

Федеральное Агентство по образованию
Амурский государственный университет

Н.П.Семичевская, Л.А.Соловцова

Обработка изображений в среде MATLAB
с использованием
IMAGE PROCESSING TOOLBOX

Лабораторный практикум

Благовещенск, 2005
СОДЕРЖАНИЕ

Введение	3
Вводное лабораторное занятие. Справочные системы и информационные ресурсы математических пакетов	5
Лабораторная работа № 1. Преобразование и вычисление функций ...	6
Лабораторная работа № 2. Расчет реакций опор нагруженной балки ...	8
Лабораторная работа № 3. Моделирование траектории полета тела ...	11
Лабораторная работа № 4. Моделирование процесса работы нагревательного устройства с терморегулятором	14
Лабораторная работа № 5. Обработка данных эксперимента	17
Лабораторная работа № 6. Имитационное моделирование вещественной частотной характеристики комплексной диэлектрической проницаемости композиционного материала	21
Требования к оформлению отчетов	28
Рекомендуемая литература	28
Приложение А. Математический пакет <i>MathCAD 200 Professional</i>	29
А.1. Интерфейс пользователя	29
А.2. Построение графиков	32
А.3. Математические расчеты	43
А.4. Программирование	53
А.5. Системный интегратор <i>MathConnex</i>	56
А.6. Справочная система и информационные ресурсы	63
Приложение Б. Пакет символьной математики <i>Maple V</i>	65
Б.1. Интерфейс пользователя	65
Б.2. Математические расчеты	70
Б.3. Графика и анимация	75
Б.4. Процедурное программирование	81
Б.5 Справочная система	84
Приложение В. Матричная лаборатория <i>MATLAB 6.5</i>	85
В.1. Характеристика пакета	85
В.2. Запуск и командный режим работы	87
В.3. Интерфейс пользователя	88
В.4. Решение дифференциальных уравнений	91
В.5. Справочная система	93

Введение

Персональные компьютеры (ПК или ПЭВМ) класса *IBM PC, XT, AT* и *PS-2* – быстро развивающееся направление вычислительной техники. Сферы их применения поистине неисчерпаемы и охватывают подготовку и обработку текстов, финансово-экономические расчеты, ведение баз данных, подготовку проектных документов и т.д. Среди всего обилия областей применения математические и научно-технические расчеты, т.е. та сфера, ради которой компьютеры (от слова *compute* – вычисления) и были созданы, уже не обладает неоспоримым приоритетом. Тем не менее, такие расчеты продолжают занимать видное место в применении ПК, особенно в научно-исследовательских организациях и учебных заведениях. При этом автоматизации решения прикладных задач может выполняться с привлечением целого ряда различных программных средств.

Термин «пакеты прикладных программ» (ППП) обычно применяется к комплексам программ различной сложности и назначения, которые представляют собой совокупность совместимых программных продуктов для решения задач определенного класса – текстовых, графических, математических, обработки данных и т.п. В процессе освоения учебных дисциплин таких как «Электротехника и электроника», «Основы теории управления», «Теоретические основы автоматизированного управления», «Надежность АСОИУ», а также ряда других студенты сталкиваются с большим количеством разнообразных, но, в общем, типовых вычислений. Для автоматизации выполнения подобных расчетов неоценимую помощь оказывает знание пакетов математических программ и умение применять их для решения практических задач.

Настоящее пособие предназначено для обучения автоматизации решения инженерно-физических задач с помощью пакетов *MathCAD 2000 Professional*, *Maple V* и *MatLAB 5.2*, а также системного интегратора *MathConnex*. Структура пособия включает непосредственно сам лабораторный практикум, а также приложения, содержащие справочную информацию по каждому из разбираемых программных продуктов, необходимую для выполнения предлагаемых лабораторных работ. Содержание практикума направлено на формирование навыков в автоматизации решения инженерно-физических задач из ряда областей, в частности – статики, кинематики, термодинамики, обработки данных эксперимента и физики диэлектриков. Все лабораторные работы выполняются с привлечением возможностей различных математических пакетов при одних и тех же исходных данных решаемой задачи, что позволяет студенту независимо оценить достоинства и недостатки каждого из изучаемых программных средств.

ВВОДНОЕ ЛАБОРАТОРНОЕ ЗАНЯТИЕ

Тема. Справочные системы и информационные ресурсы математических пакетов.

Цель. Ознакомление с наполнением встроенных электронных справочников и примерами решения задач.

Задание 1

1. Активизируйте программную среду *MathCAD 2000 Professional*.
2. Войдите в позицию *Help (?)* главного меню и выберите команду *Центр Ресурсов (Resource Center)* или запустите его с помощью соответствующей пиктограммы на инструментальной панели. В появившемся окне центра информационных ресурсов войдите в раздел *Справочный стол и краткое руководство (QuickSheets and Reference Tables)*.
3. Просмотрите примеры использования возможностей *MathCAD* при решении задач, приведенные в подразделе *Шпаргалки (QuickSheets)*.
4. Активизируйте подраздел *Справочный стол (Reference Tables)*. Ознакомьтесь с составом представленных в справочнике сведений.
5. Запустите системный интегратор *MathConnex*. Найдите папку *Samples (Примеры)* и откройте в ней директорию *MathConnex*. Просмотрите примеры использования возможностей системного интегратора *MathConnex*.

Задание 2

2. Активизируйте среду пакета символьных вычислений *Maple V*.
3. Войдите в позицию *Help (Помощь)* главного меню, выберите команду *Contents (Содержание)*. В появившемся окне встроенного справочника просмотрите интересующие разделы.

Задание 3

1. Активизируйте программную среду пакета *MatLAB 5.2*.
2. Загрузите демонстрационные файлы, отражающие особенности работы в системе, с помощью команды *demo* (*Примеры и демонстрации*).
3. Просмотрите содержание *Окна помощи (helpwin)* и *Стола помощи (helpdesk)*.

ЛАБОРАТОРНАЯ РАБОТА № 1

Тема. Преобразование и вычисление функций.

Цель. Использование возможностей математических пакетов при нахождении символьного и численного решений для функциональных зависимостей в режиме прямых вычислений.

Вопросы текущего контроля знаний

1. Дайте определение производной от функции. Поясните ее физическую сущность.
2. Раскройте понятия определенного и неопределенного интегралов.

Исходные данные

1. Выполнить дифференцирование функций:
 - a) $f(x) = \sin^2(x)$ – найти первую производную;
 - b) $f(x, y) = \cos(x)y + \sin(y)x$ – найти первые производные по x и y ;
 - c) $f(x) = x^2 \cos(x^3)$ – найти вторую производную.

1. Вычислить интегралы:

a) $\int x^2 dx$;

b) $\int_0^{10} x dx$;

c) $\int_{-\pi}^{\pi} \cos(x^2) dx$;

1. Найти пределы:

a) $\lim_{x \rightarrow \infty} \frac{1}{x^2}$;

b) $\lim_{x \rightarrow 0^-} \frac{1}{x}$.

Пакет MathCAD

Задание

1. Выполните требуемые преобразования с использованием возможностей встроенного символьного процессора – команд из позиции *Символы (Symbolic)* главного меню.

2. Выполните требуемые преобразования с помощью шаблонов из палитры *Математический анализ*.

3. Оформите отчет согласно требований, приведенных на стр. 28.

Пакеты Maple или MatLAB

Задание

Выполните предлагаемые преобразования и вычисления всеми способами, доступными в используемой программной среде.

ЛАБОРАТОРНАЯ РАБОТА № 2

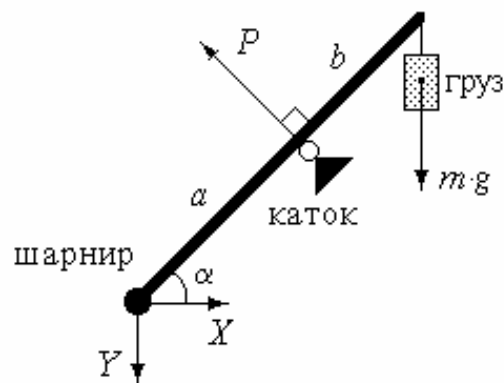
Тема. Расчет реакций опор нагруженной балки.

Цель. Решение систем линейных алгебраических уравнений.
Использование графических возможностей математических пакетов при исследовании функциональных зависимостей.

Вопросы текущего контроля знаний

1. Охарактеризуйте основные аналитические и численные методы решения систем линейных алгебраических уравнений.
2. В какой размерности в математических пакетах используется значение аргумента тригонометрической функции по умолчанию?

Исходные данные



Рассматривается балка, шарнирно закрепленная у своего левого края. В средней части она подперта катком. Справа подвешен груз. Требуется: найти реакции опор X , Y и P , если известна масса груза m , а также геометрия балки (плечи a , b и угол α); исследовать зависимость величины реакции опор от изменения α и m .

Задача сводится к нахождению условий равновесия балки, т.е. решению системы линейных алгебраических уравнений, описывающих балансы действующих на нее сил и их моментов:

$$\begin{cases} P \cdot \sin(\alpha) = X & \text{– баланс сил по оси } X \\ Y + m \cdot g = P \cdot \cos(\alpha) & \text{– баланс сил по оси } Y \\ P \cdot a = m \cdot g \cdot (a + b) \cdot \cos(\alpha) & \text{– момент сил по шарниру} \end{cases} \quad (2.1)$$

где g – ускорение свободного падения.

Таблица 1

Первая цифра номера варианта			Вторая цифра номера варианта		
№	a (м)	b (м)	№	α (град)	m (кг)
1	0,1	0,9	1	20	5
2	0,2	0,8	2	25	6
3	0,3	0,7	3	30	7
4	0,4	0,6	4	35	8
5	0,5	0,5	5	40	9
6	0,6	0,4	6	45	8
7	0,7	0,3	7	50	7
8	0,8	0,2	8	55	6
9	0,9	0,1	9	60	5
10	1,0	0,5	10	65	4

Пакет MathCAD

Задание 1

1. Решите систему уравнений (2.1), используя расчетный блок “Given... Find” и физические размерности исходных данных.
2. Проверьте достоверность полученного результата расчетами балансов сил по оси P и балке, а также момента сил относительно катка.

Замечание. Каждое новое задание рекомендуется выполнять, создавая отдельный файл.

Задание 2

1. Представьте систему (2.1) в векторно-матричной форме записи.
2. Вычислите вектор искомых значений реакций опор.
3. Повторите решение, используя функцию *lsolve*.

Замечание. Для ввода встроенных функций можно использовать соответствующий Мастер из панели инструментов.

Задание 3

1. Используя меню символьных преобразований, получите аналитическое решение системы уравнений (2.1).
2. Сформируйте функции $X(\alpha)$, $Y(\alpha)$ и $P(\alpha)$.
3. На основании своих исходных данных о геометрии балки и массе груза, постройте для полученных функций прямоугольный декартов и полярный графики при изменении α от 0 до 360 град.
4. Сформируйте функцию $X(\alpha, m)$.
5. Задайте число и нумерацию узловых точек по осям. Рассчитайте векторы аргументов функции для узловых точек и матрицу, содержащая результаты вычислений аппликат поверхности в узловых точках
6. На основании полученной матрицы аппликат постройте для функции $X(\alpha, m)$ график поверхности, карту линий уровня, векторное поле, трехмерный точечный график, а также трехмерную столбчатую гистограмму при $m = 0, 0.5, \dots, 10$ кг и $\alpha = 0, 9, \dots, 180$ град.

Пакеты Maple или MatLAB

Задание 1

1. Решите систему уравнений (2.1) в символьном виде.
2. Найдите численное решение системы, используя соответствующие программные средства.
3. Решите систему векторно-матричным способом.

Задание 2

1. Постройте для функций $X(\alpha)$, $Y(\alpha)$ и $P(\alpha)$ декартов и полярный графики при изменении α от 0 до 360 град.
2. Постройте для функций $X(\alpha, m)$ трехмерный график поверхности при изменении α от 0 до 180 град и m от 0 до 10 кг.

Замечание: Графики должны быть оформленными, т.е. содержать координатную сетку, метки осей и т.п.

ЛАБОРАТОРНАЯ РАБОТА № 3

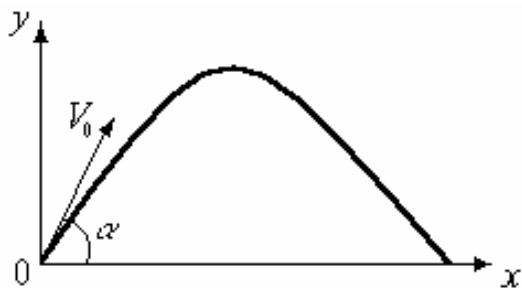
Тема. Моделирование траектории полета тела.

Цель. Реализация анимационных возможностей математических пакетов. Нахождение оптимального решения.

Вопросы текущего контроля знаний

1. Поясните назначение системной переменной *FRAME*.
2. Что означает существование значений аргумента функции, при которых значение ее первой производной равно нулю?

Исходные данные



Тело, рассматриваемое как материальная точка, брошено под углом α к горизонту с начальной скоростью V_0 . Требуется: найти время его полета и дальность броска, считая, что сопротивление воздуха отсутствует; определить значение α , при котором дальность броска будет максимальной.

Решение рассматриваемой задачи кинематики базируется на уравнениях траектории полета тела, описывающих его координаты в любой момент времени:

$$\begin{cases} x(t) = V_0 \cdot \cos(\alpha) \cdot t \\ y(t) = V_0 \cdot \sin(\alpha) \cdot t - g \cdot \frac{t^2}{2} \end{cases} \quad (3.1)$$

где g – ускорение свободного падения.

Таблица 2

Первая цифра номера варианта				Вторая цифра номера варианта			
№	α (град)	№	α (град)	№	V_0 (м/с)	№	V_0 (м/с)
1	30	6	43	1	15	6	27
2	33	7	47	2	17	7	30
3	35	8	50	3	20	8	33
4	37	9	53	4	23	9	35
5	40	10	55	5	25	10	37

Пакет MathCAD

Задание 1

1. На основании условия $y(t) = 0$, используя команду *solve* из меню символьных преобразований, найдите время полета тела.
2. Рассчитайте дальность броска.
3. Создайте анимацию движения тела в реальном масштабе времени.

Замечание. Сценарий вывода анимации должен отражать изображения, как траектории полета, так и непосредственно самого тела.

Задание 2

1. Сформируйте функциональную зависимость $x(\alpha)$ дальности броска от угла бросания.
2. Постройте график $x(\alpha)$ для значений угла α от 0 до 90 град.
3. Используя расчетный блок “*Given... Minerr*”, найдите оптимальный угол бросания, при котором дальность полета тела будет максимальной.

Задание 3

1. Постройте график первой производной функции $x(\alpha)$.
2. Рассчитайте оптимальный угол бросания с помощью корней уравнения первой производной функции $x(\alpha)$, используя функцию – *root*.

Пакеты Maple или MatLAB

Задание 1

1. Найдите символьное и численное решения для времени и дальности.
2. Реализуйте анимацию полета тела.

Задание 2

1. Найдите первую производную функции $x(\alpha)$.
2. Определите оптимальный угол для наибольшей дальности броска.

ЛАБОРАТОРНАЯ РАБОТА № 4

Тема. Моделирование характеристик работы нагревательного устройства с терморегулятором.

Цель. Применение типовых элементов программирования математических пакетов. Использование операторов решения обыкновенных дифференциальных уравнений.

Вопросы текущего контроля знаний

1. Раскройте понятие функционального программирования.
2. Поясните назначение и структуру типовых конструкций процедурного программирования.

Исходные данные

Требуется получить временную зависимость температуры нагревательного устройства, при мощности нагревателя P , массе устройства m и его удельной теплоемкости $C = 400 \text{ Дж}/(\text{кг}\cdot\text{K})$. В устройстве есть терморегулятор, который включает его, если $T < T_{min}$, и выключает, если $T > T_{max}$. Потери тепла происходят по закону:

$$\frac{dQ_1}{dt} = A \cdot (T - T_{cp}), \quad (4.1)$$

где T – температура нагревательного устройства, T_{cp} – температура окружающей среды (например, $20^\circ \text{C} = 293^\circ \text{K}$), $A = 2 \text{ Вт}/\text{K}$, а также за счет излучения в соответствии с законом Стефана-Больцмана:

$$\frac{dQ_2}{dt} = S \cdot (T^4 - T_{cp}^4), \quad (4.2)$$

где $S = 3,4 \cdot 10^{-9} \text{ Вт}/\text{K}^4$. Дифференциальное уравнение, описывающее нарастание температуры с течением времени, можно получить, исходя из того,

что подводимая к устройству энергия $P \cdot dt$ расходуется на его нагрев dQ и на потери в окружающую среду:

$$\begin{aligned} P \cdot dt &= dQ + dQ_1 + dQ_2, \\ dQ &= C \cdot m \cdot dT. \end{aligned} \quad (4.3)$$

Данная задача представляет интерес с точки зрения применения для ее решения программного модуля, учитывающего действие терморегулятора и использующего итерационную формулу Эйлера:

$$T_i = T_{i-1} + \Delta t \cdot f\left(T_{i-1} + \Delta t \cdot \frac{f(T_{i-1}, P)}{2}, P\right), \quad (4.4)$$

где Δt – шаг приращения времени; $f(T, P)$ – функция, определяющая производную температуры по времени с учетом мощности нагревателя.

Таблица 3

Первая цифра номера варианта			Вторая цифра номера варианта		
№	P (кВт)	m (кг)	№	T_{min} ($^{\circ}C$)	T_{max} ($^{\circ}C$)
1	1,5	2,0	1	300	350
2	1,6	2,1	2	300	345
3	1,7	2,2	3	300	340
4	1,8	2,3	4	290	340
5	1,9	2,4	5	290	335
6	2,0	2,5	6	290	330
7	2,1	2,6	7	310	360
8	2,2	2,7	8	310	355
9	2,3	2,8	9	310	350
10	2,4	2,9	10	300	400

Пакет MathCAD

Задание 1

1. Введите ваши исходные данные, учитывая, что температура должна задаваться в градусах Кельвина ($0^\circ C = 273^\circ K$).

2. На основании уравнений (4.1)-(4.3) запишите дифференциальное уравнение, описывающее нарастание температуры нагревательного устройства с течением времени.

3. Задайте шаг приращения времени $\Delta t = 1$ и число шагов решения $N = 500$. Сформируйте функцию $f(T, P)$.

4. Создайте программный модуль, реализующий вычисление температуры устройства по итерационной формуле (4.4) с учетом работы терморегулятора.

5. Постройте график временной зависимости температуры в диапазоне заданных шагов решения.

Замечание. Программные операторы должны вводиться с панели инструментов, набирать их на клавиатуре не рекомендуется.

Задание 2

1. Решите рассматриваемую задачу без учета работы терморегулятора, используя встроенную функцию $Bulstoer(y, x1, x2, n, F)$.

2. Учитывая состав данных в полученной матрицы решений, постройте график временной зависимости температуры для диапазона заданных шагов решения.

Задание 3

1. Выведите совместно графики временных зависимостей температуры устройства с нагревателем и без него.

Пакеты Maple или MatLAB

Задание 1

1. Сформируйте программу вычисления температуры устройства по итерационной формуле (4.4) с учетом работы терморегулятора.
2. Постройте график временной зависимости температуры в диапазоне заданных шагов решения.

Задание 2

1. Решите рассматриваемую задачу без учета работы терморегулятора, используя встроенные решатели ОДУ.
2. Постройте график решения.

Задание 3

2. Выведите совместно графики временных зависимостей температуры устройства с нагревателем и без него.

ЛАБОРАТОРНАЯ РАБОТА № 5

Тема. Обработка данных эксперимента.

Цель. Использование возможностей математических пакетов прикладных программ для аппроксимации данных.

Вопросы текущего контроля знаний

1. Дайте определения понятий «интерполяция» и «экстраполяция».
2. Перечислите наиболее популярные методы аппроксимации данных.

Исходные данные

Для представления физических закономерностей и при проведении научно-технических расчетов часто используются зависимости вида $y(x)$, причем число точек этих зависимостей ограничено. Неизбежно возникает задача приближенного вычисления значений функций в промежутках между узловыми точками (интерполяция) и за их пределами (экстраполяция). Эта задача решается аппроксимацией исходной зависимости, т. е. ее заменой какой-либо достаточно простой функцией.

При проведении физического эксперимента были получены данные зависимости $y(x)$, приведенные в табл. 4. Требуется аппроксимировать результаты эксперимента.

Таблица 4

Первая цифра номера варианта					Вторая цифра номера варианта						
№	x_1	x_2	x_3	x_4	x_5	№	y_1	y_2	y_3	y_4	y_5
1	1	0	2	4	3	1	50	0	20	10	5
2	1,5	0,5	2,5	4,5	3,5	2	55	0	20	10	5
3	2	1	3	5	4	3	50	5	20	10	5
4	2,5	1,5	3,5	5,5	4,5	4	55	5	20	10	5
5	3	2	4	6	5	5	50	0	25	10	5
6	3,5	2,5	4,5	6,5	5,5	6	55	0	25	10	5
7	4	3	5	7	6	7	50	0	20	15	5
8	4,5	3,5	5,5	7,5	6,5	8	55	0	20	15	5
9	5	4	6	8	7	9	50	0	20	15	10
10	5,5	4,5	6,5	8,5	7,5	10	55	0	20	15	10

Пакет MathCAD

Задание 1

1. Введите ваши исходные данные в виде матрицы. Используя встроенную функцию $csort(A, n)$, отсортируйте матрицу в порядке возрастания значения аргумента x .
2. Создайте векторы X и Y , содержащие значения аргумента и соответствующие им значения функции. Задайте i – число элементов в X и Y , используя встроенную функцию $length(v)$.
3. Выполните линейную и сплайн-кубическую интерполяции. Проведите пробные расчеты для проверки полученных функций. Сравните их результаты в интерполирующей и экстраполирующей областях.
4. Постройте графики, отражающие вид кривых аппроксимирующих функций с наложением на них данных эксперимента.

Задание 2

1. Введите векторы X и Y . Задайте ранжированные переменные i и j по числу элементов вектора X .
2. Найдите аппроксимирующую функцию, выразив ее в виде полинома на основании общей формулы интерполяции Лагранжа (5.1), задающей в явном виде значение интерполирующей функции f для любого значения x . Проведите пробные расчеты и постройте график.

$$f(x) := \sum_i Y_i \cdot \prod_j \text{if} \left(i = j, 1, \frac{x - X_j}{X_i - X_j} \right). \quad (5.1)$$

Замечание. Полином Лагранжа является достаточно сложной функцией. В ряде случаев для аппроксимации исходной зависимости применяют обычный степенной многочлен – полиномиальную аппроксимацию.

Задание 3

1. Введите векторы X и Y . Вычислите степень аппроксимирующего полинома n .

2. Сформируйте матрицу XI : $i = 0, \dots, n$; $j = 0, \dots, n$; $XI_{j,i} = X_j^i$, $XI_{j,0} = 1$. Просмотрите, что она собой представляет.

3. Вычислите коэффициенты полинома $a = XI^{-1} \cdot Y$.

4. Задайте формулу аппроксимирующего полинома в виде зависимости:

$$P(x) = \sum_k a_k \cdot x^k, k = n, \dots, 0.$$

Проведите пробные расчеты. Введите ранжированную переменную: $x_1 = x_1 - 1$, $x_2 = 0.9$, $x_3 = 1$ и постройте график $P(x_i)$.

Замечание. Поведение полинома между узловыми точками и при экстраполяции зачастую трудно предсказуемо. Кроме того, с ростом степени (n до 8-10) погрешность полиномиальной аппроксимации резко возрастает.

Пакеты Maple или MatLAB

Задание 1

1. Аппроксимируйте исходную зависимость отрезками полиномов третьего порядка.

2. Постройте аппроксимирующий график.

Задание 2

1. Выполните полиномиальную аппроксимацию для данных исходной зависимости.

2. Постройте соответствующий аппроксимирующий график.

ЛАБОРАТОРНАЯ РАБОТА № 6

Тема. Имитационное моделирование вещественной частотной характеристики комплексной диэлектрической проницаемости композиционного материала в области упругой электронной поляризации.

Цель. Совмещение возможностей нескольких пакетов для решения прикладной задачи в рамках использования системного интегратора.

Вопросы текущего контроля знаний

1. Дайте определение передаточной функции.
2. Каково максимальное число вводов и выводов одного программного блока при использовании системного интегратора *MathConnex*?

Исходные данные

Рассматривается композиционный диэлектрический материал, состоящий из двух компонентов равной концентрации, параметры которых приведены в таблице 5. Требуется построить график вещественной частотной характеристики диэлектрической проницаемости материала в области упругой электронной поляризации, а именно: в диапазоне частот от 10^{15} до 10^{18} рад/сек.

Смещение электронных облаков относительно атомных ядер, вызванное действием внешнего электрического поля – наиболее общий вид поляризации, имеющий место во всех без исключения диэлектриках, независимо от их структуры и агрегатного состояния. Поскольку внутренние электроны взаимодействуют с ядром гораздо сильнее внешних, то при рас-

смотрении смещения электронного облака можно ограничиться учетом поляризации только оптической (внешней) электронной оболочки иона.

В качестве математического описания процесса поляризации материала под действием переменного электрического поля $E_0(t)$ с малой амплитудой для исследуемого частотного диапазона может быть использована следующая система уравнений:

$$\begin{aligned} \frac{d^2 \mu_k(t)}{dt^2} + 2b_k \frac{d\mu_k(t)}{dt} + \omega_{0k}^2 \mu_k(t) &= \frac{N_k e^2}{m_e} E(t), k = \overline{1, K}, \\ E(t) &= E_0(t) - \frac{2}{3\varepsilon_0} \sum_{i=1}^K n_i \mu_i(t), \end{aligned} \quad (6.1)$$

где k – индекс разновидности иона; $\mu_k(t)$ – его индуцированный дипольный момент; b_k и ω_{0k} – соответственно, коэффициент затухания и частота собственных колебаний электронного облака иона; N_k – число электронов, составляющих оптическую оболочку иона; e и m_e – заряд и масса электрона; ε_0 – электрическая постоянная; K – число разновидностей ионов, имеющих место в конкретном диэлектрике; n_i – их концентрации.

С позиций технической кибернетики, уравнения (6.1) представляют собой математическую модель замкнутой линейной системы с отрицательной обратной связью. Выполнив для данной системы уравнений прямое интегральное преобразование Лапласа, можно эквивалентно изобразить ее в виде структурной схемы, показанной на рис. 1, или записать в стандартной форме записи через передаточные функции:

$$\begin{aligned} \mu_k(s) &= W_k(s) E(s), k = \overline{1, K}, W_k(s) = \frac{N_k e^2 / m_e}{s^2 + 2b_k s + \omega_{0k}^2}, \\ E(s) &= E_0(s) - \frac{2}{3\varepsilon_0} \sum_{i=1}^K \mu_i(s) n_i, \end{aligned} \quad (6.2)$$

где s – комплексная переменная; $\mu_k(s)$, $E(s)$ и $E_0(s)$ – изображения по Лапласу функций $\mu_k(t)$, $E(t)$ и $E_0(t)$; $W_k(s)$ – передаточные функции.

