

Н. К. Смоленцев

МАТЛАВ

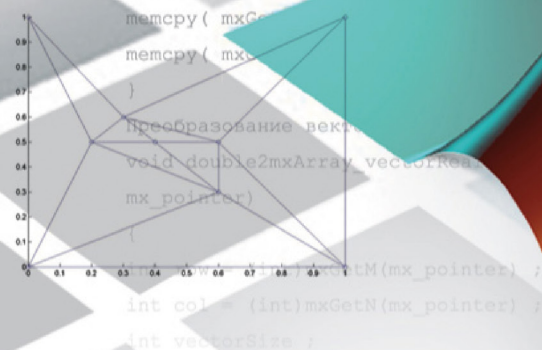
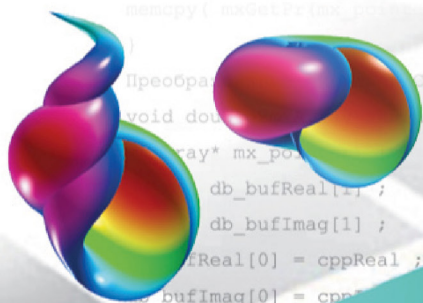
Программирование на **Visual C#**
Borland JBuilder, VBA

УЧЕБНЫЙ КУРС



CD-ROM

с примерами



ПИТЕР

ОМК
ИЗДАТЕЛЬСТВО

РЕКОМЕНДОВАНО
УЧЕБНО-МЕТОДИЧЕСКИМ СОВЕТОМ

Н. К. Смоленцев

MATLAB

**Программирование на Visual C#,
Borland JBuilder, VBA**

Учебный курс

*Рекомендовано научно-методическим советом
по математике и механике УМО
по классическому университетскому образованию РФ
в качестве учебного пособия для студентов высших учебных заведений,
обучающихся по группе математических и механических направлений
и специальностей*



Москва

УДК 519.68
ББК 32.913
С51

С51 Смоленцев Н. К.

MATLAB: Программирование на Visual C#, Borland C#, JBuilder, VBA: Учебный курс (+CD) – М.: ДМК Пресс; Спб.: Питер. – 464 с.: ил.

ISBN 978-5-388-00524-3

Данная книга посвящена изложению методов использования математических процедур MATLAB® при создании Windows-приложений, работающих независимо от MATLAB. Книга содержит введение в MATLAB и описание пакетов расширения MATLAB, позволяющих создавать компоненты, которые могут быть использованы при программировании на C++, Borland JBuilder, VBA в Excel и Visual Studio 2005. Кратко изложены необходимые сведения по языкам программирования Java и C#. Подробно рассматриваются примеры создания программ на Borland JBuilder, дополнений к Excel и программ на Visual C#, которые используют математические процедуры, разработанные на MATLAB. Освоение технологии использования математических возможностей MATLAB в других языках программирования позволит создавать полноценные Windows-приложения с развитой графической средой, в которых возможна реализация сложных математических алгоритмов.

Книга предназначена студентам и преподавателям ВУЗов по специальностям, близким к прикладной математике, профессиональным программистам, которые сталкиваются с проблемами реализации математических алгоритмов, и MATLAB-программистам, которые хотят использовать другие языки программирования для реализации алгоритмов MATLAB в виде законченных и независимых от MATLAB приложений.

MATLAB® is a trademark of The MathWorks, Inc. and is used with permission. The MathWorks does not warrant the accuracy of the text or exercises in this book. This book's use or discussion of MATLAB® software or related products does not constitute endorsement or sponsorship by The MathWorks of a particular pedagogical approach or particular use of the MATLAB® software.

УДК 519.68
ББК 32.913

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-5-388-00524-3

© Смоленцев Н. К.
© Оформление, издание, ДМК-Пресс

Краткое содержание

Предисловие	16
Глава 1. ОСНОВЫ РАБОТЫ В СИСТЕМЕ MATLAB®	19
Глава 2. КОМПИЛЯТОР MATLAB® ВЕРСИИ 4.6	85
Глава 3. СОЗДАНИЕ КОМПОНЕНТОВ ДЛЯ JAVA ПРИ ПОМОЩИ JAVA BUILDER	171
Глава 4. MATLAB BUILDER ДЛЯ EXCEL	291
Глава 5. СОЗДАНИЕ КОМПОНЕНТОВ ДЛЯ .NET ПРИ ПОМОЩИ .NET BUILDER	341
Предметный указатель	450
Содержание компакт-диска	451
Литература	455

Содержание

Предисловие	16
Глава 1. Основы работы в системе MATLAB®	19
1.1. Система компьютерной математики MATLAB®	20
1.1.1. Основные компоненты системы MATLAB	21
1.1.2. Инструментальные средства рабочего стола MATLAB	21
1.1.3. Константы и системные переменные MATLAB	27
1.1.4. Типы данных MATLAB	28
1.2. Основы работы с MATLAB®	31
1.2.1. Запуск MATLAB и начало работы	31
1.2.2. Задание массивов	33
Задание одномерных массивов	33
Задание двумерных массивов	34
1.2.3. Операции над массивами	36
1.2.4. Решение систем линейных уравнений	39
Символьная математика пакета расширения Symbolic Math	40
1.2.5. М-файлы	42
1.2.6. Чтение и запись текстовых файлов	44
1.2.7. Операции с рабочей областью и текстом сессии	47
1.3. Массивы символов	49
1.3.1. Задание массива символов	49
1.3.2. Общие функции	49
1.3.3. Проверка строк	51

1.3.4. Операции над строками	51
1.3.5. Преобразование чисел в символы и обратно	52
1.3.6. Функции преобразования систем счисления	54
1.3.7. Вычисление строковых выражений	55
1.4. Массивы ячеек	55
1.4.1. Создание массивов ячеек	56
1.4.2. Доступ к данным в ячейках	58
1.4.3. Вложенные массивы ячеек	60
1.4.4. Массивы ячеек, содержащих структуры	61
1.4.5. Многомерные массивы ячеек	62
1.5. Массивы структур	62
1.5.1. Построение структур	63
1.5.2. Доступ к полям и данным структуры	64
1.5.3. Многомерные массивы структур	67
1.6. Программирование в среде MATLAB	67
1.6.1. М-функции	67
1.6.2. Операторы системы MATLAB	73
1.6.3. Управление последовательностью исполнения операторов	76
1.6.4. Вычисление символьных выражений	80
1.6.5. Ошибки и предупреждения	81
1.6.6. Повышение эффективности обработки М-файлов ...	82

