

А. М. Половко
П. Н. Бутусов

MATLAB

ДЛЯ
СТУДЕНТА



УДК 681.3.06(075.8)
ББК 32.973.26-018.2я73
П52

Половко А. М., Бутусов П. Н.

П52 MATLAB для студента. — СПб.: БХВ-Петербург,
2005. — 320 с.: ил.

ISBN 5-94157-595-5

Содержится описание компьютерных технологий решения математических задач с помощью системы MATLAB. Приводятся примеры на все методы, изложенные в книге. Представлены варианты задач для индивидуального обучения. Описаны методики решения задач управления и создания приложений для решения типовых задач.

*Для студентов, аспирантов, преподавателей технических вузов
и специалистов, применяющих математические вычисления
в профессиональной деятельности*

УДК 681.3.06(075.8)
ББК 32.973.26-018.2я73

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Людмила Еремеевская</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн обложки	<i>Игоря Цырульниковца</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 12.07.05.

Формат 60×90^{1/16}. Печать офсетная. Усл. печ. л. 20.

Тираж 4000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-595-5

© Половко А. М., Бутусов П. Н., 2005
© Оформление, издательство "БХВ-Петербург", 2005

Оглавление

Введение	11
Глава 1. Основы интерфейса MATLAB	15
1.1. Окна системы MATLAB	16
1.1.1. Окно <i>Command Window</i>	16
1.1.2. Окно <i>Workspace</i>	18
1.1.3. Окно <i>Current Directory</i>	19
1.1.4. Окно <i>Command History</i>	20
1.1.5. Окно <i>Launch Pad</i>	20
1.2. Главное меню системы	21
1.2.1. Меню <i>File</i>	21
1.2.2. Меню <i>Edit</i>	23
1.2.3. Меню <i>View</i>	24
1.2.4. Меню <i>Web</i>	24
1.2.5. Меню <i>Window</i>	25
1.2.6. Меню <i>Help</i>	25
1.3. Панель инструментов.....	25
Глава 2. Язык общения с MATLAB	27
2.1. Символы и операторы языка	27
2.1.1. Специальные символы.....	27
2.1.2. Операторы отношения.....	29
2.1.3. Логические операторы.....	31
2.2. Числа, переменные, функции языка	32
2.2.1. Числа в MATLAB.....	32
2.2.2. Переменные и константы	34

2.3. Функции и команды общения	35
2.3.1. Команды управления окном.....	36
2.3.2. Сообщение об ошибках и их исправление.....	36
2.3.3. Сохранение результатов вычислений.....	37
2.3.4. Завершение работы	38

Глава 3. Создание приложений для решения типовых задач..... 39

3.1. Постановка задачи.....	39
3.2. Знакомство с инструментом	41
3.2.1. Меню и панель инструментов.....	43
Меню <i>Layout</i>	43
Меню <i>Tools</i>	44
3.2.2. Панель элементов управления	44
3.3. Файлы, генерируемые системой в процессе создания приложения	45
3.4. Работа над приложением	47
3.4.1. Первый этап.....	48
3.4.2. Второй этап	51
3.4.3. Третий этап.....	54

Глава 4. Специальные вычисления 61

4.1. Табулирование функции.....	61
4.2. Вычисление суммы элементов массива чисел.....	64
4.3. Вычисление произведения элементов чисел	65
4.4. Вычисление производных	66
4.5. Вычисление пределов	69
4.6. Разложение функции в степенной ряд.....	72
4.7. Определение экстремумов функции.....	77
4.7.1. Функция $fmin('fun', x1, x2)$	80
4.7.2. Функция $fmin('fun', x1, x2, options)$	82
4.8. Интегральные преобразования.....	83
4.8.1. Преобразование Лапласа.....	83
Функция $Laplace(F)$	85
Функция $Laplace(F,s)$	87
Функция $Laplace(F,w,s)$	87
4.8.2. Решение дифференциальных уравнений с помощью преобразования Лапласа.....	88
4.8.3. Обратное преобразование Лапласа	90