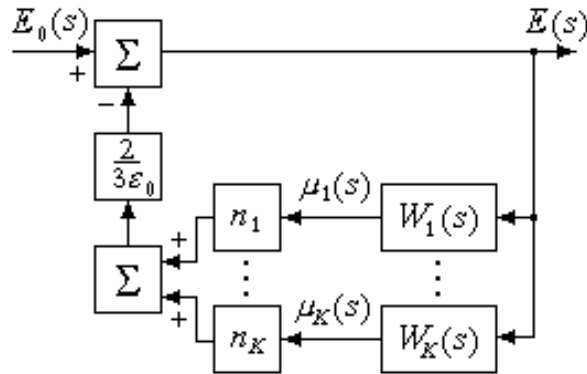


Рисунок 1. Структурная схема процесса поляризации материала

Замена $s \rightarrow j\omega$ в передаточных функциях $W_k(s)$ позволяет перейти к их частотным аналогам. Таким образом, на основании описанных математических моделей и определения комплексной диэлектрической проницаемости ($\varepsilon(j\omega) = E_0(j\omega)/E(j\omega)$) для ее представления может быть использована следующая функциональная зависимость:

$$\varepsilon(j\omega) = 1 + \frac{2}{3\varepsilon_0} \sum_{i=1}^K n_i W_i(j\omega), \quad (6.3)$$

$$W_k(j\omega) = \frac{N_k e^2 / m_e}{\omega_{0k}^2 - \omega^2 + j2b_k \omega}, k = \overline{1, K}.$$

Для имитационного моделирования спектров комплексной диэлектрической проницаемости конкретного материала необходимы численные значения параметров, участвующих в описании всех поляризационных процессов, протекающих в диэлектрике. В рассматриваемом диапазоне частот внешнего поля такими параметрами являются – b_k , ω_{0k} , N_k и n_k ионов каждой разновидности.

Количественный состав оптических (внешних) электронных оболочек ионов достаточно просто установить на основании справочных данных. Для всех ионов, входящих в состав исследуемых композитов, $N_k=8$.

При нахождении ω_{0k} можно использовать формулу:

$$\omega_{0k} = \sqrt{\frac{Q_k e^2}{4\pi\epsilon_0 m_e r_k^3}}, \quad (6.4)$$

где Q_k – эффективный заряд ядра, действующий на внешнюю оболочку и учитывающий экранизацию его полного заряда внутренними оболочками (для всех положительных ионов рассматриваемых материалов $Q_k=9$, для отрицательных – $Q_k=7$); r_k – ионный радиус.

Значения b_k , принимая во внимание силу внутреннего трения, действующую на электронный диполь с зарядом $N_k e$ при электромагнитном излучении с фазовой скоростью, равной скорости света в вакууме – c , можно рассчитать как:

$$b_k = \frac{\mu_0 N_k e^2 \omega_{0k}^2}{12\pi m_e c}, \quad (6.5)$$

где μ_0 – магнитная постоянная.

Поскольку исследуются исключительно композиты равного процентного содержания, состоящие только из двух ионов, их концентрации в материале рассчитываются по следующей формуле:

$$n_k = \frac{\rho}{(m_1 + m_2) \cdot aem} \cdot 0,5, \quad (5.6)$$

где ρ – плотность композита; m_1 и m_2 – атомные массы составляющих его ионов; aem – атомная единица массы; 0,5 – процентное содержание каждого композита согласно исходным данным.

Таблица 5

Первая цифра номера варианта					Вторая цифра номера варианта				
№	Композит	ρ (кг/м ³)	$r +$ (10 ⁻¹⁰ м)	$r -$ (10 ⁻¹⁰ м)	№	Композит	ρ (кг/м ³)	$r +$ (10 ⁻¹⁰ м)	$r -$ (10 ⁻¹⁰ м)
1	<i>LiF</i>	2632	0,68	1,33	1	<i>KF</i>	2480	1,33	1,33
2	<i>LiCl</i>	2072	0,68	1,81	2	<i>KCl</i>	1990	1,33	1,81
3	<i>LiBr</i>	3462	0,68	1,96	3	<i>KBr</i>	2752	1,33	1,96
4	<i>LiI</i>	3500	0,68	2,20	4	<i>KI</i>	3120	1,33	2,20
5	<i>NaF</i>	2790	0,98	1,33	5	<i>RbF</i>	3560	1,49	1,33
6	<i>NaCl</i>	2165	0,98	1,81	6	<i>RbCl</i>	2760	1,49	1,81
7	<i>NaBr</i>	3210	0,98	1,96	7	<i>RbBr</i>	2780	1,49	1,96
8	<i>NaI</i>	3665	0,98	2,20	8	<i>RbI</i>	3550	1,49	2,20
9	<i>CsF</i>	3590	1,65	1,33	9	<i>CsBr</i>	4440	1,65	1,96
10	<i>CsCl</i>	3970	1,65	1,81	10	<i>CsI</i>	4512	1,65	2,20

Для имитационного моделирования вещественной частотной характеристики диэлектрической проницаемости материала удобно использовать системный интегратор *MathConnex*, поскольку он позволяет совместить различные программные продукты с целью привлечения ряда их особенностей. В частности, исходные данные наиболее наглядно и целесообразно представлять в виде таблиц *Excel*, а для описания процессов поляризации через передаточные функции привлекать специализированные функции *MatLAB*; кроме того, для промежуточных расчетов можно использовать *MathCAD*. Таким образом, автоматизацию решения рассматриваемой задачи можно свести к реализации функциональной схемы, представленной на рис 2.

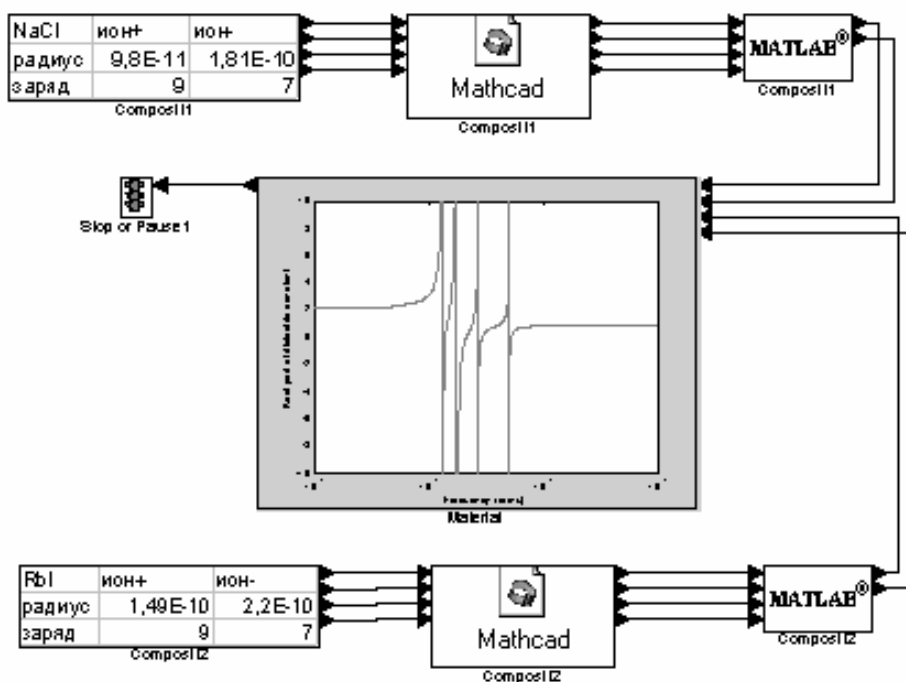


Рисунок 2. Интеграционная схема автоматизации решения

Системный интегратор MathConnex

Задание

1. Внесите в окно проектирования *MathConnex* блок электронных таблиц *Excel* в виде чистой заготовки, а также укажите отсутствие вводов и наличие 4 выводов. Заполните таблицу параметров первого композита, включив в нее эффективные заряды ядер ионов и их радиусы. Отредактируйте метку блока и области выходных данных в соответствии с индексами ячеек *Excel*.

2. Введите блок *MathCAD*, зафиксируйте его изображение в виде иконки, задайте 4 ввода и 4 вывода. Соедините созданные блоки связями. Активизируйте среду *MathCAD* внутри проектируемого блока. Присвойте переменным r_1 , Q_1 , r_2 и Q_2 соответствующие входные данные, используя имена интерфейсных функций ($in0... in3$). Введите формулы (6.4) и (6.5) для расчета значений b_k , ω_{0k} электронных оболочек ионов композита, при вводе

фундаментальных констант используйте Справочный стол (обратите внимание, что результаты вычислений будут интегрированы с *MatLAB*, т.е. размерности величин копировать не следует). Присвойте функциям вывода (*out0... out3*) вычисленные данные.

3. Создайте блок *MatLAB* (иконка, 4 ввода, 2 вывода). Введите формулу (6.6) для вычисления концентрации ионов первого композита исследуемого материала. На основании уравнения (6.2) опишите передаточные функции процессов поляризации ионов, задав коэффициенты полиномов в их числителях и знаменателях. Используя встроенную функцию *parallel*, найдите общую передаточную функцию поляризации композита. Присвойте функциям вывода ее числитель и знаменатель.

4. Спроектируйте аналогичные блоки для второго композита.

0. Введите блок *MatLAB* (иконка, 4 ввода, 1 вывод с блоком остановки). Найдите общую передаточную функцию процесса поляризации материала. Задайте оператор вычисления ее вещественной части, для исследуемого частотного диапазона с итерационным шагом равным 10^{14} рад/сек (используйте функцию *nyquist*). На основании функциональной зависимости (6.3) опишите вычисление ВЧХ диэлектрической проницаемости и задайте построение ее графика в полупологарифмическом масштабе (функция *semilogx*). Запустите имитацию.

ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТОВ

После практического выполнения заданий каждой лабораторной работы, за исключением вводного занятия, каждый обучаемый должен подготовить отчет о ее выполнении на основании индивидуального варианта.

Содержание отчета должно соответствовать следующей структуре:

1. Ответы на вопросы текущего контроля знаний.
2. Описание рассматриваемой задачи и исходных данных.
3. Отображение программных средств используемого математического пакета, необходимых для автоматизации решения.
4. Пошаговое описание выполнения практических заданий с отражением полученных экранных форм.

Отчеты выполняются и представляются к защите в электронном виде (файл формата *Microsoft Word*). В титульном листе, помимо стандартной информации указываются: названия лабораторных работ и номера вариантов индивидуальных заданий, приведенных в таблицах 1-5.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. *Очков В.Ф.* MathCAD PLUS 6.0 для студентов и инженеров. М.: ТОО фирма “Компьютер Пресс”, 1996.
2. *Дьяконов В.П., Абраменкова И.В.* MathCAD 7.0 в математике, физике и в Internet. М.: Изд-во “Нолидж”, 1999.
3. *Прохоров Г.В., Леденев М.А., Колбеев В.В.* Пакет символьных вычислений Maple V. М.: Компания “Петит”, 1997.
4. *Потемкин В.Г.* Система MatLAB 5 для студентов: Справочное пособие. М.: ДИАЛОГ-МИФИ, 1998.

Математический пакет *MahtCAD 2000 Professional*

Среди систем автоматизации математических расчетов особое место занимает система *MahtCAD*. Отличительная черта пакетов этого класса – входной язык, максимально приближенный к обычному математическому языку или языку научных статей и книг.

А.1. Интерфейс пользователя

Под интерфейсом пользователя подразумевается совокупность средств графической оболочки *MahtCAD*, обеспечивающих легкое управление системой как с клавиатуры, так и с помощью мыши. Под управлением понимается, как просто набор необходимых символов, формул, текстовых комментариев и т.д., так и возможность полной подготовки в среде *MahtCAD* документов (*WorkSheets*) и электронных книг, с последующим их запуском в реальном времени. Экранный интерфейс системы создан таким образом, что пользователь, имеющий элементарные навыки работы с *Windows*-приложениями, может непосредственно начать работу с программной средой *MahtCAD*.

Главное меню системы

Верхняя строка окна системы содержит указание на имя системы или текущего открытого окна. Следующая строка содержит позиции главного меню:

<i>Файл (File)</i>	– работа с файлами, <i>Internet</i> и <i>E-mail</i> ;
<i>Правка (Edit)</i>	– редактирование документов;
<i>Вид (View)</i>	– изменение средств обзора и включения/выключения элементов интерфейса;
<i>Вставка (Insert)</i>	– установка вставок объектов и их шаблонов;
<i>Формат (Format)</i>	– изменение формата (параметров) объектов;
<i>Математика (Math)</i>	– управление процессом вычислений;
<i>Символы (Symbolic)</i>	– выбор операций символьного процессора;
<i>Окно (Window)</i>	– управление окнами системы;
<i>? (Help)</i>	– работа со справочной базой данных о системе.

Инструментальные и наборные панели

Работа с документами *MahtCAD* обычно не требует обязательного использования возможностей главного меню, т. к. основные из них дублируются кнопками быстрого управления. Их можно выводить на экран или убирать с него с помощью опций позиции *View* главного меню *Windows*. Чаще всего используются две такие панели: панель инструментов, дублирующая ряд наиболее распространенных команд и операций, и панель форматирования для выбора типа и размеров шрифтов и способа выравнивания текстовых комментариев.

Кроме рассмотренных панелей, непосредственно после запуска системы в виде наборной панели появляется меню математических палитр (*Palletes*), предназначенных для вывода заготовок – шаблонов математических знаков, операций и операторов (рис. А.1). На рисунке А.2 представлены все математические палитры. Поскольку их достаточно много, то рекомендуется использовать только необходимые для работы палитры, а ненужные – закрывать.

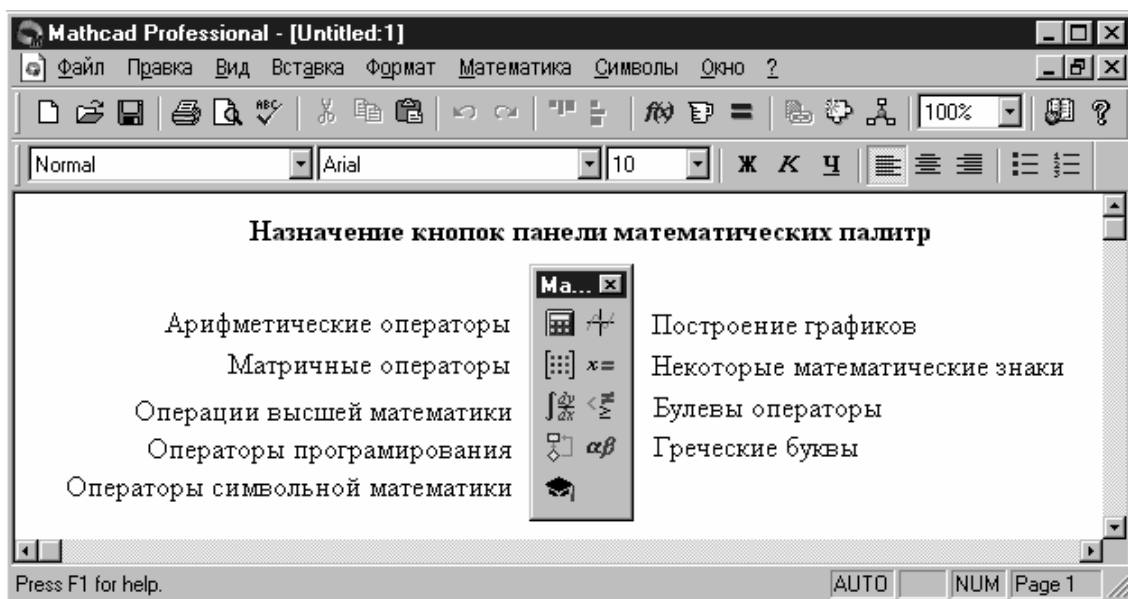


Рисунок А.1. Меню математических палитр

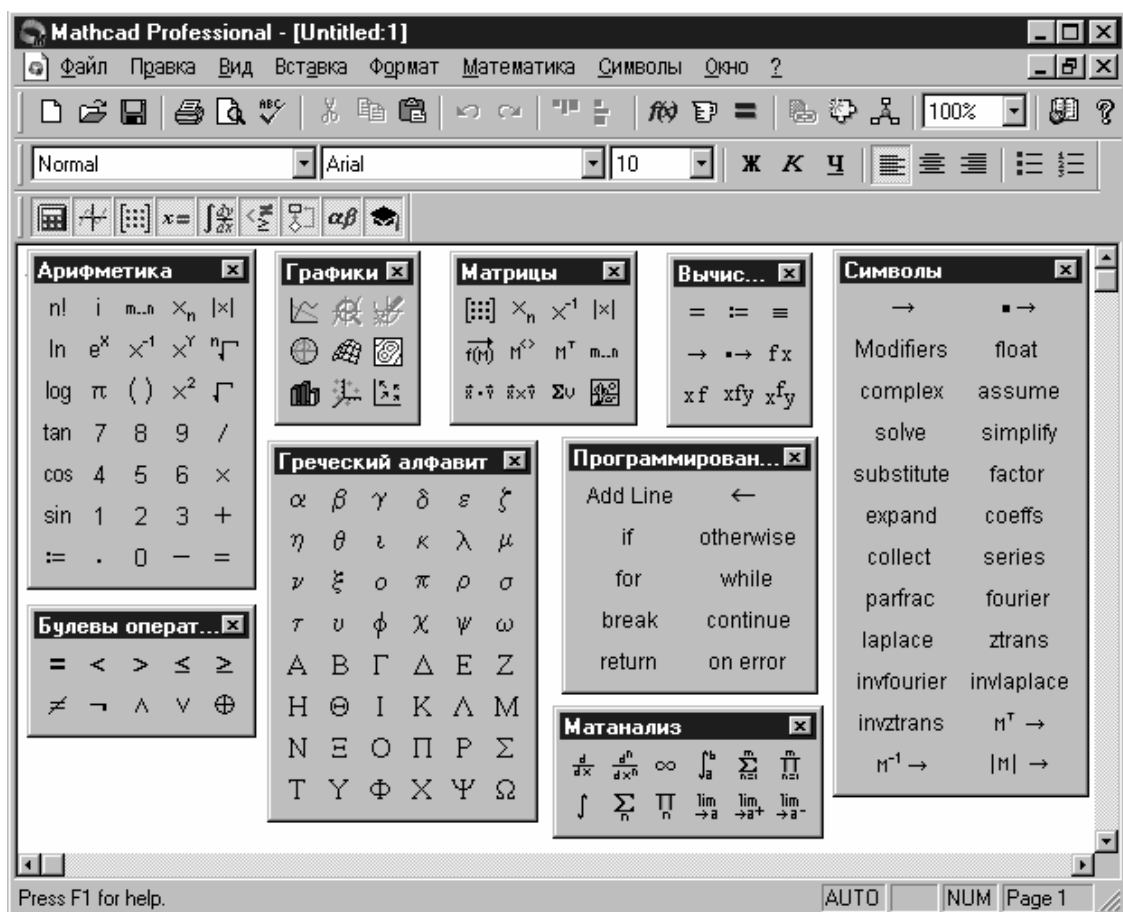


Рисунок А.2. Математические палитры

Следует отметить, что панель инструментов можно свернуть в виде наборной панели и поместить в любое место экрана, а панель математических палитр жестко закрепить. Таким образом, интерфейс системы легко модифицируется, и пользователь имеет возможность подстроить его под свой вкус.

А.2. Построение графиков

Для создания графиков в системе *MahtCAD* имеется программный графический процессор. Основное внимание при его разработке уделено обеспечению простоты задания графиков и их модификации с помощью соответствующих опций. Большинство необходимых параметров графического процессора по умолчанию задается автоматически.

MahtCAD позволяет применять шаблоны семи типов графиков, которые приведены в подменю *График (Graph)* позиции *Вставка (Insert)* главного меню:

- X-Y Plot* – двухмерный график в декартовой системе координат;
- Polar Plot* – график в полярной системе координат;
- Surface Plot* – трехмерный график;
- Contour Plot* – контурный график;
- 3D Scatter Plot* – трехмерный точечный график;
- Vector Field Plot* – карта векторного поля;
- 3D Bar Chart* – трехмерная столбчатая гистограмма.

Для вызова шаблонов также можно использовать палитру графиков, показанную на рисунке А.3. Графики любого вида можно переносить в нужное место документа, изменять их размер с помощью маркеров, появляющихся при выделении графика, а также редактировать границы областей аргументов и функций.

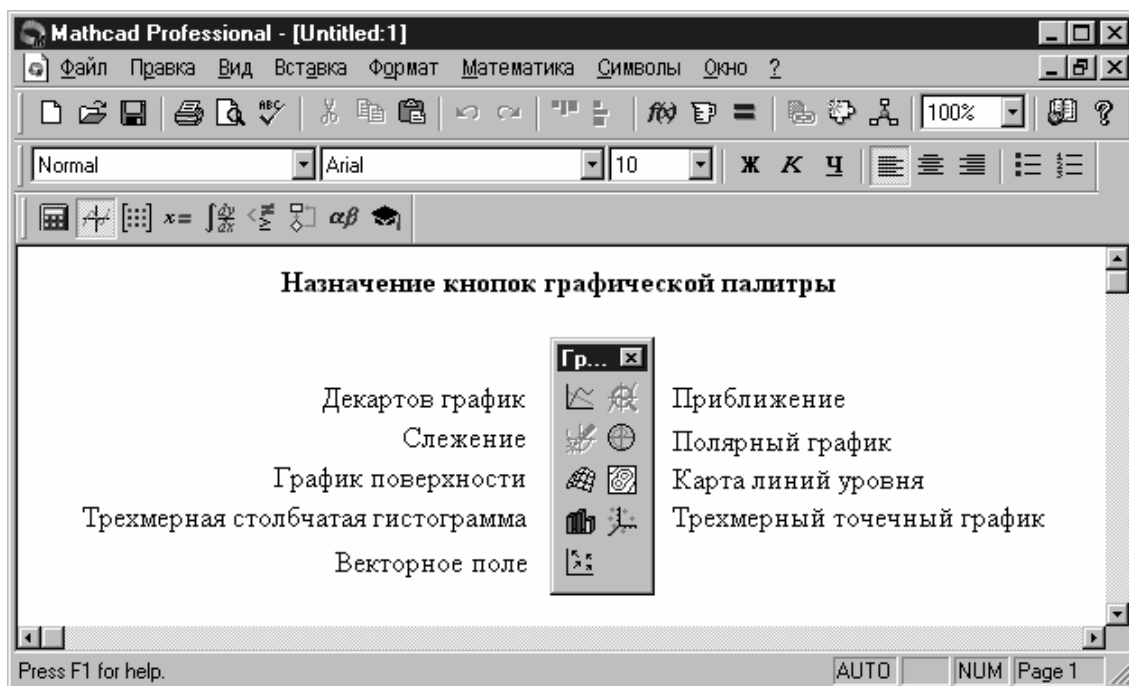


Рисунок А.3. Палитра инструментов графики

Для редактирования заданных по умолчанию графических параметров используется окно задания форматов, для вызова которого достаточно дважды быстро щелкнуть левой кнопкой мыши, установив курсор в область графика.

Двухмерный декартов график

Пакет *MahtCAD 2000 Professional* позволяет использовать два способа построения плоских графиков в декартовой системе координат. При первом – наиболее простом способе построения – достаточно вызвать шаблон *X-Y Plot*, в который следует ввести имя переменной x по оси X и выражение, описывающее функцию $f(x)$ по оси Y .

Для второго способа необходимо вначале задать ранжированную переменную – например x , определив диапазон ее изменения и шаг d , который указывается вводом начального значения переменной x_0 и, через запятую, значения $x_0 + d$. После чего через две точки указывается конечное значение x , см. рис. А.4. Затем надо задать соответствующие функции и

вызвать графический шаблон, при заполнении которого используются имена переменных и функций. Если в одном шаблоне строится несколько графиков, то для их разделения следует использовать запятые.

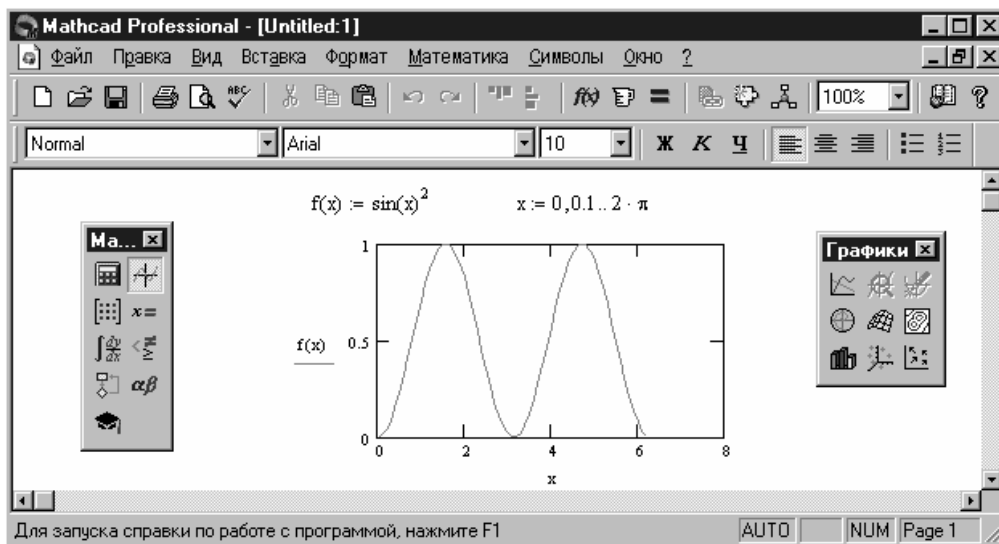


Рисунок А.4. Пример построения двухмерного декартова графика

Полярный график

В полярной системе координат каждая точка задается углом W и модулем радиус-вектора $R(W)$. График функции обычно строится в виде линии, которую описывает конец радиус-вектора при изменении угла W в определенных пределах, чаще всего от 0 до 2π (рис. А.5).

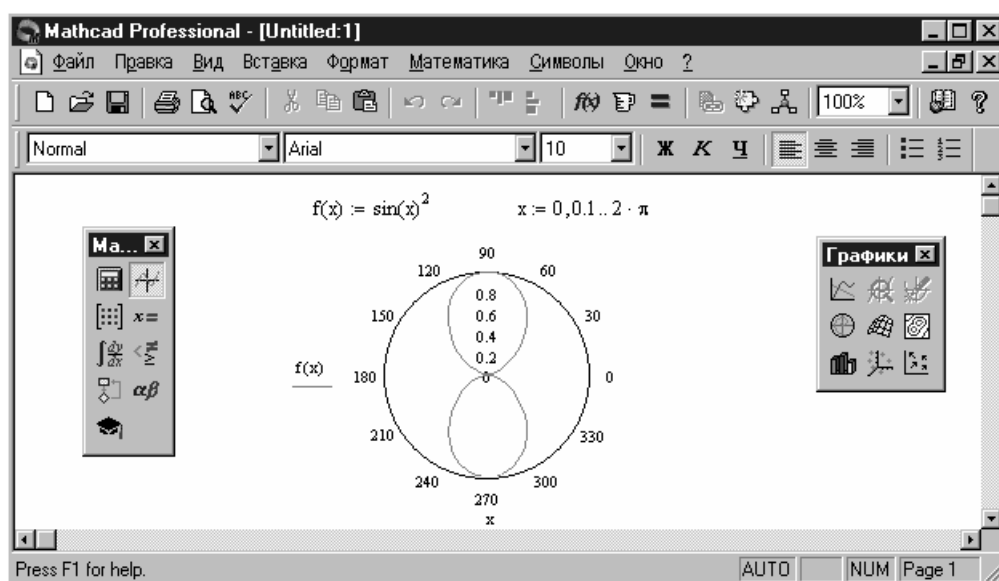


Рисунок А.5. Пример построения полярного графика

Трехмерный график

Эта операция служит для построения поверхности вида $Z(X, Y)$, предварительно представленной матрицей M значений аппликат Z . В единственный шаблон данных вводится имя матрицы со значениями аппликат поверхности (рис. А.6).

