

Смоленцев Н. К.

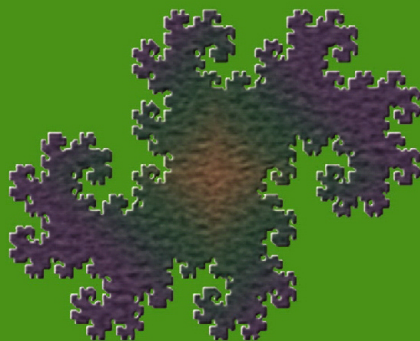
MATLAB.

Программирование на C++, C#, JAVA и VBA

Работа в системе MATLAB

*Создание компонентов
для JAVA*

*MATLAB Builder NE
для создания
компонентов .NET и Excel*



Н. К. Смоленцев

MATLAB.

Программирование на C++, C#, Java и VBA

3-е издание, электронное

*Рекомендовано УМС по математике и механике УМО
по классическому университетскому образованию РФ
в качестве учебного пособия для студентов высших учебных
заведений, обучающихся по направлениям и специальностям
«Математика», «Прикладная математика и информатика», «Механика»*



Москва, 2023

Смоленцев, Николай Константинович.

С51 **MATLAB. Программирование на C++, C#, Java и VBA / Смоленцев Н.К.. — 3-е изд., эл. — 1 файл pdf: 499 с. — Москва : ДМК Пресс, 2023. — Систем. требования: Adobe Reader XI либо Adobe Digital Editions 4.5 ; экран 10". — Текст : электронный.**

ISBN 978-5-89818-594-7

Всех, кто работал с системой MATLAB, поражает удивительная легкость написания программ на языке MATLAB для решения самых разнообразных задач. MATLAB предлагает классы, которые представляют основные типы данных MATLAB в других языках программирования: C/C++, Java, VBA, .NET. В системе имеется также возможность создания компонентов для этих языков, которые включают функции, написанные в MATLAB.

Изложению этой тематики посвящена данная книга. В ней подробно рассматривается работа Компилятора MATLAB, примеры создания независимых от MATLAB приложений на C++, Java, C#, VBA. Кроме того рассмотрена работа с MATLAB Production Server, что позволяет исполнять трудоемкие процедуры MATLAB на сервере MATLAB.

Освоение технологии использования колоссальных математических возможностей MATLAB в других языках программирования позволит создавать полноценные приложения с развитой графической средой для реализации сложных математических алгоритмов.

Издание предназначено студентам и преподавателям вузов по математическим специальностям, а также программистам, которые сталкиваются с проблемами реализации математических алгоритмов на MATLAB.

УДК 004.432
ББК 32.973.22

Электронное издание на основе печатного издания: MATLAB. Программирование на C++, C#, Java и VBA / Н. К. Смоленцев. — 2-е изд., перераб. и доп. — Москва : ДМК Пресс, 2015. — 498 с. — ISBN 978-5-97060-282-9. — Текст : непосредственный.

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.

В соответствии со ст. 1299 и 1301 ГК РФ при устранении ограничений, установленных техническими средствами защиты авторских прав, правообладатель вправе требовать от нарушителя возмещения убытков или выплаты компенсации.

ISBN 978-5-89818-594-7

© Смоленцев Н. К., 2015
© Оформление, издание, ДМК Пресс, 2015



ОГЛАВЛЕНИЕ

Предисловие	12
Глава 1.	
Система компьютерной математики MATLAB	15
Введение	15
1.1. Общее описание системы MATLAB	16
1.1.1. Инструментальные средства рабочего стола MATLAB	17
Меню ленты инструментов	18
Окна, используемые в работе MATLAB	20
Редактор массивов	21
Редактор m-файлов (MATLAB Editor)	22
Анализатор кода (Code Analyzer)	23
Профилировщик (Profiler)	24
Окно для просмотра графиков	24
1.1.2. Справочная система MATLAB	25
1.1.3. Константы и системные переменные MATLAB	28
1.1.4. Типы данных MATLAB	28
1.2. Основы работы с MATLAB	32
1.2.1. Запуск MATLAB и начало работы	32
1.2.2. Задание массивов	34
Задание одномерных массивов	35
Задание двумерных массивов	36
1.2.3. Операции над массивами	38
1.2.4. Решение систем линейных уравнений	42
1.2.5. Решение дифференциальных уравнений	43
1.2.6. Символьная математика пакета расширения Symbolic Math	48
1.2.7. М-файлы	51
1.2.8. Чтение и запись текстовых файлов	53
1.2.9. Операции с рабочей областью и текстом сессии	57
1.2.10. Графика в MATLAB	59
1.3. Программирование в среде MATLAB	64
1.3.1. Операторы системы MATLAB	64
Арифметические операторы	65
Операторы отношения	65
Логические операторы	66
Логические функции	67
1.3.2. Управление последовательностью исполнения операторов	68
1.3.3. М-функции	72
Подфункции	74
Частные функции	74
Вызов функции	75

Рабочая область функции.....	75
Проверка количества аргументов.....	76
Формирование входного массива varargin	77
Формирование выходного массива varargout	77
Локальные и глобальные переменные	78
1.3.4. Вычисление символьных выражений	79
1.3.5. Ошибки и предупреждения	80
1.3.6. Повышение эффективности обработки М-файлов	81
1.3.7. Пример. Огибающая семейства нормалей.....	83
1.4. Создание графического интерфейса пользователя в MATLAB.....	86
1.4.1. Среда разработки GUIDE графического интерфейса пользователя.....	86
Свойства инспектора свойств	91
Управления событиями GUI	92
Виды обратных вызовов	94
Структура m-файла приложения	95
Создание меню	97
1.4.2. Пример создания GUI «Предельные циклы. Границы хаоса».....	99
Постановка задачи	100
Создание GUI	104
Упражнение. Создания GUI «Предельные циклы. Границы хаоса»	107
Постановка задачи	107
1.5. Взаимодействие MATLAB и Microsoft Excel.....	108
1.5.1. Установка продукта и конфигурирование.....	109
Конфигурирование Microsoft Excel 2003	109
Конфигурирование Microsoft Excel 2007 и 2010	110
Установка предпочтений надстройки Spreadsheet Link EX	111
1.5.2. Функции Spreadsheet Link EX.....	112
Запуск и закрытие	112
Настройка	112
Экспорт данных в MATLAB	113
Импорт данных из MATLAB	113
Команды MATLAB в Microsoft Excel	113
1.5.3. Использование Spreadsheet Link EX	113
1.5.4. Использование Мастера функций (MATLAB Function Wizard).....	114
1.6. Массивы символов, ячеек и структур	117
1.6.1. Массивы символов	117
Общие функции.....	118
Преобразование чисел в символы и обратно	119
Функции преобразования систем счисления	120
Вычисление строковых выражений	121
1.6.2. Массивы ячеек.....	122
Создание массивов ячеек	122
Доступ к данным в ячейках	124
Вложенные массивы ячеек	126
1.6.3. Массивы структур	127
Построение структур	128
Доступ к полям и данным структуры	129

Глава 2.