Глава 5. Вычисление математических функций	93
5.1. Элементарные функции.....	93
5.1.1. Алгебраические и арифметические функции	94
5.1.2. Тригонометрические функции.....	98
5.1.3. Обратные тригонометрические функции.....	99
5.1.4. Гиперболические функции.....	100
5.1.5. Обратные гиперболические функции.....	101
5.1.6. Функции комплексного аргумента	102
5.2. Специальные математические функции.....	104
5.2.1. Гамма-функция.....	104
5.2.2. Бета-функция (Эйлеров интеграл первого рода).....	106
5.2.3. Функции ошибок.....	107
5.2.4. Интегральная показательная функция	109
5.2.5. Функции Эйри.....	109
5.2.6. Функции Лежандра	112
5.2.7. Функции Бесселя.....	113
5.3. Функции пользователя.....	114
Глава 6. Алгебра векторов и матриц	116
6.1. Создание векторов и матриц	116
6.2. Преобразование матриц.....	118
6.2.1. Вызов на экран и замена элементов матрицы	118
6.2.2. Изменение размера вектора или матрицы	119
6.2.3. Математические операции с векторами и матрицами.....	121
Определитель матрицы	121
Транспонирование матрицы	122
След матрицы.....	122
Обратная матрица.....	123
Единичная матрица.....	123
Образование матрицы с единичными элементами	124
Образование матрицы с нулевыми элементами.....	125
Вектор равноотстоящих точек.....	126
Перестановка элементов матрицы	126
Создание матрицы с заданной диагональю	127
Создание массивов со случайными элементами.....	129
Поворот матрицы.....	132
Выделение треугольных частей матрицы	133
Вычисление магического квадрата	134
6.3. Математические операции над векторами и матрицами	135
6.3.1. Примеры образования функций от векторов и матриц	138

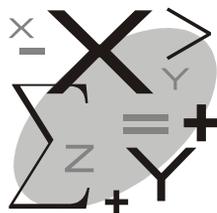
Глава 7. Визуализация вычислений.....	140
7.1. Двухмерная графика	140
7.1.1. Функция <i>plot(x,y)</i>	141
7.1.2. Функция <i>plot(x,y,s)</i>	142
7.1.3. Функция <i>plot(x1,y1,s1, x2,y2,s2, ...,xn,yn,sn)</i>	146
7.1.4. Функции построения графиков в логарифмическом масштабе	150
7.1.5. Графики в полярной системе координат.....	151
7.1.6. Создание гистограмм.....	152
7.2. Трехмерная графика.....	153
Глава 8. Алгоритмы и технологии решения уравнений	156
8.1. Алгоритмы решения алгебраических и трансцендентных уравнений.....	156
8.1.1. Метод дихотомии (половинного деления).....	156
8.1.2. Метод хорд	157
8.1.3. Метод касательных	159
8.1.4. Комбинированный метод (метод хорд и касательных)	161
8.1.5. Метод итераций.....	162
8.2. Технология решения алгебраических и трансцендентных уравнений в среде MATLAB	166
8.2.1. Технология решения уравнений с помощью функции <i>solve()</i>	166
8.2.2. Технология определения вещественных корней уравнения с помощью функции <i>fzero()</i>	169
8.2.3. Технология определения корней многочлена с помощью функции <i>roots()</i>	174
8.2.4. Варианты алгебраических и трансцендентных уравнений для индивидуальных заданий или решений.....	175
8.3. Методы решения систем алгебраических уравнений	175
8.3.1. Решение систем линейных алгебраических уравнений.....	182
Выбор начальных приближений	186
Условие сходимости итерационного процесса	186
Признак окончания вычислений	188
8.3.2. Алгоритмы метода итерации	188
8.3.3. Сравнительная оценка точных и итерационных методов.....	190
8.4. Компьютерные технологии решения систем линейных алгебраических уравнений в среде MATLAB.....	190
8.4.1. Решение системы линейных уравнений с помощью определителей	191