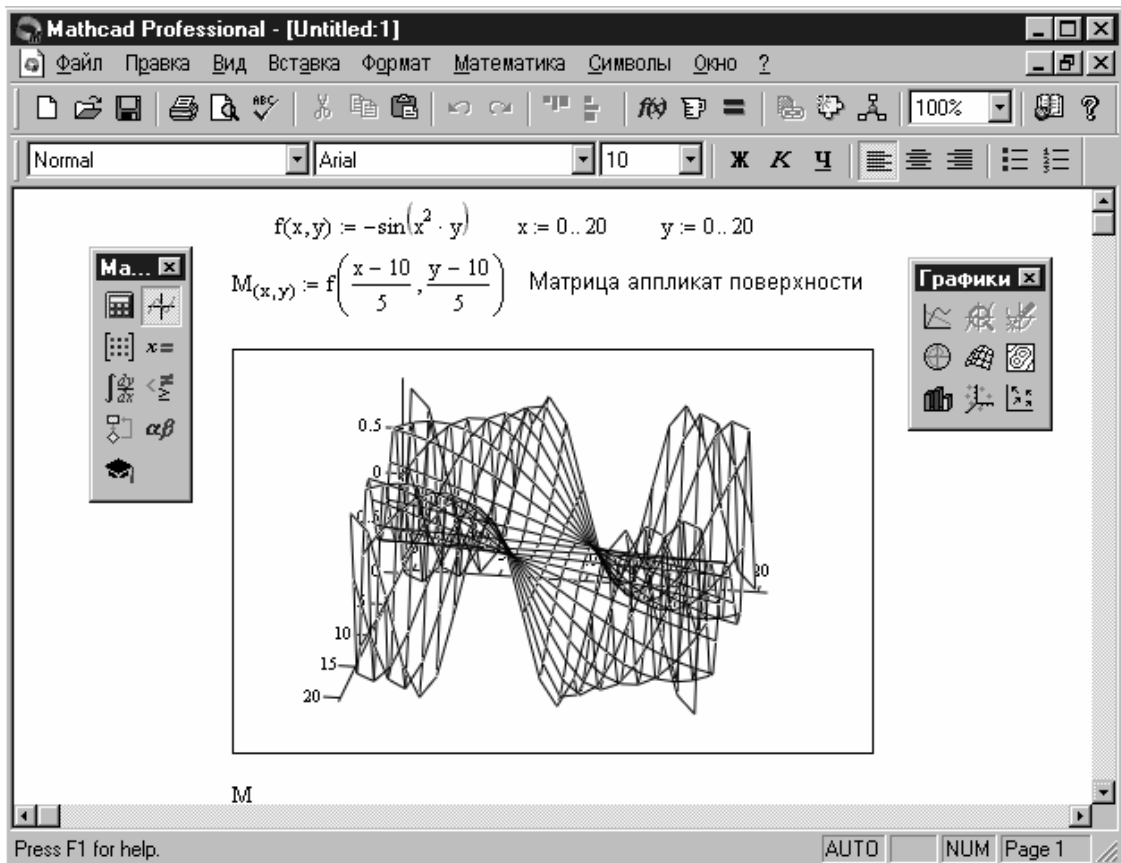


Рисунок А.6. Пример построения трехмерного графика поверхности

Прежде чем приступить к построению графика поверхности, ее необходимо сформировать матрицу ее аппликат с помощью следующей последовательности действий:

- ввести выражение для расчета аппликат поверхности в виде функции двух аргументов $Z(x, y) := f(x, y)$;
- задать векторы числа узловых точек по осям $i := 0..n$, $j := 0..m$;
- сформировать векторы значений аргументов функции в узловых точ-

ках $x_i = i \cdot (x_k - x_0) / n$, $y_j = j \cdot (y_k - y_0) / m$;

- рассчитать матрицу M , содержащую результаты вычислений для аппликата поверхности в узловых точках – $M_{i,j} := Z(x_i, y_j)$.

Поскольку график строится на основании матрицы, содержащей только координаты высот фигуры (аппликаты), то истинные масштабы по осям X и Y неизвестны и на рисунках не проставляются. Могут, впрочем, выводиться просто порядковые номера элементов матриц в заданном направлении. Большие возможности дает несколько иной способ задания поверхности – в параметрическом виде. При этом приходится формировать три матрицы и указывать их в шаблоне в виде X, Y, Z .

Контурный график

Еще один широко распространенный тип графиков для представления поверхностей – с помощью линий равного уровня (рис. А.7). Используемый для их построения шаблон подобен описанному выше трехмерному графику.

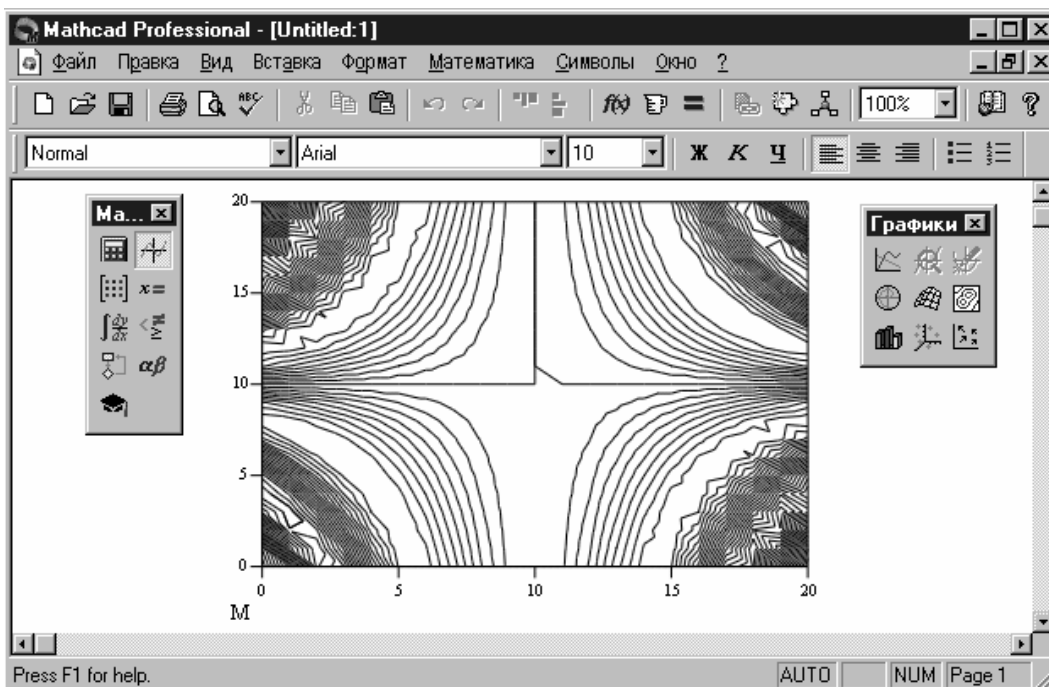


Рисунок А.7. Пример построения контурного графика поверхности

Зачастую контурные графики более информативны, чем просто поверхности, поскольку у последних нередко одни части закрывают другие. Например, пик на переднем плане может закрыть задний план, чего нет на контурных графиках.

Карта векторного поля

Если линии контурного графика заменить векторами, стрелки которых обращены острием в сторону нарастания высоты поверхности, а плотность расположения зависит от скорости этого нарастания, то получится карта векторного поля (рис. А.8).

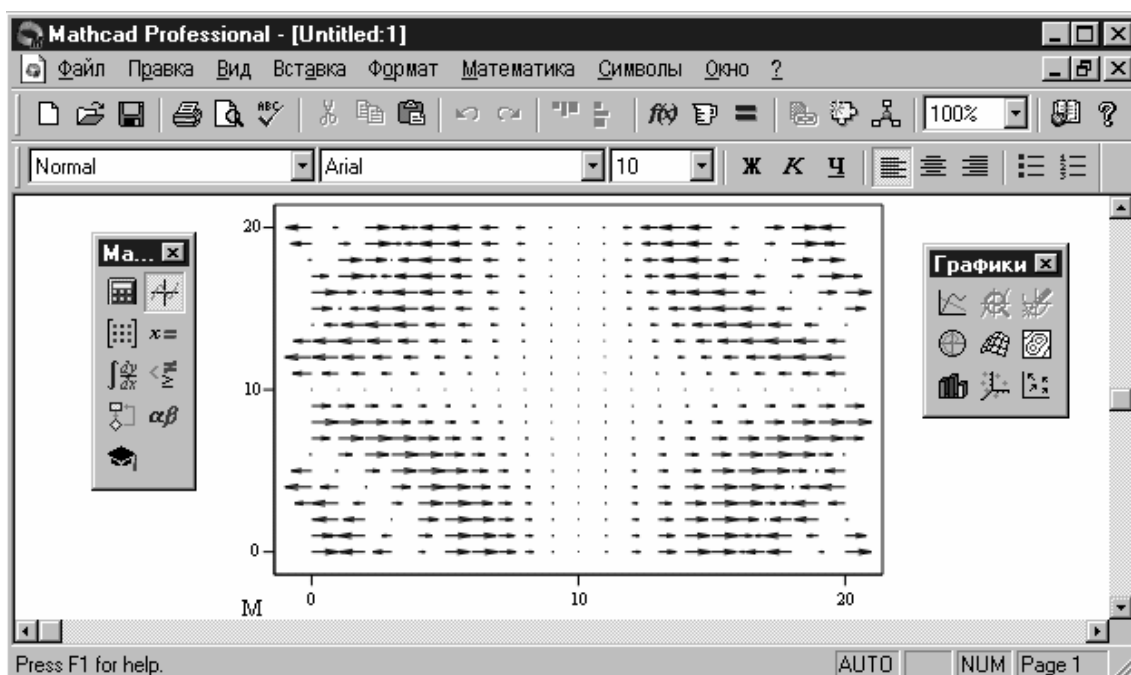


Рисунок А.8. Пример построения векторного графика поверхности

Трехмерный точечный график

Нередко поверхности представляют в виде точек или иных фигур, находящихся в трехмерном пространстве, каждая из которых несет информацию о положении ее геометрического центра (рис. А.9).

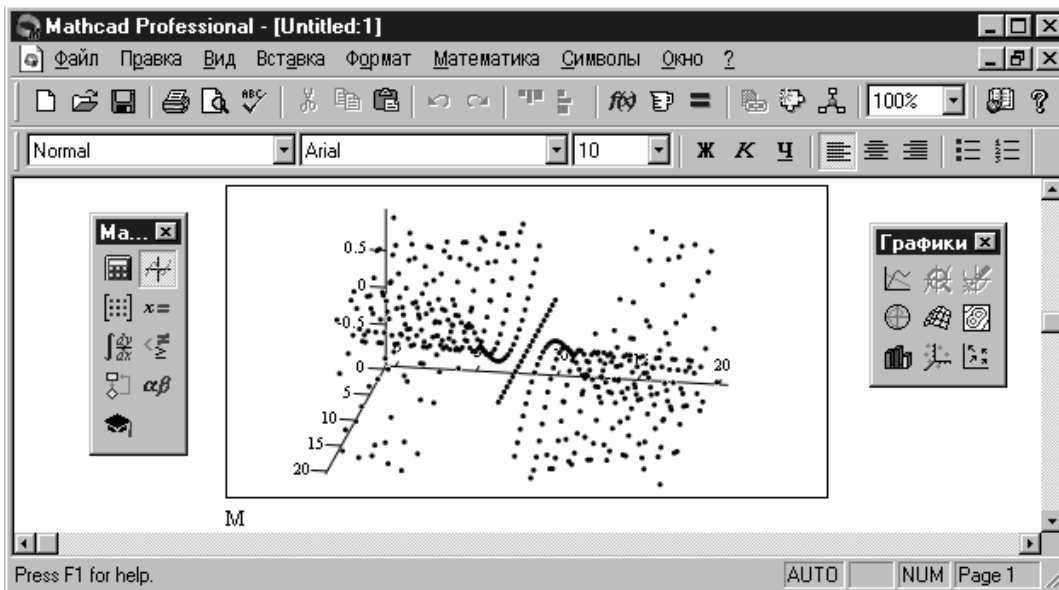


Рисунок А.9. Пример построения трехмерного точечного графика

Трехмерная столбчатая гистограмма

Весьма распространенной формой изображения поверхностей является представление ее рядом трехмерных столбиков, высота которых определяется значением координаты $z(x, y)$, см. рис. А.10.

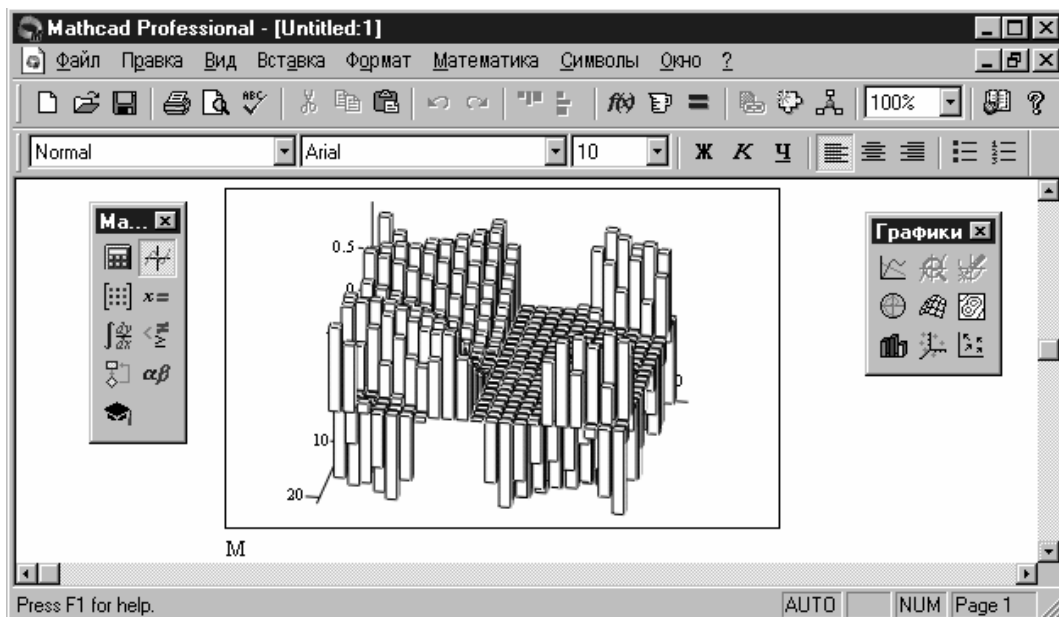


Рисунок А.10. Пример построения трехмерной столбчатой гистограммы

Подобные графики широко применяются при представлении сложных статистических данных – например, представленных тремя независимыми переменными. Обычно их построение достаточно трудоемко, *MahtCAD* же превращает данную процедуру в обыденную операцию.

Анимация изображений

При визуализации различных процессов достаточный интерес вызывает создание анимационных графиков, имитирующих динамику рассматриваемой физической системы или объекта, реализация которых в *MahtCAD* осуществляется с помощью команды *Анимация (Animate)* позиции *Вид (View)* главного меню. Разберем последовательность анимации на примере моделирования движения тела, брошенного со скоростью 10 м/с под углом 45° к горизонту.

Подготовка исходного графика

Перед тем как анимировать поведение какого-либо физического объекта или течение процесса, его следует представить в виде соответствующей функции времени (или ряда временных функций). Для рассматриваемого примера координаты тела в любой момент времени описываются следующими уравнениями:

$$\begin{cases} x(t) = V_0 \cdot \cos(\alpha) \cdot t \\ y(t) = V_0 \cdot \sin(\alpha) \cdot t - g \cdot \frac{t^2}{2} \end{cases}$$

где g – ускорение свободного падения.

Чтобы анимационный график отражал полет тела в реальном масштабе времени, надлежит предварительно рассчитать время его полета, поскольку это необходимо в дальнейшем при определении числа кадров и скорости их воспроизведения (число кадров = время полета \times скорость воспроизведения). Кроме того, следует найти также дальность полета, т.к.

при построении графика нужно будет задать границы изменения координат тела, чтобы зафиксировать изображение рабочей области, в противном случае размер графического окна будет меняться в процессе воспроизведения кадров. Чтобы соразмерить масштаб осей, можно ввести в шаблон графика одинаковые значения границ по X и Y , а также отрегулировать размер области графических построений до вида квадрата. Следующим шагом является задание аргумента временной функции через системную переменную *MahtCAD* – *FRAME* (рис. А.11).

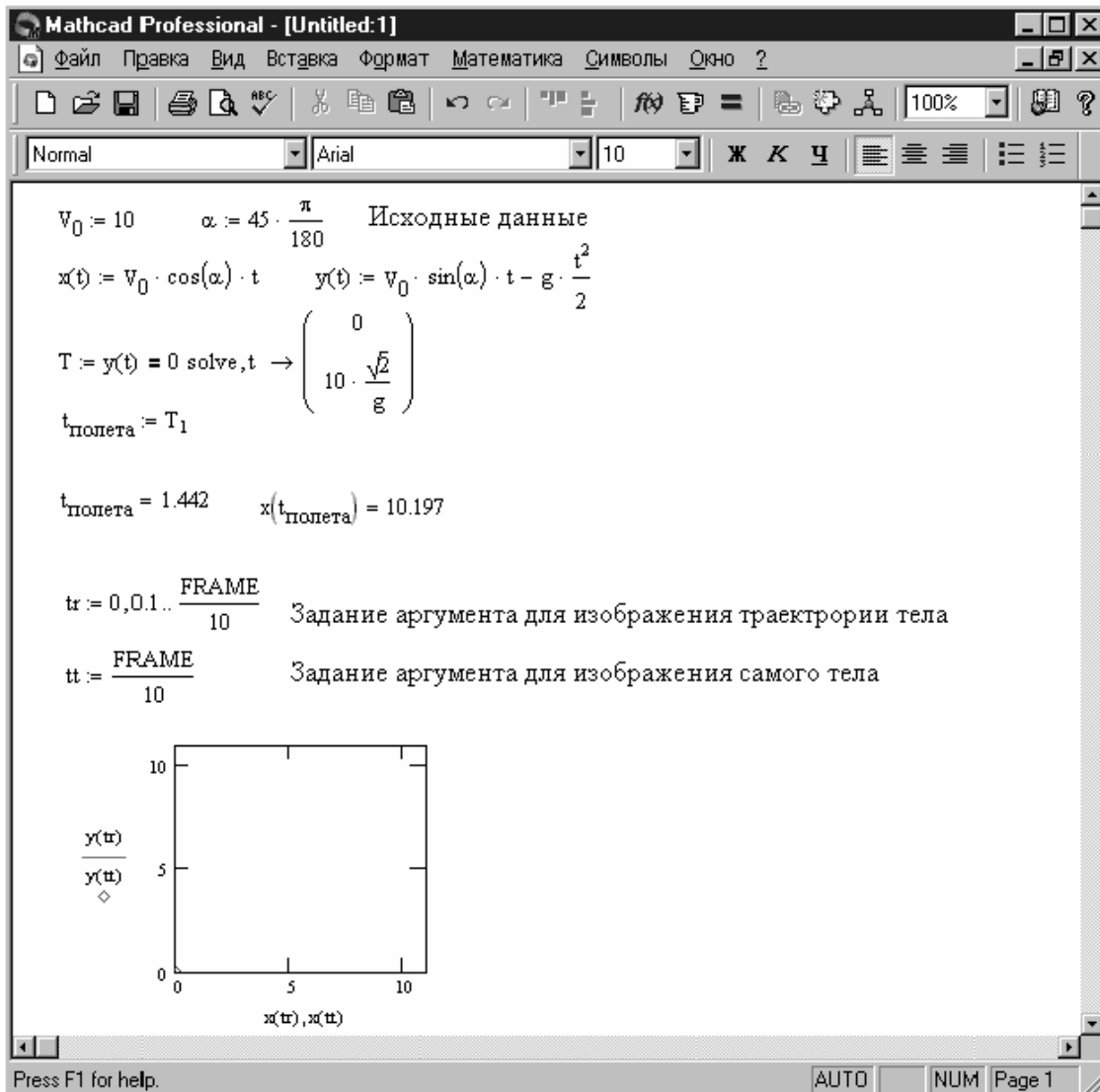


Рисунок А.11. Пример подготовки исходного графика
При этом следует учитывать сценарий вывода изображений, а имен-

но: если аргумент задается как ранжированная функция, то картинка отражает график траектории полета, если же аргумент задан как единственное значение, то кадры положения тела в каждый момент времени будут последовательно сменять друг друга. В заключение подготовительного этапа строится непосредственно сам исходный график, который будет анимирован. Естественно, что при задании его параметров следует учитывать изложенные выше рекомендации.

Реализация анимационного графика

Для анимации исходного графика следует запустить команду *Анимация* (*Animate*). В появившемся окне задаются начальное и конечное значение переменной *FRAME* (число кадров) и скорость воспроизведения. Для рассматриваемого примера установим скорость – 10 кадр/с, таким образом, минимальное число кадров для отражения всего времени полета тела будет равно 15. Следующим шагом является выделение области анимации, т. е. самого графика (рис. А.12).

После выделения анимационной области пользователю остается нажать кнопку “*Анимация*”, при этом, если все подготовительные операции выполнены успешно, в окне активизируется небольшой встроенный дисплей, на который будут выводиться изображения создаваемых кадров и их соответствующие номера рядом с надписью “*FRAME*”.

По готовности анимационного клипа для его просмотра появится окно *Playback* (рис. А.13). Для запуска клипа достаточно нажать кнопку с изображением треугольника в левом нижнем углу окна просмотра. Рядом с кнопкой запуска находится еще одна кнопка, которая позволяет настраивать параметры просмотра: изменять размеры окна, варьировать скорость воспроизведения и т.д.

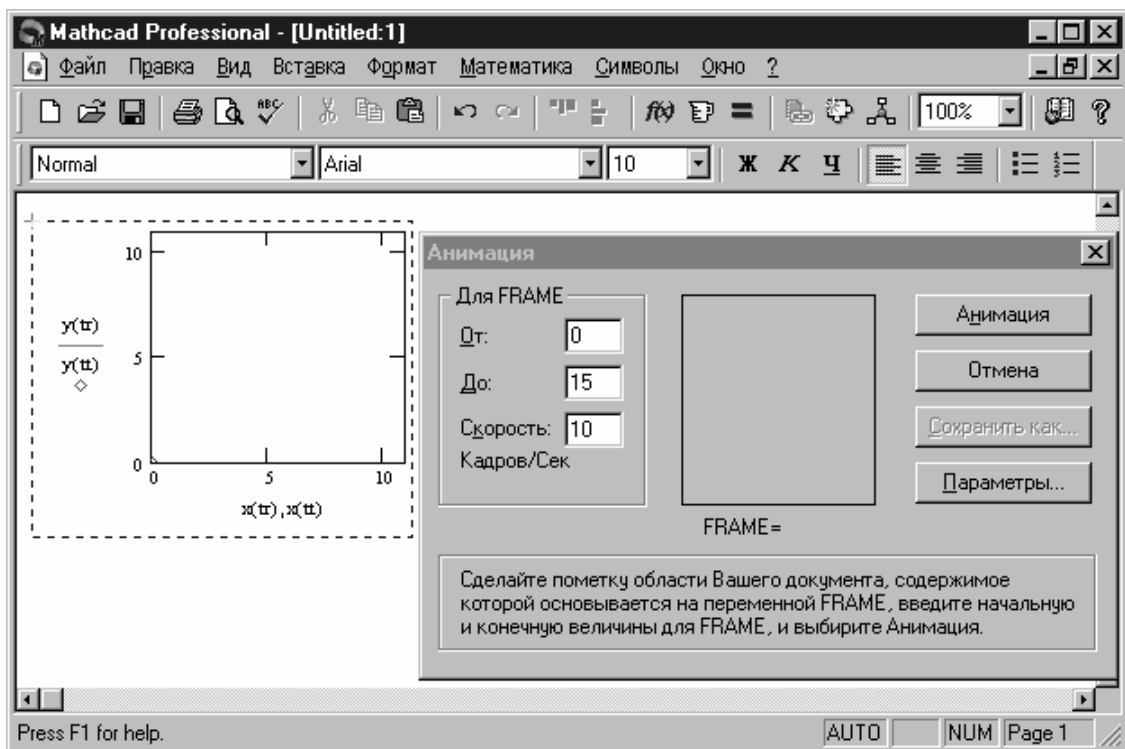


Рисунок А.12. Пример задания параметров анимационного графика

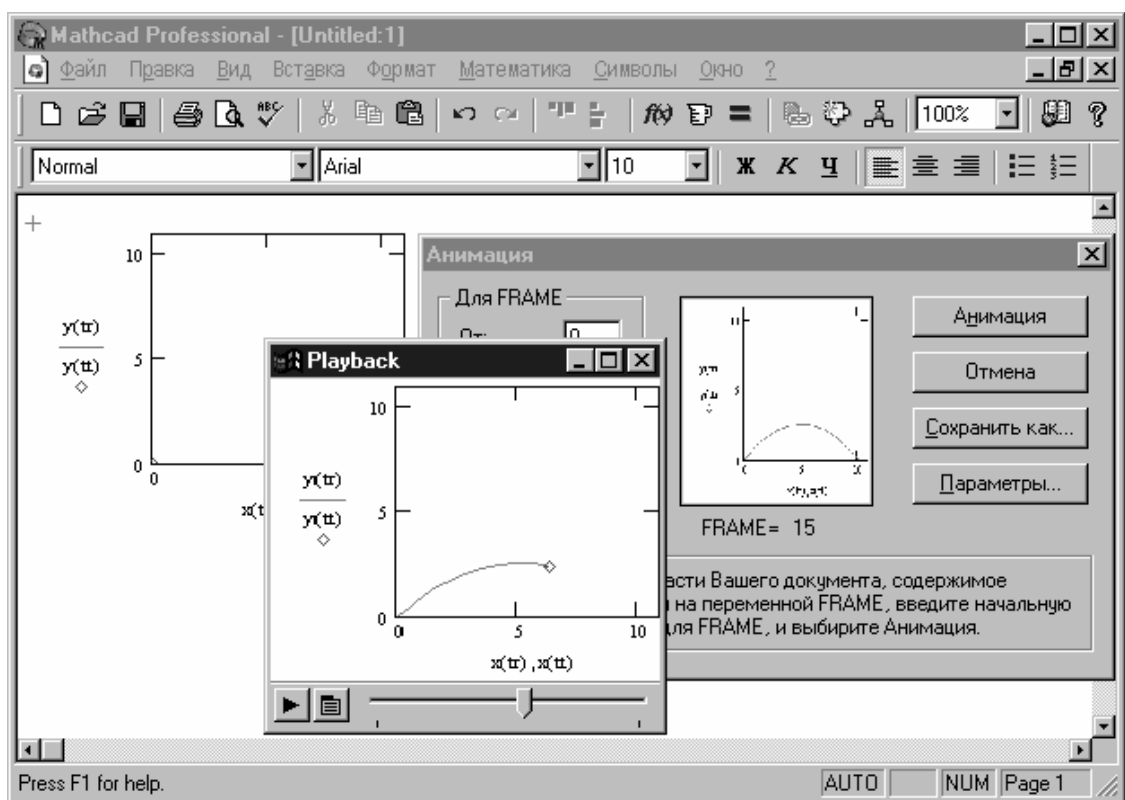


Рисунок А.13. Пример реализации анимационного графика

А.3. Математические расчеты

Будучи автоматизированной системой, предназначенной для проведения физико-математических расчетов, *MahtCAD* существенно облегчает и ускоряет выполнение самых разнообразных вычислительных процедур. В данном разделе приводится только та часть его возможностей, использование которой необходимо для успешного выполнения лабораторного практикума.

