

А. М. Половко  
П. Н. Бутусов

# MATLAB

ДЛЯ  
СТУДЕНТА



**А. М. Половко  
П. Н. Бутусов**

# **МАТЛАВ**

**ДЛЯ СТУДЕНТА**

Санкт-Петербург

«БХВ-Петербург»

2005

УДК 681.3.06(075.8)  
ББК 32.973.26-018.2я73  
П52

**Половко А. М., Бутусов П. Н.**

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

ISBN 5-94157-595-5

Содержится описание компьютерных технологий решения математических задач с помощью системы МАТЛАВ. 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 Цырульниковой</i>
Зав. производством	<i>Николай Тверских</i>

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

Формат 60×90<sup>1</sup>/<sub>16</sub>. 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

<b>Введение</b> .....	<b>11</b>
<b>Глава 1. Основы интерфейса MATLAB</b> .....	<b>15</b>
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
<b>Глава 2. Язык общения с MATLAB</b> .....	<b>27</b>
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 <i>fmin</i> ('fun', $x1$ , $x2$ ) .....	80
4.7.2. Функция <i>fmin</i> ('fun', $x1$ , $x2$ , <i>options</i> ) .....	82
4.8. Интегральные преобразования.....	83
4.8.1. Преобразование Лапласа.....	83
Функция <i>Laplace(F)</i> .....	85
Функция <i>Laplace(F,s)</i> .....	87
Функция <i>Laplace(F,w,s)</i> .....	87
4.8.2. Решение дифференциальных уравнений с помощью преобразования Лапласа.....	88
4.8.3. Обратное преобразование Лапласа .....	90

<b>Глава 5. Вычисление математических функций .....</b>	<b>93</b>
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
<b>Глава 6. Алгебра векторов и матриц .....</b>	<b>116</b>
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

<b>Глава 7. Визуализация вычислений.....</b>	<b>140</b>
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
<b>Глава 8. Алгоритмы и технологии решения уравнений .....</b>	<b>156</b>
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
Условия 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

# Введение

Компьютерная алгебра — новое научное направление в информатике. Его появление тесно связано с созданием универсальных математических программных средств символьной математики, таких как Mathematica, Maple, Derive, Mathcad, MATLAB и др.

Каждая из этих систем является уникальной. В ней имеется свой язык общения, наборы математических функций, алгоритмы и методы решения математических задач.

Уникальность системы MATLAB определяется следующими ее особенностями:

- ◆ система ориентирована на матричные операции;
- ◆ наличие большого числа библиотечных функций, делающих ее одновременно специализированной математической системой, предназначенной для решения ряда научных и инженерных задач (анализ и синтез систем управления, теория нечетких множеств, планирование эксперимента и многих других задач);
- ◆ возможность диалога с другими математическими системами (Maple, Mathcad, MS Excel) расширяет возможности MATLAB, ликвидирует один из ее недостатков — слабую, по сравнению с другими системами, символьную математику.

В результате этих особенностей MATLAB — одна из наиболее мощных математических систем, пользующаяся большой популярностью пользователей.

За рубежом изданы десятки книг по системе MATLAB. В нашей стране их очень мало. При этом большинство из них недоступны широкому читателю, т. к. изданы малыми тиражами в основном внутривузовскими издательствами для своих внутренних потребностей.

Настоящая книга предназначена для широкого круга пользователей. Ее особенностями являются:

- ◆ подробное изложение компьютерных технологий решения математических задач, а не только перечисление функций MATLAB, предназначенных для решения математических задач;
- ◆ большое количество примеров: практически на каждый из математических методов;
- ◆ краткое изложение сущности математических методов решения задач в среде MATLAB;
- ◆ отсутствие в книге сведений о системе MATLAB, не имеющих прямого отношения к решению математических задач (история создания, инсталляция системы, специальная графика и т. п.);
- ◆ краткость и одновременно ясность и достаточная полнота изложения компьютерных технологий решения математических задач;
- ◆ ориентация на студента технического вуза.

Книгу целесообразно использовать при проведении упражнений на персональном компьютере, например, по вычислительной математике, информатике и многим другим предметам.

При проведении лабораторных работ по общетехническим и специальным дисциплинам, требующим элементов научных исследований, без универсальных программных систем не обойтись. Здесь будет полезна система MATLAB.