8.4.2. Матричный метод решения систем линейных уравнений....	192
8.4.3. Решение систем линейных уравнений с помощью функции <i>solve()</i>	194
8.5. Компьютерные технологии решения систем нелинейных уравнений.....	196
Функция <i>cgs()</i>	198
8.6. Варианты уравнений для индивидуального решения	199
Задание 1. Решение систем линейных алгебраических уравнений.....	199
Задание 2. Решение систем нелинейных алгебраических уравнений.....	204
Глава 9. Решение дифференциальных уравнений.....	205
9.1. Формулировка задачи	205
9.2. Приближенные аналитические методы решения дифференциальных уравнений	206
9.2.1. Метод последовательного дифференцирования	206
9.2.2. Метод неопределенных коэффициентов.....	208
9.2.3. Метод последовательных приближений.....	209
9.3. Численные методы решения дифференциальных уравнений	210
9.3.1. Метод Эйлера.....	210
9.3.2. Усовершенствованные методы Эйлера.....	212
Метод Эйлера — Коши	212
Усовершенствованный метод Эйлера.....	212
Усовершенствованный метод Эйлера — Коши с итерационной обработкой результатов.....	213
9.3.3. Метод Рунге — Кутты	213
9.4. Компьютерные технологии решения дифференциальных уравнений.....	216
Глава 10. Алгоритмы и технологии вычисления интегралов	220
10.1. Методы и алгоритмы вычисления интегралов	220
10.1.1. Формулы прямоугольников	221
10.1.2. Формула трапеций	221
10.1.3. Формула парабол (Симпсона).....	222
10.2. Численные методы вычисления интеграла в системе MATLAB .	223
10.2.1. Метод трапеций.....	223
Функция <i>cumtrapz(x,y)</i>	226
Функция <i>trapz(y)</i>	228
Функция <i>trapz(x,y)</i>	229

10.2.2. Численное интегрирование с помощью квадратурных формул	231
Метод парабол (Симпсона).....	231
Функция <i>quad('fun', a, b)</i>	232
Функция <i>quad('fun', a, b, tol)</i>	232
Функция <i>ablquad('fun', a, b, c, d)</i>	233
Функция <i>quad8('fun', a, b)</i>	234
10.3. Аналитические методы вычисления интеграла	236
10.3.1. Функция <i>int()</i> вычисления неопределенного и определенного интегралов	236
10.3.2. Вычисление кратных интегралов	239
10.3.3. Вычисление несобственных интегралов	240
10.4. Примеры вычисления интегралов.....	241
Глава 11. Методы и компьютерные технологии интерполяции.....	248
11.1. Элементы теории.....	248
11.1.1. Выбор вида функции интерполяции.....	250
Графоаналитический способ.....	250
Способ линеаризации нелинейных функций.....	254
Анализ табличных разностей.....	255
Использование специальных программ автоматизации интерполяции	256
11.1.2. Определение коэффициентов функции интерполяции	257
11.1.3. Проверка адекватности модели	257
11.2. Интерполяция точная в узлах. Универсальный метод.....	258
11.2.1. Интерполяция линейными функциями.....	258
11.2.2. Интерполяция нелинейными функциями.....	262
11.2.3. Сплайн-интерполяция	264
11.2.4. Интерполяция точная в узлах.....	265
11.3. Интерполяция, приближенная в узлах (аппроксимация).....	267
11.3.1. Функция <i>lsqcurvefit()</i>	267
11.3.2. Полиномиальная аппроксимация	270
11.3.3. Интерполяция кубическими полиномами	272
Глава 12. Компьютерные технологии решения задач управления	274
12.1. Задачи управления	274
12.2. Функции MATLAB для создания передаточных функций звеньев системы	275
12.2.1. Функция <i>tf()</i>	275
12.2.2. Функции <i>pole()</i> и <i>zero()</i>	277

12.2.3. Функции <i>roots()</i> и <i>poly()</i>	279
12.2.4. Функция <i>conv()</i>	280
12.2.5. Функция <i>polyval()</i>	281
12.3. Операции с передаточными функциями звеньев.....	283
12.3.1. Сложение передаточных функций.....	283
12.3.2. Функция <i>pzmap()</i>	284
12.3.3. Функция <i>series()</i>	286
12.3.4. Функция <i>parallel()</i>	287
12.3.5. Функция <i>feedback()</i>	288
12.3.6. Функция <i>minreal()</i>	292
12.4. Исследование переходных процессов в системах управления.....	293
12.4.1. Функция <i>step()</i>	295
12.5. Частотные характеристики системы.....	296
12.5.1. Амплитудно-фазовая характеристика системы.....	299
12.5.2. Диаграмма Никольса	300
12.6. Пример анализа динамики системы управления.....	302
12.6.1. Образование передаточной функции разомкнутой системы	303
12.6.2. Определение нулей и полюсов передаточной функции $G(S)$	303
12.6.3. Расположение нулей и полюсов на комплексной плоскости.....	304
12.6.4. Анализ устойчивости системы.....	304
12.6.5. Исследование качества переходного.....	304
12.6.6. Получение передаточной функции замкнутой системы ...	306
12.6.7. Определение нулей и полюсов передаточной функции замкнутой системы и расположение их на комплексной плоскости.....	306
12.6.8. Переходные процессы замкнутой системы с жесткой отрицательной обратной связью.....	308
12.6.9. Исследование устойчивости и качества переходных процессов системы управления при гибкой отрицательной обратной связи	308
12.7. Индивидуальные задания для исследования динамики систем управления	309
12.7.1. Задание 1.....	309
12.7.2. Задание 2.....	314
Постановка задачи.....	314
Варианты индивидуальных заданий и передаточных функций	315
Литература	318