А.3.1. Решение алгебраических уравнений и систем

Функции root и polyroots

Многие уравнения и системы из них не имеют аналитических решений. Однако они могут решаться численными методами с заданной погрешностью (не более значения, заданного системной переменной *TOL*). Для простейших уравнений вида $F(x) = 0$ решение находится с помощью функции *root* (Выражение, Имя переменной).

Эта функция возвращает значение переменной, при котором выражение дает 0. Функция реализует вычисления итерационным методом, причем можно задать начальное значение переменной, что бывает особенно полезным при существовании нескольких решений. Рисунок А.14 иллюстрирует технику применения функции *root* для вычисления корней кубического полинома. Для поиска корней обычного полинома $p(x)$ степени n пакет *MahtCAD 2000 Professional* содержит функцию *polyroots(V)*, возвращающую вектор корней полинома, коэффициенты которого находятся в векторе V с длиной, равной $n+1$. При этом будут найдены как вещественные, так и комплексные корни (рис. А.15.). Не рекомендуется пользоваться данной функцией, если степень полинома выше пятой-шестой, т.к. в этом случае трудно получить малую погрешность вычисления корней.

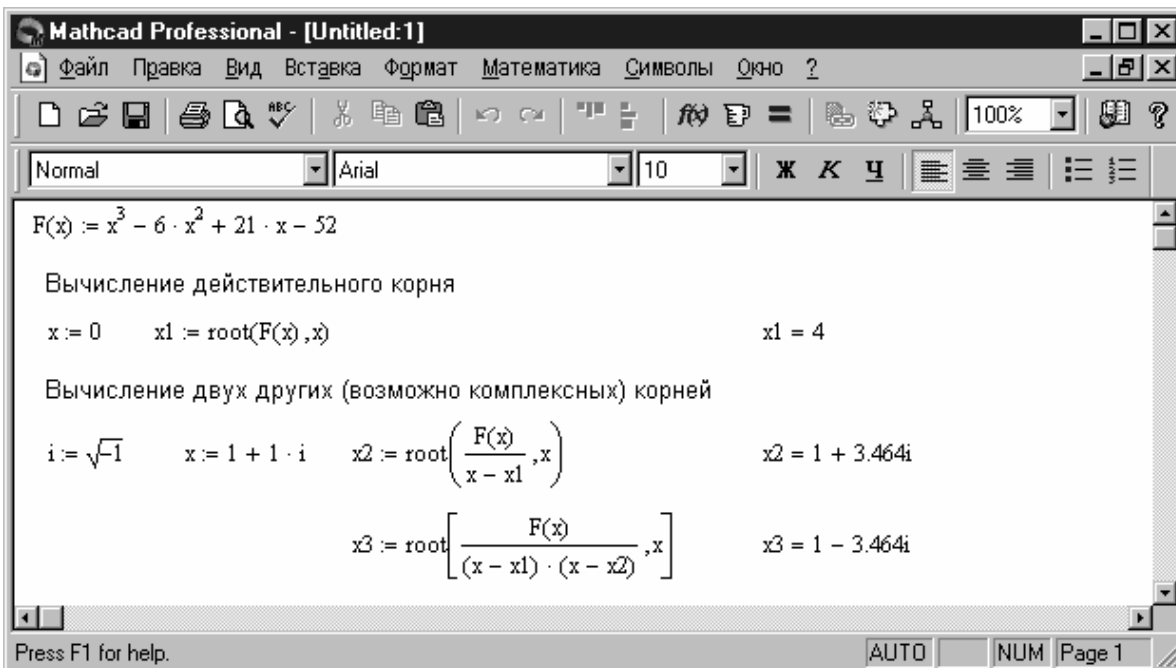


Рисунок А.14. Пример вычисления корней функцией *root*

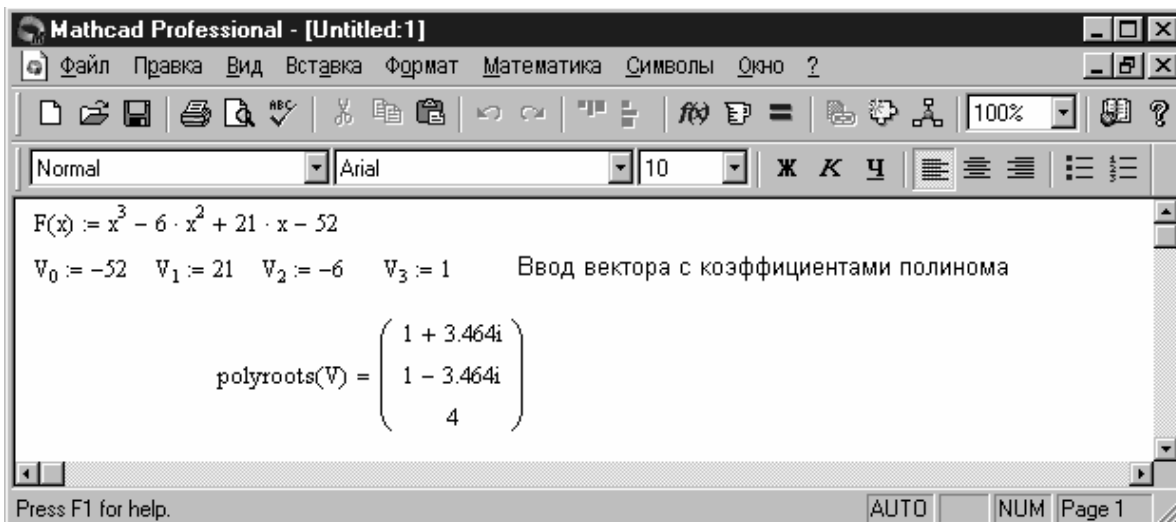


Рисунок А.15. Пример поиска корней с помощью *polyroots*

Обратите внимание на задание элементов вектора. При задании номера элемента массива следует использовать шаблон из палитры «Матрицы», либо клавишу клавиатуры «[». Для ввода текстового индекса переменной используется клавиша точки.

Встроенные блоки Given... Find и Given... Mineer

При решении систем уравнений *MathCAD* предоставляет для использования специальный вычислительный блок, открываемый служебным словом – директивой *Given* и имеющий следующую структуру:

- начальные приближения для искомых решений;
- служебное слово – *Given*;
- решаемые уравнения, в которых должен использоваться знак символического (логического) равенства;
- выражения с функциями *Find* и *Mineer* (рис. А.16).

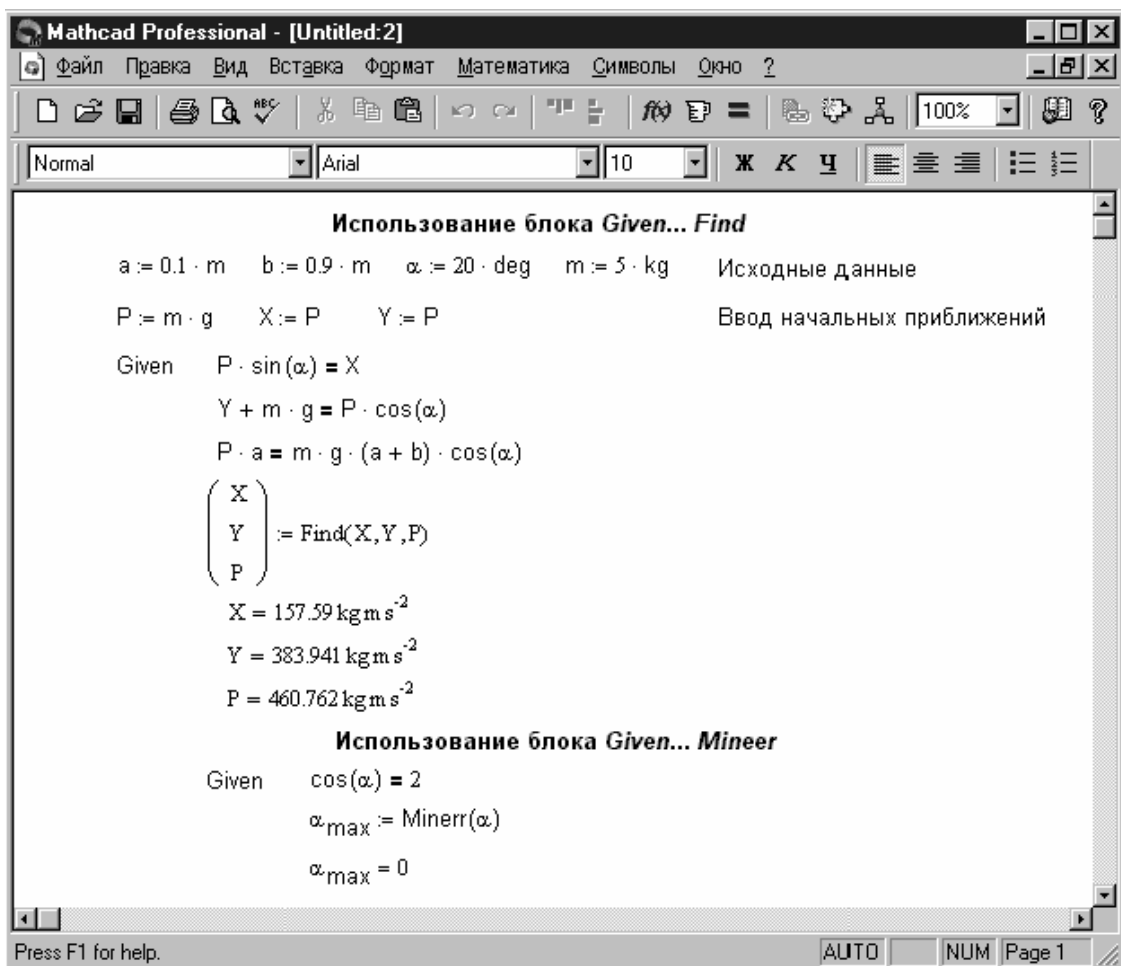


Рисунок А.16. Примеры использования расчетных блоков

В заключение решения рекомендуется дополнить блок проверкой решения системы. В блоке может использоваться одна из функций:

Find (*var1*, *var2*, ...) – возвращает значение для точного решения;

Mineer (*var1*, *var2*, ...) – находит приближенное решение.

Первая используется, когда решение реально существует (хотя может и не являться аналитическим). Вторая пытается найти максимальное приближение к несуществующему решению путем минимизации среднеквадратичной погрешности решения.

Векторно-матричное решение системы линейных уравнений

MahtCAD позволяет решать широкий круг задач линейной алгебры, используя встроенные векторно-матричные функции и операторы математической палитры «Матрицы». К примеру, если задана матрица A и вектор B для системы линейных уравнений в матричной форме записи $A \cdot X = B$, то вектор решения можно получить из выражения $X = A^{-1} \cdot B$. Если уравнений n , размерность вектора B должна быть тоже n , а матрицы A – $n \times n$. Поскольку решение систем линейных уравнений – довольно распространенная задача, то в *MahtCAD* введена встроенная функция *lsolve* (A , B). Варианты векторно-матричного решения системы уравнений, рассмотренной в предыдущем примере, приведены на рис. А.17.

Аналитическое решение уравнений

Для проведения аналитических выкладок и преобразований *MahtCAD 2000 Professional* дополнен символьным ядром (процессором), аналогичным применяемому в одной из самых мощных систем компьютерной алгебры – *Maple*. Операции, относящиеся к работе символьного процессора, содержатся в подменю позиции *Символы* (*Symbolic*) главного меню, для реализации которых можно также использовать соответствующую математическую палитру.

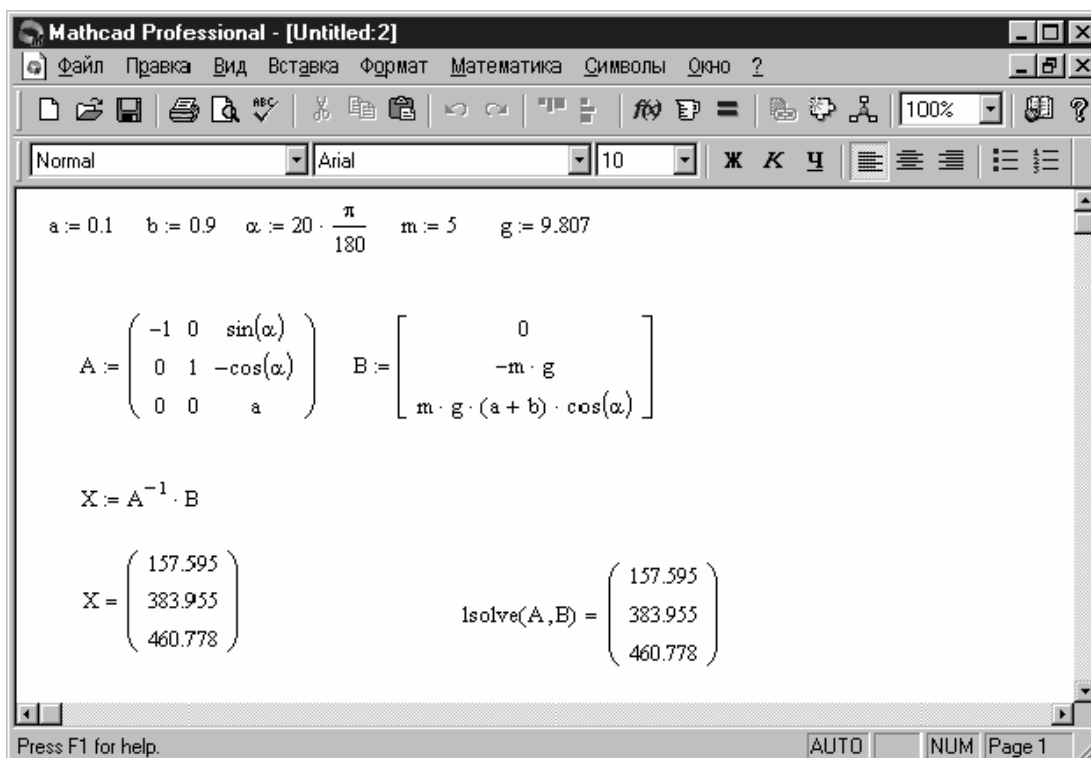


Рисунок А.17. Примеры векторно-матричного решения

При нахождении символьного решения системы уравнений может использоваться блок *Given... Find* (рис. А.18).

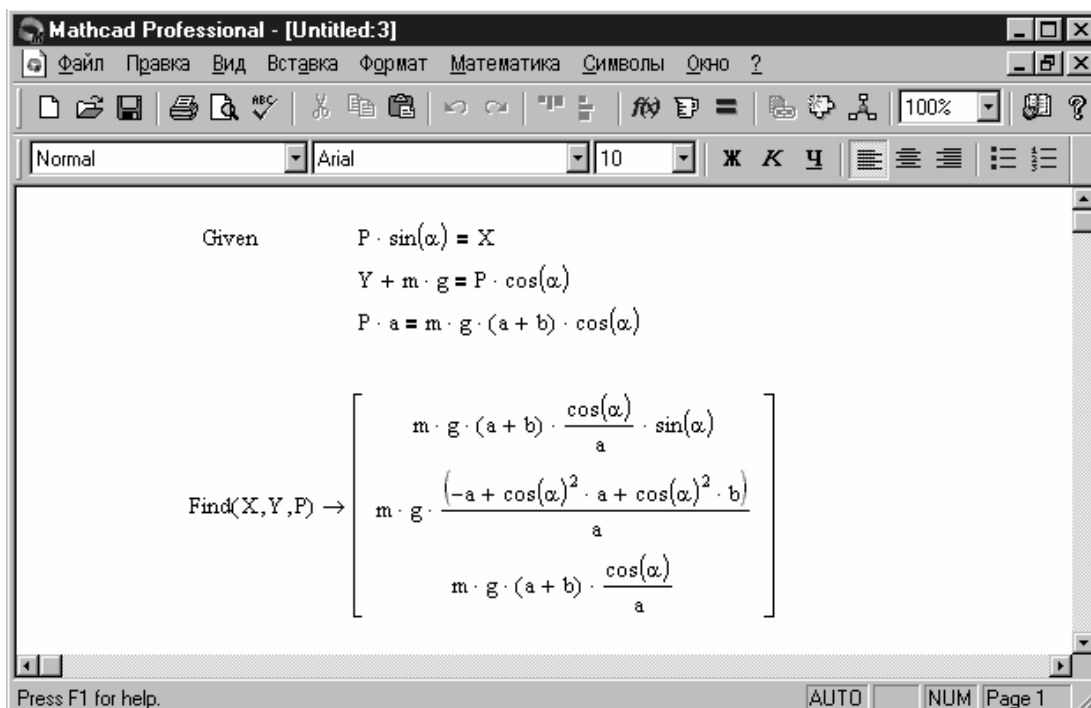


Рисунок А.18. Пример аналитического решения

Отличие от применения данного блока для численного решения заключается в том, что в этом случае начальные приближения для решений не задаются и вместо оператора присваивания “:=” употребляется оператор символического вывода “→”, вводимый из соответствующей палитры.

А.3.2. Решение обыкновенных дифференциальных уравнений

Для решения динамических задач в *MahtCAD* существует целый ряд встроенных функций. Ограничимся рассмотрением тех из них, которые дают решения обыкновенных дифференциальных уравнений:

rkadapt($y, x1, x2, acc, n, F, k, s$) – возвращает матрицу, содержащую таблицу значений решения задачи Коши на интервале от $x1$ до $x2$ для системы обыкновенных дифференциальных уравнений, вычисленную методом Рунге-Кутты с переменным шагом и начальными условиями в векторе y (правые части системы записаны в векторе F , acc – задаваемая точность решения, n – число шагов, k – максимальное число промежуточных точек решения, s – минимально допустимый интервал между точками);

Rkadapt($y, x1, x2, n, F$) – возвращает матрицу решений методом Рунге-Кутты с переменным шагом;

rkfixed($y, x1, x2, n, F$) – возвращает матрицу решений методом Рунге-Кутты при фиксированном числе шагов;

bulstoer($y, x1, x2, acc, n, F, k, s$) – возвращает матрицу решения методом Булириш-Штера с переменным шагом;

Bulstoer($y, x1, x2, n, F$) – возвращает матрицу решения методом Булириш-Штера с постоянным шагом.

Для формирования вектора F каждое из решаемых дифференциальных уравнений необходимо представить в виде системы уравнений первого порядка. При этом обозначению самой функции должна соответствовать

некоторая переменная с индексом 0, т.е. первые элементы массива решений. Тогда первая производная функции будет обозначаться переменной с индексом 1 и т.д. Примеры использования функций приведены на рис. А.19 и А.20.

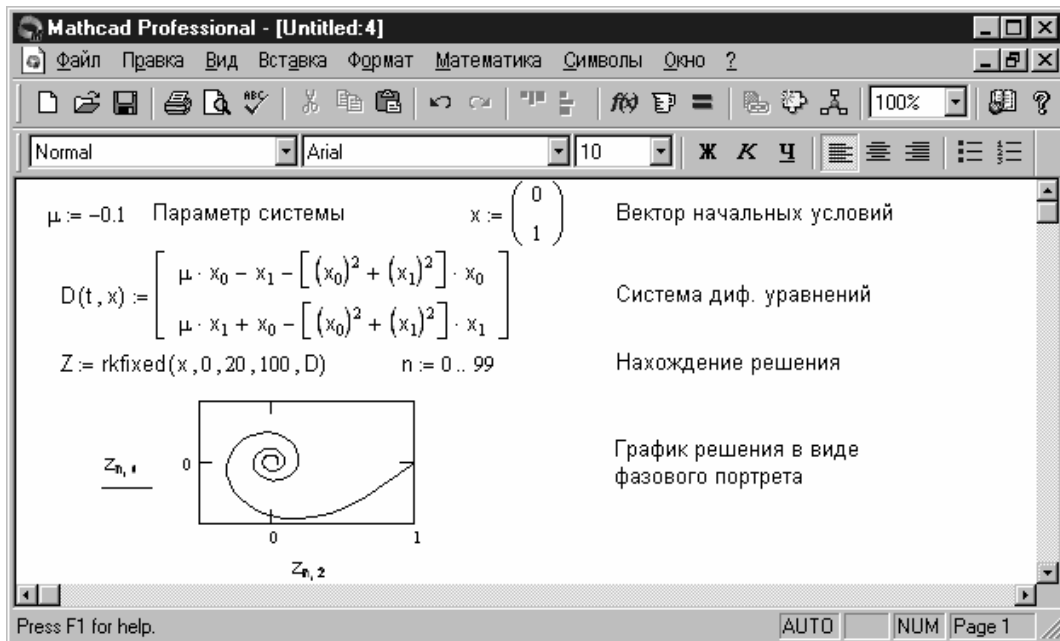


Рисунок А.19. Решение методом Рунге-Кутта с фиксированным шагом

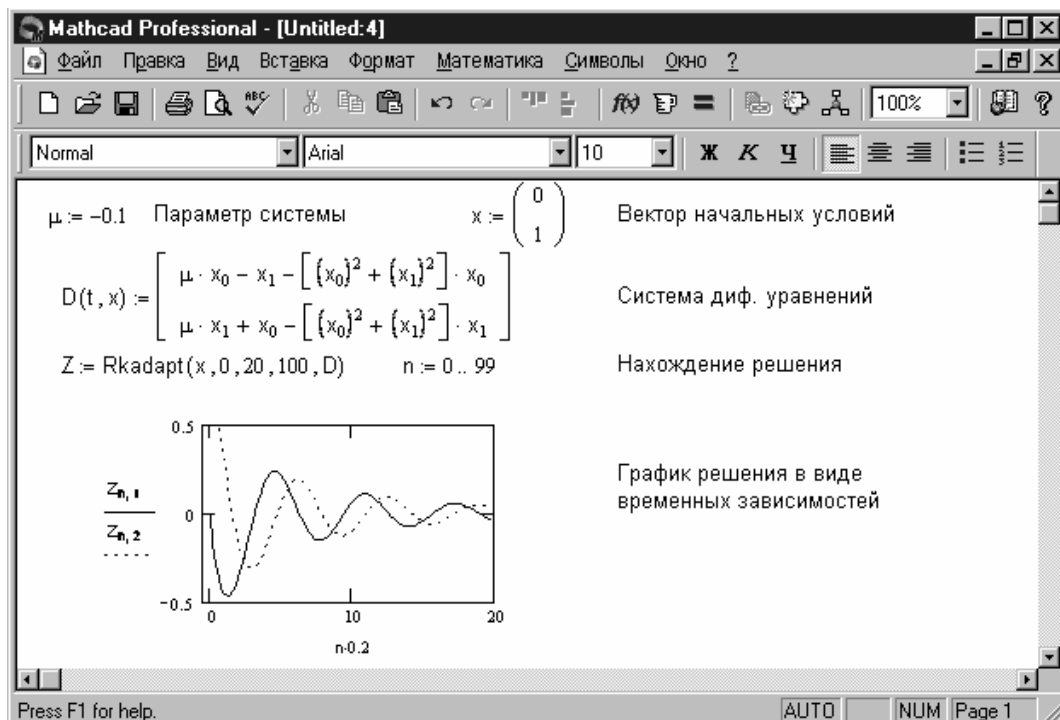


Рисунок А.20. Решение методом Рунге-Кутта с переменным шагом

А.3.3. Аппроксимация зависимостей вида $y(x)$

Для представления физических закономерностей и при проведении научно-технических расчетов часто используются зависимости вида $y(x)$, причем число точек этих зависимостей обычно ограничено. В этом случае неизбежно возникает задача приближенного вычисления значений функций в промежутках между узловыми точками (интерполяция) и за их пределами (экстраполяция). Эта задача решается аппроксимацией исходной зависимости, т. е. ее подменой какой-либо простой функцией, что также дает возможность проведения над ней обычных математических операций.

Одномерная кусочно-линейная аппроксимация

При кусочно-линейной интерполяции вычисления дополнительных точек выполняются по линейной зависимости. Графически это означает простое соединение узловых точек отрезками прямых, для чего используется функция *linetrp* (VX, VY, x).

Для заданных векторов VX и VY узловых точек и заданного аргумента x *linterp* возвращает значение функции при ее линейной интерполяции. При экстраполяции используются отрезки, проведенные через две крайние точки (рис. А.21).

Одномерная сплайн-аппроксимация

При линейной интерполяции уже первая производная аппроксимирующей функции терпит разрывы в узловых точках. В тех случаях, когда есть основания полагать, что не только аппроксимирующая функция, но и ряд ее производных непрерывны, существенно лучшие результаты дает сплайн-аппроксимация. При ней исходная функция заменяется отрезками кубических полиномов, проходящих через три смежные узловые точки.

Коэффициенты полиномов рассчитываются так, чтобы непрерывными были первая и вторая производные.

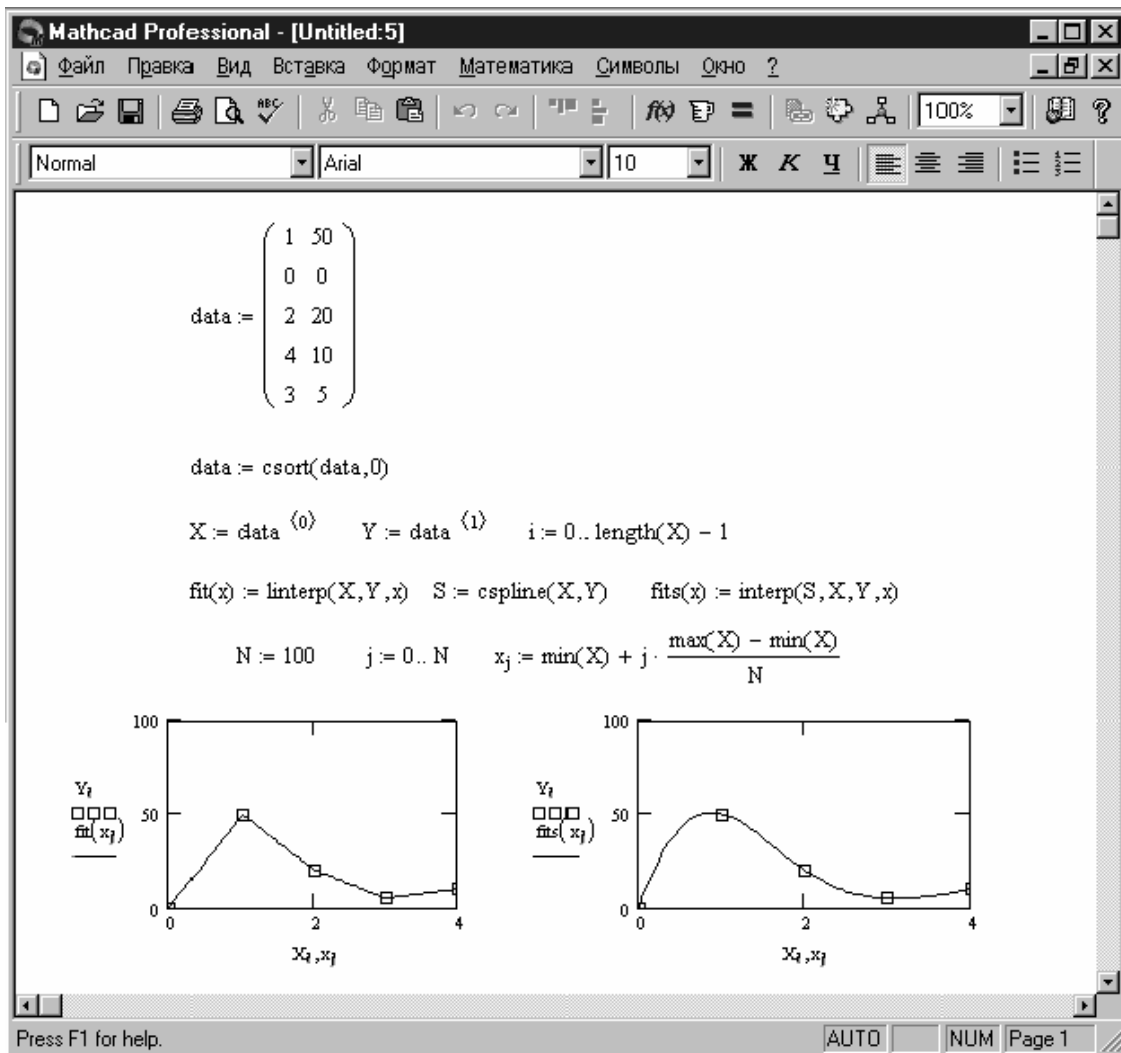


Рисунок А.21. Примеры аппроксимации

Для осуществления сплайновой аппроксимации *MathCAD* предлагает три встроенные функции для получения векторов вторых производных:

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 в опорных точках к параболической кривой;

lspline (*VX*, *VY*) – возвращает вектор *VS* вторых производных при при-

ближении к опорным точкам прямой.