Систему MATLAB целесообразно использовать при обработке результатов лабораторных работ по любому из предметов, где они проводятся.

Книга просто необходима студенту при курсовом и дипломном проектировании. Она должна занять достойное место в библио-

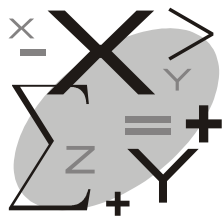
теке студента уже с первого курса, где он проходит обучение по математике и информатике.

Не следует только думать, что если книга адресована студенту, то она ограничена по содержанию. Это далеко не так. Ее специфика лишь в методике изложения компьютерных технологий решения математических задач: большое число примеров, индивидуальных заданий, наличие сведений из теории решения математических задач, простота и краткость изложения.

По полноте излагаемых методов и особенно компьютерных технологий решения математических задач в среде MATLAB она не уступает другим, например, приведенным в списке литературы в конце книги.

Книга будет полезна преподавателям, аспирантам и соискателям, научным работникам и всем, кому по роду деятельности придется решать математические задачи.

# ГЛАВА 1



## Основы интерфейса MATLAB

Общение пользователя с системой MATLAB, как и с любой другой математической системой, происходит с помощью клавиатуры и мыши. Набирая соответствующие символы, являющиеся кодами операций, и щелкая мышью по кнопкам, пользователь вводит данные, обращается к функциям и командам, получает решение.

Обширный пользовательский интерфейс включает множество операций и требует знания (вернее запоминания) большого количества функций, команд и кнопок. К ним, в частности, относятся:

- ◆ главное меню системы;
- ◆ кнопки панели инструментов;
- ◆ окна системы;
- ◆ редактор файлов;
- ◆ общение с приложениями;
- ◆ графическая система;
- ◆ справочная система

и многое другое.

Все это можно изучить и активно им пользоваться только в процессе решения задачи. Изучать элементы интерфейса в полном объеме в начале работы с системой смысла не имеет.

В данной главе в конспективной форме описываются только изначально необходимые окна системы, главное меню и панель инструментов.

## 1.1. Окна системы MATLAB

Общение пользователя с системой происходит посредством пяти окон. Рассмотрим каждое из них.

### 1.1.1. Окно *Command Window*



Окно **Command Window** (Окно команд) является для пользователя наиболее важным. Посредством этого окна (рис. 1.1) вводятся математические выражения, получаются результаты вычислений, а также выдаются сообщения, посылаемые системой. Данное окно становится доступным пользователю сразу же после запуска программы. Математические выражения пишутся в командной строке после знака приглашения `>>`.

Наберем в строке ввода выражение

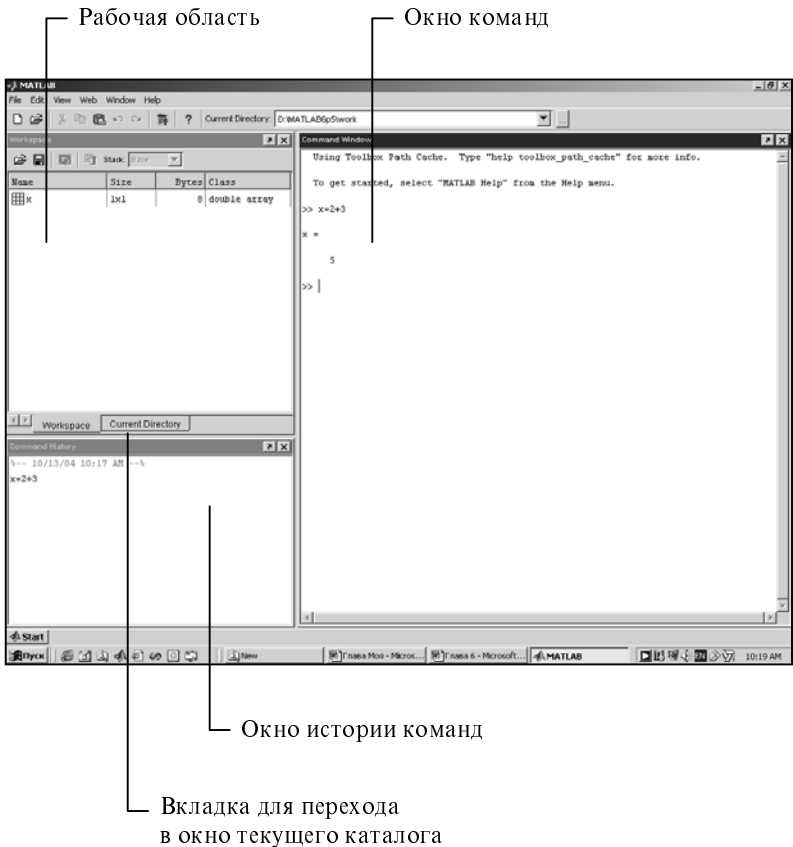
```
>> x=2+3
```

Для выполнения действия нажмем клавишу `<Enter>`. Результат виден на рис. 1.1.