ГЛАВА 2



Язык общения с MATLAB

2.1. Символы и операторы языка

Операторы языка — это символы операций над данными, называемыми *операндами*. В MATLAB применяются все общепринятые операнды. Однако некоторые из них имеют ряд особенностей. Следует всегда иметь в виду, что большинство операторов языка MATLAB относится к матричным операциям. Например, операторы $*$ и $/$ вычисляют произведение и частное от деления двух массивов векторов и матриц. Если же необходимо вычислить почленное умножение или деление массивов, то следует применять операторы $.*$ и $./$. В MATLAB также используется оператор деления справа налево (\backslash или $\. \backslash$).

Подробно эти и другие, часто используемые операторы языка с большим числом примеров, рассмотрены в *главе 6*.

Полный список операторов можно получить по команде `help ops`.

2.1.1. Специальные символы

Специальными являются следующие символы языка MATLAB:

- ◆ () — круглые скобки;
- ◆ [] — квадратные скобки;
- ◆ { } — фигурные скобки;
- ◆ . — десятичная точка;

- ◆ ; — точка с запятой;
- ◆ : — двоеточие;
- ◆ , — разделитель (запятая);
- ◆ .. — родительский каталог;
- ◆ ... — продолжение строки;
- ◆ % — комментарий;
- ◆ ! — вызов команды операционной системы;
- ◆ = — присвоение;
- ◆ ' — кавычка.

Рассмотрим назначение специальных символов.

- ◆ : — оператор образования массива данных переменной; формирует из векторов и матриц подвекторы и подматрицы. Представляется в следующих формах:
 - $i:k$ — аналог вектора $[i, i+1, i+2, \dots, k]$, например, $1:5$ — $[1\ 2\ 3\ 4\ 5]$;
 - $i:j:k$ — аналог оператора $i:k$, существует при $j > 0, k > i$ или при $j < 0, i > k$;
 - $M(:, i)$ — выбирается i -й столбец из матрицы M ;
 - $M(i, :)$ — выбирается i -я строка из матрицы M ;
 - $M(:, :)$ — аналогичен $M(:)$;
 - $M(i:k)$ — аналогичен $M(i), M(i+1), M(i+2), \dots, M(k)$;
 - $M(:, i, k)$ — аналогичен $M(:, i), M(:, i+1), M(:, i+2), \dots, M(:, k)$;
 - $M(:)$ — представление массива M в виде столбца;
 - $M(:, :, k)$ — k -я строка трехмерного массива M .
- ◆ Оператор круглые скобки $()$ используется для задания аргументов функции, порядка выполнения операций в математических выражениях, указания индексов элементов вектора или матрицы. Например: $\sin(x)$, $(x-1)/(x+1)$, $x(v)$, $x(1)$, $M(A, B)$, $M(:, i)$.