Глава 2. Компилятор MATLAB® версии 4.6

85

2.1. Основы работы с Компилятором MATLAB®	86
2.1.1. Назначение Компилятора MATLAB	86
2.1.2. Инсталляция и конфигурирование	87
2.1.3. Пример использования Компилятора	88

Среда разработки Deployment Tool	88
Создание приложения	90
Использование команды msc	93
2.1.4. Среда выполнения компоненты MATLAB, библиотека MCR	94
2.1.5. Файлы, создаваемые Компилятором	95
Технологический файл компоненты (CTF)	96
Файлы обертки	96
2.2. Процесс создания компонента MATLAB®	97
2.2.1. Процесс создания компонента	97
2.2.2. Управление путями при компиляции	98
2.3. Работа с msc и mbuild	100
2.3.1. Работа с msc	101
Обычное использование msc	101
Опции msc	102
Порядок использования опций	105
Использование файлов групп	106
Создание файлов обертки	107
2.3.2. Использование псевдокомментариев	108
2.3.3. Несколько полезных замечаний	109
2.3.4. Функция mbuild	110
2.4. Примеры создания автономных приложений и библиотек	111
2.4.1. Библиотеки совместного использования	111
Библиотека совместного использования C	111
Функции, создаваемые из m-файлов	118
Использование varargin и varargout в интерфейсе m-функции	119
C++ библиотека совместного использования	119
2.4.2. Создание автономных приложений	122
Создание кода только из m-файлов	123
Объединение M-файлов и кода C или C++	124

2.5. Классы C++ Компилятора 4.6 MATLAB®	127
2.5.1. Основные типы данных	127
2.5.2. Класс mxArray	128
Конструкторы	128
Методы копирования	130
Методы получения информации о массиве	130
Методы сравнения	132
Методы доступа к элементам массива mxArray	132
Операторы	134
Статические методы	135
2.5.3. Класс mwString	136
Конструкторы	136
Методы	136
Операторы	136
2.5.4. Класс mxArrayException	137
Конструкторы	137
Методы	138
Операторы	138
2.6. Внешние интерфейсы	138
2.6.1. Процедуры доступа к MAT-файлам	139
2.6.2. Операции с массивами mxArray	139
2.7. Передача значений между C/C++ double, mxArray и mxArray	141
2.7.1 Преобразование значений между C/C++ double и mxArray	142
Преобразование скаляров	142
Преобразование векторов	142
Преобразование матриц	143
2.7.2 Преобразование значений из C/C++ double в mxArray	143
Преобразование скаляров	143
Преобразование векторов	143
Преобразование матриц	143

2.7.3 Преобразование значений из mxArray в C/C++ double	144
Преобразование скаляров	145
Преобразование векторов	145
Преобразование матриц	145
2.7.4. Вспомогательные функции преобразования данных ..	145
Преобразование значений из C/C++ double в mxArray	146
Преобразование значений из mxArray в C/C++ double ..	147
Преобразование из C/C++ double в mxArray	149
Преобразование mxArray в C/C++ double	149
Пример создания заголовочного файла	150
2.8. Математическая библиотека C++ MATLAB® 6.5	151
2.8.1. Расположение файлов математической библиотеки C++	152
2.8.2. Документация Математической библиотеки MATLAB C++	153
2.8.3. Знакомство с Математической библиотекой MATLAB C++	154
2.8.4. Работа с массивами mxArray	155
Числовые массивы	156
2.8.5. Подключение математических библиотек к Borland C++ Builder	160
2.8.6. Примеры приложений использующих математические библиотеки	161
Чтение, обработка и запись данных	162
Построение графиков данных mxArray	166

Глава 3. Создание компонентов для Java при помощи Java Builder

3.1. Язык программирования Java	172
3.1.1. Основные элементы программирования на Java ...	173
Первая программа на Java	173
Комментарии и имена	175
Константы	175

Типы данных	176
Операции	182
Операторы	184
Массивы	187
3.1.2. Классы в Java	189
Понятие класса	189
Как описать класс и подкласс	191
Окончательные члены и классы	193
Класс Object	193
Опертор new	194
Конструкторы класса	194
Статические члены класса	195
Метод main()	196
Где видны переменные	196
Вложенные классы	197
Пакеты и интерфейсы	197
Структура Java-файла	200
3.2. Введение в Java Builder	201
3.2.1. Общие сведения о MATLAB Builder для Java	201
3.2.2. Графический интерфейс пользователя MATLAB Builder для Java	203
3.2.3. Создание компонента Java	205
3.2.4. Использование командной строки для создания компонента	208
3.2.5. Разработка приложения, использующего компонент	210
3.2.6. Обсуждение примера магического квадрата	213
3.3. Массивы MATLAB в Java	214
3.3.1. Использование методов класса MWArray	215
Построение MWArray	216
Методы получения информации о MWArray	216
Методы получения и задания данных в MWArray	218
Методы копирования, преобразования и сравнения массивов MWArray	220

Методы для использования на разреженных массивах MWArray	221
3.3.2. Использование MWNumericArray	222
Построение различных типов числовых массивов	223
Методы уничтожения MWNumericArray	227
Методы для получения информации о MWNumericArray	227
Методы доступа к элементам и задания элементов MWNumericArray	228
Методы копирования, преобразования и сравнения массивов MWNumericArray	232
Методы возвращения значений специальных констант	233
Методы toTypeArray и getTypeArray преобразования массивов данных	234
Методы работы с разреженными массивами MWNumericArray	235
3.3.3. Работа с логическими, символьными и массивами ячеек	237
3.3.4. Использование MWClassID	239
Поля MWClassID	240
Методы класса MWClassID	240
3.3.5. Использование класса MWComplexity	240
3.4. Примеры приложений Java	241
3.4.1. Пример спектрального анализа	241
Построение компонента	241
Разработка приложения, использующего компонент	244
3.4.2. Пример матричной математики	248
Построение компонента	251
Разработка приложения, использующего компонент	250
3.5. Некоторые вопросы программирования	255
3.5.1. Импорт классов и создание экземпляра класса	255

