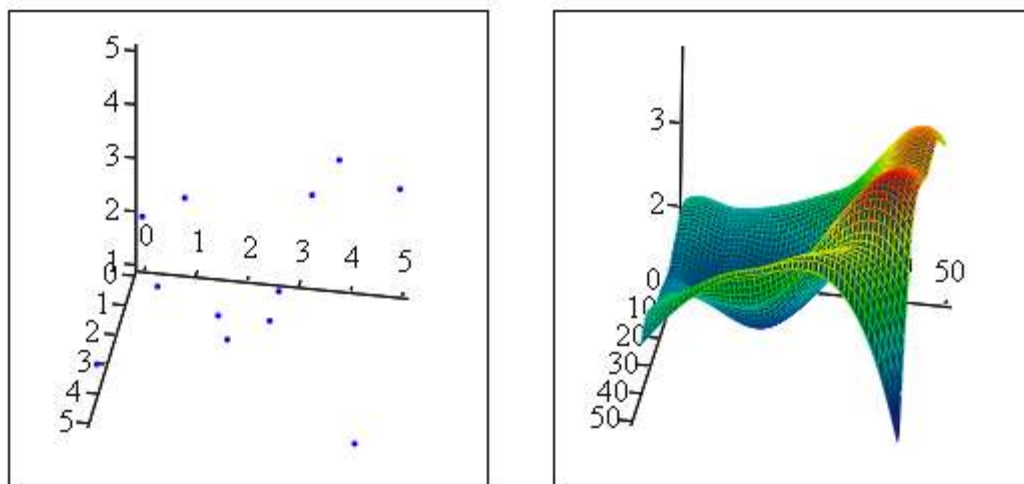
(X, Y, Z)

Z3

Рисунок 6.7 – Вид функции $Regress(Mxy, Vz, n)$ для многомерной полиномиальной регрессии

$VS1 := loess(Mxy, VZ, 0.85)$

$$Z1_{i,j} := \text{interp} \left[VS1, Mxy, VZ, \begin{pmatrix} x_i \\ y_j \end{pmatrix} \right]$$



(X, Y, Z)

Z1

Рисунок 6.8 – Вид функции $loess(Mxy, Vz, span)$ для многомерной полиномиальной регрессии

6.2.4 Функция для нелинейной регрессии общего вида

Под нелинейной регрессией общего вида подразумевается нахождение вектора K параметров произвольной функции $F(x, K1, K2, \dots, Kn)$, при котором обеспечивается минимальная среднеквадратичная погрешность приближения облака исходных точек.

Для проведения нелинейной регрессии общего вида используется функция $genfit(VX, VY, VS, F)$. Эта функция возвращает вектор K параметров

функции F , дающий минимальную среднеквадратичную погрешность приближения функцией $F(x, K1, K2, \dots, Kn)$ исходных данных.

F должен быть вектором с символьными элементами, содержащими уравнение исходной функции и ее производных по всем параметрам. Вектор VS должен содержать начальные значения элементов вектора K , необходимые для решения системы нелинейных уравнений регрессии итерационным методом.

Пример:

$$XX := \begin{pmatrix} 1.5 \\ 4 \\ 5 \\ 7 \\ 8.5 \\ 10 \\ 11 \\ 12.5 \end{pmatrix} \quad YY := \begin{pmatrix} 5 \\ 4.5 \\ 7 \\ 6.5 \\ 9 \\ 11 \\ 5 \\ 9.5 \end{pmatrix} \quad \begin{array}{l} ff(x,r,w) := r \cdot \ln(w \cdot x) \\ \left(\frac{d}{dr} ff(x,r,w) \rightarrow \right) \ln(w \cdot x) \\ \left(\frac{d}{dw} ff(x,r,w) \rightarrow \right) \frac{r}{w} \end{array}$$

$$F(x,U) := \begin{pmatrix} U_0 \cdot \ln(U_1 \cdot x) \\ \ln(U_1 \cdot x) \\ U_0 \\ \frac{1}{U_1} \end{pmatrix} \quad VG := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$k := 0.. \text{rows}(XX) - 1$$