Четвертая функция *interp* (VS, VX, VY, x) возвращает значение векторов VS, VX, VY и заданного значения x .

Таким образом, сплайн-аппроксимация проводится в два этапа. На первом с помощью одной из приведенных выше функций отыскивается вектор вторых производных функции $y(x)$, заданной векторами значений ее абсцисс и ординат; на втором для каждой искомой точки вычисляется значение $y(x)$ с помощью встроенной функции *interp*. Примеры линейной и кубической сплайновой аппроксимации приведены на рис. А.21.

Нетрудно заметить, что график при линейной аппроксимации оказывается слишком грубым, отчетливо видны “углы” – точки разрыва производной. В то же время сплайн-аппроксимация, несмотря на малое число точек, дает прекрасные результаты – график функции оказывается плавным, поскольку обеспечена непрерывность первой и второй производных.

Полиномиальная аппроксимация

В ряде случаев, для повышения точности аппроксимации, бывает необходимо использовать полином выше третьей степени, т.е. n -степенной многочлен. Число узловых точек ($n + 1$) в этом случае определяют степень полинома n . Использование данного вида аппроксимации содержит ряд подготовительных операций: формирование матрицы XI (см. рис. А.22), на основании которой вычисляются коэффициенты полинома. К сожалению, поведение полинома между узловыми точками и за пределами области интерполяции, т. е. при экстраполяции, зачастую трудно предсказуемо. Кроме того, с ростом степени аппроксимирующего полинома (n до 8-10) погрешность полиномиальной аппроксимации резко возрастает. Приемлемые результаты достигаются для достаточно гладких функций.

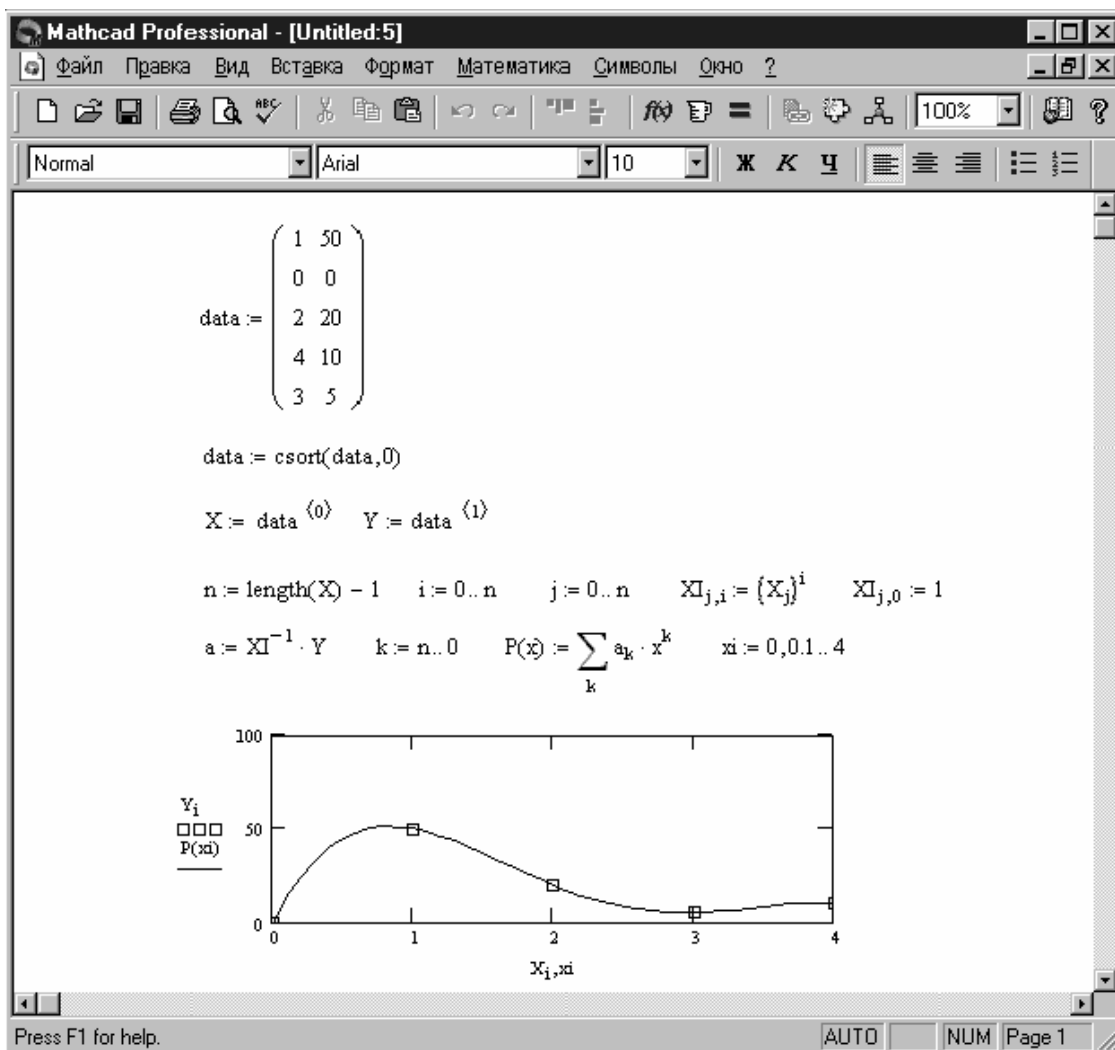


Рисунок А.22. Пример полиномиальной аппроксимации

А.4. Программирование

Вплоть до появления последних версий системы *MathCAD* возможности программирования в них были крайне ограниченными. Фактически его среда позволяла реализовать лишь линейные программы, осуществляя функциональное программирование, в основе которого лежит понятие функции. Отсутствовала возможность задания завершенных программных модулей. Эти возможности появились, начиная с версии *MathCAD PLUS 6.0*, и в *MahtCAD 2000 Professional* имеются уже в расширенном варианте. Они сосредоточены в палитре программных элементов.

Оператор Add Line

Выполняет функции расширения программного блока. Расширение фиксируется удлинением вертикальной черты программных блоков или их древовидным развитием. Благодаря этому могут быть созданы сколь угодно большие программы.

Оператор внутреннего присваивания ←

Выполняет функции внутреннего локального присваивания. Локальный характер присваивания означает, что присвоенное значение переменной сохраняется только в теле программы.

За пределами тела программы данная переменная может быть не определена, либо равна заданному ей значению вне программного блока.

Оператор создания условных выражений if (если)

Является оператором для создания условных выражений. Он задается в виде: *Выражение if Условие*. Если *Условие* выполняется, то возвращается значение *Выражения*.

Совместно с этим оператором часто используются оператор прерывания *break* и оператор иного выбора *otherwise*.

Оператор for (для)

Служит для организации циклов с заданным числом повторений. Он записывается в виде: *for Var ∈ N_{min} .. N_{max}*. Эта запись означает, что если переменная *Var* меняется с шагом +1 от значения *N_{min}* до *N_{max}*, то выражение, помещенное в шаблон, будет выполняться. Переменную счетчика *Var* можно использовать в выражения программы.

Оператор while (в то время как)

Служит для организации циклов, действующих до тех пор, пока выполняется некоторое условие. Он записывается в виде: *while Условие*. Выполняемое выражение записывается на место шаблона.

Оператор otherwise (иначе)

Обычно используется совместно с оператором *if* для присваивания значений при условиях, противоположных используемым в операторе *if*.

Оператор break (прервать)

Вызывает прерывание работы программы всякий раз, как он встречается.

Чаще всего данный оператор используется с операторами *if*, *while* и *for*, обеспечивая переход в конец тела цикла.

Оператор continue (продолжить)

Используется для продолжения работы после прерывания выполнения программы.

Оператор также обычно применяется совместно с операторами *while* и *for*, обеспечивая возврат в начало цикла.

Оператор-функция возврата return

Особый оператор-функция *return* прерывает выполнение программы и возвращает значение своего операнда, стоящего следом за ним.

Например: *return 0 if x < 0*.

Оператор on error (по ошибке)

Оператор обработки ошибок позволяет создавать конструкции обработчиков ошибок. Он задается в виде: *Выражение_1 on error Выражение_2*.

Если при выполнении *Выражения_1* возникает ошибка, то выполняется *Выражение_2*.

Пример реализации программного блока для вычисления температуры нагревательного устройства с терморегулятором, рассматриваемого в лабораторной работе № 3, приведен на рис. А.23.

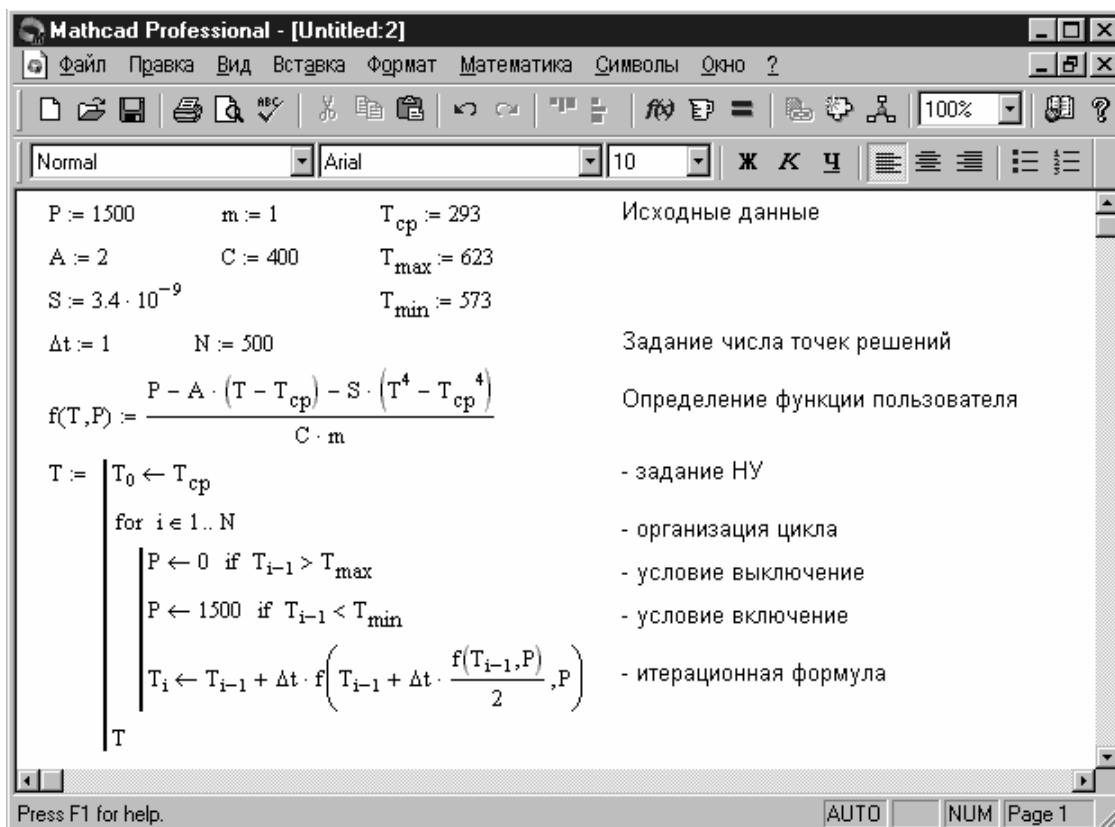


Рисунок А.23. Пример использования программного блока

А.5. Системный интегратор *MathConnex*

MathConnex – новое средство, появившееся в последних версиях системы *MahtCAD*. Оно выполняет две важные функции:

- служит для интеграции различных приложений с пакетом *MahtCAD* и обеспечения их совместной работы с использованием объектных связей;
- выполняет функции имитационного моделирования систем, представленные типовыми блоками в виде функциональной схемы.

Благодаря системному интегратору, возможно простое и наглядное установление сложных взаимосвязей между различными программными продуктами – математической системой *MahtCAD*, матричной системой *MatLAB*, графической системой *Axum*, электронными таблицами *Excel* из пакета *Microsoft Office* и др. Мощность такой объединенной системы воз-

растает многократно, поскольку она позволяет использовать для решения задач пользователя целый арсенал программных систем, включая встроенные в них специфические и подчас уникальные функции.

На рис. А.24 приведен один из типовых примеров применения *MathConnex* – на основании данных, вводимых из таблицы *Excel*, документ *MahtCAD* выполняет полиномиальную регрессию, вычисляет коэффициенты полинома и коэффициент корреляции и выводит их с помощью специальных инспектирующих блоков.

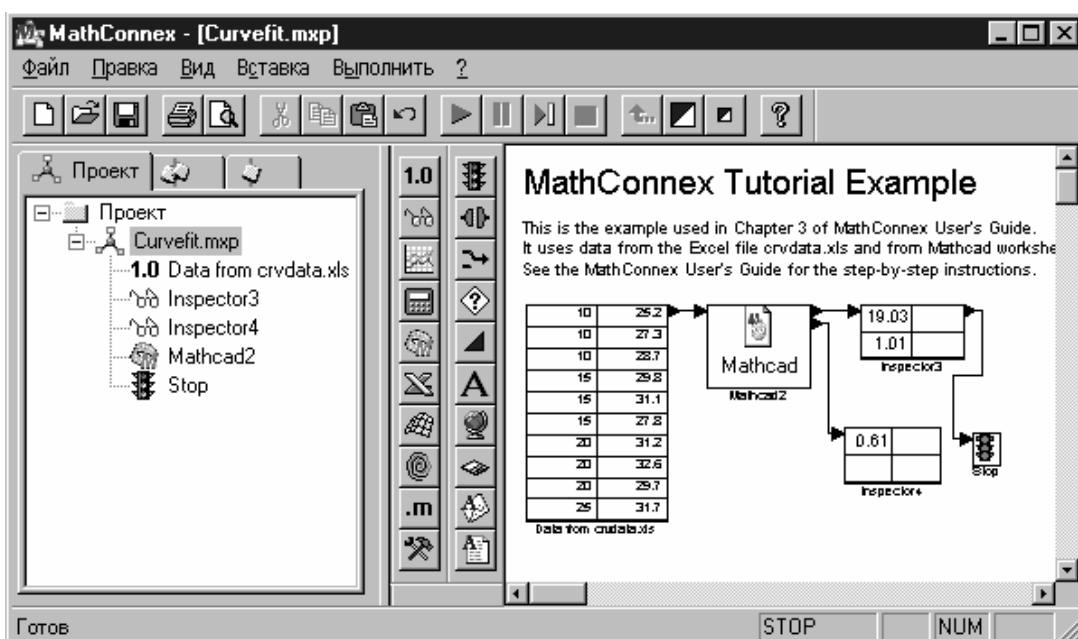


Рисунок А.24. Пример использования системного интегратора

А.5.1. Интерфейс пользователя

Для запуска системы *MathConnex* используется кнопка в инструментальной панели *MahtCAD* с пиктограммой в виде шарика и трех прямоугольников. Появившееся окно *MathConnex* имеет строку своего главного меню и расположенную под ним панель инструментов. В правой части экрана системы находится большое окно для рабочего документа и палитра компонентов, в левой – малое окно проектирования (рис. А.24).

Главное меню MathConnex

Главное меню системы имеет типичный для *Windows*-приложений набор позиций и относящихся к ним подменю. Отметим лишь, что все операции его относятся к деталям интерфейса *MathConnex* и к документам данного приложения, которые имеют расширение *mhr*.

Кроме того, следует отметить специфическую для *MathConnex* позицию *Выполнить (Run)*, которая выводит подменю, содержащее следующие операции:

<i>Run</i>	– пуск имитации работы документа;
<i>Pause</i>	– пауза в имитации;
<i>Step</i>	– имитация для следующего блока в режиме пошаговой работы;
<i>Stop</i>	– остановка имитации;
<i>Single Step Mode</i>	– переход к пошаговой симуляции;
<i>Highlight</i>	– установка подсветки исполняемого компонента.
<i>Components</i>	

Инструментальная панель MathConnex

Инструментальная панель содержит ряд кнопок типичного вида и назначения, кроме того, на ней есть две новые группы кнопок, которые имеют следующее предназначение (рис. А.25):

- *кнопки управления имитацией*

- a* – пуск имитации;
- b* – пауза в имитации;
- c* – пошаговое выполнение имитации;
- d* – остановка имитации;

- *кнопки управления уровнем исполнения и размерами документа*

- e – переход к предыдущему по уровню блоку;
- f – увеличение размеров документа;
- g – уменьшение размеров документа.

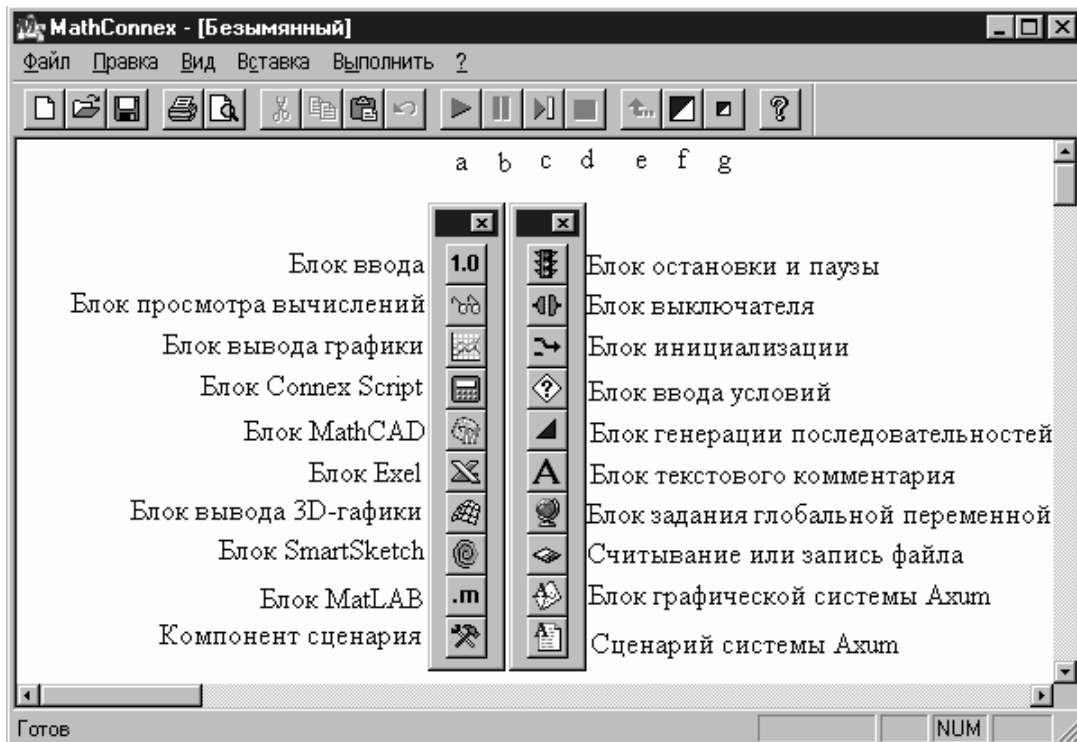


Рисунок А.25. Назначение кнопок палитры компонентов

А.5.2. Подготовка документов

Подготовка документов в системе *MathConnex* сводится к созданию блок-схемы решаемой задачи, которая может содержать рассмотренные выше компоненты, связи между ними и текстовые комментарии. Все это размещается в окне документов. Для ввода блоков необходимо установить курсор на кнопку требуемого блока. Затем, нажав и удерживая левую клавишу мыши, перетащить блок в нужное место окна редактирования и, отпустив клавишу, зафиксировать его. В ряде случаев система выводит диалоговые установочные окна для задания числа вводов и выводов, обозначаемых на блоках треугольниками, и некоторых других параметров.

Например, при использовании блока таблиц *Excel*, первоначально появляется окно, изображенное на рис. А.26.

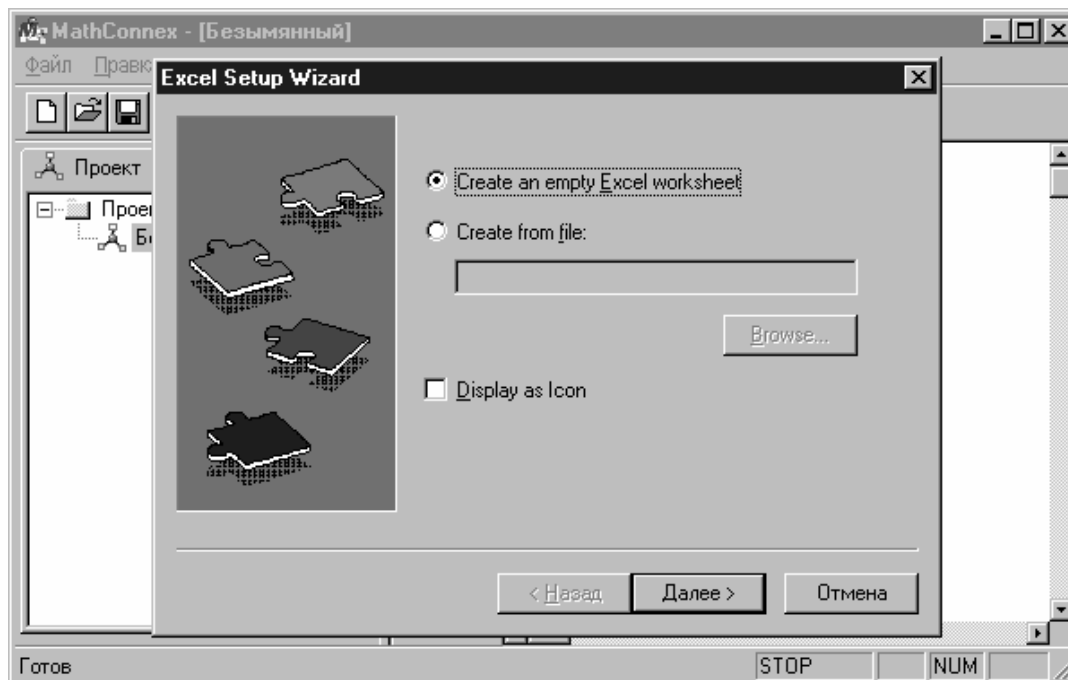


Рисунок А.26. Первое установочное окно блока *Excel*

Первая (верхняя) строка которого предлагает использовать таблицу в виде чистой заготовки, заполняемой пользователем непосредственно при работе с документом *MathConnex*. Активизация второй строки позволяет вставить в блок *Excel* заранее подготовленный *xls*-файл, используя клавишу *Browse*. Третья строка фиксирует вывод блока в виде иконки либо рабочей области. По готовности установок пользователя нажимается кнопка *Далее*, которая вызывает следующее установочное окно (рис. А.27).

Данное окно предназначено для определения параметров вводов и выводов рассматриваемого блока. Отметим, что их максимальное число не может превышать четырех для каждого отдельного блока. Кнопка *Готово* вставляет таблицу *Excel* в разрабатываемый документ *MathConnex*, при этом в окне проектирования появляется соответствующая пиктограмма.

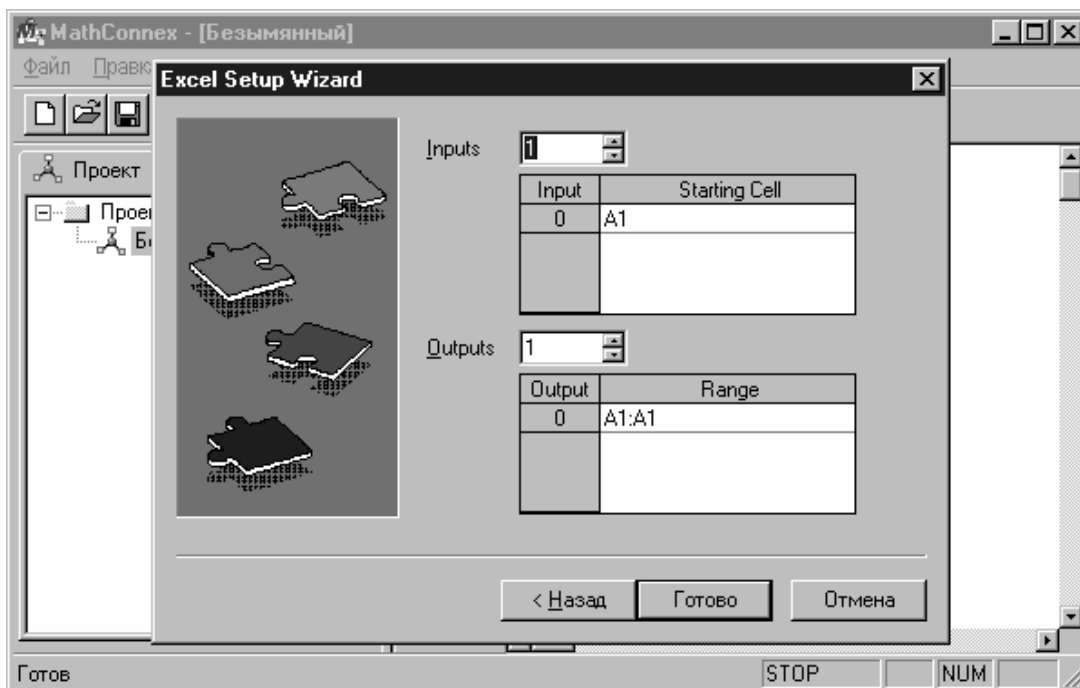


Рисунок А.27. Второе установочное окно блока *Excel*

Вводы и выходы блоков в общем случае соединяются соответствующим образом с помощью активизации их курсором мыши, превращающимся в кисть руки, держащей карандаш. Кроме того, вводы и выходы каждого блока должны быть описаны в них интерфейсными функциями $in0$, ..., $in3$ (вводы) и $out0$, ... $out3$ (выводы). По готовности документа для пуска имитации достаточно активизировать соответствующую кнопку на инструментальной панели.

А.5.3. Редактирование блоков и документов

В окне проектирования (левая часть экрана) приводится полный перечень блоков вставленных, в окне документов. Установив курсор мыши на любую позицию данного перечня, и нажав левую клавишу мыши, можно выделить заштрихованной рамкой соответствующий блок в окне документов. Уцепившись курсором за эту рамку, блок можно переставить в любое место рабочего окна. Эту же процедуру можно выполнить, выделив

блок непосредственно в окне документов. Кроме того, используя шаблоны в виде маленьких темных прямоугольников по сторонам рамки, изменяют размеры блока. Двойной щелчок левой клавиши вводит режим редактирования блока. При этом автоматически запускается создавшее блок приложение. После редактирования нужно установить курсор на свободное поле рабочего окна и щелкнуть левой клавишей мыши. При этом выделение исчезнет, а документ предстанет в виде, готовом для его симуляции.