Если мы захотим исправить одну из цифр, то у нас ничего не получится. Невозможность редактирования ранее введенной команды простой установкой курсора в нужную строку является одной из особенностей системы MATLAB. Для того чтобы отредактировать ранее введенную команду, необходимо установить курсор в строку ввода и воспользоваться клавишами `<↑>` и `<↓>`. Эти клавиши позволяют пролистать стек введенных ранее команд и оставить в строке ту команду, которая необходима. Команду можно выполнить сразу (нажав клавишу `<Enter>`) или после редактирования.

Окно команд имеет еще два управляющих элемента, в правом верхнем углу. Это кнопка закрытия окна , а расположенная левее — кнопка отделения окна от интерфейса системы . По-

сле нажатия на эту кнопку окно становится автономным, имеющим собственное меню. Возврат окна в общий интерфейс происходит посредством выбора из его главного меню пунктов **View | Dock Command Window** (Вид | Пристыковать окно команд). Заметим, что такими же кнопками снабжены оставшиеся четыре окна, поэтому о них (кнопках) в дальнейшем мы упоминать не будем.



**Рис. 1.1.** Главное окно системы

Рассмотренное окно является основным и, в принципе, при работе с системой MATLAB можно было бы обойтись без других.



Однако наличие остальных окон делают диалог с компьютером легким и комфортным.

## 1.1.2. Окно *Workspace*

В процессе работы используются переменные различных типов. Созданные переменные хранятся в специально отведенной области памяти компьютера. Они не исчезают сами по себе, а только при выходе из программы или с помощью специальных команд.

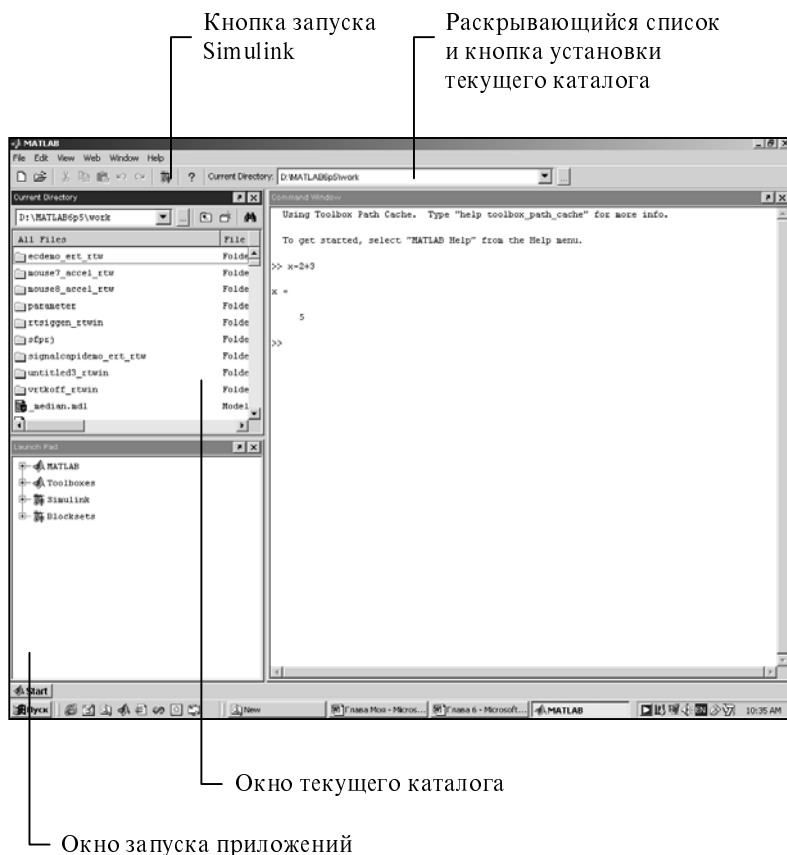


Рис. 1.2. Окно *Workspace*

При этом переменные (точнее их значения) можно использовать в любом вводимом нами математическом выражении. Окно **Workspace** (Рабочая область) предоставляет пользователю список всех переменных, хранящихся в рабочем пространстве (рис. 1.2). Выбрать можно любую переменную, просмотреть ее содержимое или выполнить какие-либо другие действия.