$$U := \text{genfit}(XX, YY, VG, F) \quad U := \begin{pmatrix} 2.051 \\ 5.288 \end{pmatrix}$$

$$y(t) := U_0 \cdot \ln(U_1 \cdot t)$$

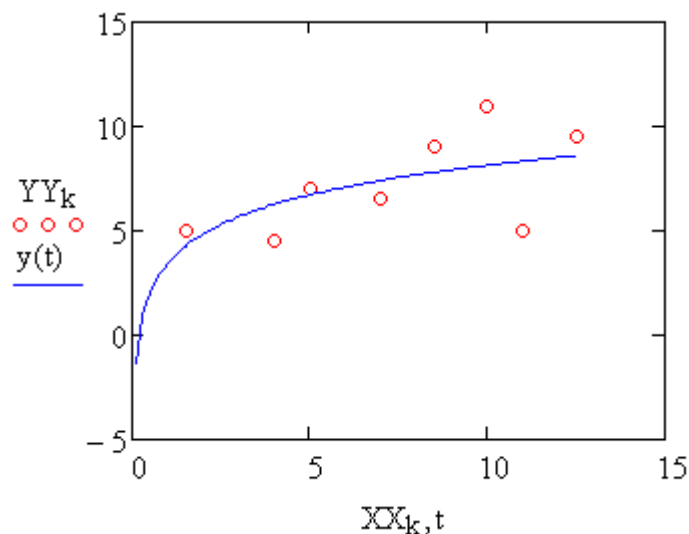


Рисунок 6.9 – Вид функции для нелинейной регрессии общего вида

6.3 Функция сглаживания данных

Данные большинства экспериментов имеют случайные составляющие погрешности. Поэтому часто возникает необходимость статистического сглаживания данных. Ряд функций Mathcad предназначен для выполнения операций сглаживания данных различными методами. Вот перечень этих функций:

- $medsmooth(VY, n)$ – для вектора с m действительными числами возвращает m -мерный вектор сглаженных данных по методу скользящей медианы, параметр n задает ширину окна сглаживания (n должно быть нечетным числом, меньшим m);
- $ksmooth(VX, VY, b)$ – возвращает n -мерный вектор сглаженных VY , вычисленных на основе распределения Гаусса. VX и VY – n -мерные векторы действительных чисел. Параметр b (полоса пропускания) задает ширину окна сглаживания (b должно в несколько раз превышать интервал между точками по оси x);
- $supsmooth(VX, VY)$ – возвращает n -мерный вектор сглаженных VY , вычисленных на основе использования процедуры линейного сглаживания методом наименьших квадратов по правилу k -ближайших соседей с адаптивным выбором k . VX и VY – n -мерные векторы действительных чисел. Элементы вектора VX должны идти в порядке возрастания.

Пример:

$i := 1..50$

$VX_i := i$

$VY_i := rnd(1)$

$n := rows(VY)$

$n = 51$

$SM1 := medsmooth(VY, 9)$

$SM2 := ksmooth(VX, VY, 5)$

$SM3 := supsmooth(VX, VY)$

$i := 0..n - 1$

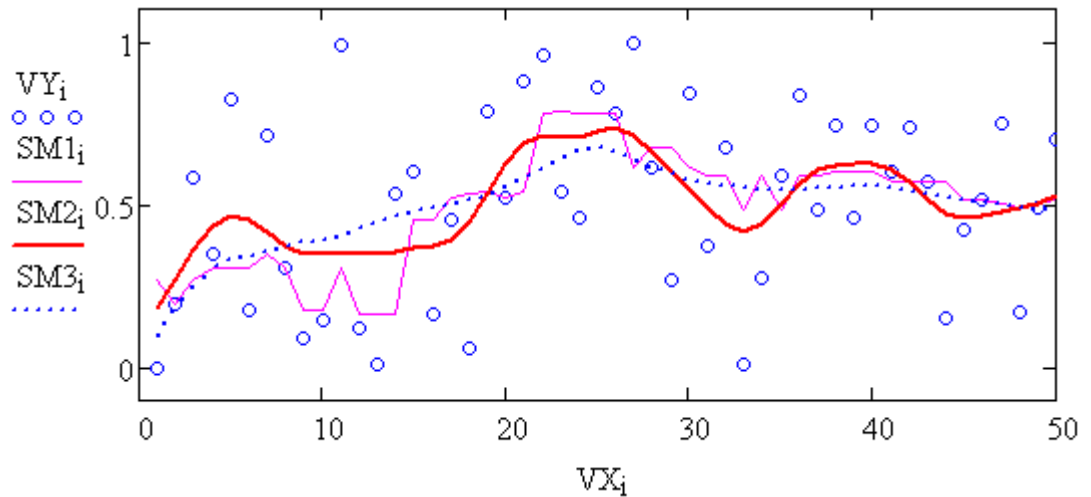


Рисунок 6.10 – Вид для функции сглаживания данных

6.4 Функция предсказания

Весьма интересной является функция предсказания $predikt(data, k, N)$, где $data$ – вектор данных, где $data$ – вектор данных, k – число последних точек существующих данных, на основе которых происходит расчет предсказываемых точек; и N – число точек, в которых необходимо предсказать данные. Она по ряду заданных равномерно расположенных точек позволяет рассчитать некоторое число N последующих точек, т. е. по существу выполняет экстраполяцию произвольной (но достаточно гладкой и предсказуемой) зависимости.

Функция предсказания обеспечивает высокую точность при монотонных исходных функциях или функциях, представляемых полиномом невысокой степени.

Пример:

$k := 0..100$

$$DAT_k := \exp\left(\frac{-k}{200}\right) \cdot \sin\left(\frac{k}{10}\right)$$

$P := predict(DAT, 3, 100)$

$i := 0,5..99$

$n := 100, 105..200$

$$DANN_n := \exp\left(\frac{-n}{200}\right) \cdot \sin\left(\frac{n}{10}\right)$$

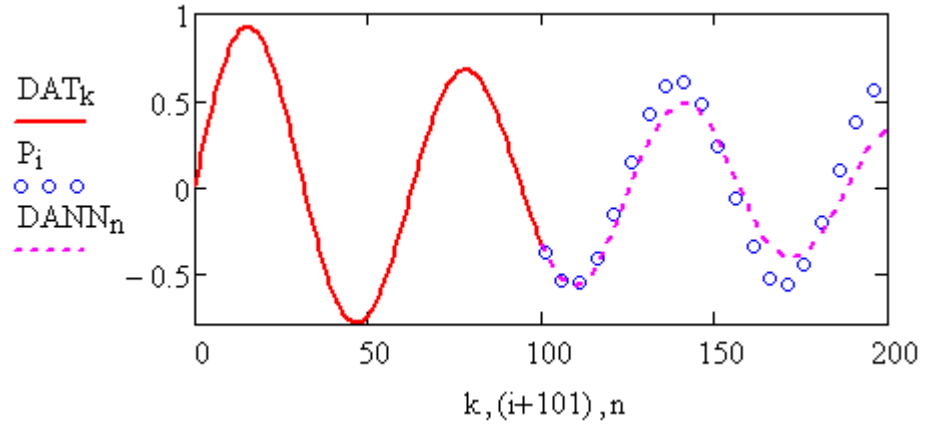


Рисунок 6.11 – Вид для функции предсказания

Лабораторная работа № 7 «Интерполяция и регрессия, функции сглаживания данных и предсказания»

Цель работы: получить навыки работы в Mathcad при использовании интерполяции и регрессии, функции сглаживания данных и предсказания.

Задание на лабораторную работу:

Изучить материал, представленный в разделе 6 (пункт 6.1, 6.2, 6.3, 6.4) соответственно.

1. Введите матрицу координат точек плоскости:

x	0.1	5	1.1	0.5	3.4	2.7	1.9	4	4.7
y	10	1.3	5	7	3	4.5	6	2.5	1.5

а) Постройте линейную и сплайновую интерполяцию для данного облака точек;

б) Постройте функции линейной, полиномиальной и обобщенной регрессии для этих же точек.

2. С помощью функции *rnd* введите 50 случайных чисел из отрезка $[0,2]$. Постройте функции сглаживания данных (с помощью различных встроенных функций).