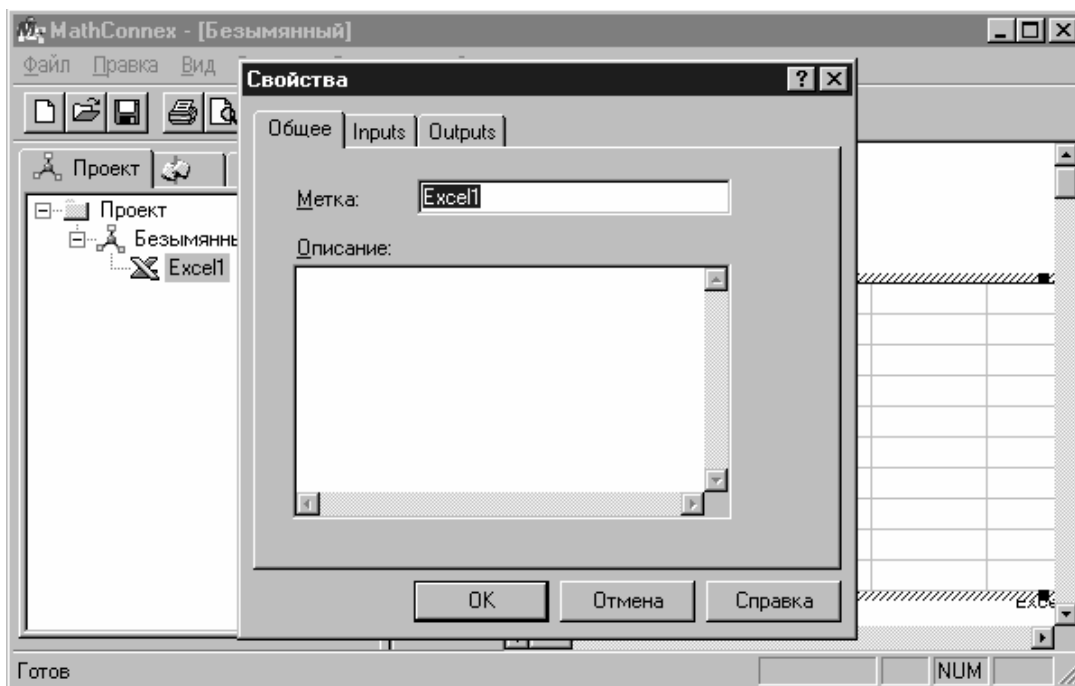


Рисунок А.28. Окно редактирования блока *Excel*

Для редактирования блоков полезно также использовать контекстно-зависимые меню, появляющиеся при нажатии правой клавиши мыши. На рис. А.28 показано окно редактирования блока *Excel*, в котором можно изменить его название (метку), ввести необходимый комментарий, а также, используя соответствующие закладки, отредактировать параметры вводов и выводов. При отладке документов полезен режим пошаговой работы, при котором на каждом шаге выполняется один блок. Обычно выполнение блока можно отличить лишь по изменению входящих в него данных, ре-

зультатов вычислений и графических построений. Однако, включив режим подсветки (*Highlight Components*), можно наблюдать выделение исполняемого блока (компонента) ярко-зеленой рамкой.

А.6. Справочная система и информационные ресурсы

К информационным ресурсам пакета *MathCAD 2000 Professional* относятся подсказка первоначального уровня (рис. А.29) – *Полезные советы (Tip of the Day)*, электронный справочник, обучающая программа, примеры применения – шпаргалки, электронные книги, справочные таблицы, доступ в *Internet* и др. Справочная система содержит весьма обширную информацию. Она полезна, например для выявления деталей применения того или иного оператора системы или функции, а также освоения работы с интерфейсом. Работа с ней аналогична правилам типичным для справочников (*Help*) любых других *Windows*-приложений.

Общий доступ ко всем справочным материалам открывает Центр информационных ресурсов – *Resource Center*. Его окно представлено на рис. А.30. Шпаргалки – *QuickSheets* являются профессионально выполненными примерами применения *MathCAD* при решении разнообразных математических задач. Кроме того, к услугам пользователя сотни примеров применения, расположенных на *WEB*-сервере корпорации *MathSoft*, доступном при работе с сетью *Internet*.

Особо следует отметить Справочный стол – *Reference Table*, встроенный в систему справочник математических, физических и химических формул и таблиц. Любые справочные данные (константу, формулу и т. д.) можно скопировать в буфер обменов и перенести в нужное место Вашего *MathCAD*-документа для использования в расчетах.

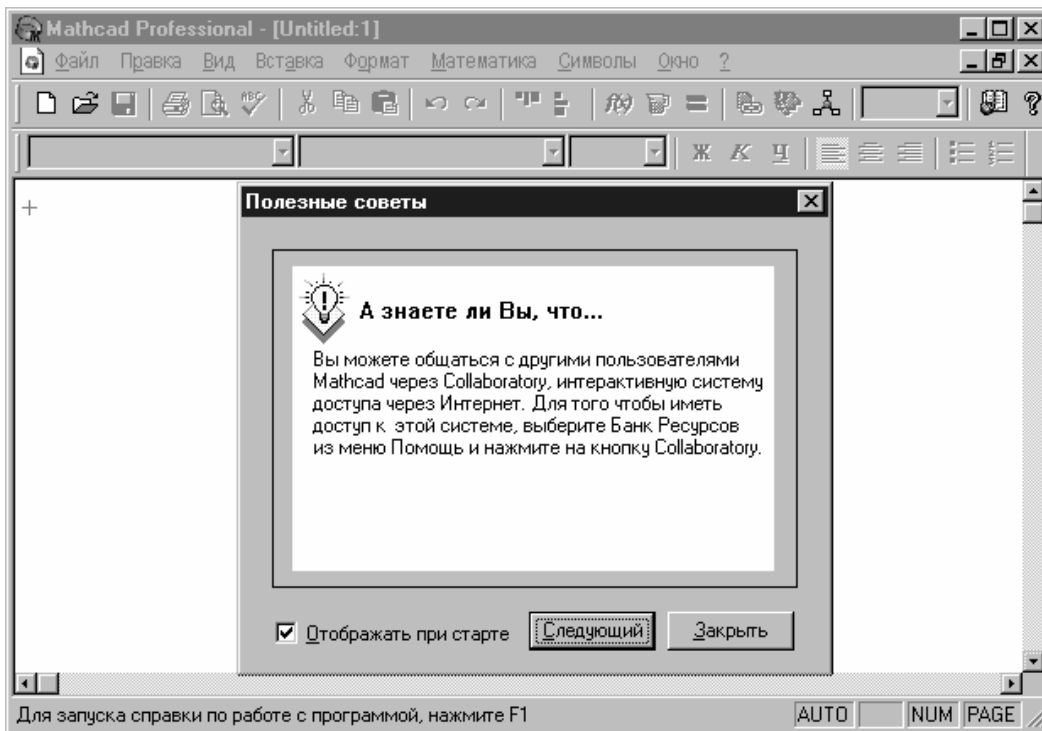


Рисунок А.29. Окно с подсказкой первоначального уровня

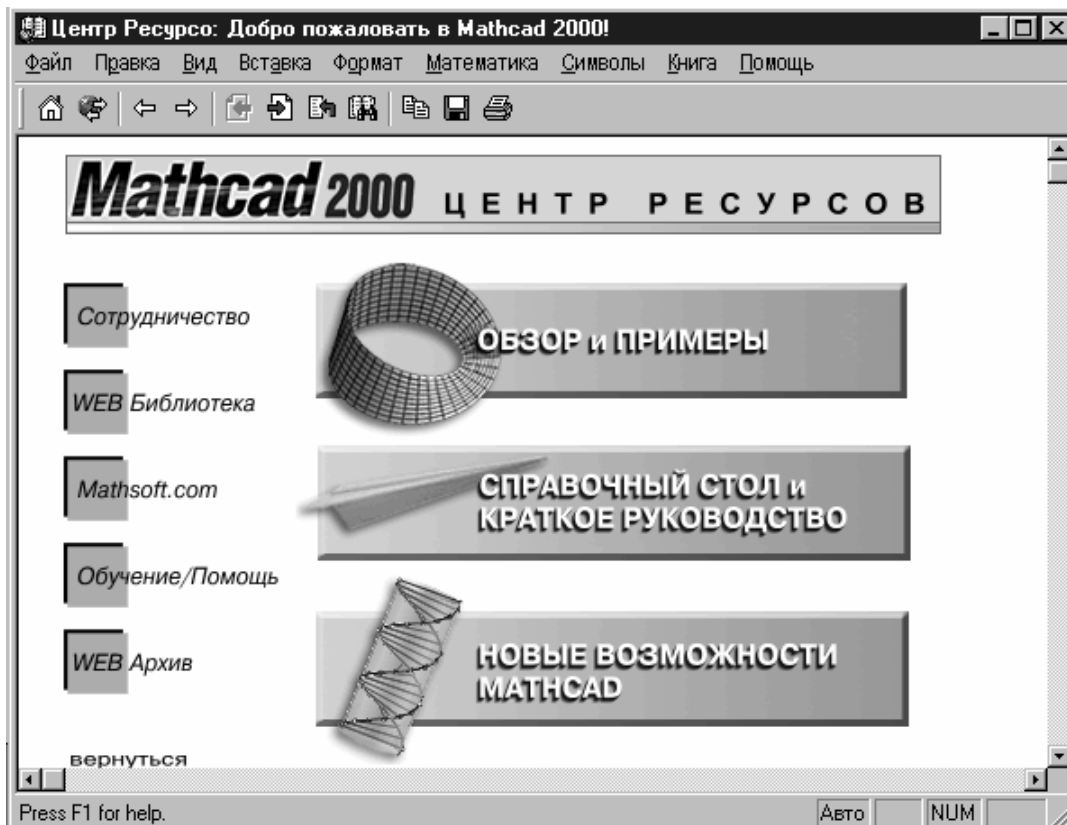


Рисунок А.30. Окно центра информационных ресурсов

Пакет символьных вычислений

Maple V

Пакет прикладных программ *Maple* ориентирован на решение задач в наиболее точной форме – аналитическом (символьном) виде. Однако его возможности не исключают проведения численных и графических вычислений.

Среда *Maple* состоит из трех компонентов: ядра, библиотек и экранного интерфейса. Ядро – это “математическое орудие”, выполняющее все виды расчетов. Оно написано и откомпилировано на языке программирования *C* и выполняет основную часть вычислений, производимых системой. Библиотеки включают в себя встроенные процедуры, написанные в среде *Maple* на ее собственном языке программирования и сохраняемые в отдельном файле. Текст команд и программ, написанный в *Maple*, не компилируется, а интерпретируется, что позволяет пользователю создавать собственные процедуры интерактивно в пределах среды.

Б.1. Интерфейс пользователя

Maple V присущи все особенности *Windows*-приложений (рис. Б.1). Его экранный интерфейс включает традиционные составные части:

- верхнюю строку с указанием имени системы и текущего окна;
- главное меню системы;
- инструментальную панель;

- контекстное меню;
- рабочее окно, в котором отсутствует полоса горизонтальной прокрутки, т.к. *Maple* форматирует содержимое в пределах ширины рабочей области;
- нижнюю строку статуса, в которой размещается ресурсная информация об использованной и свободной памяти, временные характеристики и т.п.

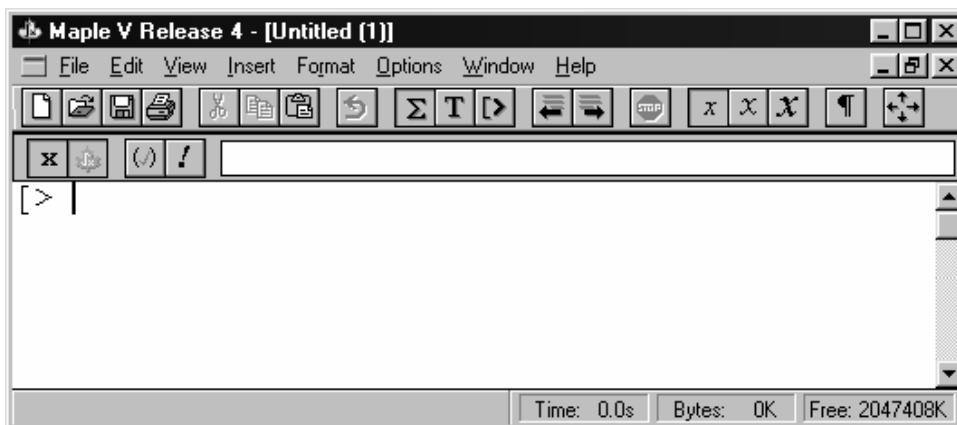


Рисунок Б.1. Окно системы *Maple V*

Главное меню системы

Главное меню *Maple V* содержит следующие позиции, обращение к которым вызывает выпадающие меню:

- | | |
|-------------------------|---|
| <i>File</i> (Файл) | – работа с файлами; |
| <i>Edit</i> (Правка) | – редактирование документов; |
| <i>View</i> (Вид) | – изменение средств обзора и включения/выключения элементов интерфейса; |
| <i>Insert</i> (Вставка) | – установка вставок объектов и их шаблонов; |
| <i>Format</i> (Формат) | – изменение формата (параметров) объектов; |
| <i>Options</i> (Опции) | – выбор операций контроля режима работы; |
| <i>Window</i> (Окно) | – управление окнами системы; |
| <i>Help</i> | – работа со справочной базой данных о системе. |

Инструментальная и контекстная панели

Кнопки панели инструментов *Maple V* разбиты на девять групп и обозначены цифрами на рисунке Б.2.

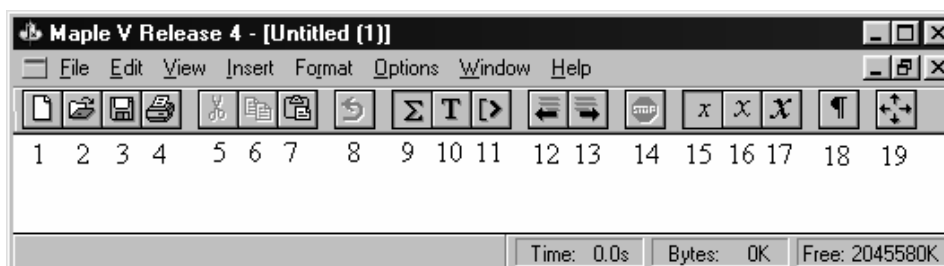


Рисунок Б.2. Назначение кнопок инструментальной панели

Ряд из них дублирует некоторые команды главного меню. Первая группа из четырех пиктограмм отражает общепринятое обозначение стандартных процедур: открыть новый файл, загрузить файл с диска, быстрое сохранение файла и вывод его на печать. Следующие три – соответственно вырезать, скопировать и вставить фрагмент, используя промежуточный буфер *Windows*. Восьмая пиктограмма отменяет последнюю операцию (если это возможно), девятая – активизирует режим ввода вычислений, десятая – ввод текстовой области, одиннадцатая – включает и выключает знак “>” в начале строки ввода. Следующая группа из двух кнопок предназначена для повышения/понижения уровней структуры документа. Четырнадцатая – позволяет прервать вычислительные процессы системы. Пиктограммы 15-17 изменяют масштаб шрифтов. Восемнадцатая включает и выключает скрытые символы. Последняя кнопка уменьшает рабочее окно.

Объекты контекстного меню предназначены (рис. Б.3):

- a* – переключение дисплея между текстовой и стандартной нотацией;
- b* – переключение типа выражения между активным режимом вычислений и инертным комментарием;
- c* – автокоррекция синтаксиса выражения;

- d* – выполнение текущего выражения;
- e* – строка редактирования выражений.

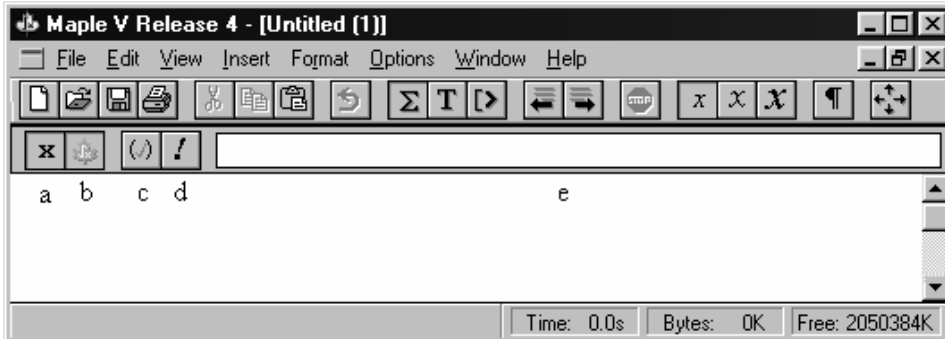


Рисунок Б.3. Назначение объектов математического контекстного меню

Рабочее окно

Интерфейс *Maple V* поддерживает концепцию рабочих листов (*worksheets*), которые объединяют текст, входные команды, вывод результатов и графику в одном документе (см. рис. Б.4).

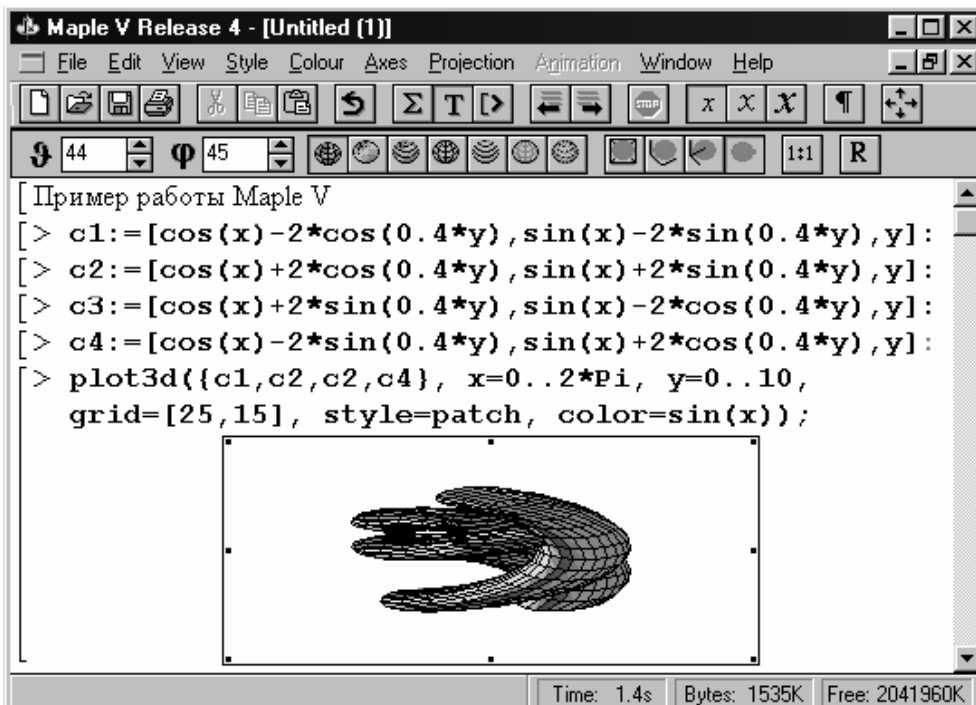


Рисунок Б.4. Пример рабочего документа

Программа позволяет работать одновременно с несколькими рабо-

чими листами и устанавливая между ними динамические связи, т. е. переводить результаты вычислений с одного листа на другой. Можно также запускать несколько программ одновременно, что дает возможность проводить сравнение вычислений при различных начальных значениях.

При вводе выражения и его выполнении в рабочем окне, исходная запись и полученный результат всегда охвачены общей скобкой слева. После выполнения команды вслед за полем вывода появляется новая командная строка с расположенным на ней курсором. Любое введенное выражение в *Maple* может заканчиваться символом « : » или символом « ; ». В первом случае результат не будет выводиться на экран, во втором – будет. Если пользователь желает употребить ранее введенное выражение, то необходимо воспользоваться символом « ” ». Причем ввод:

- ” – означает, что следует использовать выражение, введенное непосредственно перед текущим;
- ” ” – использовать введенное за одно выражение ранее;
- ” ” ” – воспользоваться введенным за два выражения ранее.

В зависимости от того, на каком объекте документа установлен курсор, изменяется вид контекстного меню в соответствии с операциями редактирования объекта. Активизация десятой пиктограммы инструментального меню вызывает область ввода текста (рис. Б.5).

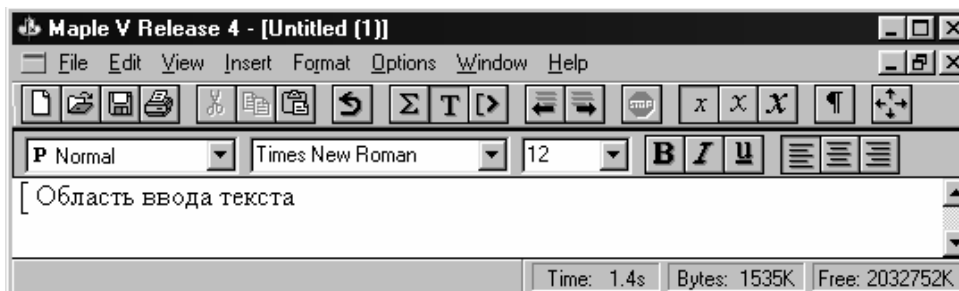


Рисунок Б.5. Область ввода текста

Б.2. Математические расчеты

Будучи автоматизированной системой, *Maple* существенно упрощает проведение символьных, численных и графических вычислений. Он может решать как простые, так и довольно сложные задачи. В настоящем приложении рассматривается только доля его возможностей, знание которых необходимо для успешного выполнения практической части пособия.

Ознакомится с потенциалом пакетов данного класса в более полном объеме можно из источников, приведенных в списке рекомендуемой литературы.

Б.2.1. Решение алгебраических уравнений и систем

Решение линейных и нелинейных алгебраических уравнений и их систем в общем виде *Maple* осуществляет с помощью команды *solve* (решить). Если системе не удастся найти решение в символьном виде, то можно использовать команду *fsolve*, которая найдет его в численном виде. Примеры решений системы уравнений приведены на рисунке Б.6.

Среда *Maple* позволяет выполнять все стандартные операции, определенные в линейной алгебре. Они становятся доступными при подключении библиотеки *linalg*. Любая из библиотек подключается через команду *with* с указанием ее имени. В нашем случае следует ввести выражение:

> with(linalg):

Таким образом, для системы линейных уравнений пользователь может применить векторно-матричный вариант решения (рис. Б.7). Обратите внимание, в последнем примере встречаются команды: *matrix* – для ввода матрицы; *vector* – для ввода вектора; *multiply* – для умножения матриц; *evalf* – для представления выражения в численном виде.

```

Maple V Release 4 - [Untitled (2)]
File Edit View Insert Format Options Window Help
C 2D Output Times New Roman 12 B I U
[> sist:={p*sin(alpha)=x,
[> y+m*g=p*cos(alpha),
[> p*a=m*g*(a+b)*cos(alpha)}:
[> con1:=solve(sist, {x,y,p}):
con1 := {p =  $\frac{m g (a+b) \cos(\alpha)}{a}$ , y =  $\frac{m g (-a + \cos(\alpha)^2 a + \cos(\alpha)^2 b)}{a}$ ,
x =  $\frac{m g (a+b) \cos(\alpha) \sin(\alpha)}{a}$ }
[> a:=0.1: b:=0.9: m:=5: g:=9.807: alpha:=20*Pi/180:
[> con2:=fsolve(sist, {x,y,p}):
con2 := {y = 383.9549463, p = 460.7782766, x = 157.5954522}
Time: 0.0s Bytes: OK Free: 2043540K

```

Рисунок Б.6. Примеры решения системы алгебраических уравнений

```

Maple V Release 4 - [Untitled (2)]
File Edit View Insert Format Options Window Help
C 2D Output Times New Roman 12 B I U
[> with(linalg):
[> A:=matrix([[-1,0,sin(alpha)],[0,1,-cos(alpha)],[0,0,a]]):
[> B:=vector([0,-m*g,m*g*(a+b)*cos(alpha)]):
[> a:=0.1: b:=0.9: alpha:=20*Pi/180: m:=5:
g:=9.807:
[> X:=multiply(A^(-1),B):
X :=  $\begin{bmatrix} 490.3500000 \sin\left(\frac{1}{9}\pi\right) \cos\left(\frac{1}{9}\pi\right), \\ -49.03500000 + 490.3500000 \cos\left(\frac{1}{9}\pi\right)^2, 490.3500000 \cos\left(\frac{1}{9}\pi\right) \end{bmatrix}$ 
[> evalf("):
[ 157.5954522, 383.9549464, 460.7782766 ]
Time: 3.4s Bytes: 1215K Free: 2033916K

```

Рисунок Б.7. Векторно-матричный вариант решения системы уравнений

Б.2.2. Решение дифференциальных уравнений с визуализацией результатов

Графическая библиотека *DEtools* содержит средства для построения решений обыкновенных дифференциальных уравнений и некоторых типов уравнений в частных производных, включая поля решений и фазовый портрет. Данная библиотека также содержит средства визуализации решений. Ее подключение происходит после выполнения команды:

> with(DEtools):

В состав данной библиотеки входят следующие семь функций:

- DEplot* – построение решений дифференциальных уравнений и систем;
- DEplot1* – построение решения уравнения первого порядка;
- DEplot2* – построение решения системы из двух уравнений;
- Dchangerav* – подстановка новых переменных в уравнение;
- PDEplot* – построение решения квазилинейного уравнения первого порядка в частных производных;
- dfieldplot* – построение поля решения;
- phaseportrait* – построение фазового портрета.

Рассмотрим детали использования функции *DEplot*.

Формат вызова:

DEplot (*diffeq*, *vars*, *trange*, *inits*, *<options>*);

DEplot (*diffeq*, *vars*, *trange*, *xrange*, *yrange*, *<options>*).

Параметры:

diffeq – система дифференциальных уравнений;

vars – имена переменных;

trange – область определения независимых переменных;

- inits* – начальные условия;
- xrange* – область построения первой независимой переменной;
- yrange* – область построения второй независимой переменной;
- options* – опции.

Параметр *inits* – набор начальных условий в виде одной из форм:

$\{[t0, x0, y0], [t1, x1, y1], [t2, x2, y2], \dots\}$ или

$\{[x(t0)=x0, y(t0)=y0], [x(t1)=x1, y(t1)=y1], [x(t2)=x2, y(t2)=y2], \dots\}$.

Все параметры $\langle options \rangle$ указываются в виде: параметр = значение:

- stepsize* = *h* – значение *stepsize* используется для задания шага в выбранном методе решения поставленной задачи, по умолчанию используется $stepsize = (b - a)/20$, где $a..b$ – область определения независимой переменной (т.е. параметр *trange*);
- iteration* = $\langle integer \rangle$ – число шагов итерации между двумя точками, по умолчанию используется $iteration = 1$;
- limitrange* = *true* – прекратить интегрирование, если интегральные кривые выходят за пределы указанного промежутка для *x* и *y*, по умолчанию используется $limitrange = false$;
- scene* = [$\langle name \rangle$, $\langle name \rangle$, $\langle name \rangle$] – указывает вид построения, можно также использовать для изменения положения осей: [*x*, *y*] – будет построен двухмерный график $y(x)$, [*x*, *y*, *t*] – трехмерное изображение всех переменных, причем [*x*, *y*, *t*] *t* – вертикальная ось;
- metod* = $\langle scheme \rangle$ – определяет метод решения задачи, может быть: 'euler', 'backeuler', 'impeuler' или 'rk4'.

Пример использования функции *Deplot* приведен на рис. Б.8.