Упомянутые действия выполняются посредством контекстного меню (нужно щелкнуть правой кнопкой мыши по имени переменной в списке).

### 1.1.3. Окно *Current Directory*

Окно **Current Directory** (Текущий каталог) является аналогом известной программы Проводник, но имеет для MATLAB свое особое предназначение (рис. 1.3).

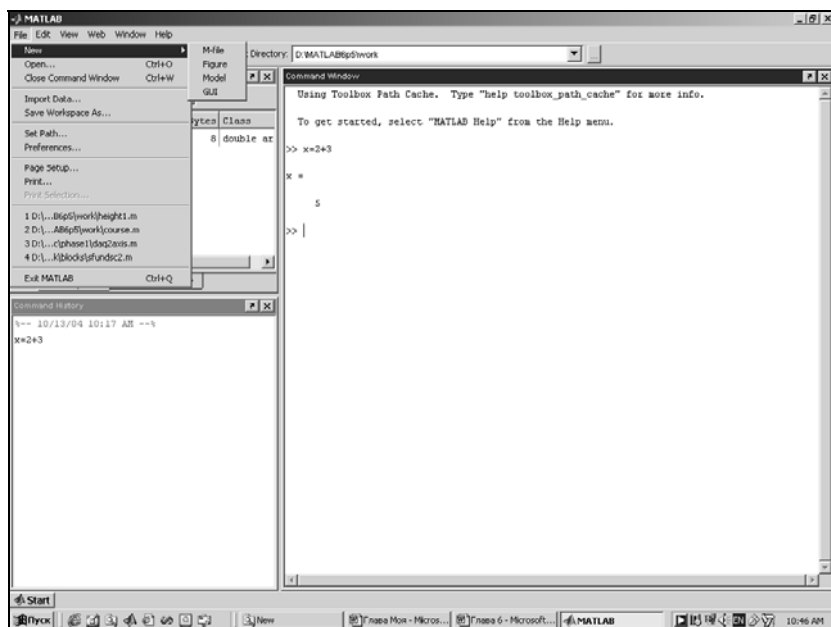


Рис. 1.3. Окно *Current Directory*

Дело в том, что, кроме работы с математическими выражениями из командного окна, пользователь также может работать с файлами (об этом речь пойдет далее). К тому же математические функции, которые мы используем, физически представляют собой файлы, названные по именам функций. В этих файлах записаны программы, реализующие функции. Таким образом, пользователь постоянно использует файлы. Например, указывая встроенную функцию, мы фактически пишем имя файла (без расширения), в котором хранится текст программы. И где в таком случае система должна искать требуемый файл? Она будет искать его в текущем каталоге или в пути доступа.

### 1.1.4. Окно *Command History*

Все команды, которые набираются в командной строке **Command Window** (Окно команд), автоматически образуют список, который и выводится в окне **Command History** (История команд). Чем полезен этот список? Если появилась необходимость повторить ранее выполненную команду, то она отыскивается в списке **Command History** (История команд), и, дважды щелкнув по ней левой кнопкой мыши, можно команду выполнить.

Можно выполнить нужную последовательность команд из командной строки и получить соответствующую последовательность команд в **Command History** (История команд). Несколькими простыми действиями полученная последовательность может быть преобразована в программу. Содержимое данного окна не теряется после выхода из системы и выключения компьютера. Удалить список команд можно только с помощью меню.

### 1.1.5. Окно *Launch Pad*

Окно **Launch Pad** (Панель запуска) содержит дерево файловой системы, где отображены только установленные на компьютере программные продукты, входящие в систему MATLAB. С помощью этого окна можно запустить любой из них. Те же действия легко осуществить посредством кнопки **Start** (Пуск), расположенной в левом нижнем углу окна MATLAB.