- ◆ Оператор квадратные скобки [] формирует векторы и матрицы, например: [1 2 3 4], [1,2,3; 3 5 2].
- ◆ Оператор фигурные скобки { } применяется для формирования массивов ячеек.
- ◆ Десятичная точка (.) служит для отделения целой части числа от дробной. Кроме того, она применяется как знак указания операций над элементами символьных переменных. Примеры: 3.2, .15, 2.*log(x)+x.^2-x./cos(x).
- ◆ Точка с запятой (;) применяется в конце операторов для подавления вывода информации на экран, а также внутри круглых скобок для отделения строк матрицы.
- ◆ Запятая (,) используется для разделения элементов вектора и матрицы.
- ◆ Знак процента (%) воспринимается программой как начало комментария.
- ◆ Символ равенства (=) является знаком присваивания имени математическому выражению: x=[1 2 3 4 5], x=cos(a), x=2.5.
- ◆ Одиночная кавычка (') применяется для указания того, что математическое выражение содержит символьные переменные, например: Y='x+exp(-a)+1=0'.
- ◆ Знак восклицания (!) указывает, что за ним следует команда операционной системы.
- ◆ Знак из двух точек (..) определяет, что осуществляется переход на один уровень вверх по дереву каталогов (родительский каталог).
- ◆ Знак три точки (и более) (...) — продолжение строки. Его используют при переносе текста на другую строку.

2.1.2. Операторы отношения

Операторы отношения служат для сравнения двух операндов. Если операнды одинаковы, то программа возвращает 1 (True), в противном случае — 0 (False).

Правила записи операндов приведены в табл. 2.1.

Таблица 2.1. Операторы отношения

Функция	Имя оператора	Обозначение	Примеры
eq	Равно	==	a=b
ne	Не равно	~=	a~b
lt	Меньше	<	x<y
gt	Больше	>	x>y
le	Меньше или равно	<=	x<=y
ge	Больше или равно	>=	x>=y

Операторы = и ~= сравнивают действительные и комплексные переменные. При этом сравниваются действительные и комплексные части числа.

Операторы <, <=, >, >= при сравнении комплексных чисел сравнивают только действительные части числа.

Примеры представлены в табл. 2.2.

Таблица 2.2. Примеры использования операторов отношения

Выражение	Функция	Результат
>> 5==5	>> eq(5, 5)	ans = 1
>> 3~=3	>> ne(3, 3)	ans = 0
>> 2+3i==2+i	>> eq(2+3i, 2+i)	ans = 0
>> 2+3i==2+3i	>> eq(2+3i, 2+3i)	ans = 1
>> 2+3i~=2+3i	>> ne(2+3i, 2+3i)	ans = 0

Таблица 2.2 (окончание)

Выражение	Функция	Результат
>> 3.2<3.21	>> lt(3.2<3.21)	ans = 1
>> 2.3+8i<2.4+i	>> lt(2.3+8i, 2.4+i)	ans = 1
>> 3.8-3i>5+i	>> gt(3.8-3i, 5+i)	ans = 0
>> 3<2.999	>> le(3, 2.999)	ans = 0
>> 3>=2.999	>> ge(3, 2.999)	ans = 1

2.1.3. Логические операторы

Логические операторы предназначены для реализации логических операций: дизъюнкции (ИЛИ), конъюнкции (И), инверсии (НЕ), исключающего ИЛИ.

Правила записи операторов приведены в табл. 2.3.

Таблица 2.3. Логические функции и операторы

Функция	Имя	Оператор
and	И	&
or	ИЛИ	
not	НЕ	~
xor	ИЛИ НЕ	-

Пример 2.1

```
>> 3 | 'x'
ans =
    1
```

```

>> 'x' | ~'x'
ans =
     1

>> 'x' & ~'x'
ans =
     0

>> x1=[1,2,a,4];
>> x2=[1,0,0,1];
>> and(x1,x2)           или           >> x1&x2
ans =
     1     0     0     1

>> or(x1,x2)           или           >> x1|x2
ans =
     1     1     1     1

>> not(x1)             или           >> ~x1
ans =
     0     0     0     0

>> not(x2)             или           >> ~x2
ans =
     0     1     1     0

>> xor(x1,x2)
ans =
     0     1     1     0

```

Из примеров видно, что программа числа и символы, отличные от нуля, воспринимает как 1.

2.2. Числа, переменные, функции языка

2.2.1. Числа в MATLAB

Числа в MATLAB могут быть положительными и отрицательными, целыми и дробными, действительными и комплексными. Они могут представляться с фиксированной и плавающей точкой, с мантиссой и порядком (в научной форме).

Особенности представления чисел в MATLAB:

- ◆ мнимая единица кодируется с помощью двух символов: i или j ;
- ◆ целая часть числа от дробной отделяется точкой;
- ◆ отделение порядка числа от мантииссы осуществляется символом e ;
- ◆ знак $+$ положительного числа не ставится, знаки $+$ и $-$ положительного и отрицательного чисел называются *унарными*.