Компилятор MATLAB 133

2.1. Общие сведения о Компиляторе MATLAB 133

2.1.1. Назначение Компилятора MATLAB	134
2.1.2. Инсталляция и конфигурирование	134
2.1.3. Среда выполнения компоненты MATLAB, библиотека MCR	135
2.1.4. Среда разработки Deployment Tool	136
2.2. Создание автономных приложений и библиотек	139
2.2.1. Создание автономного приложения.....	140
Подготовка к созданию приложения.....	140
Создание приложения.....	140
Установка приложения на другую машину	142
2.2.2. Библиотеки совместного использования С и обращение к ним из программы	143
Подготовка к созданию библиотеки	143
Создание библиотеки	143
Установка библиотеки на другую машину	146
Создание С-приложения, использующего библиотеку	146
Тестирование приложения	149
2.2.3. Библиотека совместного использования С++	150
2.2.4. Функции библиотеки, создаваемые из m-файлов	151
Использование varargin и varargout в интерфейсе m-функции	152
2.3. Программный интерфейс C/C++ API Компилятора MATLAB.....	153
Примеры	154
2.3.1. Классы С++ Компилятора 5.1 MATLAB	155
2.3.2. Класс mxArray.....	156
Основные типы данных	157
Конструкторы.....	157
Методы копирования	159
Методы получения информации о массиве	159
Методы доступа к элементам массива mxArray.....	160
Статические методы	162
Операторы	162
2.3.3. Класс mxArray	163
Конструкторы.....	164
Методы	164
Операторы.....	164
2.3.4. Класс mxArray.....	165
2.3.5. Внешние интерфейсы	165
Процедуры доступа к MAT-файлам.....	166
Операции с массивами mxArray.....	167
2.4. Передача значений между C/C++ double, mxArray и mxArray.....	173
2.4.1. Преобразование значений между C/C++ double и mxArray	173
Преобразование скаляров	174
Преобразование векторов	174
Преобразование матриц	174
2.4.2. Преобразование значений из C/C++ double в mxArray	174
Преобразование скаляров	175
Преобразование векторов	175
Преобразование матриц	175
2.4.3. Преобразование значений из mxArray в C/C++ double	176
Преобразование скаляров	176
Преобразование векторов	176
Преобразование матриц	177

2.4.4. Вспомогательные функции преобразования данных.....	177
Преобразование значений из C/C++ double в mxArray	177
Преобразование значений из mxArray в C/C++ double	179
Преобразование из C/C++ double в mxArray.....	180
Преобразование mxArray в C/C++ double	181
Пример создания заголовочного файла	181

Глава 3.

Создание компонентов для Java при помощи MATLAB Builder JA ... 183

3.1. Введение в Java Builder	183
3.1.1. Необходимое программное обеспечение Java.....	184
3.1.2. Установка и конфигурирование MATLAB Builder JA.....	185
Установка и настройки совместимости MATLAB Builder JA с Java.....	185
3.2. Создание и использование пакетов MATLAB Builder JA	188
3.2.1. Создание пакета Java средствами MATLAB Builder JA	189
Подготовка к созданию проекта	189
Создание компонента	190
3.2.2. Разработка приложения, использующего компонент	193
Создание кода приложения Java	193
Компиляция приложения.....	195
Запуск приложения	196
Упаковка и распространение приложения Java	196
3.2.3. Обсуждение примера Java-программы	197
3.2.4. Объем n -мерного шара и площадь $(n - 1)$ -мерной сферы	198
Создание компонента Java Builder.....	198
Создание кода приложения Java	199
Компиляция приложения.....	200
Запуск приложения	201
3.3. Создание оконных приложений в среде NetBeans.....	201
3.3.1. Среда проектирования IDE NetBeans	201
3.3.2. Объем n -мерного шара и площадь $(n - 1)$ -мерной сферы	206
Подключение библиотек MATLAB и созданного пакета Volume	207
Создание окна приложения	209
Задание элементов окна приложения	209
Создание программы приложения	210
Распространение приложения	212
3.3.3. Магический квадрат.....	213
Подключение библиотек MATLAB и пакета magicsquare.....	213
Создание окна приложения	214
Задание элементов окна приложения	214
Создание программы приложения	214
3.4. Некоторые вопросы программирования с Java Builder	216
3.4.1. Импорт классов и создание экземпляра класса	217
3.4.2. Правила обращения к методам Java Builder	217
Стандартный интерфейс	218
Интерфейс mx	219
3.4.3. Правила преобразования данных MATLAB и Java	220
Автоматическое преобразование в тип MATLAB	220
Преобразование типов данных вручную	221
3.4.4. Аргументы методов Java Builder	224
Передача неопределенного числа параметров	224

Получение информации о результатах методов	226
Передача объектов Java по ссылке.....	228
3.4.5. Обработка ошибок.....	228
Обработка исключений MWException.....	228
Обработка общих исключений.....	229
3.4.6. Управление собственными ресурсами.....	230
Использование «сборки мусора» JVM.....	231
Использование метода dispose	231
3.5. Массивы MATLAB в Java	232
3.5.1. Использование методов класса MWArray	233
Построение и удаление MWArray	234
Методы получения информации о MWArray	234
Методы получения и задания данных в MWArray	236
Методы копирования, преобразования и сравнения массивов MWArray.....	238
Методы для использования на разреженных массивах MWArray.....	240
3.5.2. Использование MWNumericArray.....	241
Построение различных типов числовых массивов	242
Методы уничтожения MWNumericArray	246
Методы для получения информации о MWNumericArray	246
Методы доступа к элементам и задания элементов MWNumericArray	247
Методы копирования, преобразования и сравнения массивов MWNumericArray	252
Методы возвращения значений специальных констант	254
Методы toTypeArray и getTypeArray преобразования массивов данных	254
Методы работы с разреженными массивами MWNumericArray	255
3.5.3. Работа с логическими, символьными и массивами ячеек.....	257
Логические массивы	257
Символьные массивы.....	258
Массивы ячеек	259
3.5.4. Использование MWClassID	260
Поля MWClassID	260
Методы класса MWClassID	261
3.5.5. Использование класса MWComplexity	261
3.6. Язык программирования Java	262
Общие сведения	262
3.6.1. Основные элементы языка Java	263
Комментарии и имена	264
Константы	265
Типы данных.....	266
Преобразования типов.....	270
Преобразование строки в число (STRING to NUMBER).....	270
Преобразование числа в строку (NUMBER to STRING)	271
Преобразования чисел.....	272
Преобразования символа char	272
Преобразования простых типов	273
Операции	273
Операторы	276
Массивы	279
3.6.2. Классы в Java	281
Понятие класса	281
Как описать класс и подкласс.....	283
Окончательные члены и классы	284
Класс Object.....	285
Оператор new	285

Конструкторы класса.....	286
Статические члены класса.....	286
Метод main()	287
Где видны переменные	288
Вложенные классы	288
Пакеты и интерфейсы	289
Структура Java-файла	291

Глава 4.