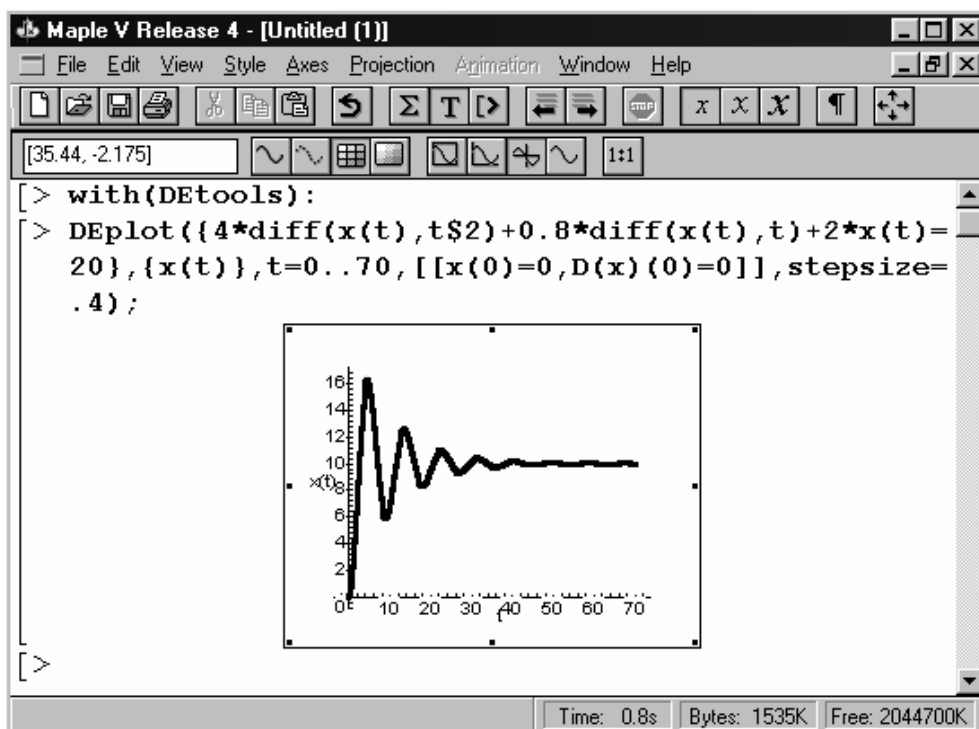


Рисунок Б.8. Пример решения дифференциального уравнения

Обратите внимание на использование оператора повторения – \$, в примере он применяется для задания степени производной.

Б.2.3. Аппроксимация зависимостей вида $y(x)$

Maple V содержит мощную библиотеку *stats*, поддерживающую разнообразные статистические вычисления и генерирующую реализации случайных последовательностей, с заданными законами распределения. Подключение данной библиотеки осуществляется командой:

> **with(stats):**

Подбиблиотека fit

Предназначена для нахождения корреляционных отношений и аппроксимации статистических данных выбранными зависимостями.

Формат вызова:

stats [*fit, leastsquare* [*vars, eqn, parms*]] (*data*), или

$fit [leastsquare [vars, eqn, parms]] (data),$

где $data$ – список данных; $vars$ – список переменных, в порядке представления данных; eqn – аппроксимирующее уравнение (по умолчанию линейное); $parms$ – множество параметров, которые будут заменены вычисленными значениями. Пример аппроксимации данных приведен на рис. Б.9.

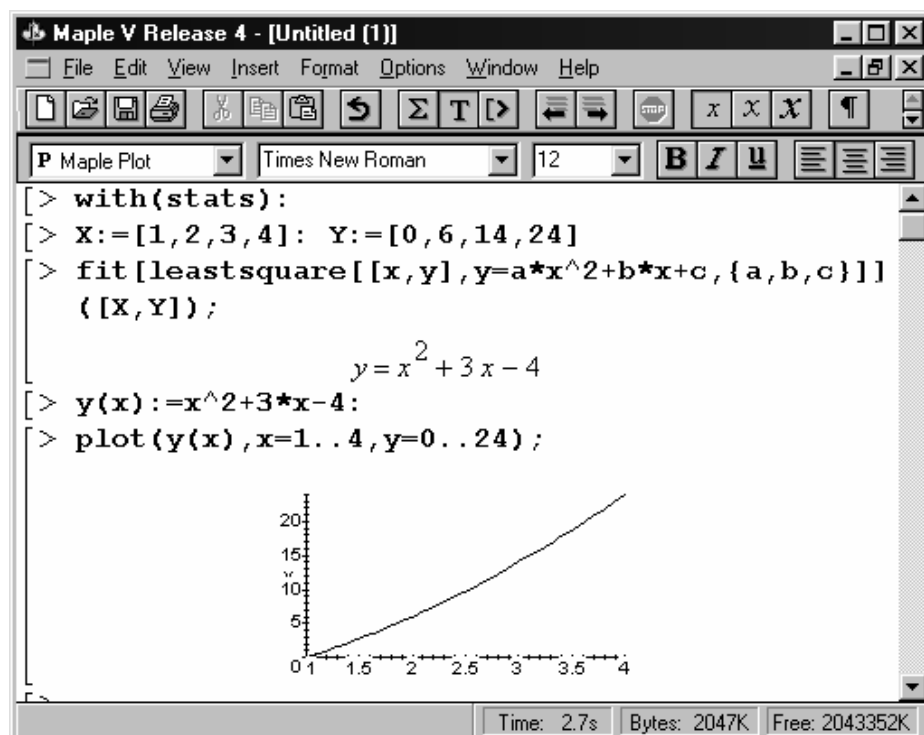


Рисунок Б.9. Пример аппроксимации данных

Б.3. Графика и анимация

Maple является прекрасным инструментом для визуализации информации об исследуемой функции. Используя мышь, можно просмотреть локальные минимумы и максимумы, нулевые точки.

Б.3.1. Построение двумерных графиков

Задание областей

Область – это окно декартовой системы координат, в которой строится график. Она определяется следующим образом:

$x = \text{нижняя граница}.. \text{верхняя граница}.$

Числа, определяющие границы, должны быть действительными. Области можно определять с помощью констант – *infinity* (бесконечность), *Pi* (π).

Стили

При построении графиков можно выбрать стиль (тип) интерполирования. Задается стиль с помощью ключевого слова *style* (см. рис. Б.10). В *Maple* реализуются три стиля:

point – построение в виде точек;

line – линейная интерполяция;

patch – построение в виде многоугольников.

По умолчанию используется *line*.

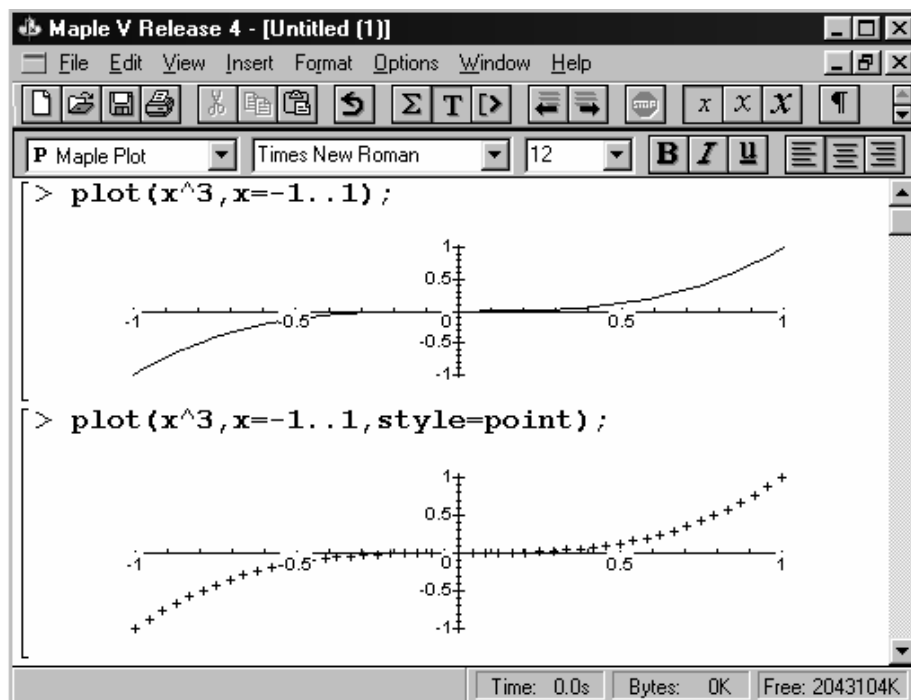


Рисунок Б.10. Примеры использования стилей

Параметрическая графика

При построении параметрических функций (см. рис. Б.11) используется следующий синтаксис команды *plot*:

plot ($[x(t), y(t), t = \text{диапазон изменения}]$, *h*, *v*, *options*).

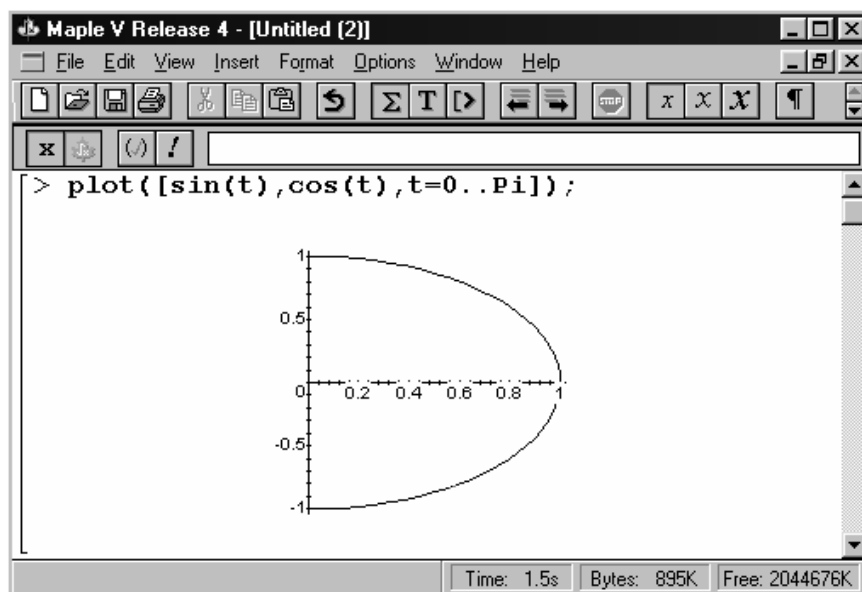


Рисунок Б.11. Пример построения параметрических функций

Совмещение графиков

На одной координатной плоскости можно построить несколько графиков. При таком построении *Maple* автоматически выбирает разные цвета для цвета графиков функций. Допустимо совмещать обычную и параметрическую графику (рис. Б.12). При построении графика функции в полярных координатах используется команда $coords = polar$. В этом случае первый параметр функции – радиус, второй – угол (см. рис. Б.13).

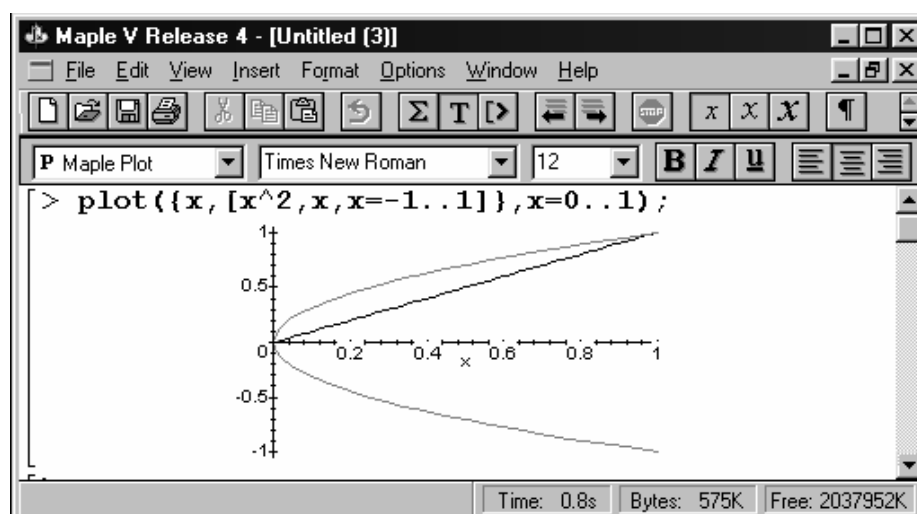


Рисунок Б.12. Пример совмещения графиков

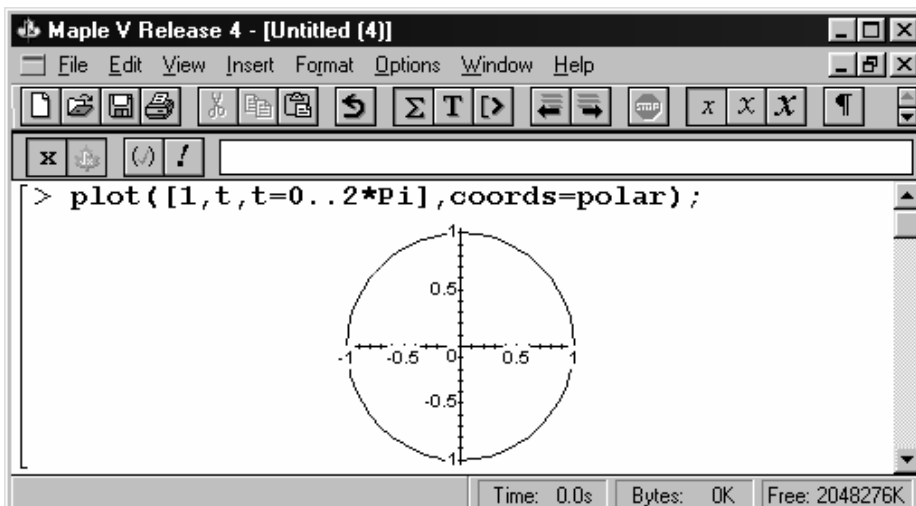


Рисунок Б.13. Пример построения графика в полярных координатах

Б.3.2. Построение трехмерных графиков

Для построения графиков $3D$ используется функция *plot3d*.

Синтаксис:

plot3d (*expr1*, *x=a..b*, *y=c..d*)

plot3d (*f*, *a..b*, *c..d*)

plot3d (*[f,g,h]*, *a..b*, *c..d*), где

f, *g*, *h* – отображаемые функции (или функция);

expr1 – выражение, зависящее от *x* и *y* (функция двух переменных);

a, *b*, *c*, *d* – действительные константы;

x, *y* – имена.

Пример построения трехмерного графика приведен на рис. Б.4.

Б.3.3. Оформление графиков

Параметры, используемые для оформления графиков, перечисляются в команде *plot* (*plot3d*) после указания областей в следующей форме:

< имя параметра > = < значение >.

Рассмотрим назначение некоторых из них:

- coords* – устанавливает тип системы координат (*cartesian* – декартова, *spherical* – сферическая, *cylindrical* – цилиндрическая, *polar* – полярная), по умолчанию устанавливается декартова;
- numpoints = n* – определяет минимальное количество точек графика, по умолчанию $n = 49$;
- color* – определяет цвет построения (*black, blue, brawn, green, red*);
- title = t* – определяет заголовок чертежа, где *t* – текст названия;
- thickness = n* – определяет толщину линий, $n = 0, 1, 2, 3$ (0 – по умолчанию);
- linestyle = n* – стиль линии, 0 и 1 – сплошная линия, $n > 1$ – различные шаблоны заполнения точками;
- symbol = s* – символ для точек чертежа, *s* – одно из выражений: *box* (квадрат), *point* (точка), *diamond* (ромб), *cross* (крест), *circle* (круг);
- font = l* – шрифт для текстовых объектов графика, *l* – список [*family, style, size*], где *family* – одно из выражений *times, courier, helvetica, symbol*; для *times* стиль может быть одним из выражений *roman, bold, italic, bolditalic*; для *courier* и *helvetica* стиль может быть выбран из *bold, oblique, boldoblique*; для *symbol* стиль не указывается;
- size = n* – определяет размер точек шрифта *n*;
- titlefont = l* – определяет шрифт для заголовка (так же, как для *font*);
- axesfont = l* – определяет шрифт для координатных меток координатных осей (так же, как для *font*);

- $labelfont = l$ – определяет шрифт для меток на координатных осях (так же как для $font$);
- $scaling = s$ – управляет масштабированием, s может быть *constrained* (сдавленный) или *unconstrained* (несдавленный), по умолчанию – *unconstrained*.

Примеры оформления графиков приведены на рис. Б.14 и Б.15.

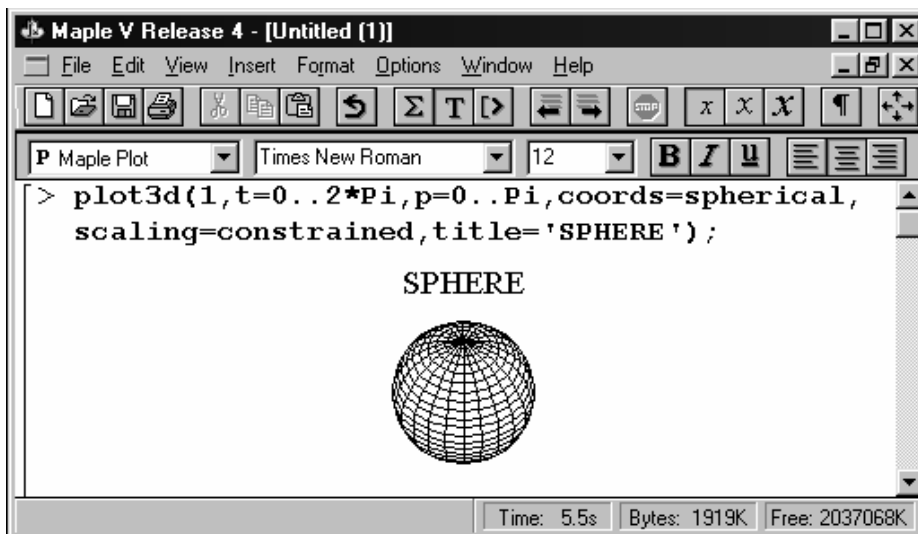


Рисунок Б.14. Пример оформления трехмерного графика

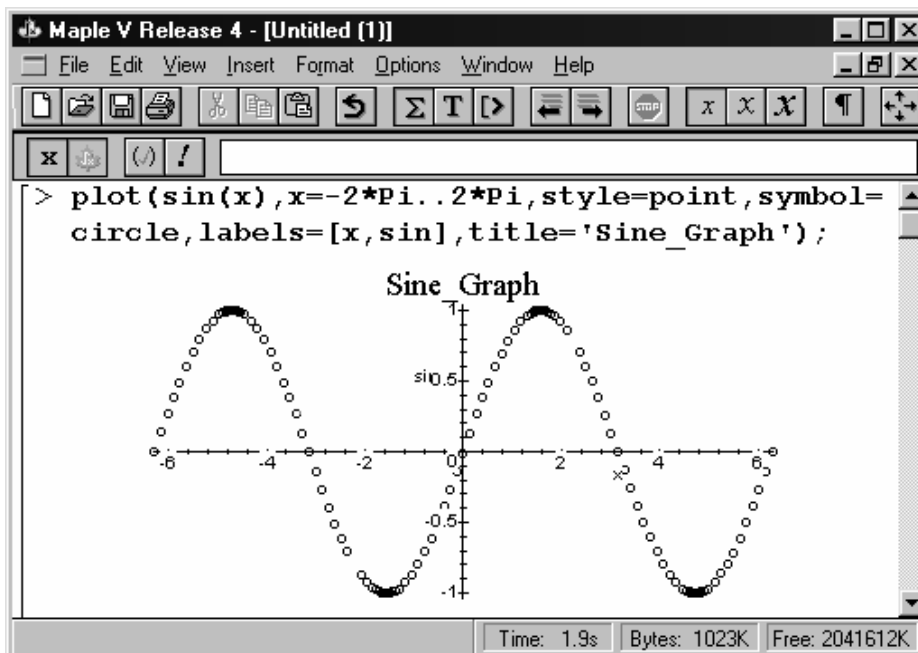


Рисунок Б.15. Пример оформления двухмерного графика

Б.3.4. Анимация

В *Maple* возможна анимация двухмерных и трехмерных графиков с помощью команд *animate* и *animate3d*. Для ее реализации необходимо предварительно подключить библиотеку *plots*.

Синтаксис команд:

animate ($F(x, t), x, t$); *animate3d* ($F(x, y, t), x, y, t$),

где x, y, t – диапазоны изменения величин.

При анимации происходит следующее: изменяется значение t и при его фиксированных значениях строится график функции F . Количество выводимых кадров можно установить параметром *frames* (по умолчанию *frames* = 16). Для просмотра анимационного графика следует выделить его область, при этом появится измененное соответствующим образом для работы с графиками главное меню, в котором достаточно выбрать команду *Animation*, а затем запустить просмотр анимации, см. рис. Б.16.

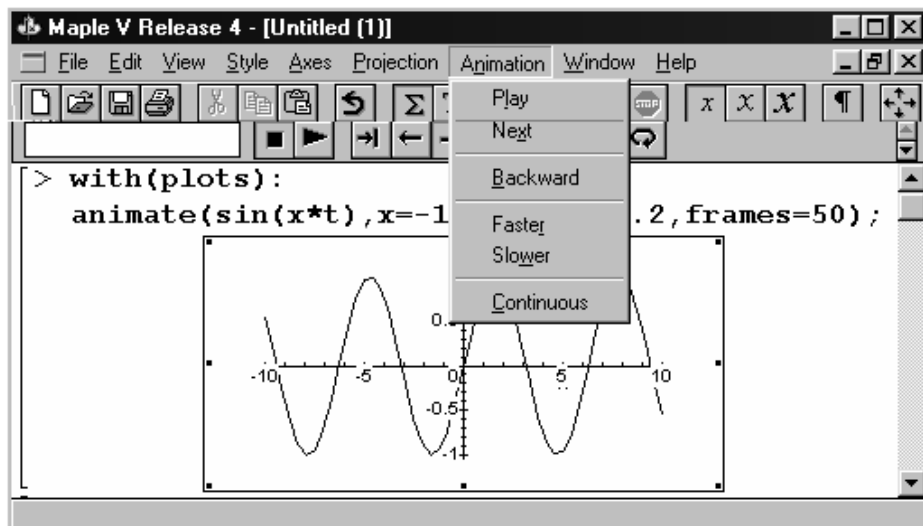


Рисунок Б.16. Реализация и просмотр анимационного графика

Б.4. Процедурное программирование

Программирование в *Maple* аналогично соответствующим конструкциям языков высокого уровня, рассмотрим ряд из них.

If <условие> then <операторы> else <операторы> fi

При помощи данной конструкции обеспечиваются условные переходы, например см. рис. Б.17.

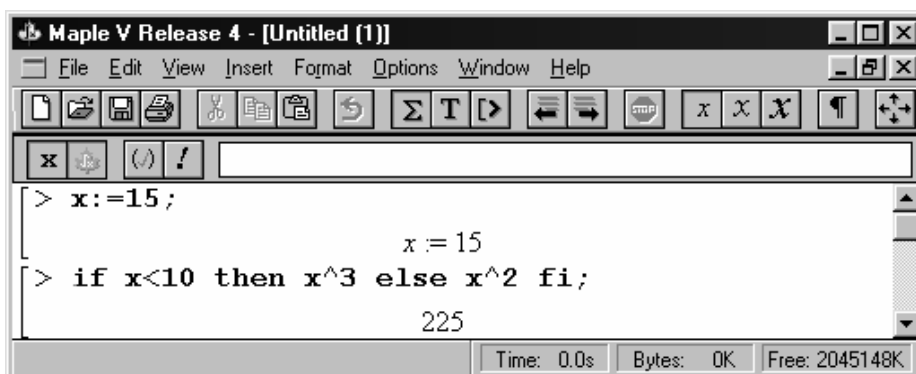


Рисунок Б.17. Пример использования конструкции *if / then / else / fi*

Обратите внимание на использование выражения «*fi*» (*if* в обратном чтении) в конце сочетания операторов, он указывает на завершение программной конструкции. После каждой из лексем *then* и *else* не обязательно использование одного оператора, как в приведенном примере, может быть реализовано любое количество программных кодов.

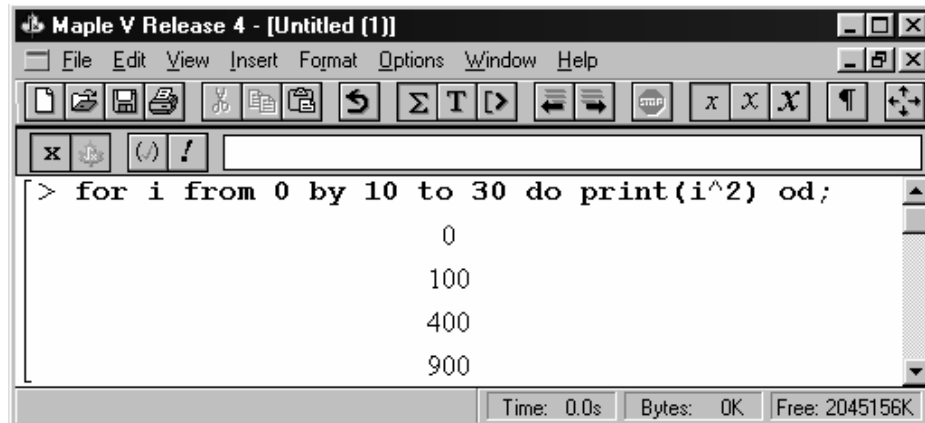
Конструкция условного перехода может быть расширена введением блока дополнительных условий *elif / then: if <условие> then <операторы> elif <условие> then <операторы> else <операторы> fi*.

Maple реализует обработку циклов при помощи различных программных конструкций.

***For <имя переменной> from <значение> by <значение>
to <значение> do <операторы> od***

При задании цикла данной конструкцией используются верхняя и нижняя границы и величина шага для переменной, меняющейся в задаваемом диапазоне от нижнего к верхнему значению. Цикл выполняет команду

(набор команд), заключенную между *do* и *od* соответствующее число раз. Переменная диапазона также может быть включена в процесс вычислений. В примере, приведенном на рис. Б.18, переменная *i* меняется от 0 до 30 с шагом 10, при этом для каждого шага выполняется команда *print(i^2)*.

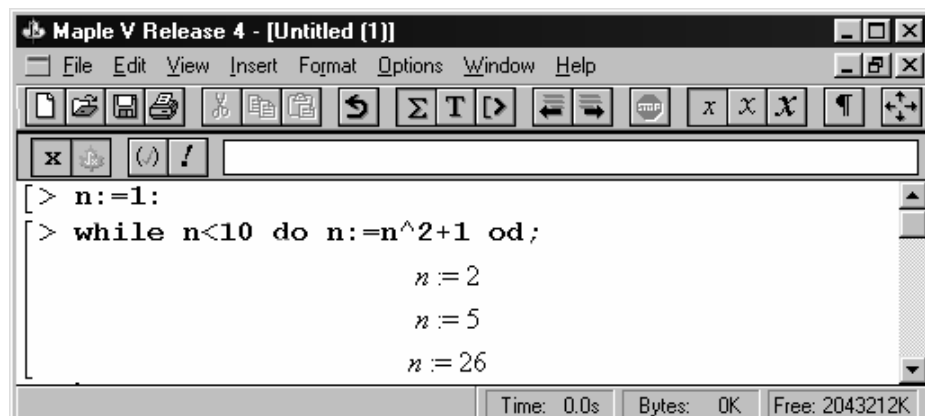


```
Maple V Release 4 - [Untitled (1)]
File Edit View Insert Format Options Window Help
[Icons]
x ( ) !
[>] for i from 0 by 10 to 30 do print(i^2) od;
      0
      100
      400
      900
Time: 0.0s Bytes: OK Free: 2045156K
```

Рисунок Б.18. Пример конструкции *For / from / by / to / do / od*

While <условие> do <операторы> od

Для такой конструкции цикла *Maple* будет повторять команду (набор команд), заключенную между *do* и *od*, пока логическое соотношение, записанное между лексемами *while* и *do*, не станет истиной (рис. Б.19).



```
Maple V Release 4 - [Untitled (1)]
File Edit View Insert Format Options Window Help
[Icons]
x ( ) !
[>] n:=1:
[>] while n<10 do n:=n^2+1 od;
      n := 2
      n := 5
      n := 26
Time: 0.0s Bytes: OK Free: 2043212K
```

Рисунок Б.19. Пример использования конструкции *While / do / od*

Б.5. Справочная система

Справочная система *Maple* удовлетворит самого взыскательного пользователя. В ней помещены подробные справки по всем функциям, языку программирования, командам управления ресурсами. Практически во все разделы включены многочисленные примеры.