3. С помощью функции *predict* предскажите поведение функции $f(x) = \sin(2 \cdot x)$ на $[2\pi; 4\pi]$, если предположить, что она задана на отрезке $[0; 2\pi]$. Изменится ли предсказанное поведение, если в качестве исходных данных взять функцию, определенную на $[0; \pi]$?

Составьте отчет по лабораторной работе №7 (см. Приложение А – Пример оформления отчета). Сделайте соответствующие выводы.

7 Учебная задача «Разработка и исследование алгоритма оценки временного положения сигнала при наличии шума»

В современном мире задача обнаружения объектов каким-либо устройством с помощью различных сигналов (определение расстояния до объекта, скорость объекта и т.д.) играет большую роль и часто оказывается очень сложной. Из-за наличия всевозможных шумов и помех, большого расстояния до цели достаточно трудно оценить характеристики сигнала. Шумы обуславливают случайный характер результатов наблюдения, и поэтому наиболее подходящим способом изучения этих явлений является статистический метод. Каждое приемное устройство является фильтром, выделяющим полезную информацию из принимаемой смеси сигнала и шума.

Целью данной учебной задачи является оценка временного положения сигнала при наличии шума, нахождение энергии полезного сигнала при заданных значениях, определение среднеквадратичной ошибки оценки (СКО), построение гистограммы распределения оценок, улучшение навыков работы в среде Mathcad.

7.1 Полезный сигнал и его параметры

Пусть полезный сигнал задается уравнением:

$$s(t) = a \cdot (t - t_0)^m \cdot \exp\left[\frac{-(t - t_0)^q}{2 \cdot b}\right], t \geq t_0,$$

где a – константа, которая задается пользователем (она определяет амплитуду сигнала, то есть его максимальное значение);

t_0 – временное положение сигнала;

t – временной параметр функции, определяющий сигнал.

$$s(t) := \begin{cases} \left[a \cdot (t - t_0)^m \cdot \exp \left[\frac{-(t - t_0)^q}{2 \cdot b} \right] \right] & \text{if } t \geq t_0 \\ 0 & \text{otherwise} \end{cases}$$

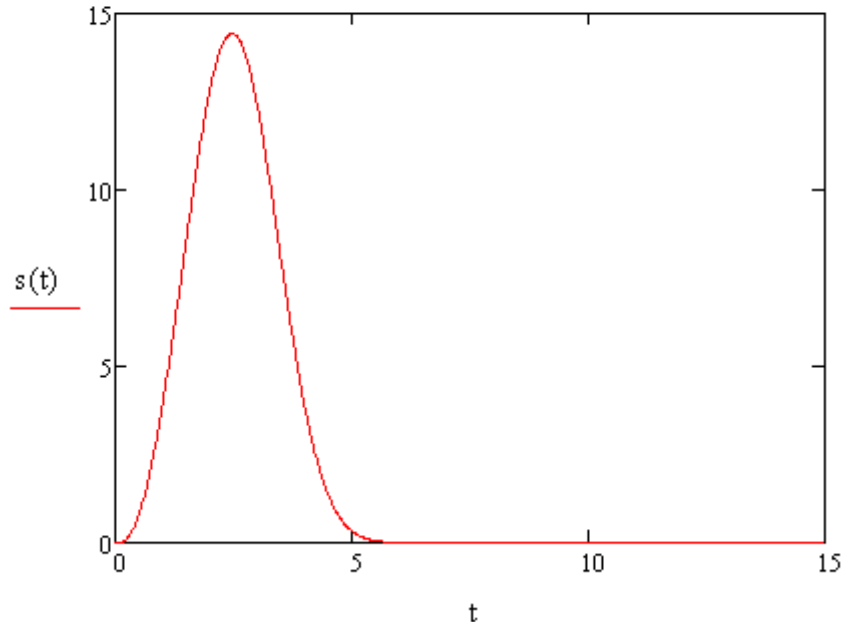


Рисунок 7.1 – График полезного сигнала

Пока речь идет о постоянном времени, на самом деле для наложения шума на сигнал и определения временного положения сигнала нужно перейти к дискретному времени.