MATLAB Builder NE для создания компонентов .NET292

4.1. Среда разработки Microsoft Visual Studio .NET	293
4.1.1. Основные элементы платформы Microsoft .NET	293
Основные понятия платформы .NET	293
Среда выполнения .NET Framework	297
Стандартная система типов	298
Общая спецификация языков программирования	300
4.1.2. Среда разработки Visual Studio 2013	300
Запуск и вход в Visual Studio	300
4.1.3. Создание простого приложения	304
Создание проекта	304
Настройка проекта	306
Конструирование пользовательского интерфейса	307
Обработка событий	308
Отладка и тестирование приложения	309
Сборка окончательной версии.....	309
4.2. Начало работы с .NET Builder: создание консольных приложений	310
4.2.1. Назначение Компилятора MATLAB Builder NE.....	310
4.2.2. Установка и конфигурирование	311
4.2.3. Создание .NET сборки в среде разработки Deployment Tool.....	312
4.2.4. Разработка приложения для библиотеки матричной математики.....	316
Открытие и настройка проекта	316
Файл приложения	317
Обсуждение кода	319
Запуск приложения	320
Распространение сборки и приложения	321
4.2.5. Использование командной строки для создания .NET сборки	321
4.2.6. Создание COM-компонентов	323
Пример COM-компонента и приложения.....	325
4.3. Примеры Windows-приложений, использующих математические процедуры MATLAB	327
4.3.1. Вычисление интегралов.....	328
Разработка m-функций	328
Создание .NET-компонента.....	330
Создание приложения.....	330
4.3.2. Решение обыкновенных дифференциальных уравнений	337
Разработка m-функций	339
Создание .NET-компонента ODE	342
Создание Windows-приложения	343
4.3.3. Вейвлет-анализ сигналов. Открытие, обработка и сохранение файлов....	352
Вейвлет-анализ сигнала	352
Разработка m-функций	353
Создание .NET-компонента	356

Создание приложения.....	356
4.4. Введение в программирование с .NET Builder	368
4.4.1. Библиотека классов .NET MWArray	369
4.4.2. Правила преобразования данных.....	371
4.4.3. Интерфейсы, создаваемые .NET Builder.....	375
4.4.4. Задание сборки компонента и пространства имен	378
4.4.5. Обязательные элементы программы	378
4.4.6. Передача входных параметров	380
Примеры передачи входных параметров.....	382
Передача массива вводов	382
Обработка глобальных переменных MATLAB	383
Обработка возвращаемых значений.....	383
Использование запросов MWArray	385
4.4.7. Обработка ошибок.....	386
4.4.8. Управление родными ресурсами	386
4.4.9. Преобразования между типами C# и MWNumericArray	389
Преобразование скаляров	389
Преобразование векторов	390
Преобразование матриц	390
4.5. Основы языка C#	391
4.5.1. Элементы синтаксиса языка C#	392
Алфавит и слова C#.....	392
Структура программы C#	393
Переменные и константы C#	395
Объявление переменных. Область видимости и время жизни	396
4.5.2. Система типов	397
Тип object	397
Типы значений и ссылочные типы	397
Системные встроенные типы	399
Приведение типов	400
Логический тип	401
Строковые и символьные типы	401
Организация системы типов	403
4.5.3. Массивы	404
4.5.4. Операции и выражения	407
4.5.5. Класс и структура	409
Классы	409
Интерфейсы.....	414

Глава 5.

MATLAB Builder для Excel415

5.1. Установка MATLAB Builder EX	416
5.2. Создание компонента Excel для Мастера функций.....	418
5.2.1. Построение компонента матричной математики.....	418
Использование командной строки для построения компонент	421
5.2.2. Установка созданного компонента	422
5.2.3. Общие вопросы создания компонента Excel Builder	423
Процедура создания компонента	423
Регистрация компонента.....	424
Разработка новых версий.....	425
Получение информации о компоненте	426

5.3. Использование созданного приложения в Excel.....	427
5.4. Создание дополнения с пользовательским интерфейсом.....	433
5.4.1. Построение компонента	433
Подготовка файлов	433
Создание компонента	435
5.4.2. Разработка пользовательского интерфейса дополнения	435
Регистрация библиотеки Fourier_1_0.dll компонента	435
Выбор библиотек, необходимых для разработки дополнения.....	436
Создание кода VBA главного модуля приложения	436
Создание формы Visual Basic	439
Добавление пункта меню Spectral Analysis в Excel.....	442
Сохранение дополнения	443
5.4.3. Тестирование дополнения	443
5.4.4. Упаковка и распространение дополнения	445
5.4.5. Инсталляция приложения и его интегрирование в Excel.....	446
5.4.6. Обсуждение программы VBA	447
5.4.7. Использование флагов	450
5.5. Библиотека утилит Excel Builder	452
5.5.1. Функции MATLAB Builder для Excel	453
5.5.2. Библиотека утилит Excel Builder	454
Класс MWUtil	455
Класс MWFlags	456
Class MWStruct	458
Класс MWField	458
Класс MWComplex	459
Class MWSparse	460
Класс MWArg	460
Перечисления	460
5.6. Справка по VBA.....	461

Глава 6.

MATLAB Production Server.....469

6.1. Общие сведения о MATLAB Production Server	469
6.1.1. Назначение MATLAB Production Server	470
6.1.2. Инсталляция и конфигурирование сервера.....	471
Установка MATLAB Production Server	471
Конфигурирование.....	472
Создание локального экземпляра MATLAB Production Server	473
Запуск сервера	474
Проверка состояния сервера	475
Остановка сервера.....	475
Заключительные замечания	476
6.1.3. Подготовка программ MATLAB для MATLAB Production Server	477
Содержание readme-файла.....	479
6.2. Работа с MATLAB Production Server	480
6.2.1. Клиентское программирование на Java	481
Общие требования к Java-коду.....	481
Листинг Java-кода	483
Компиляция и запуск приложения	485
Распространение клиентского приложения.....	485

6.2.2. Клиентское программирование на C# .NET	486
Создание проекта Microsoft Visual Studio	486
Создание ссылки на клиентскую библиотеку	486
Разработка .NET интерфейса в C#	487
Написание построение и запуск .NET приложения	487
Список литературы	490
Перечень примеров программ	492
Предметный указатель	495



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. Однако программы, написанные на m-языке MATLAB, работают только в среде MATLAB. Поэтому хотелось бы иметь нечто подобное для других языков программирования. Оказывается это возможно! Система MATLAB предлагает классы, которые представляют основные типы данных MATLAB для многих других языков программирования. Первоначально такие классы были разработаны для языка C/C++ в составе математической библиотеки C/C++ MATLAB. Эта библиотека включает огромное количество математических функций, которые можно использовать в C/C++. Начиная с выпуска MATLAB R14 (2007 г.), корпорация MathWorks отказалась от дальнейшего развития математических библиотек C/C++, существенно изменив и расширив возможности пакета MATLAB Compiler – Компилятора MATLAB. При этом были разработаны такие расширения MATLAB Compiler, как: MATLAB Builder для Java – пакет расширения для создания и использования компонентов для языка Java; MATLAB Builder для Excel – пакет расширения для создания и использования дополнений (Add-Ins) Excel; MATLAB Builder для .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.

Данная книга является обновленным и переработанным изданием книги [См2]. Рассмотрим кратко содержание книги по главам.