Формат чисел определяет их вид на экране монитора. А все вычисления в MATLAB осуществляются в формате двойной точности. Ввод чисел выполняется в любом формате по желанию пользователя.

Устанавливается формат чисел с помощью следующих команд:

- ◆ `format short` — короткое представление (5 знаков числа);
- ◆ `format short e` — короткое представление в экспоненциальной форме (5 знаков мантииссы, 3 знака порядка);
- ◆ `format long` — длинное представление числа (15 знаков);
- ◆ `format long e` — длинное представление в экспоненциальной форме (15 знаков мантииссы, 3 знака порядка);
- ◆ `format hex` — шестнадцатеричный формат;
- ◆ `format bank` — представление в денежном выражении (2 знака после точки).

Пример 2.2

Необходимо представить число $5/7$ во всех форматах.

Решение:

- ◆ `format short` — 0.7143;
- ◆ `format short e` — 7.1429e-001;
- ◆ `format long` — 0.71428571428571;
- ◆ `format long e` — 7.142857342857143e-001;
- ◆ `format bank` — 0.71.

2.2.2. Переменные и константы

Переменные — это символы, используемые для обозначения некоторых хранимых данных. Переменная имеет имя, называемое *идентификатором*. Данные могут менять свои значения, а идентификатор остается прежним. В MATLAB количество символов идентификатора ограничено и равно 31.

Имя переменной начинается с буквы и может состоять из букв, цифр и некоторых (допустимых) символов.

В процессе решения задач переменные могут занимать много памяти компьютера. Для очистки памяти от переменных в MATLAB используется функция `clear`, которая имеет синтаксис:

- ◆ `clear` — уничтожение всех переменных;
- ◆ `clear x` — уничтожение переменной `x`;
- ◆ `clear a b c` — уничтожение семейства переменных.

Константа — это численное значение уникального имени, имеющего математический смысл. Наиболее часто в MATLAB используются следующие константы:

- ◆ `pi` — число π ;
- ◆ `inf` — машинная бесконечность;
- ◆ `ans` — имя переменной, хранящей результат вычисления и вывешиваемой на экране в следующем виде: `ans=`;
- ◆ `NaN` — нечисловой характер данных (Not a Number);
- ◆ `eps` — погрешность операций с числами с плавающей точкой (2^{-52});
- ◆ `realmin` — минимальное число с плавающей точкой (2^{-1022});
- ◆ `realmax` — максимальное число с плавающей точкой (2^{1022}).

Пример 2.3

```
>> pi/2
ans =
    1.5708
```

```
>> sin(1.2)/sin(0)
ans =
      inf
>> (2-2)/(1-1)
ans =
      NaN
>> eps
ans =
  2.2204e-016
>> realmin
ans =
  2.2251e-308
>> realmax
ans =
  1.7977e308
```

Математическое выражение, взятое в одиночные кавычки, не вычисляется. Оно воспринимается как сочетание символов. Такое выражение может быть преобразовано в числовое. Об этом мы узнаем в дальнейшем при решении задач.

2.3. Функции и команды общения

Понятие "функция" весьма обширно. В математике — это любое выражение, у которого имеется имя, например $y = 2x + \sin(x) - 1$. Математические функции могут быть *элементарными*, *специальными* и *функциями пользователя*. В компьютерной алгебре функцией часто называют процедуру решения стандартной задачи, например, `solve` — имя функции решения уравнений. Математические функции достаточно подробно описаны в *главе 5*. Функции компьютерной алгебры описываются в процессе решения математических задач. В следующих разделах будут представлены функции общения с компьютером, независимо от того, какая решается задача. Такими функциями являются команды управления окнами, редактирования, сохранения результатов решения задачи, завершения работы.

2.3.1. Команды управления окном

Для очистки экрана набирается команда и нажимается клавиша <Enter>. Такими командами в MATLAB являются:

- ◆ `clc` — очищает окно, оставляя лишь знак приглашения `>>` в верхнем левом углу экрана;
- ◆ `home` — аналогична команде `clc`;
- ◆ `echo on all` — вывод на экран текста m-файлов;
- ◆ `echo off all` — отключение вывода на экран текста m-файлов;
- ◆ `echo <имя_файла> on` — вывод на экран файла сценария;
- ◆ `echo <имя_файла> of` — выключение режима вывода на экран файла сценария;
- ◆ `echo <имя_файла>` — смена режима вывода на противоположный;
- ◆ `more on` — режим постраничного вывода;
- ◆ `more off` — отключение режима постраничного вывода.