Сигнал в дискретном времени будет выглядеть следующим образом:

$$s(k) = a \cdot (k \cdot \Delta t - v \cdot \Delta t)^m \cdot \exp \left[\frac{-(k \cdot \Delta t - v \cdot \Delta t)^q}{2 \cdot b} \right], k \geq \frac{t_0}{\Delta t},$$

где k – дискретное время;

v – дискретное значение временного положения;

Δt – шаг, с которым идет разбиение интервала.

Переход к дискретному времени выполнить самостоятельно.

7.1.1 Длительность сигнала

Важной характеристикой сигнала является его длительность. Для начала нужно определить t_u : для этого надо взять уровень 0.1 от максимального, методом перебора определить точки пересечения t_1 и t_2 , тогда длительность сигнала будем определять как разность между этими значениями. Максимальный уровень сигнала найдем, взяв производную от функции сигнала и приравняв ее к нулю, получим значения t , подставив в функцию, найдем максимум.

$$L = 0.1 \cdot a$$

Решить простое уравнение:

$$s(t) = L \text{ или } a \cdot (t - t_0)^m \cdot \exp\left[\frac{-(t - t_0)^q}{2 \cdot b}\right] = 0.1 \cdot a$$

Решить данное уравнение можно несколькими способами:

- Используя функцию $root(f(x), x)$;
- Используя функцию $solve$.

Находим корни уравнения: t_1 и t_2 . Далее находим $tu = t_1 - t_2$.

7.1.2 Энергия сигнала

Как и любая электромагнитная волна, сигнал имеет энергию. Эта энергия определяется по следующей формуле (как площадь подынтегральной кривой):

$$E(s) = \int_{t_1}^{t_2} (s(t))^2 dt$$

7.2 Случайный шумовой процесс

В реальности вместе с полезным сигналом всегда присутствует шум. Из-за него и возникают помехи, трудности передачи и соответственно затрудняется прием, также нужно учесть, что шум всегда разный, следовательно, смесь сигнала с шумом будет всегда разной – в этом и состоит вся сложность данной работы.

7.2.1 Параметры, определяющие шум

Шум имеет следующий вид:

$$n_{(k+1)} = (1 - p) \cdot n_k + \sigma \cdot \sqrt{2 \cdot p} \cdot x_k,$$

где p – интенсивность шумового процесса;

σ – амплитуда шумового процесса;

x_k – случайная составляющая шумового процесса, задается с помощью функции rnd .

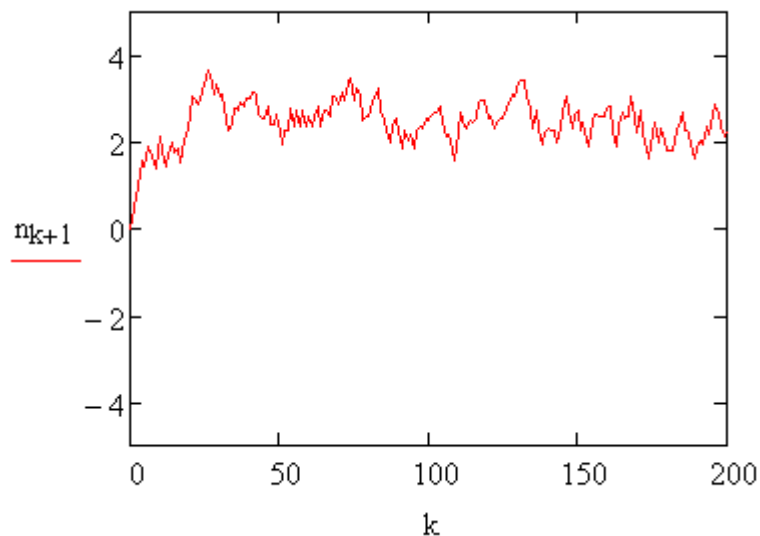


Рисунок 7.2 – График шумового процесса

7.3 Генерация смеси сигнала и шума

Как было сказано раньше, всегда имеется смесь положительного сигнала с шумом. Смесь определяется по следующей формуле:

$$z(k) = s(k) + n_{(k+1)},$$

где $s(k)$ – полезный сигнал;

$n_{(k+1)}$ – шумовой процесс.