Первая глава содержит первоначальные сведения о системе MATLAB R2014a. Глава предназначена для читателей, которые владеют программированием, но не на MATLAB. Глава содержит описание работы с числовыми массивами, мас-

сивами символов, ячеек и структур, а также основы программирования в среде MATLAB, включая работу с таблицами Excel и создание графического интерфейса пользователя.

Вторая глава посвящена описанию пакета расширения MATLAB Compiler версии 5.1 (для MATLAB R2014a). Возможности Компилятора огромны, он 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 простых примеров. Предполагается, что читатель знаком с основами языка C++.

Глава 3 посвящена созданию компонентов для Java и приложений Java, которые используют компоненты MATLAB. Рассмотрены следующие вопросы: установка и конфигурирование Java Builder для создания пакетов Java; создание из m-функций MATLAB пакетов Java Builder и их использование в консольных Java приложениях; создание приложений с графическим интерфейсом пользователя в среде проектирования NetBeans, в которых используются пакеты, созданные на Java Builder. Кроме того, рассмотрены некоторые особенности программирования на Java при использовании пакетов Java Builder, описание классов MWArray MATLAB для Java и справочные сведения по языку Java.

Глава 4 посвящена созданию .NET-компонентов и приложений .NET, которые используют компоненты MATLAB. Вначале кратко излагаются необходимые сведения о платформе .NET Framework и среде проектирования Visual Studio 2013. Далее подробно на примерах рассматривается создание компонентов и консольных C#-приложений, которые используют созданные компоненты. Подробно на примерах рассмотрены создания приложений с графическим интерфейсом пользователя в среде проектирования Visual Studio, в которых используются компоненты, созданные на .NET Builder. Разобрано построение следующих приложений:

- вычисление однократных и двойных интегралов;
- решение обыкновенных дифференциальных уравнений и систем дифференциальных уравнений;
- вейвлет-анализ сигналов: открытие, обработка и сохранение файлов.

В конце главы обсуждаются некоторые особенности программирования на C# при использовании компонентов .NET Builder и справочные сведения по языку C#.

В главе 5 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 с графическим интерфейсом для решения различных математических задач с данными Excel. Также рассмотрены некоторые особенности программирования на Visual Basic при использовании компонентов

Excel Builder и классов MWArray Компилятора MATLAB и справочные сведения по языку Visual Basic.

Глава 6 посвящена описанию работы с пакетом расширения MATLAB Production Server (MPS). Этот пакет позволяет запускать любые MATLAB-программы по сети, имея на локальной машине лишь установленное программное обеспечение MPS и среду исполнения MCR. Веб-приложения, базы данных и корпоративные приложения используют MATLAB-программы, которые исполняются на MATLAB Production Server через простую клиентскую библиотеку. Можно использовать MATLAB и MATLAB Compiler для создания своих приложений и внедрять их непосредственно на MATLAB Production Server без перекодирования или создания специальной инфраструктуры для управления ими. Схема работы MATLAB Production Server достаточно простая. Программа на MATLAB упаковывается с помощью Компилятора MATLAB и размещается на сервере MPS. Теперь к этой программе можно обратиться по сети из других приложений. Запрашиваемая программа будет выполняться на сервере MPS, что обеспечивает экономию машинных ресурсов.

На сайте «ДМК Пресс» (www.dmkpress.com) имеются приложения с исходными текстами примеров программ, рассматриваемых в данной книге.

При написании книги использовалось свободное программное обеспечение jdk1.8.0_25 и NetBeans 8.0.1, а также ознакомительные версии Visual Studio 2013 и MATLAB 2014a.



ГЛАВА 1.

Система компьютерной математики MATLAB

В данной главе мы рассмотрим начальные вопросы системы MATLAB – крупнейшей и старейшей системы компьютерной математики:

- общее описание системы MATLAB, инструментальные средства, типы данных MATLAB;
- основы работы с MATLAB: операции с массивами, решение математических задач, графика в MATLAB;
- программирование в MATLAB;
- создание графического интерфейса пользователя в MATLAB;
- совместная работа Microsoft Excel и MATLAB;
- массивы символов, ячеек и структур MATLAB.

Введение

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. напр. [В], [ККШ], [По], [ЧЖИ], [Кр], [Д], [ГЦ], [Ан] и [См2]. Отметим также Internet-ресурсы [MW].

Система MATLAB была разработана в конце 70-х гг. и широко использовалась на больших ЭВМ. В дальнейшем были созданы версии системы MATLAB для пер-

сональных 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), namного расширивших сферы применения системы. Пакеты расширений представляют собой обширные библиотеки функций MATLAB (m-файлов), которые созданы для использования MATLAB в решении специальных задач. Пакеты расширения (их число более 50) включают такие интересные 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. Общее описание 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, рабочее пространство и браузер для просмотра помощи.

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 допускает 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 для двумерной и трехмерной визуализации данных, обработки изображений, анимации, и презентационной графики, а также включает функции, которые позволяют полностью настроить вид графики и создавать законченные графические интерфейсы пользователя на ваших приложениях MATLAB.

MATLAB API для других языков (API – Application Program Interface, интерфейс прикладного программирования). Позволяет взаимодействовать с MATLAB и типами данных MATLAB из приложений на других языках; позволяет писать программы на C, C++ и Fortran для вызова их из MATLAB совместно с MATLAB. API включает следующие средства:

- MATLAB Engine API – вызов MATLAB из программ на C/C++ и Fortran;
- MATLAB COM Automation Server – вызов MATLAB из COM компонент и приложений;
- MAT-File API – чтение и запись данных MATLAB из программ на C/C++ и Fortran;
- MEX-File Creation API – создание MATLAB-функций их функций на C/C++ и Fortran;
- C/C++ Matrix Library API – написание программ на C/C++, которые работают с данными типа mxArray MATLAB;
- Fortran Matrix Library API – написание программ на Fortran, которые работают с данными типа mxArray MATLAB.

1.1.1. Инструментальные средства рабочего стола MATLAB

Начиная с выпуска R2012b существенно изменился интерфейс MATLAB. Вместо обычного меню и панели инструментов появилась лента инструментов (**HOME**) на главном окне MATLAB и добавлены еще две ленты-закладки: **PLOTS** – для выбора способов отображения графиков и **APPS** – это галерея приложений MATLAB. Кроме того, существенно изменилась справочная система MATLAB.

При запуске MATLAB, появляется *рабочий стол* MATLAB. Он содержит инструменты для управления файлами, переменными и приложениями, связанными с MATLAB. Рабочий стол MATLAB имеет вид как на рисунках. 1.1.1 и 1.1.2.

Инструментальные средства рабочего стола MATLAB включают следующие компоненты:

- меню ленты инструментов;
- командное окно (Command Window);
- браузер рабочей области (Workspace);
- история команд (Command History);
- браузер текущего каталога (Current Folder);
- редактор массива (Array Editor);
- редактор (MATLAB Editor);
- профилировщик (Profiler);
- окно для изображения графиков.

Меню ленты инструментов

Вверху справа (рис. 1.1.2) имеется обычное меню: «сохранить», «вырезать», «скопировать», «вставить», «отменить» и кнопка выбора активного окна. Рядом – строка поиска справки.