3.5.2. Правила обращения к методам Java Builder	255
Стандартный интерфейс	256
Интерфейс mlx	257
3.5.3. Правила преобразования данных MATLAB и Java ...	258
Автоматическое преобразование в тип MATLAB	259
Преобразование типов данных вручную	260
3.5.4. Аргументы методов Java Builder	262
Передача неопределенного числа параметров	262
Получение информации о результатах методов	264
Передача объектов Java по ссылке	266
3.5.5. Обработка ошибок	266
Обработка исключений MWEException	266
Обработка общих исключений	268
3.5.6. Управление собственными ресурсами	269
Использование «сборки мусора» JVM	269
Использование метода dispose	269
3.6. Среда проектирования JBuilder	271
3.7. Примеры создания приложений	
с использованием классов Java Builder	277
3.7.1. Объем n-мерного шара	
и площадь (n-1)-мерной сферы	278
Создание компонента Java Builder	278
Создание приложения JBuilder	279
Создание пакета	
для распространения приложения	283
3.7.2. Магический квадрат	285
Глава 4. MATLAB Builder для Excel	291
4.1. Введение	292
4.1.1. Создание компонента для Excel	294
4.1.2. Установка компонента на другие машины	297

4.1.3. Мастер функций	298
4.1.4. Работа с компонентами в Excel	303
4.2. Общие вопросы создания компонент Excel Builder	305
4.2.1. Процедура создания компонента	305
4.2.2. Регистрация компонента	306
4.2.3. Разработка новых версий	307
4.3. Пример создания дополнения для спектрального анализа	308
4.3.1. Построение компонента	308
4.3.2. Подключение компонента к Excel с использованием VBA	311
4.3.3. Создание формы Visual Basic	314
4.3.4. Добавление пункта меню Spectral Analysis в Excel	317
4.3.5. Тестирование дополнения	319
4.3.6. Упаковка и распространение дополнения	320
4.3.7. Обсуждение программы VBA	321
4.3.8. Использование флагов	324
4.4. Библиотека утилит Excel Builder	326
4.4.1. Функции MATLAB Builder для Excel	326
4.4.2. Библиотека утилит Excel Builder	327
Класс MWUtil	328
Класс MWFlags	328
Class MWStruct	330
Класс MWField	330
Класс MWComplex	330
Class MWSparse	331
Класс MWArg	331
Перечисления	332
4.5. Справка по VBA	333

Глава 5. Создание компонентов для .NET при помощи .NET Builder	341
5.1. Среда разработки Microsoft .NET	342
5.1.1. Основные элементы платформы Microsoft .NET	342
Новые понятия	343
5.1.2. Среда выполнения .NET Framework	346
5.1.3. Стандартная система типов	347
5.1.4. Общая спецификация языков программирования	349
5.2. Основы языка C#	349
5.2.1. Элементы синтаксиса языка C#	350
Алфавит и слова C#	350
Структура программы C#	351
Переменные и константы C#	352
Объявление переменных. Область видимости и время жизни	353
5.2.2. Система типов	354
Значимые и ссылочные типы	355
Системные встроенные типы	356
Приведение типов	357
Логический тип	359
Строковые и символьные типы	359
Перечисления	360
Организация системы типов	361
5.2.3. Массивы	362
5.2.4. Операции и выражения	365
5.2.5. Управление последовательностью выполнения операторов	367
Оператор if...else условного перехода	367
Оператор switch	368
Оператор цикла while	369
Оператор цикла do... while	369

Оператор цикла for	370
Операторы break и continue	370
5.2.6. Класс и структура	371
Классы	371
Структуры	375
Интерфейсы	375
5.2.7. Отражение	376
5.3. Введение в .NET Builder	377
5.3.1. Библиотека классов .NET MWArray	379
5.3.2. Правила преобразования данных	381
5.3.3. Интерфейсы, создаваемые .NET Builder	384
5.3.4. Задание сборки компонента и пространства имен ...	387
5.4. Создание консольный приложений	387
5.4.1. Пример магического квадрата	388
Создание .NET компонента	388
Использование компонента в приложении	391
5.4.2. Пример матричной математики	395
Создание .NET компонента	395
Использование компонента в приложении	396
5.4.3. Использование командной строки для создания компоненты .NET	400
5.5. Некоторые вопросы программирования с компонентами .NET Builder	402
5.5.1. Обязательные элементы программы	402
5.5.2. Передача входных параметров	404
Примеры передачи входных параметров	405
Передача массива вводов	406
Обработка глобальных переменных MATLAB	407
Обработка возвращаемых значений	407
Использование запросов MWArray	408
5.5.3. Обработка ошибок	409
5.5.4. Управление родными ресурсами	410

5.5.5. Преобразования между типами C# и <code>MWNumericArray</code>	412
Преобразование скаляров	412
Преобразование векторов	413
Преобразование матриц	414
5.6. Среда разработки Visual Studio 2005	415
5.6.1. Создание нового проекта	419
5.7. Программирование на Visual Studio 2005 с использованием математических процедур MATLAB	420
5.7.1. Вычисление интегралов	421
Создание .NET компонента	421
Создание приложения	422
5.7.2. Решение обыкновенных дифференциальных уравнений	429
Создание .NETкомпонента ODE	429
Создание Windows-приложения	432
5.7.3. Открытие, обработка и сохранение файлов	440
Создание .NETкомпонента	440
Создание приложения	442
Предметный указатель	450
Содержание компакт-диска	451
Литература	455

Предисловие

Как известно, система MATLAB® является одной из наиболее мощных универсальных систем компьютерной математики. Возможности системы MATLAB уникальны. Список основных функций MATLAB (не включая специализированных функций пакетов расширений) содержит более 1000 наименований. Кроме встроенных процедур, система MATLAB имеет чрезвычайно легкий в использовании язык программирования высокого уровня, основанный на таких мощных типах данных, как многомерные числовые массивы, массивы символов, ячеек и структур MATLAB. Программы, написанные на m-языке MATLAB работают только в среде MATLAB, однако в системе MATLAB предусмотрены возможности создания приложений на других языках программирования, которые используют процедуры, написанные на m-языке MATLAB. До выпуска MATLAB 6.5 для этих целей предназначались математические библиотеки C/C++ MATLAB, которые позволяли создавать автономные C/C++ приложения, использующие функции MATLAB.