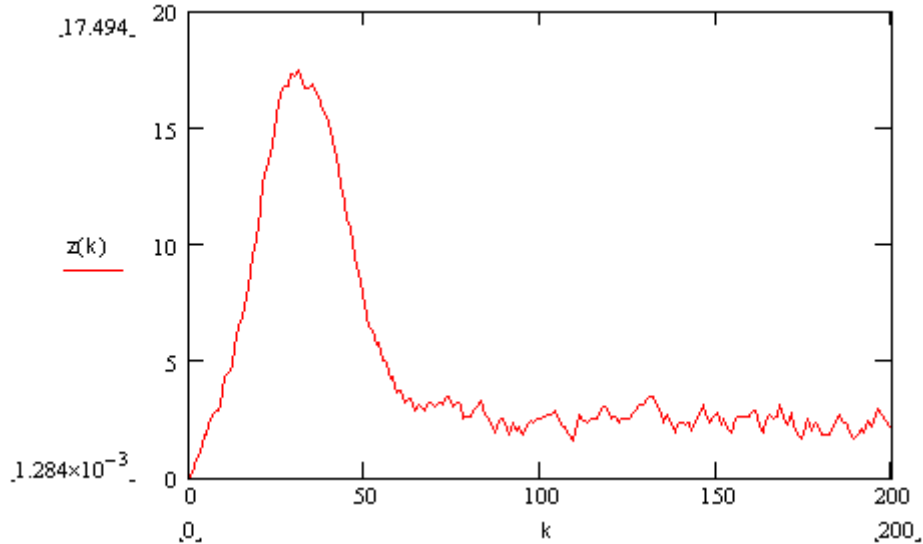


Рисунок 7.3 – График смеси полезного сигнала с шумом

Получившаяся смесь трудна для дальнейшего анализа из-за влияния шума. Можно увеличить полезный сигнал (амплитуду) или уменьшить шум, тогда влияние шума будет меньше, но в действительности этого не происходит, т.к. пока еще не научились передавать полезный сигнал, значительно превышающий шум, и полностью избавляться от влияния шума, поэтому применяют сглаженную смесь, определяемую следующей формулой:

$$z(k) = \frac{1}{m} \cdot \sum_{j=1}^m z(k-j),$$

где j – начальное значение отсчета;

m – значение, которое определяет «качество» сглаживания. Чем оно больше, тем сглаживание лучше.

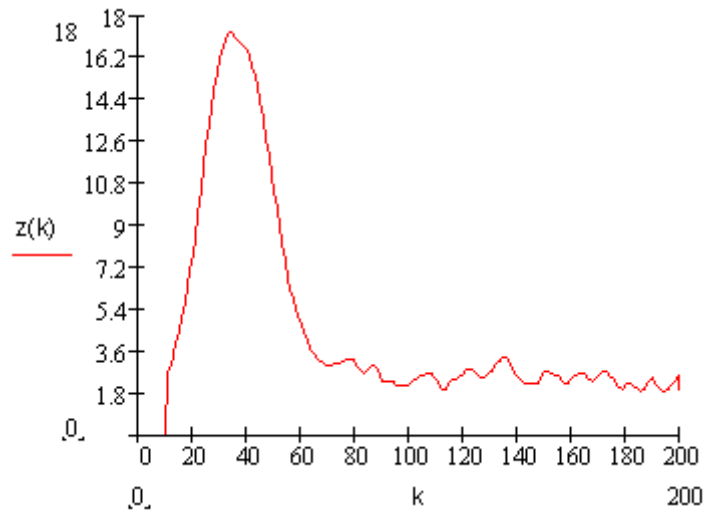


Рисунок 7.4 – График сглаженной функции

7.4 Оценка временного положения полезного сигнала

7.4.1 Алгоритм нахождения временного положения полезного сигнала

Существует несколько способов определения временного положения, самый качественный – метод наименьших квадратов. Кстати, от того, насколько хорош используемый метод, зависит качество определения временного положения.

Предложить свой метод обработки сигнала самостоятельно.