Основную верхнюю часть интерфейса MATLAB занимает лента инструментов из трех вкладок. Вкладка **PLOTS** предлагает выбрать удобные формы отображения графиков. Вкладка **APPS** содержит серию приложений в системе MATLAB. Основная вкладка **HOME** состоит из нескольких блоков меню: FILE, VARIABLE, CODE, ENVIRONMENT, RESOURCES (рис. 1.1.1 и 1.1.2).

Первый блок **FILE** состоит из пяти кнопок:

- **New Script** – открывает редактор для написания программы на m-языке MATLAB;
- **New** – создание нового документа MATLAB. Это может быть: скрипт, функция, пример, класс, системный объект, график, создание графического интерфейса пользователя, задание комбинации клавиш для быстрого вызова;
- **Open** – открыть имеющийся документ;
- **Find Files и Compare** – найти и сравнить файлы.

Второй блок **VARIABLE** также состоит из пяти кнопок:

- **Import Data** – для импорта данных в систему MATLAB;
- **Save Workspace** – сохранение данных из рабочего пространства;
- **New Variable** – открывает специальный редактор для создания новой переменной;
- **Open Variable** – открывает специальный редактор для просмотра и редактирования имеющейся в рабочем пространстве переменных;
- **Clear Workspace** – удаление переменных из рабочего пространства.

Третий блок **CODE** позволяет обратиться к анализатору кода, профилировщику и провести чистку командного окна и истории команд.

Четвертый блок **ENVIRONMENT** позволяет провести пользовательскую настройку системы:

- **Layont** – вид рабочего стола MATLAB;
- **Preferences** – открывает диалоговое окно для подробной настройки MATLAB;
- **Set Path** – установка путей для папок и файлов, которые предполагаются использовать в работе системы MATLAB;
- **Parallel** – некоторые возможности для параллельных вычислений.

Пятый блок **RESOURCES** – справки и поддержки:

- **Help** – справочная система MATLAB;
- **Community** – открывает сайт MATLAB Central для общения по вопросам системы MATLAB (<http://www.mathworks.com/matlabcentral/>);
- **Request Support** – запрос поддержки системы MATLAB;
- **Add Ons** – запрос на дополнительные возможности поддержки MATLAB (<http://www.mathworks.com>).

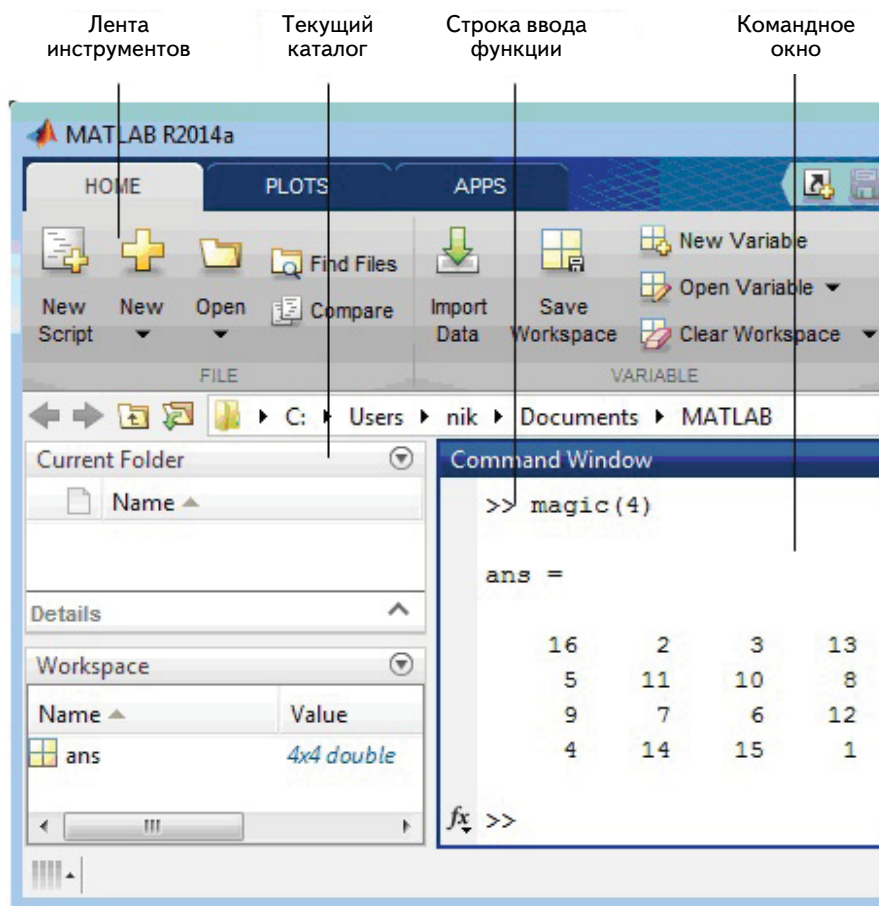


Рис. 1.1.1. Рабочий стол MATLAB (левая часть)

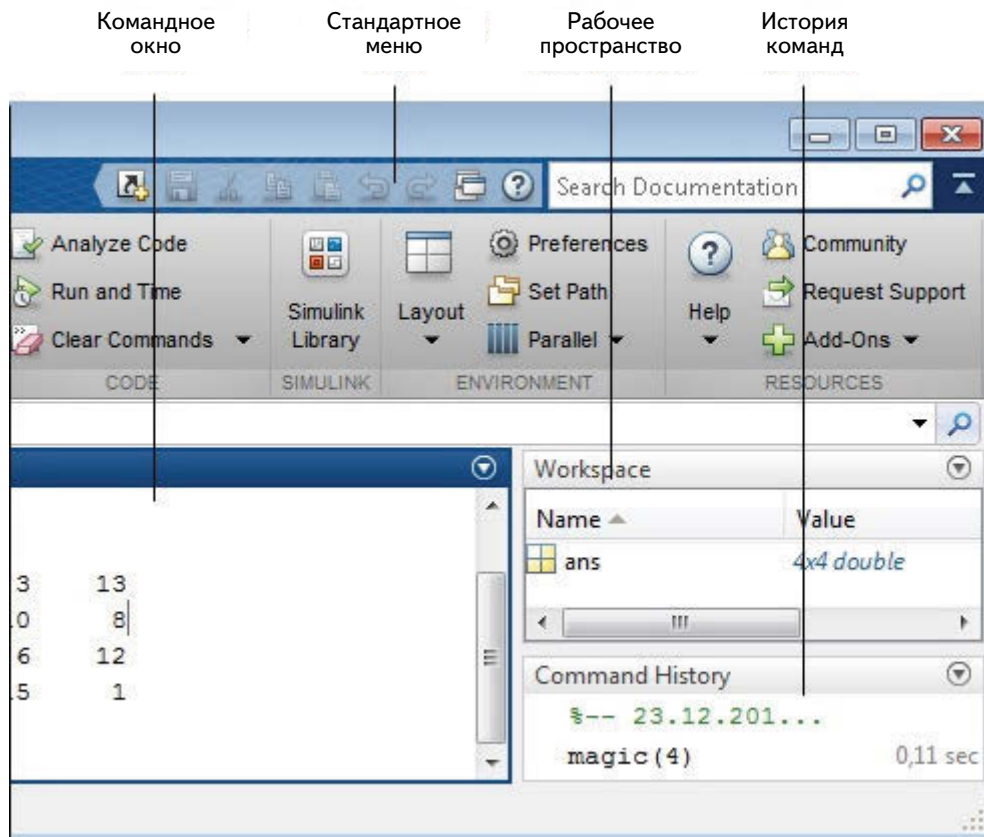


Рис. 1.1.2. Рабочий стол MATLAB (правая часть)

Окна, используемые в работе MATLAB

Ниже ленты инструментов расположена строка текущего (активного) каталога. Основную часть рабочего стола MATLAB занимают несколько окон. По умолчанию при загрузке MATLAB открываются четыре окна.