Начиная с выпуска MATLAB 7, корпорация MathWorks отказалась от дальнейшего использования математических библиотек C/C++, существенно изменив и расширив возможности пакета расширения MATLAB Compiler – Компилятора MATLAB. При этом были разработаны такие расширения MATLAB Compiler, как: MATLAB Builder for Java – пакет расширения для создания и использования компонентов для языка Java; MATLAB Builder for Excel – пакет расширения для создания и использования дополнений (Add-Ins) Excel; MATLAB Builder for .NET – пакет расширения для создания и использования компонентов в среде .NET Framework. Для обеспечения работы компонентов, созданных Компилятором MATLAB, разработана универсальная среда MCR исполнения компонентов MATLAB. Программа, созданная на других языках программирования и использующая скомпилированные функции MATLAB, выполняется только с MCR. Сама система MATLAB для работы приложения не требуется. Созданные компоненты MATLAB и приложения, их использующие, могут свободно распространяться вместе со средой исполнения MCR.

В данной книге рассматривается использование Компилятора MATLAB и его расширений: MATLAB Builder for Java, MATLAB Builder for Excel и MATLAB Builder for .NET. Данное издание является продолжением книги [ППС], в которой изложены математические библиотеки C/C++ MATLAB и показано их использование для создания Windows-приложений на Borland C++ Builder. Особенности программирования для систем, отличных от Windows, можно найти в документации MATLAB®.

Рассмотрим кратко содержание данной книги по главам.

Первая глава содержит первоначальные сведения о системе MATLAB. Она предназначена для читателей, которые владеют программированием, но не на MATLAB. Глава содержит описание работы с числовыми массивами, массива-

ми символов, ячеек и структур, а также основы программирования в среде MATLAB.

Вторая глава посвящена описанию пакета расширения MATLAB Compiler версии 4.6 (для MATLAB R2007a). Возможности Компилятора огромны. Компилятор MATLAB поддерживает почти все функциональные возможности MATLAB. Компилятор MATLAB из m-файлов MATLAB может создать C или C++ автономные консольные приложения и библиотеки общего доступа (dll). Изложение материала сопровождается обсуждением тестовых примеров MATLAB (эти примеры входят в инсталляционный пакет MATLAB Compiler). В конце главы дано краткое описание математических библиотек C/C++ MATLAB 6.5 и приведены примеры их использования.

Глава 3 посвящена созданию компонентов для Java и приложений Java, которые используют компоненты MATLAB. Вначале кратко излагаются необходимые сведения о языке Java. Подробно на примерах рассматривается создание компонентов и консольных Java-приложений, которые используют созданные компоненты (учебные примеры MATLAB). В конце главы подробно рассмотрено создание Windows-приложений на Borland JBuilder, которые используют упакованные в компоненты процедуры MATLAB.

В главе 4 рассматривается создание компонентов для Excel и VBA-приложений, которые используют эти компоненты MATLAB. Система MATLAB предлагает свое, фирменное, дополнение к Excel для использования при работе в Excel скомпилированных функций MATLAB. Это дополнение называется «Мастер функций». Рассмотрена работа с Мастером функций. Кроме того, рассмотрено создание на VBA собственных дополнений для решения различных математических задач с данными Excel.

Глава 5 посвящена созданию .NET-компонентов и приложений .NET, которые используют компоненты MATLAB. Вначале кратко излагаются необходимые сведения о .NET Framework и языке программирования C#. Подробно на примерах рассматривается создание компонентов и консольных C#-приложений, которые используют созданные компоненты (учебные примеры MATLAB). В конце главы подробно рассмотрено создание Windows-приложений на Visual Studio 2005, которые используют процедуры MATLAB.

Книга имеет приложение в виде компакт-диска с исходными текстами примеров программ, рассматриваемых в данной книге.

Книга предназначена студентам и преподавателям ВУЗов по специальностям, близким к прикладной математике, профессиональным программистам, которые сталкиваются с проблемами реализации математических алгоритмов и MATLAB-программистам, которые хотят использовать другие языки программирования для реализации алгоритмов MATLAB в виде законченных и независимых от MATLAB приложений.

Освоение технологии использования колоссальных математических возможностей MATLAB в других языках программирования позволит создавать полноценные Windows-приложения с развитой графической средой, в которых возможна

реализация сложных математических алгоритмов для решения научно-технических задач.

Книга написана при содействии корпорации MathWorks в соответствии с программой MathWorks поддержки книг, посвященных MATLAB®. Автор выражает благодарность компании MathWorks за предоставленную возможность использования документации и лицензионного программного обеспечения MATLAB® R2007a для написания этой книги.

Основы работы в системе MATLAB®

1.1. Система компьютерной математики MATLAB®	20
1.2. Основы работы с MATLAB®	31
1.3. Массивы символов	49
1.4. Массивы ячеек	55
1.5. Массивы структур	62
1.6. Программирование в среде MATLAB	67

MATLAB® – это одна из старейших систем компьютерной математики, построенная на применении матричных операций. Название MATLAB происходит от слов *matrix laboratory* (матричная лаборатория). Матрицы широко применяются в сложных математических расчетах. Однако в настоящее время MATLAB далеко вышла за пределы специализированной матричной системы и стала одной из наиболее мощных универсальных систем компьютерной математики. В MATLAB используются такие мощные типы данных, как многомерные числовые массивы, массивы символов, ячеек и структур, что открывает широкие возможности применения системы во многих областях науки и техники. В данной главе мы кратко рассмотрим некоторые вопросы работы в системе MATLAB.

Описание системы MATLAB® и ее применения к решению различных задач математического анализа, обработки данных, решения дифференциальных уравнений и к графике можно найти в *Help MATLAB* и в любом руководстве по MATLAB, см. например [ККШ], [Пот], [ЧЖИ], [Кр], [Ма], [Ко], [Д], [ГЦ], [Ан] и [ППС]. Отметим также интернет-ресурсы [W].

1.1. Система компьютерной математики MATLAB®

Система MATLAB® была разработана в конце 70-х гг. и широко использовалась на больших ЭВМ. В дальнейшем были созданы версии системы MATLAB для персональных компьютеров с различными операционными системами и платформами. К расширению системы были привлечены крупнейшие научные школы мира в области математики, программирования и естествознания. Одной из основных задач системы является предоставление пользователям мощного языка программирования высокого уровня, ориентированного на математические расчеты и способного превзойти возможности традиционных языков программирования для реализации численных методов.