В результате должно получиться одно временное положение. Конечно, по одному измерению нельзя сделать соответствующие выводы и расчеты. Поэтому данный алгоритм повторяется много раз, т.е. составляется большой цикл (так называемые N реализации). В результате работы программы должен получиться массив значений, размер которого определяется числом измерений (реализаций). Эти значения далее подвергаются обработке.

7.5 Статистическая обработка результатов временного положения сигнала

Получив массив, состоящий из значений временного положения смеси сигнала с шумом, можно начать его анализ, а именно: построить гистограмму распределения оценок временного положения и рассчитать среднеквадратичную погрешность вычислений.

7.5.1 Построение гистограммы

В результате реализаций шума N раз получают различные оценки положения сигнала. Все значения попадают в интервалы, лежащие рядом с истинной оценкой, определенной нами из графика функции сигнала. Для того чтобы увидеть, в какие интервалы попадают значения оценок, необходимо построить гистограмму распределения. Воспользуемся функцией $histogram(M)$, которая определяет интервал между минимальным и максимальным значениями оценок, затем разбивает этот интервал на M

интервалов и показывает, какое количество оценок положения в какой интервал попало. Гистограмма представлена на рисунке 7.5.

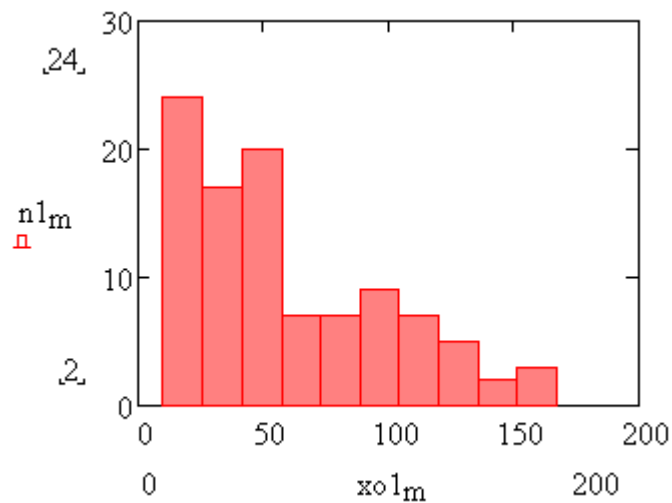


Рисунок 7.5 – Гистограмма распределения оценок положения сигнала

7.5.2 Расчет среднеквадратичного отклонения (СКО)

Т.к. временное положение смеси сигнала с шумом всегда разное, из-за влияние шума, то найденные значения будут «колебаться» возле действительного, определяемого из графика сигнала, следовательно, временное положение будет определяться с некоторой погрешностью, определяемой по следующей формуле:

$$\delta = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^N (\varepsilon_i)^2},$$

где N – число опытов;

ε_i – погрешность измерения одного опыта.

7.5.3 Зависимость СКО от отношения сигнал/шум

Увеличивая амплитуду сигнала, увеличивается его максимальное значение, поэтому выделить сигнал из шума становится легче, следовательно, ошибка определения положения сигнала должна уменьшаться, что и показано на рисунке 7.6.

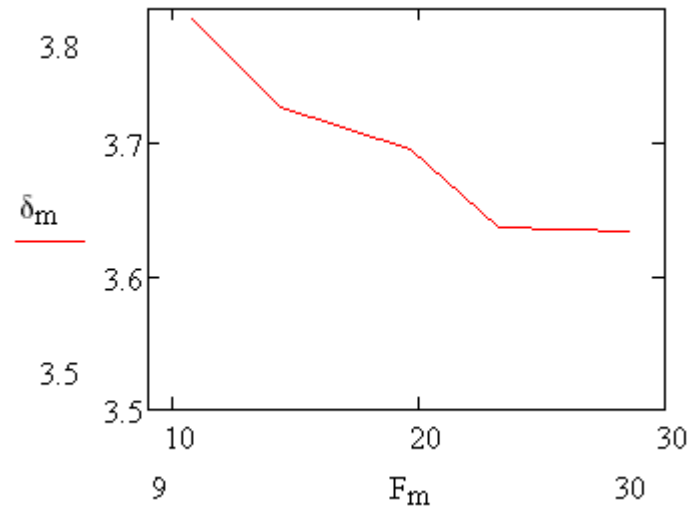


Рисунок 7.6 – Зависимость СКО от отношения сигнал/шум

Лабораторная работа № 8 «Разработка и исследование алгоритма оценки временного положения сигнала при наличии шума»

Примечание: лабораторная работа рассчитана на 8 часов.