- **Командное окно (Command Window).** Основное окно MATLAB. 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 исполненные команды можно ввести в командную строку либо из окна истории команд, либо пролистывая их в командной строке клавишами «стрелка вверх/вниз».
- **Окно текущего каталога (Current Folder).** Обычно расположено слева. Используется для просмотра содержания текущего каталога. Позволяет

также менять каталог, искать файлы, открывать файлы и делать изменения, используя правую кнопку мыши. Внизу окна отображается информация о типе выбранного файла. Напомним, что 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 также использовать также строку текущего каталога.

- **Окно рабочего пространства (Workspace).** Обычно расположено справа (рис. 1.1.2). *Рабочее пространство* MATLAB состоит из набора переменных (массивов) созданных в течение сеанса MATLAB и сохраненных в памяти. Переменные добавляются к рабочей области в результате выполнения функций, m-файлов, или при загрузке сохраненных ранее рабочих областей. В рабочей области отображается информация о типе каждой переменной. Содержимое рабочего пространства 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.

Удалить переменные из рабочей области можно по кнопке **Clear Workspace** из ленты инструментов. Можно также выбрать переменную и воспользоваться открывающимся контекстным меню правой кнопки мыши. Чтобы сохранить рабочую область в файле, который может быть загружен в следующем сеансе MATLAB, достаточно выбрать **Save Workspace** из ленты инструментов, или использовать функцию `save`. Рабочая область сохраняется в бинарном MAT-файле. Чтобы прочитать данные из MAT-файла, нужно выбрать **Import Data** из ленты инструментов.

- **История команд (Command History).** Инструкции и команды, которые вводятся в 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`.

Замечание. Окна могут быть расположены пользователем по-своему. В частности, они могут быть вынесены с рабочего стола MATLAB в отдельные окна. Для этого достаточно навести указатель мыши на заголовок окна и правой кнопкой мыши открыть контекстное меню и выбрать тип расположения окна. Например, чтобы вынести окно как отдельное, достаточно в контекстном меню выбрать "Unlock".

Редактор массивов

Если дважды щелкнуть мышкой по переменной в рабочей области, то эта переменная отобразится в *редакторе массива* (рис. 1.1.3) Он используется для визуального просмотра и редактирования одно- или двумерных числовых массивов, массивов строк и массивов ячеек строк, которые находятся в рабочей области.

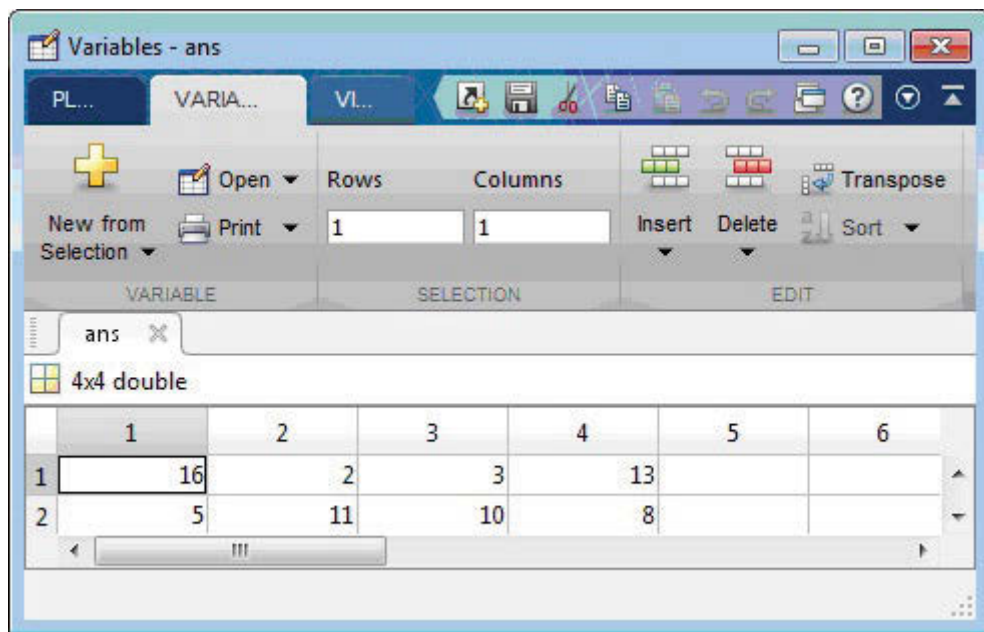


Рис. 1.1.3. Редактор массива

Редактор m-файлов (MATLAB Editor)

Он используется для создания и отладки m-файлов, т.е. программ, написанных на языке MATLAB. *Редактор-отладчик* представляет собой текстовый редактор с возможностями запуска и отладки программы m-файла. Редактор/отладчик вызывается либо из меню **New** или **Open** инструментальной панели MATLAB, либо двойным щелчком по соответствующему m-файлу. При этом он может быть либо встроенным в рабочий стол MATLAB, либо открыться в виде отдельного окна (рис. 1.1.4). Если он открывается на рабочем столе MATLAB, то к инструментальным панелям MATLAB добавляются еще две – для работы с редактором. В правом верхнем углу окна редактора есть кнопка **Show Editor Actions**, которая содержит возможность **Undock** переключения редактора в виде отдельного окна.

Основную верхнюю часть интерфейса редактора занимает лента инструментов из трех вкладок. Основная вкладка **EDITOR** позволяет открыть/сохранить m-файл, редактировать, запустить и имеет средства отладки. Вкладка **PUBLISH** содержит средства обработки файла для его публикации в формате html: выбор типа шрифта, гиперссылки, inline LaTeX, формулы в формате LaTeX, вставка рисунком и т. д. Вкладка **VIEW** дает возможности открыть несколько окон редактора и выбора организации текста программы.

Основная вкладка **EDITOR** состоит из нескольких блоков меню: FILE, EDIT, NAVIGATE, BREAKPOINTS, RUN (рис. 1.1.4).