Система MATLAB® объединяет вычисление, визуализацию и программирование в удобной для работы окружающей среде, где задачи и решения выражаются в привычном математическом виде. Обычные области использования MATLAB: математика и вычисления, разработка алгоритмов, моделирование, анализ данных и визуализация, научная и техническая графика, разработка приложений. В университетских кругах MATLAB® – это стандартный учебный инструмент для вводных и продвинутых курсов в математике, в прикладных исследованиях и науке. В промышленности, MATLAB – это инструмент высокой производительности для исследований, анализа и разработки приложений.

Поразительная легкость модификации системы и возможность ее адаптации к решению специфических задач науки и техники привели к созданию десятков пакетов прикладных программ (*Toolboxes*), намного расширивших сферы применения системы. Пакеты расширений представляют собой обширные библиотеки функций MATLAB® (*m-файлы*), которые созданы для использования MATLAB в решении специальных задач. Пакеты расширения (их число более 50) включают такие инте-

ресные области, как обработка сигналов, системы управления, нейронные сети, нечеткая логика, биоинформатика, вейвлеты, моделирование и много других.

Возможности системы MATLAB® уникальны. Список основных функций MATLAB® (не включая специализированных функций пакетов расширений) содержит более 1000 наименований.

1.1.1. Основные компоненты системы MATLAB

Система MATLAB состоит из пяти главных частей.

Среда разработки. Это набор инструментов и средств обслуживания, которые помогают использовать функции и файлы MATLAB. Многие из этих инструментов – графические пользовательские интерфейсы. Среда разработки включает рабочий стол MATLAB и командное окно, окно истории команд, редактор-отладчик, и браузеры для просмотра помощи, рабочего пространства, файлов и путей поиска.

Библиотека математических функций MATLAB. Это обширное собрание вычислительных алгоритмов от элементарных функций типа суммы, синуса, косинуса и комплексной арифметики, до более сложных функций типа транспонирования, обращения матриц, нахождения собственных значений матриц, функций Бесселя и быстрого преобразования Фурье.

Язык MATLAB. Это язык высокого уровня, основанный на работе с матричными массивами, с функциями управления потоками, структурами данных, вводом/выводом и объектно-ориентированным программированием. Он позволяет быстро и легко освоить создание небольших программ, а также имеется возможность создания полных и сложных прикладных программ.

Графика. MATLAB имеет обширные средства для графического отображения векторов и матриц, а также создания аннотаций и печати этих графиков. Графика MATLAB включает функции высокого уровня для двумерной и трехмерной визуализации данных, обработки изображений, анимации, и презентационной графики. Графика MATLAB также включает функции низкого уровня, которые позволяют полностью настроить вид графики и создавать законченные графические интерфейсы пользователя на ваших приложениях MATLAB.

MATLAB API (Application Program Interface, интерфейс прикладного программирования). Это библиотека, которая позволяет писать программы C и Fortran совместно с MATLAB. API включает средства для вызова подпрограмм из MATLAB (динамическая связь), вызывая MATLAB как вычислительный механизм, и для чтения и записи MAT-файлов.

1.1.2. Инструментальные средства рабочего стола MATLAB

При запуске MATLAB, появляется *рабочий стол* MATLAB. Он содержит инструменты (графические пользовательские интерфейсы) для управления файлами, переменными и приложениями, связанными с MATLAB. Рабочий стол MATLAB имеет вид как на рис. 1.1.1.

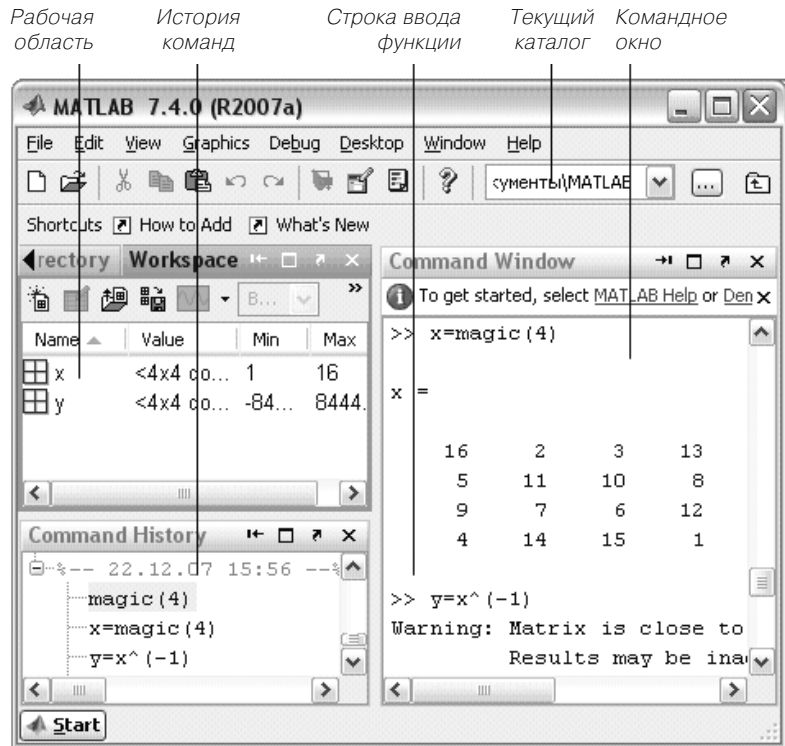


Рис. 1.1.1. Рабочий стол MATLAB

Инструментальные средства рабочего стола MATLAB включают следующие компоненты:

- командное окно (Command Window);
- браузер рабочей области (Workspace Browser);
- редактор массива (Array Editor);
- история команд (Command History);
- браузер текущего каталога (Current Directory Browser);
- кнопка старта (Start);
- браузер справки (Help Browser);
- редактор/отладчик (Editor/Debugger);
- профилировщик (Profiler).

Замечание 1. Некоторые характеристики для настольных инструментальных средств можно определить, выбирая **Preferences** из меню **File**.