Цель работы: получить навыки работы в Mathcad при моделировании полезного сигнала на фоне шумового процесса.

Задание на лабораторную работу:

1. Исследовать сигнал, в п. 7.1
2. Исследовать следующие виды сигналов:

$$s(t) = a \cdot (t - t_0)^m \cdot \sin^q \left(\frac{2\pi \cdot (t - t_0)}{b} \right); t_0 \leq t \leq \left(t_0 + \frac{b}{2} \right);$$

$$s(t) = \frac{a_1}{[a_2 + a_3 \cdot (t - t_0)^{2 \cdot m}]}, t_0 \leq t \leq 4 \cdot T_u; T_u < t_0 \leq 3.5 \cdot T_u,$$

параметры a , m , q , a_1 , a_2 , a_3 выбираются пользователем самостоятельно.

3. Составить блок-схему программы
4. Выполненное задание должно содержать следующий вид графического материала:

- а) График полезного сигнала;
- б) График полезного сигнала и порогом;
- в) График шумового процесса;
- г) График смеси полезного сигнала и шума;
- д) График сглаженной смеси полезного сигнала и шума;
- е) График гистограмма распределения оценок положения сигнала;
- ж) График зависимости СКО от отношения сигнал/шум

Составьте отчет по лабораторной работе №8 (см. Приложение А – Пример оформления отчета). Сделайте соответствующие выводы.

Список использованной литературы

1. Д. А. Гурский. Вычисления в Mathcad. – Минск: Новое знание, 2003. – 813, [3] с. : ил., табл. – Предм. указ.: с. 801-813. – ISBN 985-475-013-2
2. Дьяконов В.П. Самоучитель Mathcad. – С-Пб.: Питер, 2001, – 643 с.
3. Е. М. Кудрявцев. Mathcad 8. Символьное и численное решение разнообразных задач. – М.: ДМК, 2000. - 320 с.: ил. – ISBN 5-89818-058-3
4. Крушель Е.Г., Панфилов А.Э. Осваиваем Mathcad (первокурсникам, заочникам и не только...): Учеб. пособие/ ВолгГТУ, Волгоград, 2006 – 179с.
5. Электронный ресурс:
[<http://www.exponenta.ru/soft/mathcad/usersguide/0.asp>] – Математический сайт Exponenta.ru, раздел Mathcad.
6. Электронный ресурс:
[<http://detc.ls.urfu.ru/assets/amath0021/soder.htm>] – Электронный учебник по Mathcad

Приложение А

Пример оформления отчета

Министерство образования и науки
Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра радиотехнических систем (РТС)

НАЧАЛЬНЫЕ СВЕДЕНИЯ О МАТНСАД

Отчет по лабораторной работе № 1
по дисциплине «Информационные технологии 1»

Студент гр. 124-1
_____ П.П. Петров

Руководитель
Старший преподаватель
каф. РТС
_____ Д.О. Ноздревых

Цель работы: познакомиться с программным продуктом Mathcad: интерфейсом, произвести основные вычисления.

Ход работы:

Теоретическая часть лабораторной работы:

Описание основных команд, функций, которые использовались.

Практическая часть лабораторной работы:

Выполнение индивидуального задания/ задания на лабораторную работу.

Заключение:

Выводы о проделанной работе и полученных результатах.

Список использованных источников.

Примечание: отчет по лабораторной работе оформляется по Образовательному стандарту ТУСУРа

[http://tusur.ru/export/sites/ru.tusur.new/ru/education/documents/inside/tech_01-2013_new.pdf]

или

ТУСУР/ Образование/ Нормативное обеспечение/ Нормативные акты ТУСУРа/ Нормативные акты ТУСУРа/ Организация учебного процесса/14.1