2.3.2. Сообщение об ошибках и их исправление

При наличии ошибок в выражениях или командах MATLAB не только не выдает решение, но и указывает на наличие ошибки. Из текста иногда можно понять сущность ошибки, но чаще комментарии бывают настолько общими, что трудно установить место и содержание ошибки.

В MATLAB используются два вида информации об ошибке: предупреждение и сообщение о ней.

При предупреждении процесс вычислений не прекращается, но на экран выдается текст, предупреждающий о том, что ответ может быть ошибочным.

При сообщении об ошибке вычисления прекращаются.

В случае неопределенности результатов вычисления появляется сообщение `NaN`, что означает неопределенность, например, вида `0/0` или `∞/∞`.

При делении числа на ноль появляется сообщение "Warning: Divide by Zero".

Следует при этом иметь в виду, что машинный ноль и бесконечность имеют значения 10^{-308} и 10^{308} соответственно.

Напомним, что устранение ошибки наиболее целесообразно не путем набора нового правильного выражения, а редактированием ошибочного.

В MATLAB применяется оригинальный способ редактирования посредством повторного исполнения команд. С помощью клавиш $\langle \uparrow \rangle$ и $\langle \downarrow \rangle$ можно, перелистывая строки ранее введенных выражений или команд, найти необходимое выражение и либо отредактировать его, либо выполнить повторно. Например, пусть необходимо вычислить выражение $xe^{-x} + x^2 - 1$ при значении x от 0 до 2 с шагом 0.2.

Программа будет иметь вид:

```
>> x=0:0.2:2;  
>> y=x*exp(-x)+x^2-1
```

Пусть теперь необходимо вычислить при тех же значениях аргумента следующую функцию: $xe^x - x^2 + 1$.

Для решения задачи достаточно, нажимая многократно клавишу $\langle \uparrow \rangle$, вызвать исходное выражение и отредактировать его. После нажатия клавиши $\langle \text{Enter} \rangle$ получим ответ.

Клавиши $\langle \rightarrow \rangle$ и $\langle \leftarrow \rangle$ перемещают курсор вправо и влево на один символ соответственно. Комбинация клавиш $\langle \text{Ctrl} \rangle + \langle \rightarrow \rangle$, $\langle \text{Ctrl} \rangle + \langle \leftarrow \rangle$ перемещает курсор на одно слово вправо и влево соответственно.

2.3.3. Сохранение результатов вычислений

В рабочей области памяти хранятся результаты решения задачи. Их сохранение осуществляется с помощью функции `save`, которая имеет вид:

- ◆ `save fname` — сохранение рабочей области всех переменных в файле с именем `fname.mat`;

- ◆ `save fname x` — сохранение переменной `x`;
- ◆ `save fname x, y, z` — сохранение переменных `x, y, z`.

С помощью функции `save` нельзя сохранить весь текст, который находится на экране после решения задачи. Запрет объясняется просто. При решении задачи экран, как правило, загружен ошибочными функциями, неверными данными, сообщениями об ошибках и т. п. Все это хранить нет надобности, поэтому функция этого и не делает.

Для того чтобы сохранить необходимые строки решения задачи, пользователь должен убрать с экрана все ненужное. Для этого он должен воспользоваться редактором и отладчиком и получить текст без синтаксических и прочих ошибок. Такой текст сохраняется в виде файла с расширением `m`.

Впрочем, сохранить неотредактированный текст в полном объеме тоже можно. Для этого необходимо воспользоваться командой `diary`.

Сохраненный с помощью функции `save` текст можно вызвать для продолжения решения задачи. Для этого служит функция `load`.

2.3.4. Завершение работы

Завершение работы с системой MATLAB осуществляется с помощью команд `quit`, `exit` или нажатием комбинации клавиш `<Ctrl>+<0>`.

Все, что написано в этой главе, заучивать не нужно. Процедуры, команды и функции, приведенные здесь, запоминаются, а главное понимаются, только в процессе решения задачи. К этой главе читатель сможет обращаться в процессе решения задачи, как к справочному материалу.