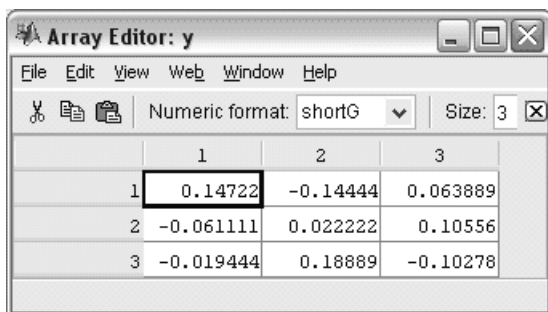
Рассмотрим подробнее инструментальные средства рабочего стола.

Командное окно (Command Window). Используется для ввода команд, переменных и выполнения функции и m-файлов. Команду можно вызвать в строке ввода – это последняя строка с символом приглашения (>>). Выполненная команда перестает быть активной, она недоступна для редактирования. Ранее ис-

полненные команды можно ввести в командную строку либо из окна истории команд, либо пролистывая их в командной строке клавишами «стрелка вверх/вниз».

Браузер рабочей области (Workspace Browser). Рабочая область MATLAB состоит из набора переменных (массивов) созданных в течение сеанса MATLAB и сохраненных в памяти (см. рис. 1.1.1). Переменные добавляются к рабочей области в результате выполнения функций, m-файлов, или при загрузке сохраненных ранее рабочих областей. В рабочей области содержится информация о каждой переменной, см. рис. 1.1.1. Содержимое этой области можно просмотреть также из командной строки с помощью команд `who` и `whos`. Команда `who` выводит только имена переменных, а команда `whos` – информацию о размерах массивов и типе переменной.

Чтобы удалить переменные из рабочей области, достаточно выбрать переменную и выполнить **Delete** в меню **Edit**, либо в, открывающемся правой кнопкой мыши, контекстном меню. Чтобы сохранить рабочую область в файле, который может быть загружен в следующем сеансе MATLAB, достаточно выбрать **Save Workspace As** в меню **File**, или использовать функцию `save`. Рабочая область сохраняется в бинарном MAT-файле. Чтобы прочитать данные из MAT-файла, нужно выбрать **Import Data** из меню **File**.



	1	2	3
1	0.14722	-0.14444	0.063889
2	-0.061111	0.022222	0.10556
3	-0.019444	0.18889	-0.10278

Рис. 1.1.2. Редактор массива

Редактор массивов. Если дважды щелкнуть мышкой по переменной в рабочей области, то эта переменная отобразится в *редакторе массива*. Он используется для визуального просмотра и редактирования одно- или двумерных числовых массивов, массивов строк и массивов ячеек строк, которые находятся в рабочей области.

История команд (Command History). Инструкции и команды, которые вводятся командном окне, регистрируются в окне истории команд. Можно рассмотреть ранее выполненные команды, копировать и выполнить выбранные команды. Чтобы сохранить вводы и выводы сессии MATLAB в файл используется функция `diary`.

Браузер текущего каталога (Current Directory). М-файл, который можно выполнить в командном окне, должен находиться или в *текущем* каталоге или на пути поиска файлов. Для быстрого изменения текущего каталога можно использовать поле **Current Directory** в инструментальной панели рабочего стола. Для просмотра содержания текущего каталога используется браузер текущего каталога. Он позволяет также менять каталог, искать файлы, открывать файлы и делать изменения.

Кнопка Start. Обеспечивает свободный доступ к инструментальным средствам, демонстрационным версиям, и документации.

Браузер справки (Help). MATLAB имеет обширную и прекрасно организованную документацию, состоящую из описания функций и серии электронных книг для более глубокого изучения методов, используемых в MATLAB. Справочный материал и электронные книги созданы в формате html, поэтому доступ к ним возможен как в среде MATLAB, так и независимо. Для поиска и изучения документации и демонстрационных версий для всех программ в среде MATLAB используется *Help-браузер* MATLAB. Он открывается из меню **Help**, или нажатием кнопки справки «?» в инструментальной панели, или из командной строки командой `helpbrowser`.

Браузер справки состоит из двух панелей, Навигатор (Help Navigator), который используется для поиска, и правая панель, где отображается выбранная информация.

Навигатор справки содержит оглавление документации в раскрывающихся списках. После выбора темы появляется следующий раскрывающийся список с содержанием документации по данной теме. При этом следует обратить внимание, что значок двух синих страниц обозначает руководство пользователя по данной теме, а значок двух желтых страниц обозначает справку по функциям. Например, на рис. 1.1.3, *Using the Symbolic Math Toolbox* – это руководство пользователя (электронная книга) по пакету символьной математики, а *Function Reference* – справка по функциям пакета. Help-навигатор имеет следующие возможности:

- **Product filter** (Фильтр программ) – устанавливается для того, чтобы показывать документацию только для заданных продуктов системы MATLAB;
- **Contents** (Содержание) – отражает заголовки и оглавления документации;
- **Index** (Индекс) – справка по ключевым словам в алфавитном порядке;
- **Demos** (Демонстрационные примеры) – представляет и выполняет демонстрации многих продуктов MATLAB;
- **Search** (Поиск) – поиск по определенному слову или фразе в документации;
- **Favorites** (Избранное) – показывает список документов, которые предварительно определены как фавориты.

В правой панели отображается содержание найденной документации. Данное окно также имеет ряд дополнительных возможностей поиска и печати (гиперссылки на близкие темы, переход на следующую страницу, в самой нижней строке отображается путь и название файла справки). Отметим, что при выборе темы в