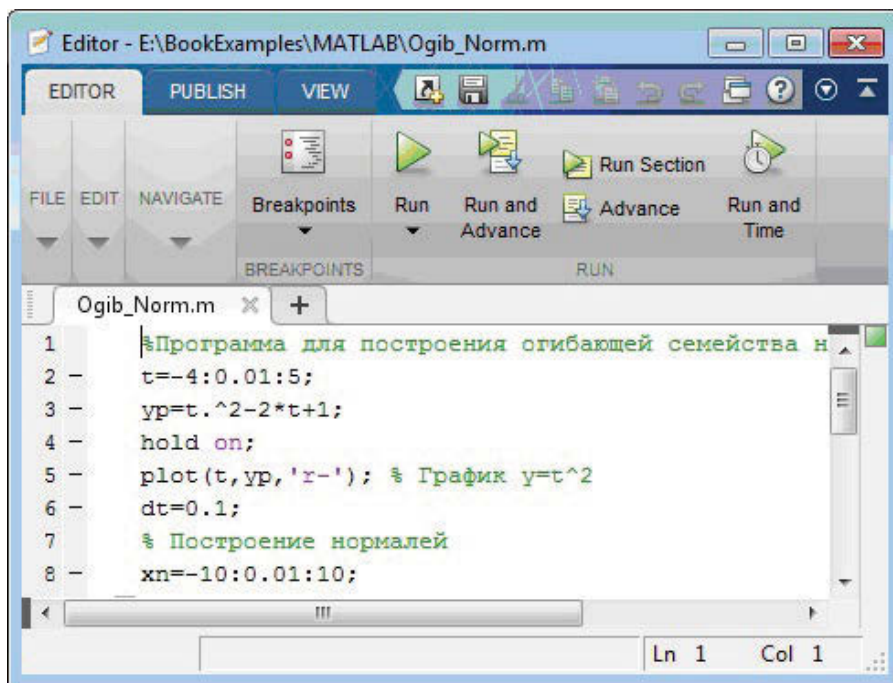


Рис. 1.1.4. Редактор-отладчик

Если в редакторе/отладчике открыт m-файл из текущего каталога, он может быть запущен в MATLAB прямо из редактора по (зеленой) кнопке **Run**. Эта кнопка записывает файл в текущий каталог и затем запускает его. Возможности редактора/отладчика достаточно большие, однако с ними лучше познакомиться практически, записывая и запуская m-файлы.

Анализатор кода (Code Analyzer)

Анализатор кода используется в редакторе m-файлов для проверки кода и минимизации ошибок при компиляции. Анализатор кода проверяет Ваш код на проблемы и рекомендует изменения. Анализатор кода работает постоянно, сообщает о проблемах и рекомендует изменения. По умолчанию отключен, но можно это проверить. Для этого нужно открыть **Preferences => Code Analyzer** на ленте инструментов **Home**. Для открытия интерфейса анализатора кода предусмотрена кнопка **Code Analyzer** на ленте инструментов **HOME** в блоке **CODE**.

Анализатор кода имеет индикатор в верхнем правом углу редактора. Если индикатор зеленый – анализатор не обнаружил проблем с кодом. Оранжевый цвет – предупреждения. Красный цвет – ошибки кода. Ошибки и предупреждения анализатора кода можно увидеть, если навести указатель мыши на цветные полоски в крайнем правом вертикальном поле редактора. Анализатор укажет на ошибки и предложит их исправить.

Профилировщик (Profiler)

Он представляет собой графический интерфейс пользователя, помогающий улучшать скорость работы m-файла. Для открытия профилировщика предусмотрена кнопка **Run and Time** на ленте инструментов **HOME** в блоке **CODE**. Либо, если Вы работаете в редакторе m-файлов, нажать кнопку **Run and Time** на ленте **EDITOR**. Откроется графический интерфейс профилировщика. Подробное описание можно найти в разделе **Help: MATLAB => Advances Software Development => Performance and Memory => Code Performance => Examples and How To => Profiling for Improving Performance**.

Окно для просмотра графиков

Если мы задали в командной строке построение графика, например, функции

$$f(x, y) = \frac{\sin(\rho + \text{eps})}{\rho + \text{eps}}, \quad \rho = \sqrt{x^2 + y^2},$$

```
>>ezsurf('sin((x^2+y^2)^(1/2)+eps)/((x^2+y^2)^(1/2)+eps)')
```

то график открывается в отдельном окне. Это окно имеет много возможностей. Укажем только некоторые из них: увеличение/уменьшение, вращение, определение координат точки на графике, палитра. Кроме того, все элементы графика допускают редактирование – для этого нужно перейти в режим редактирования. В меню **View** нужно включить **Figure Palette**, **Plot Browser** и **Property Editor** – это все можно также сделать, кнопкой справа, напоминающей окно и указанной на рис. 1.1.5 стрелкой.

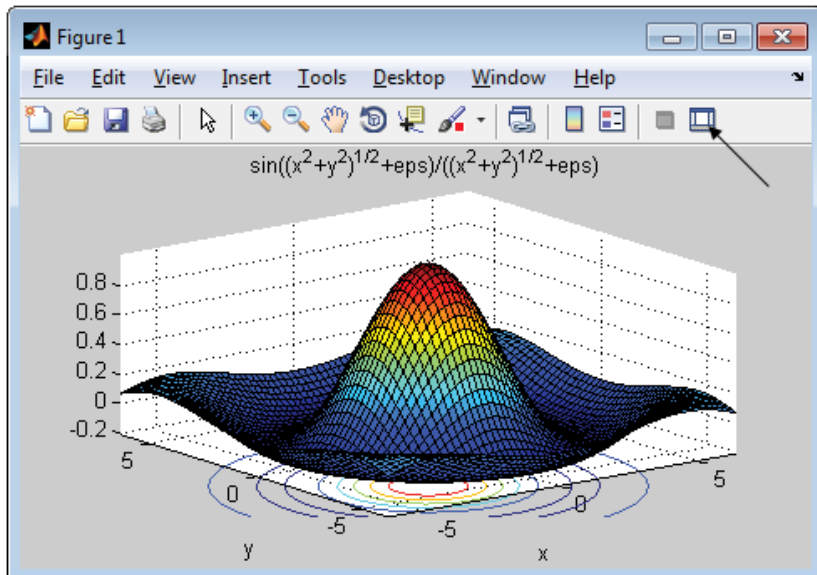


Рис. 1.1.5. Окно просмотра графиков

1.1.2. Справочная система MATLAB

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`.

В 2012 году изменился внешний вид MATLAB и, соответственно, существенно изменилась справочная система. Она стала более структурированной, однако стали недоступными непосредственно статьи, которые описывают базовые теоретические аспекты процедур MATLAB. Такие концептуальные статьи можно теперь найти при изучении справки по какой-либо функции MATLAB. Справка стала в большей степени использовать интернет-ресурсы компании MathWorks: <http://www.mathworks.com/support/2014a/matlab/>.