## 1.2. Главное меню системы

Посредством меню выполняются наиболее общие действия системы MATLAB. Меню имеет стандартный вид и организацию, присущие многим программным продуктам. Умение работать с меню может существенно облегчить диалог пользователя с компьютером. В этом разделе мы рассмотрим более подробно только те пункты меню, которые необходимы для решения поставленных в книге задач. При этом нужно иметь в виду, что глубоко изучить и активно использовать меню можно только в процессе решения задач.

Главное меню MATLAB содержит следующие шесть пунктов:

- ◆ **File** (Файл) — работа с файлами;
- ◆ **Edit** (Правка) — редактирование;
- ◆ **View** (Вид) — управление окнами;
- ◆ **Web** — связь с фирмой-разработчиком через Интернет;
- ◆ **Window** (Окно) — связь с окнами системы;
- ◆ **Help** (Справка) — связь со справочной системой MATLAB.

Рассмотрим главное меню системы, его команды и операции.

### 1.2.1. Меню *File*

Меню содержит большое число команд. Будем их называть в дальнейшем пунктами главного меню системы.

Пункт **New** (Создать) предоставляет возможность создать новый объект, а именно:

- ◆ **M-File** (M-файл) — файл с расширением m, в который записываются программы;
- ◆ **Figure** (Фигура) — специальное окно для вывода графической информации. При решении математических задач обычно используются функции, которые создают такие окна автоматически, поэтому данный пункт меню при визуализации вычислений используется редко;
- ◆ **Model** (Модель) — модель Simulink;

◆ **GUI** — графический интерфейс пользователя (Graphical User interface), используется для создания собственных приложений.

Пункт **Open** (Открыть) позволяет выполнить открытие существующего объекта посредством стандартного диалогового окна. Открытие объекта можно также осуществить из окна **Current Directory** (Текущий каталог).

Пункт **Close Current Directory** (Закрыть текущий каталог) закрывает окно текущего каталога.

Пункт **Import Data** (Импортировать данные) производит импорт в среду MATLAB разнородных данных (анимационные ролики, звуковые файлы, числовые данные в различных форматах и т. д.).

Пункт **Save Workspace As** (Сохранить рабочую область как) выполняет сохранение рабочей области. Созданные и используемые переменные хранятся в специальной рабочей области (Workspace). При выходе из программы рабочая область автоматически уничтожается вместе со всеми переменными. Чтобы этого не происходило, необходимо сохранить рабочую область на диске в виде файла с расширением `mat`. Проще всего это сделать посредством пункта меню **Save Workspace As** (Сохранить рабочую область как), можно также использовать команду `save имя_файла.mat`. В дальнейшем сохраненная рабочая область загружается либо с помощью пункта меню **Import Data** (Импортировать данные), либо командой `load имя_файла`.

Пункт **Set Path** (Задать путь) организует работу с путями доступа. При обращении к конкретному файлу (например, к функции) система MATLAB не требует указания пути. Это происходит потому, что MATLAB использует специальный список папок с путями доступа к ним. Когда вводится имя файла (с целью его поиска), система автоматически просматривает все известные ей папки, имеющиеся в списке. При инсталляции MATLAB список путей доступа строится автоматически. Для сохранения файлов во вновь созданной папке необходимо внести имя папки и путь доступа в общий список с помощью пункта меню **Set Path** (Задать путь). При вызове данного пункта появляется диалоговое окно с элементами управления (кнопками и списком).

Наиболее актуальными являются кнопки **Add Folder** (Добавить папку), **Add With Subfolders** (Добавить вместе с вложенными папками). Нажав одну из кнопок, можно указать путь к желаемой папке, после чего она будет добавлена к списку путей доступа. Кнопка **Save** сохраняет сделанные изменения.

Пункт **Preferences** (Настройка) изменяет некоторые свойства рабочей среды системы MATLAB.

Следующие три пункта меню: **Page Setup** (Параметры страницы), **Print** (Печать) и **Print Selection** (Печать выделенной области) служат для вывода информации на принтер, являются стандартными для многих пакетов и объяснений не требуют.

Меню **File** (Файл) имеет список последних открывавшихся файлов. Он содержит имена файлов с путями доступа и позволяет загрузить эти файлы двойным щелчком мыши.

Пункт **Exit MATLAB** (Выход) позволяет завершить работу с программой.