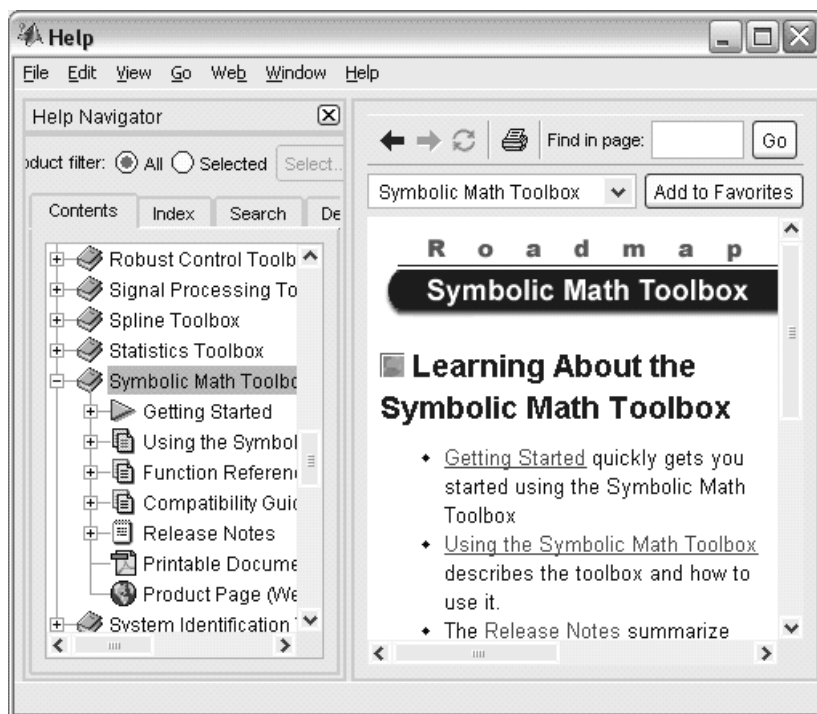


Рис. 1.1.3. Help-браузер MATLAB

Help-навигаторе, в правой панели также отражается содержание данной темы с комментариями.

Возможен прямой доступ к документации вне среды MATLAB. Для этого достаточно открыть каталог справки `C:\MATLAB\help\` и в нем открыть файл `begin_here.html`, либо в каталоге `C:\MATLAB\help\techdoc\` открыть `matlab_product_page.html`. Для справки по пакету расширения нужно открыть соответствующий каталог, например, `C:\MATLAB\help\toolbox\symbolic\` и в нем файл `symbolic.html` или `symbolic_product_page.html`.

Доступ к справке имеется и из командной строки MATLAB. Это наиболее быстрый способ выяснить синтаксис и особенности применения `m`-функции. Для этого используются команды `help <имя m-функции>` в командной строке. Соответствующая информация появляется непосредственно в командном окне. Например, команда `help magic` выведет в командное окно следующую информацию

```
help magic
```

```
MAGIC Magic square.
```

```
MAGIC(N) is an N-by-N matrix constructed from the integers
1 through N^2 with equal row, column, and diagonal sums.
```

```
Produces valid magic squares for all N > 0 except N = 2.
```

Все функции системы MATLAB организованы в логические группы, структура каталогов основана на этой организации. Например, все функции линейной алгебры находятся в каталоге `matfun`. Можно распечатать все функции этого каталога с короткими пояснениями, если использовать команду

```
help matfun
```

Команда `help` сама по себе выводит на экран список каталогов. Команда `lookfor` позволяет выполнить поиск `m`-функции по ключевому слову, при этом анализируется первая строка комментария, и она же выводится на экран, если в ней встретилось ключевое слово. Например, команда `lookfor inverse` выводит на экран большой список, начало которого представлено ниже

```
lookfor inverse
INVHILB Inverse Hilbert matrix.
IPERMUTE Inverse permute array dimensions.
ACOS    Inverse cosine.
ACOSH   Inverse hyperbolic cosine.
ACOT    Inverse cotangent.
ACOTH   Inverse hyperbolic cotangent.
ACSC    Inverse cosecant.
ACSCH   Inverse hyperbolic cosecant.
ASEC    Inverse secant.
ASECH   Inverse hyperbolic secant.
ASIN    Inverse sine. lookfor inverse
```

Дополнительные команды справочной системы. Укажем еще ряд команд, при помощи которых можно получить справочные данные в командном режиме:

- `computer` – выводит сообщение о типе компьютера, на котором установлена текущая версия MATLAB;
- `info` – выводит информацию о фирме Math Works с указанием адресов электронной почты;
- `ver` – выводит информацию о версиях установленной системы MATLAB и ее пакетах расширений;
- `version` – выводит краткую информацию об установленной версии MATLAB;
- `what` – выводит имена файлов текущего каталога;
- `what name` – выводит имена файлов каталога, заданного именем `name`;
- `whatsnew name` – выводит на экран содержимое файлов `readme` заданного именем `name` класса для знакомства с последними изменениями в системе и в пакетах прикладных программ;
- `which name` – выводит путь доступа к функции с данным именем;
- `help demos` – выводит весь список примеров в справочной системе MATLAB;
- `bench` – тест на быстродействие компьютера. Результаты теста представляются в виде таблицы и диаграммы сравнения с другими типами компьютеров.

Редактор/отладчик. Он используется для создания и отладки `m`-файлов, т.е. программ, написанных на языке MATLAB. *Редактор/отладчик* представляет со-

бой текстовый редактор с возможностями запуска и отладки программы m-файла. Редактор/отладчик вызывается либо из меню **File** \Rightarrow **New**, либо по кнопкам «новый документ», «открыть» в инструментальной панели MATLAB, либо двойным щелчком по соответствующему m-файлу. Если в редакторе/отладчике открыт m-файл из текущего каталога, он может быть запущен в MATLAB прямо из редактора по кнопке «Run». Эта кнопка записывает файл в текущий каталог и затем запускает его. Как видно на рис. 1.1.4, знак (%) означает начало текста комментария. Этот знак действует только в пределах одной строки. Возможности редактора/отладчика достаточно большие, однако с ними лучше познакомиться практически, записывая и запуская m-файлы.

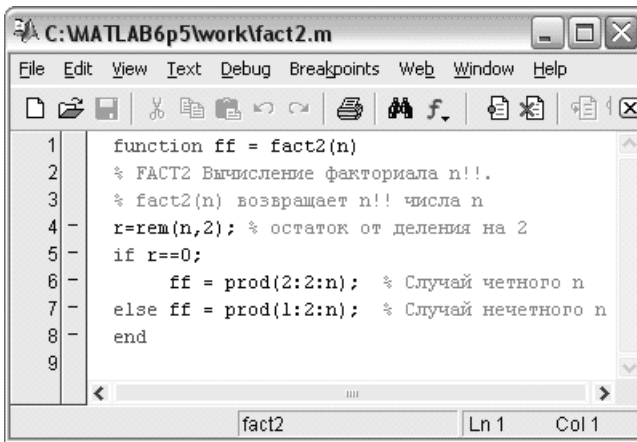


Рис. 1.1.4. Редактор/отладчик

Профилировщик (Profiler). Он представляет собой графический интерфейс пользователя, помогающий улучшить работу m-файла. Для открытия профилировщика нужно исполнить команду `profile viewer` в командной строке MATLAB.