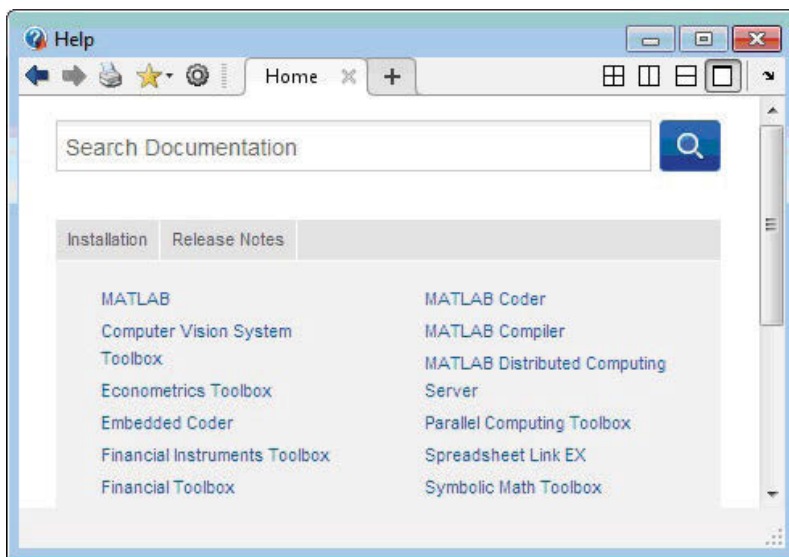


Рис. 1.1.6. Help-браузер MATLAB

Дадим краткое описание справочной системы Help MATLAB.

На верхней панели Help MATLAB имеются стрелки навигации, кнопка настройки **Preferences**, панель вкладок с кнопкой «+» – для задания новой вкладки и несколько возможностей выбора типа отображения основного поля. Ниже идет строка поиска. Еще ниже – адресная строка справки с кнопкой **Home** – для выхода в основное меню справки. Слева – поле **Contents** – для отображения содержания активной вкладки. Основное окно справки имеет три вкладки: **Getting Started**, **Examples** и **Release notes**, назначение которых не требует пояснений (рис. 1.1.7).

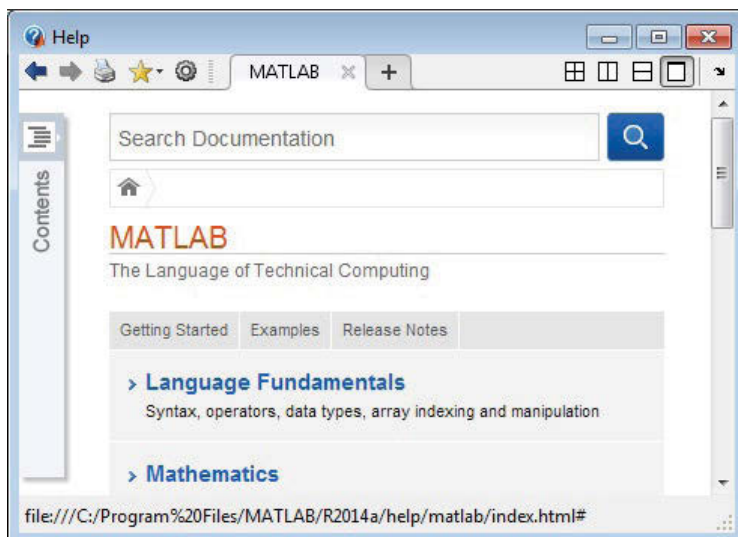


Рис. 1.1.7. Help-браузер MATLAB. Раздел MATLAB

При выборе, например первой закладки **Getting Started** вначале идет основная краткая справка, а более подробные учебные руководства собраны ниже под заголовком **Tutorials**.

Внизу Help-браузера расположены еще две кнопки: **Functions** и **PDF Documentation**. (рис. 1.1.8). Первая кнопка выводит список функций, относящихся к данному разделу справки, а вторая – дает возможность зарегистрированным пользователям обратиться к PDF-документации на сайте компании MathWorks.



Рис. 1.1.8. Help-браузер MATLAB

Возможен прямой доступ к файлам документации MATLAB. Для этого достаточно открыть каталог справки интересующего Вас раздела, например: `User:\MATLAB\help\wavelet` и в нем открыть файл `index.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
      This MATLAB function returns an n-by-n matrix constructed from
      the integers 1 through n^2 with equal row and column sums.
M = magic(n)
Reference page for magic      %Это ссылка на документацию
See also ones, rand          %Это ссылка на документацию ones и rand
```

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
ifft      - Inverse discrete Fourier transform.
ifft2     - Two-dimensional inverse discrete Fourier transform.
ifftn     - N-dimensional inverse discrete Fourier transform.
ifftshift - Inverse FFT shift.
acos      - Inverse cosine, result in radians.
acosd     - Inverse cosine, result in degrees.
acosh     - Inverse hyperbolic cosine.
acot      - Inverse cotangent, result in radian.
acotd     - Inverse cotangent, result in degrees.
```

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 Интернет-адрес и телефон для контакта с фирмой MathWorks;

- **ver** – выводит информацию о версии установленной системы MATLAB и об установленных пакетах расширений;
- **version** – выводит краткую информацию об установленной версии MATLAB;
- **what name** – выводит имена файлов каталога, заданного именем name;
- **which name** – выводит путь доступа к функции с данным именем;
- **help demos** – выводит ссылку на Help-браузер для примеров;
- **bench** – тест на быстродействие компьютера. Результаты теста представляются в виде таблицы и диаграммы сравнения с другими типами компьютеров.

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,
 $\text{realmin} = 2.225073858507202e-308$, или 2^{-1022} ;
- **realmax** – наибольшее число с плавающей точкой,
 $\text{realmax} = 1.797693134862316e+308$, 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 17 основных типов данных (или классов). Каждый из этих типов данных находится в форме массива, вообще говоря, многомерного. Все основные типы данных показаны на рис. 1.1.9. Дополнительные типы данных *user classes* и *java classes* могут быть определены пользователем как подклассы структур, или созданы из классов Java.

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.

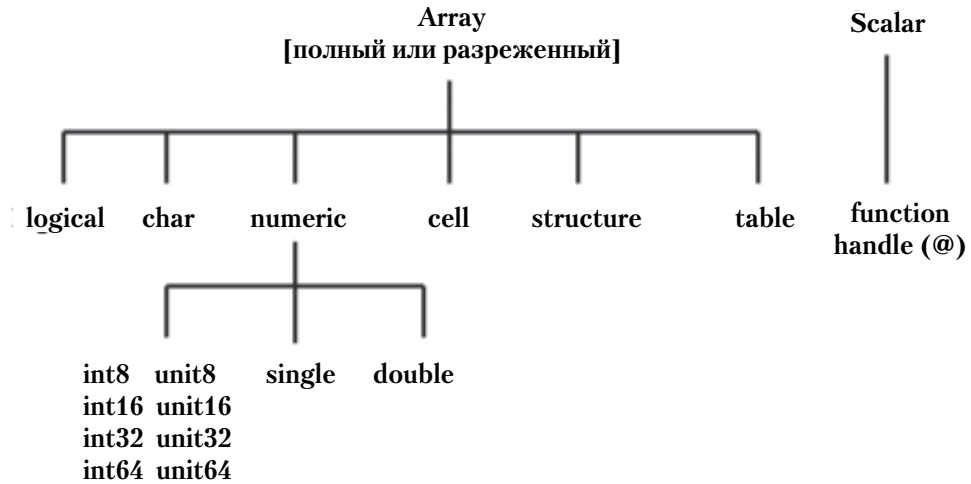


Рис. 1.1.9. Типы данных MATLAB.

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` и `string`. 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;