## 1.2.2. Меню *Edit*

Пункты **Undo** (Отменить), **Redo** (Повторить), **Cut** (Вырезать), **Copy** (Копировать), **Paste** (Вставить), **Select All** (Выделить все) и **Find** (Найти) полностью соответствуют своему стандартному предназначению и в комментариях не нуждаются. Первые пять из перечисленных пунктов меню продублированы кнопками на панели инструментов.

Пункт **Paste Special** (Специальная вставка) используется для обмена с внешними программами (например, MS Excel), числовыми данными посредством буфера обмена.

Пункт **Clear Command Window** (Очистить окно команд) очищает командное окно.

Пункт **Clear Command History** (Очистить историю команд) очищает окно предыстории.

Пункт **Clear Workspace** (Очистить рабочую область) очищает рабочую область от хранящихся в ней переменных.

### 1.2.3. Меню *View*

Среда системы MATLAB располагает несколькими рабочими окнами. Поэтому необходимо иметь инструмент, позволяющий управлять появлением, расположением и содержимым этих окон. Таким инструментом является меню **View** (Вид).

Пункт **Desktop Layout** (Разметка рабочего стола) помогает задать количество и расположение окон путем исполнения пунктов подменю.

Пункт **Undock** (Отстыковать) позволяет сделать автономным (отделить окно от интерфейса системы) выделенное в данный момент (активное) окно. После выбора данного пункта надпись меняется на **Dock** (Пристыковать) с названием активного окна. Меняется также на противоположную и функция пункта меню. Теперь при его выборе автономное окно снова прикрепляется к общему окну системы.

Следующая группа пунктов меню с названиями окон является группой переключателей. Каждый из этих пунктов может сделать видимым или невидимым соответствующее окно.

Пункт **Current Directory Filter** (Фильтр текущего каталога) имеет подменю пунктов-переключателей. С помощью этих переключателей можно выводить в окно **Current Directory** (Текущий каталог) определенные типы файлов.

Пункт **Workspace View Options** (Параметры отображения рабочей области) позволяет менять состав информации о переменных в списке окна **Workspace** (Рабочая область). Здесь можно также отсортировать список переменных по различным критериям.

### 1.2.4. Меню *Web*

Меню **Web** позволяет, при наличии подключения к Интернету, заходить в различные разделы сайта фирмы-изготовителя программного продукта. Здесь можно ознакомиться с новой информацией о системе, скачать ее, получить консультацию по интересующим вопросам. Для осуществления последних двух возможностей необходимо быть зарегистрированным пользователем.

Кроме того, выбрав пункт MATLAB **File Exchange** (Расширения), можно иметь свободный доступ к огромной библиотеке открытых примеров использования MATLAB для решения большого спектра задач.

### 1.2.5. Меню *Window*

Отображает список всех открытых дополнительных окон и служит для оперативного перехода к нужному окну. Это могут быть графические окна или окна, содержащие текст программы (рабочие окна в этом списке не отображаются). Пункт меню **Close All** (Закрыть все) позволяет закрыть все окна, кроме рабочих.

### 1.2.6. Меню *Help*

Содержит большое количество полезной информации о MATLAB на английском языке.

## 1.3. Панель инструментов

Кнопки панели инструментов обеспечивают выполнение большинства необходимых команд решения математических задач. Всплывающие подсказки при обращении к кнопкам сообщают об их содержании.

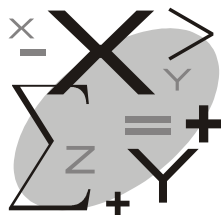
Кнопки панели инструментов имеют следующие назначения:

- ◆ **New file** (Создать) — выводит окна редактора файлов;
- ◆ **Open file** (Открыть) — открывает окна загрузки файлов;
- ◆ **Cut** (Вырезать) — вырезает выделенный файл и помещает в буфер обмена;
- ◆ **Copy** (Копировать) — копирует выделенный файл в буфер обмена;
- ◆ **Paste** (Вставить) — переносит фрагмент из буфера обмена в строку ввода;
- ◆ **Undo** (Отменить) — отменяет результат предыдущей операции;



- ◆ **Workspace Browser** (Просмотр рабочей области) — выводит окно ресурсов рабочей области;
- ◆ **Path Browser** (Просмотр пути) — выводит окно файловой структуры;
- ◆ **New Simulink Model** (Создать модель Simulink) — создает модель Simulink;
- ◆ **Help Window** (Справка) — открывает окна справки.

## 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
```