1.1.3. Константы и системные переменные MATLAB

Это следующие специальные числовые и *системные константы*:

- **i** или **j** – мнимая единица (корень квадратный из -1);
- **pi** – число $\pi = 3.141592653589793e+000$;
- **eps** – погрешность операций над числами с плавающей точкой, это расстояние от единицы до ближайшего большего числа, $\text{eps} = 2.220446049250313e-016$, или 2^{-52} ;

- **realmin** – наименьшее число с плавающей точкой, $\text{realmin} = 2.225073858507202\text{e-}308$, или 2^{-1022} ;
- **realmax** – наибольшее число с плавающей точкой, $\text{realmax} = 1.797693134862316\text{e+}308$, или 2^{1023} ;
- **inf** – значение машинной бесконечности;
- **ans** – переменная, хранящая результат последней операции и обычно вызывающая его отображение на экране дисплея;
- **NaN** – неопределенность, нечисловое значение (Not-a-Number), например $0/0$.

1.1.4. Типы данных MATLAB

В MATLAB существует 15 основных типов данных (или классов). Каждый из этих типов данных находится в форме массива, вообще говоря, многомерного. Все основные типы данных показаны на рисунке 1.1.5. Дополнительные типы данных *user classes* и *java classes* могут быть определены пользователем как подклассы структур, или созданы из классов Java.

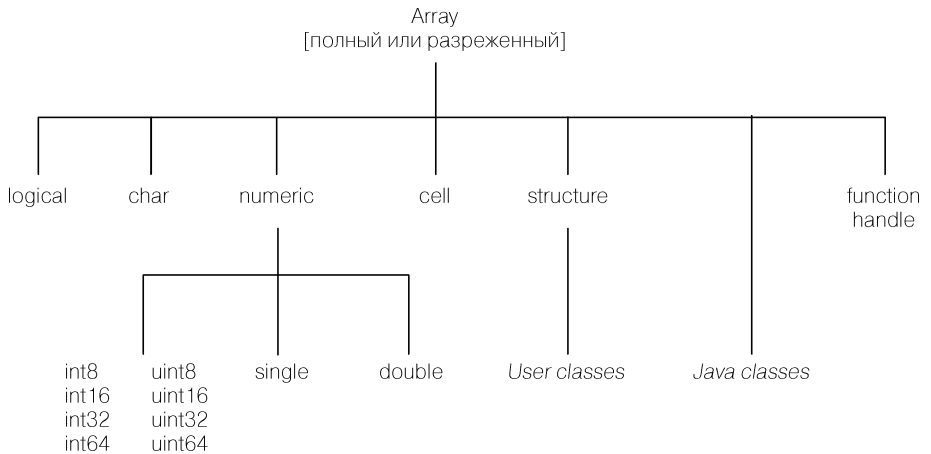


Рис. 1.1.5. Типы данных MATLAB

Типы переменных в MATLAB заранее не декларируются. Тип переменной *numeric* или *array* в MATLAB не задается. Эти типы служат только для того, чтобы сгруппировать переменные, которые имеют общие атрибуты. Матрицы типа *double* и *logical* могут быть как полными, так и разреженными. Разреженная форма матрицы используется в том случае, когда матрица имеет небольшое количество отличных от нуля элементов. Разреженная матрица, требует для хранения меньше памяти, поскольку можно хранить только отличные от нуля элементы и их индексы. Операции с разреженными матрицами требуют специальных методов.

Тип данных `logical`. *Логический массив.* Он представляет значения логических переменных `true` или `false`, используя логическую единицу (1, истина) и логический нуль (0, ложь), соответственно. Логические матрицы могут быть разреженными. MATLAB возвращает логические значения из отношений (например, `>`, `~ =`) и логических операций и функций. Например, следующая команда

```
x = magic(4) > 10
```

создает логический массив 4-на-4 из единиц и нулей, в соответствии с тем, больше элемент матрицы `magic(4)` числа 10, или нет.

Тип данных `char`. Массив символов (каждый символ 2 байта). Такой массив называют также строкой. Символьная строка – это просто массив 1-на-*n* символов. Можно создать массив *m*-на-*n* строк, если каждая строка в массиве имеет одну и ту же длину. Для создания массива строк неравной длины, используется массив ячеек. Массив символов может быть задан в командной строке в одинарных кавычках, например,

```
x = 'Привет!'
```

Числовые типы данных `numeric`. Это массивы чисел с плавающей запятой одинарной точности (`single`), массивы чисел с плавающей запятой двойной точности (`double`), массивы целых чисел со знаком (`int8`, ..., `int64`) и без знака (`uint8`, ..., `uint64`), которые имеют длину в 8, 16, 32, и 64 бита. Для числовых типов данных в MATLAB отметим следующее:

- все вычисления MATLAB делаются с двойной точностью;
- чтобы выполнять математические операции на целочисленных или массивах одинарной точности, нужно преобразовать их к двойной точности, используя функцию `double`.

Тип данных `int*`. Он содержит следующие типы:

- **`int8`** – массив 8-разрядных целых чисел со знаком (1 байт на одно число). Он позволяет хранить целые числа в диапазоне от -128 до 127;
- **`int16`** – массив 16-разрядных целых чисел со знаком (2 байта на одно число). Он позволяет хранить целые числа в диапазоне от -32 768 до 32 767;
- **`int32`** – массив 32-разрядных целых чисел со знаком (4 байта на одно число). Он позволяет хранить целые числа в диапазоне от -2 147 483 648 до 2 147 483 647;
- **`int64`** – массив 64-разрядных целых чисел со знаком (8 байт на одно число). Он позволяет хранить целые числа в диапазоне от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807.

Тип данных `uint*`. Он содержит следующие типы:

- **`uint8`** – массив 8-разрядных целых чисел без знака (1 байт на одно число). Он позволяет хранить целые числа в диапазоне от 0 до 255;
- **`uint16`** – массив 16-разрядных целых чисел без знака (2 байта на одно число). Он позволяет хранить целые числа в диапазоне от 0 до 65 535;
- **`uint32`** – массив 32-разрядных целых чисел без знака (4 байта на одно число). Он позволяет хранить целые числа в диапазоне от 0 до 4 294 967 295;