Команда *Help* главного меню системы вызывает ниспадающее меню, в котором для доступа к справочной системе следует выбрать команду *Contents* (Содержание). Пользователь может получить более 2200 подробнейших справочных статей с перекрестными ссылками и “живыми примерами”, которые могут быть скопированы в рабочую среду и проиграны со своими данными.

Кроме того работа с системой соответствует всем правилам, присущим *Help* под *Windows*, т. е. имеется возможность поиска по ключевому слову, получение справки по словоформе и т. д.

ПРИЛОЖЕНИЕ В

Матричная лаборатория

MATLAB 6.5

MATLAB «Matrix Laboratory» (лаборатория матриц) – это одна из наиболее популярных разработок фирмы *The MathWorks Inc.* Пакет *MATLAB* – мощная и весьма удобная программная система, предоставляющая пользователю возможность достаточно просто и быстро решать сложные вычислительные задачи, описываемые в матричной форме. В *MATLAB* базовую роль играют специализированные группы программ – *toolboxes*, представляющие собой соответствующие коллекции функций, позволяющие решать частные задачи.

В.1. Характеристика пакета

Для работы с системой *MATLAB* необходимо предварительно изучить его язык программирования, который в отличие от обычных алгоритмических языков достаточно прост. *Язык MATLAB* является языком матриц и массивов высокого уровня с управлением потоками, функциями, структурами данных и вводом-выводом.

Среда MATLAB – это совокупность инструментов (приспособлений), состоящая из средств, необходимых для управления переменными в рабочем пространстве, ввода и вывода данных, а также средств для создания, контроля и отладки программ *MATLAB*.

Управляемая графика MATLAB – графическая система, включающая как команды высокого уровня для визуализации данных, об работки изо-

бражений, анимации и создания иллюстраций, так и команды низкого уровня, позволяющие полностью редактировать внешний вид графики.

Пакет *MATLAB* обеспечивает решение вычислительных задач на естественном матричном языке в интерактивном режиме и опирается на возможности собственного программного обеспечения. Например, можно осуществить выполнение следующих операций: вычисление собственных векторов и собственных значений; арифметические действия над матрицами, включая их обращение; решение линейных уравнений; идентификацию по методу наименьших квадратов; декомпозицию по вырожденным значениям; быстрое Фурье-преобразование; расчет цифровых фильтров; статистические расчеты.

MATLAB ориентирован на выполнение научных и инженерных расчетов на ПЭВМ. Эти расчеты могут иметь отношение как к области вычислительной математики (численное интегрирование, дифференцирование, решение оптимизационных задач, выполнение различного рода аппроксимаций, обработка исходных данных), так и к линейной алгебре и аналитической геометрии, математической статистике или к иным научно-техническим приложениям – таким как спектральный и корреляционный анализ, расчет фильтров и прочее.

Пакет *MATLAB* выполняет операции над векторами и матрицами. Одномерный массив называется *вектором*, а двумерный – матрицей. Массивы характеризуются в общем случае размерностью и размером. *Размерность* массива определяется его структурной организацией, например, в виде: одной строки или одного столбца (размерность 1); страницы (размерность 2); куба (размерность 3) и т.д. *Размер* матрицы – это число ее строк n и столбцов m .

В.2. Запуск и командный режим работы

После запуска пакета *MATLAB* (с типовой установкой по умолчанию) появляется основное окно системы, представленное на рис. В.1.

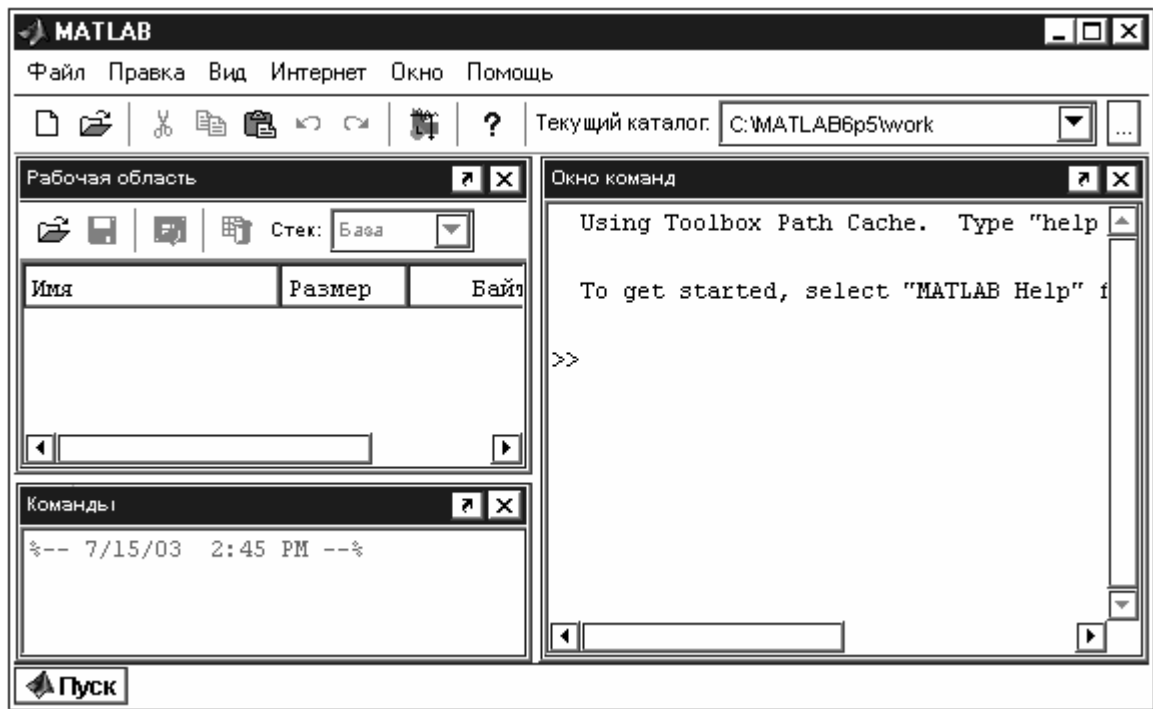


Рисунок В.1. Окно управления системы

Пакет *MATLAB* практически готов к выполнению вычислений в *командном режиме*, для работы в котором в интерактивном режиме используется простейший строчный редактор с общепринятыми командами системы *Windows*. Особое внимание следует обратить на применение клавиш \uparrow и \downarrow . Если выполнить ряд некоторых действий, то с помощью этих клавиш можно вернуть (в положение после маркера строки ввода – \gg) любую из ранее введенных строк для ее исправления или дублирования, поскольку клавиши перелистывают строки сверху вниз или снизу вверх.

Для очистки экрана и перемещения курсора в левый верхний угол окна управления в командном режиме применяются команды *clc* и *home*.

Команда *echo*, позволяет включать или выключать отображение текстов *m*-файлов при каждом обращении к ним.

Система *MATLAB* в командном режиме позволяет выполнять достаточно сложные вычисления без подготовки программы, т.е. в *интерактивном режиме прямых вычислений*. Перечислим некоторые базовые правила работы в командном режиме:

- для указания ввода исходных данных используется символ »;
- данные вводятся с помощью простейшего строчного редактора;
- для блокировки вывода результата на экран дисплея после математического выражения надо установить знак ; (точка с запятой);
- если не указана переменная со значением результата вычислений, то *MATLAB* назначает такую переменную с именем *ans*;
- знаком присваивания в *MATLAB* является знак равенства =;
- встроенные функции записываются *строчными буквами*, а их аргументы указываются в *круглых скобках* – например, *exp(2)*;
- результат вычислений выводится в строках вывода (без знака »).

В.3. Интерфейс пользователя

Как уже отмечалось, после запуска пакета *MATLAB* появляется окно управления, фрагмент которого с панелью инструментов представлен на рис. В.2а. Набор кнопок панели управления обеспечивает выполнение команд, необходимых пользователю в повседневной работе с пакетом.

Использование кнопок – Открыть файл, Вырезать, Копировать, Вставить, Отменить – не отличается от общепринятых в системе *Windows*, а кнопка Новый файл открывает окно редактора/отладчика (рис. В.2б).

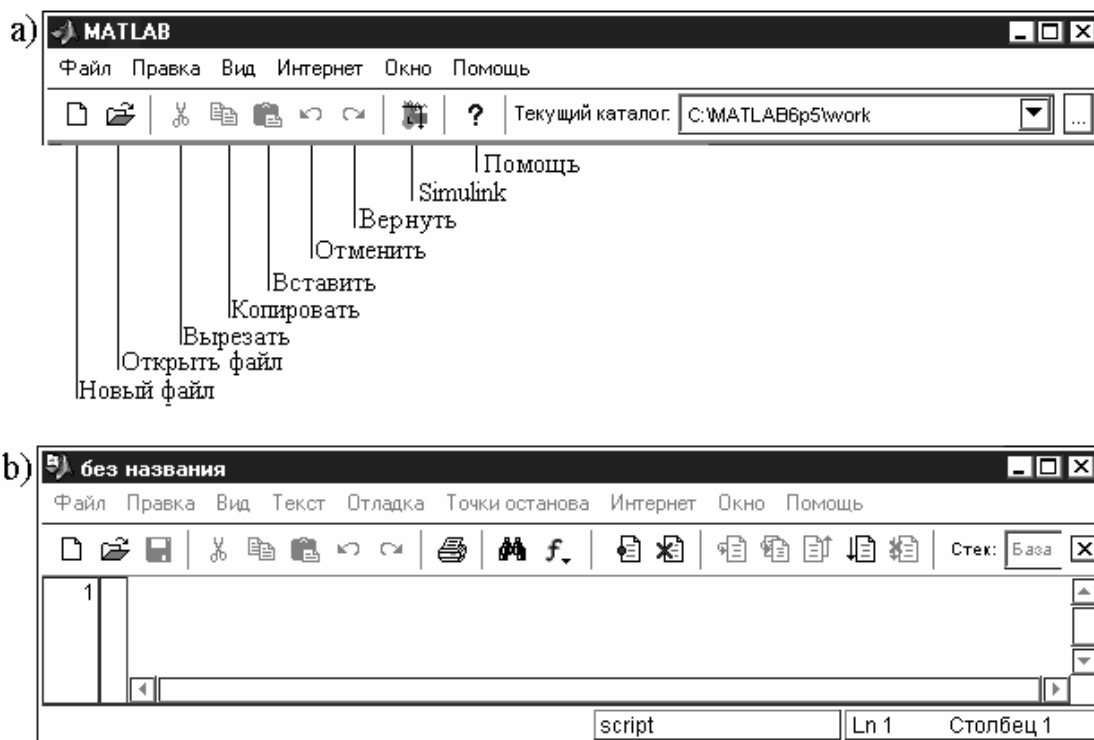


Рисунок В.2. Фрагменты окна управления и окна редактора/отладчика

По умолчанию создается *m*-файл с именем без названия, которое при сохранении файла можно изменить на любое иное. В редакторе/отладчике можно редактировать несколько *m*-файлов сразу. Каждый файл будет находиться в отдельном окне редактирования, хотя активным будет только одно, располагаемое поверх остальных. Кнопка *Simulink* вызывает окно библиотек интерактивной системы с одноименным названием, которая является сопутствующей пакету *MATLAB* программой

Редактор/отладчик *m*-файлов

В пакете *MATLAB* для создания, редактирования и отладки *m*-файлов используется специальный многооконный редактор. Все *m*-файлы делятся на два класса:

- файлы-сценарии (*Script*-файлы), не имеющие входных параметров;
- файлы-функции (*Function*-файлы), имеющие входные параметры.

Таким образом, создаваемый в окне редактора файл без внешних данных (нет списка входных параметров), – это файл-сценарий, а файл в виде функции от входных параметров, список которых указывается в круглых скобках, – это уже файл-функция.

Например, если в окне редактора функцию, вычисляющую сумму модулей элементов векторов x и y , записать в виде

```
function zz = fun(x,y)
```

```
zz = abs(x) + abs(y);
```

а затем файл с этим текстом запомнить – пусть под именем $zz.m$ (при этом окно редактора получит изображение, показанное на рис. В.3), то функция $zz(x,y)$ становится готовой к применению.

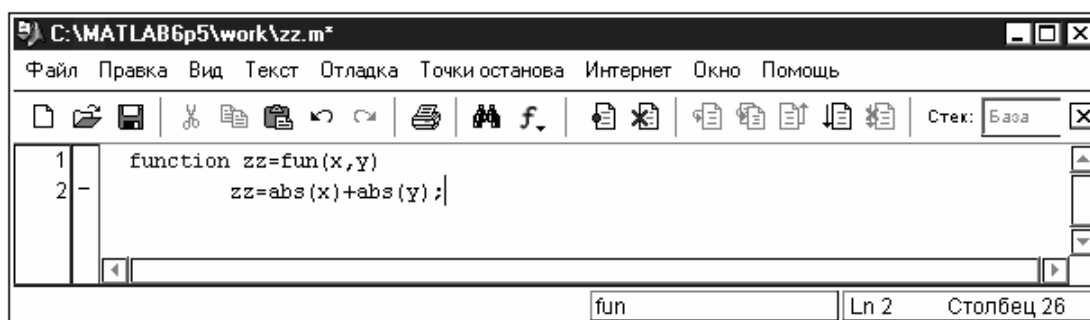


Рисунок В.3. Пример задания файла-функции

Редактор/отладчик m -файлов имеет собственную панель инструментов, ее внешний вид дан на рис. В.4, где также перечислено назначение кнопок, которые используются при отладке создаваемых программ.

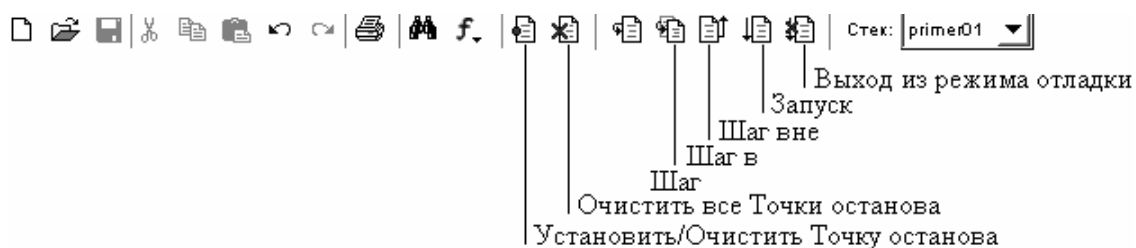


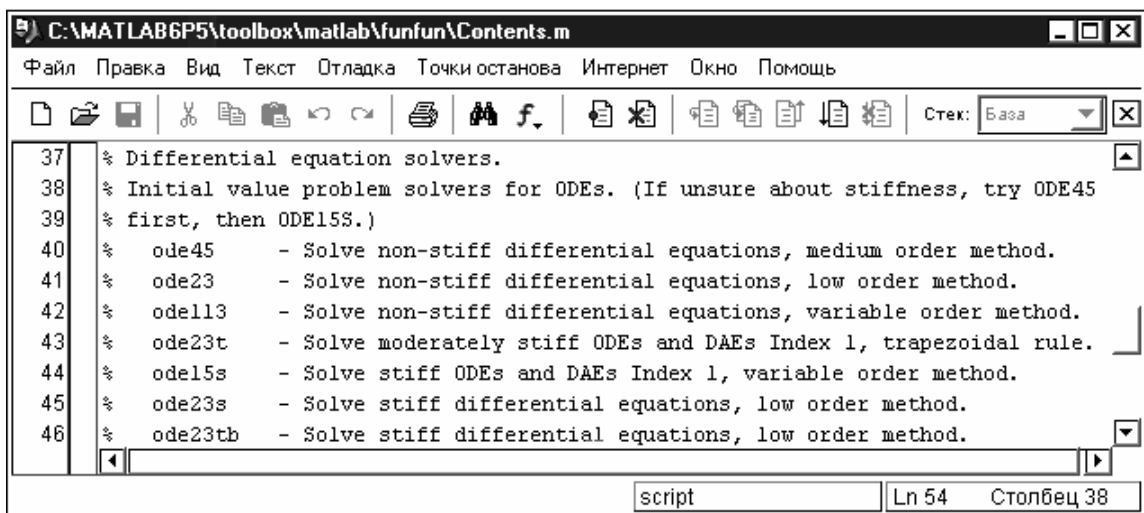
Рисунок В.4. Панель инструментов редактора/отладчика m -файлов.

Основным приемом отладки *m*-файлов является установка в их тексте контрольных точек остановки. Для этого необходимо выполнить следующую последовательность действий: во-первых, создать *m*-файл; во-вторых, написать текст создаваемой программы; в-третьих, сохранить *m*-файл на диске под оригинальным именем; в-четвертых, установить требуемые контрольные точки.

В.4. Решение дифференциальных уравнений

Решение обыкновенных дифференциальных уравнений (ОДУ) с известными начальными условиями – это решение задачи Коши. В пакете *MATLAB* имеется *специальный решатель ОДУ*, позволяющий выбрать метод решения, задать значение начальных условий и повысить эффективность используемых методов с помощью установления значений специальных опций.

Для решения систем ОДУ используются операции, функции и решающие устройства, перечень которых приведен в *m*-файле *Contents* (папка *funfun*, каталог *matlab*). Фрагмент этого файла представлен на рис. В.5.



```
37 % Differential equation solvers.
38 % Initial value problem solvers for ODEs. (If unsure about stiffness, try ODE45
39 % first, then ODE15S.)
40 % ode45 - Solve non-stiff differential equations, medium order method.
41 % ode23 - Solve non-stiff differential equations, low order method.
42 % ode113 - Solve non-stiff differential equations, variable order method.
43 % ode23t - Solve moderately stiff ODEs and DAEs Index 1, trapezoidal rule.
44 % ode15s - Solve stiff ODEs and DAEs Index 1, variable order method.
45 % ode23s - Solve stiff differential equations, low order method.
46 % ode23tb - Solve stiff differential equations, low order method.
```

Рисунок В.5. Фрагмент *m*-файла *Contents*

Решатель ОДУ использует различные численные методы (в *MATLAB* – функции), позволяющие решать уравнения в явной форме Коши $x' = F(t,x)$ и неявной, т.е. в виде $Mx' = F(t,x)$ или $M(t)x' = F(t,x)$.

Функция $[t, X] = \text{solver}('F', \text{tspan}, x0)$, где имя *solver* заменяется именем соответствующей функции; $\text{tspan} = [t0, tf]$, интегрирует ОДУ на интервале $[t0, tf]$ с начальными условиями $x0$. Функция вычисления правых частей определяется именем F . Каждая строка выходного массива X соответствует решению в моменты времени, определяемые вектором t .

Функция $[t, X] = \text{solver}('F', \text{tspan}, x0, \text{options})$ позволяет указать дополнительные опции, формируемые с помощью функции *odeset*. К часто используемым относятся опции относительной погрешности *RelTol* (по умолчанию $1e-3$) и вектора абсолютных погрешностей (по умолчанию все компоненты равны $1e-6$).

Функция $[t, X] = \text{solver}('F', \text{tspan}, x0, \text{options}, p1, p2, \dots)$ позволяет передать значения параметров $p1, p2, \dots$ при вычислении функции правых частей. Если опции используются по умолчанию, то на их месте следует указать пустой массив $[]$.

Функция $[t, X, te, xe, ie] = \text{solver}('F', \text{tspan}, x0, \text{options})$ при условии, что опция *Events* установлена в состояние 'on', позволяет зафиксировать момент, когда какая-либо из указанных переменных принимает значение 0. При этом функция вычисления правых частей должна иметь вид $F(t, x, 'events')$. Выход te – это вектор моментов времени, когда фиксировалось событие, вектор xe – перечень соответствующих переменных, а вектор ie – индексы событий, указывающие направление изменения решения в момент пересечения нуля: при (-1) – убывает; при $(+1)$ – возрастает; при (0) – только фиксируется.

Функции $[t, X, s] = \text{solver}(\dots)$ с тремя выходными аргументами позволяют получить статистические характеристики решателя. При этом вектор s имеет 6 компонент, которые отражают: количество успешных шагов; количество неудачных попыток; количество вызовов функции $F(t, x)$; количество вызовов функции Якоби dF/dx ; количество LU -разложений; количество решений вспомогательных систем линейных уравнений. Последние три параметра применяются только к решателям жестких систем.

При обращении к решателю ОДУ без указания выходных параметров $\text{solver}(\dots)$ он вызывает функцию вывода odeplot для построения графиков решения. Иной способ организации вывода графиков состоит в установлении для опции OutputFcn значения – odeplot . Значения этой опции odephas2 и odephas3 позволяют вывести двумерные или трехмерные графики фазовых траекторий.

В.5. Справочная система

Интерактивная справка реализуется в командном режиме, а именно – с помощью команды help , которая выводит весь список каталогов, содержащих m -файлы с определениями операторов, функций и иных объектов. Конкретная справка, т.е. по выбранному объекту или группе – например, просмотр каталога lang осуществляется командой help lang . Справка может быть получена и по ключевому слову, т.е. с помощью команды lookfor – Ключевое слово.

Перечислим некоторые *дополнительные справочные команды*:
 computer – выводит сообщение о типе компьютера, на котором установлена текущая версия MATLAB ;
 help script – выводит сообщение о назначении m -файлов сценариев (Script -файлов);

help function – выводит сообщение о назначении и структуре *m*-файлов функций;

what – выводит имена файлов текущего каталога;

what name – выводит имена файлов каталога с именем *name*;

which name – выводит путь доступа к функции с данным именем.

В *MATLAB* имеется мощная справочная подсистема (см. в разделе Помощь пункт – Полная Помощь по семейству продуктов), вызов которой открывает окно помощи, представленное на рис. В.6.



Рисунок В.6. Окно справочной системы – раздел Содержание

Если выбрать другой раздел справки, – например, демонстраций или демонстрационных примеров, то на экране (рис. В.7.) появится меню раздела *Getting Started with Demos* (Начало работы с демонстрационными версиями), выбор любого пункта которого раскрывает пользователю доступ к новым меню и соответствующим разделам помощи.

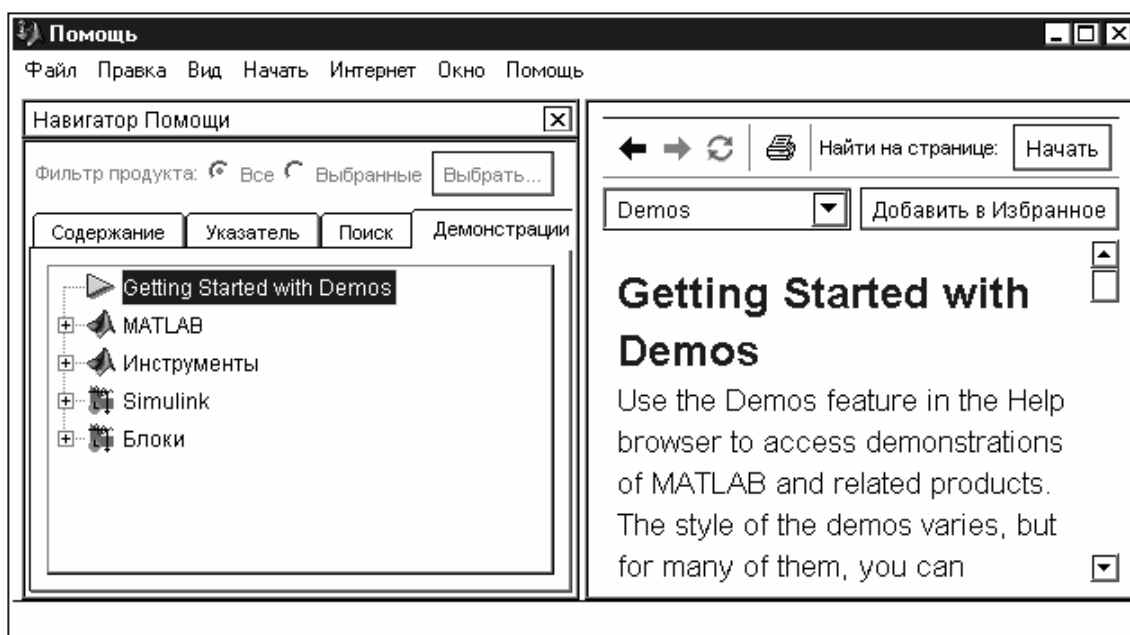


Рисунок В.7. Окно справочной системы – раздел Демонстрации

При выборе раздела справочной системы – Указатель – экранная форма получает вид, показанный на рис. В.8.

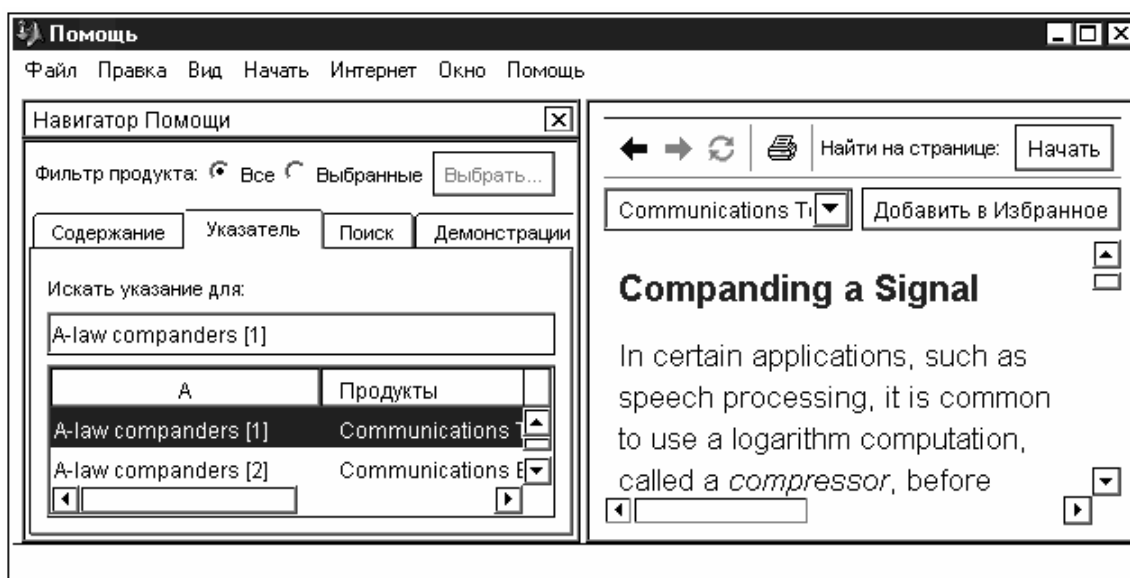


Рисунок В.8. Окно справочной системы – раздел Указатель