



第 1 章 MATLAB 入门 /1

1.1	MATLAB 简介	1
1.1.1	MATLAB 的由来	1
1.1.2	MATLAB 的发展	1
1.1.3	MATLAB 语言的特点	2
1.1.4	MATLAB 的组成	6
1.1.5	Simulink 简介	8
1.2	MATLAB 的开发环境	9
1.2.1	开发环境概览	9
1.2.2	命令窗口	10
1.2.3	工作空间	13
1.2.4	历史命令	14
1.2.5	工作目录	14
1.2.6	编辑器	15
1.2.7	设置路径	15
1.3	MATLAB 的在线帮助	17
1.3.1	利用命令窗口获得帮助	17
1.3.2	利用帮助浏览器获得帮助	18

第 2 章 MATLAB 的数据与数据类型 /21

2.1	数据的基本概念	21
2.1.1	标量与向量	21
2.1.2	数组与矩阵	21
2.1.3	实数与复数	21
2.1.4	常量与变量	22
2.1.5	变量的命名规则	22
2.2	数据的类型	23
2.2.1	使用数据时的注意事项	23
2.2.2	数值型	26
2.2.3	字符型	30



2.2.4	逻辑型	35
2.2.5	结构体型	38
2.2.6	元胞型	39
2.3	数据类型的转换	40
2.3.1	数值型数据之间的类型转换	40
2.3.2	非负整数的进制转换	41
2.3.3	数值型与字符型数据的转换	42
2.3.4	元胞型与其他数据类型的转换	42

第 3 章 数组、矩阵及其运算 /44

3.1	数组的创建	44
3.1.1	数组的创建	44
3.1.2	特殊数组的创建	45
3.1.3	数组的大小	47
3.1.4	数组的访问	47
3.1.5	数组的组合、扩充、拆分与重组	48
3.2	数组的运算	51
3.2.1	关系运算	51
3.2.2	逻辑运算	52
3.2.3	数组的四则运算	53
3.2.4	数组的集合运算	54
3.3	矩阵的运算	55
3.3.1	加减运算	55
3.3.2	乘法运算	55
3.3.3	除法运算	58
3.3.4	乘方运算	59
3.4	矩阵的求值运算	60
3.4.1	计算矩阵的行列式值	60
3.4.2	计算矩阵的逆与伪逆	61
3.4.3	计算矩阵的秩	61
3.4.4	计算矩阵的迹	61
3.4.5	计算向量和矩阵的范数	62
3.4.6	计算矩阵的条件数	63
3.4.7	计算矩阵的最大无关组	64
3.4.8	计算矩阵的特征值与特征向量	64
3.4.9	计算矩阵的正交基	65
3.5	矩阵的特殊运算	65
3.5.1	矩阵的转置	65



3.5.2	矩阵的旋转	66
3.5.3	抽取对角线元素	67
3.5.4	抽取上下三角阵	68
3.6	特殊矩阵	68
3.6.1	希尔伯特矩阵	68
3.6.2	范德蒙德矩阵	69
3.6.3	特普利茨矩阵	69
3.6.4	伴随矩阵	70
3.6.5	帕斯卡矩阵	70
3.6.6	稀疏矩阵	71
3.7	矩阵的分解	73
3.7.1	三角分解	73
3.7.2	奇异值分解	74
3.7.3	正交分解	75
3.7.4	Cholesky 分解	76
3.7.5	Schur 分解	76
3.7.6	特征值分解	77

第 4 章 MATLAB 的程序设计 /78

4.1	程序控制结构	79
4.1.1	顺序结构	79
4.1.2	选择结构	80
4.1.3	循环结构	85
4.2	程序的流程控制	87
4.2.1	continue 命令	87
4.2.2	break 命令	88
4.2.3	return 命令	88
4.3	try-catch 结构	89
4.4	M 文件	90
4.4.1	M 文件的命名规则	90
4.4.2	脚本文件	90
4.4.3	函数文件	91
4.5	函数类型	91
4.5.1	主函数	92
4.5.2	子函数	92
4.5.3	嵌套函数	93
4.5.4	私有函数	96
4.5.5	重载函数	98



4.5.6	匿名函数	98
4.5.7	利用全局变量传递参数	99
4.6	程序的优化	100
4.6.1	用数组运算取代循环	100
4.6.2	尽量调用 MATLAB 函数	101
4.6.3	使用循环时的注意事项	102
4.7	实例分析	103

第 5 章 MATLAB 的数值计算 /108

5.1	多项式的计算	108
5.1.1	生成多项式	108
5.1.2	多项式的加减乘除	109
5.1.3	多项式的求导	109
5.1.4	多项式的求值	110
5.1.5	多项式的求根	111
5.2	插值计算	111
5.2.1	一维数据的插值	112
5.2.2	二维数据的插值	113
5.2.3	一般分布的二维数据插值	117
5.3	曲线拟合	118
5.3.1	多项式拟合	118
5.3.2	最小二乘拟合	119
5.4	数据分析和统计	122
5.4.1	基本的数据分析和统计	122
5.4.2	协方差和相关系数	129
5.5	概率统计	130
5.5.1	随机变量的分布与数字特征	130
5.5.2	参数估计	133
5.5.3	假设检验	134
5.5.4	方差分析	136
5.6	数值积分	138
5.6.1	一元函数的数值积分	138
5.6.2	二元函数的数值积分	139
5.7	数值微分	140
5.8	方程的解	141
5.8.1	线性方程组的解	141
5.8.2	非线性方程的解	143
5.8.3	非线性方程组的解	144



5.8.4	无约束最优化问题	145
5.8.5	有约束最优化问题	147

第 6 章 MATLAB 的文件操作 /148

6.1	对 Excel 数据表和 Lotus 1-2-3 数据表的操作	148
6.1.1	对 Excel 数据表的操作	148
6.1.2	对 Lotus 1-2-3 数据表的操作	152
6.2	对图像文件的操作	153
6.2.1	检查图像文件的相关信息	154
6.2.2	读取图像文件	154
6.2.3	保存图像文件	155
6.3	对音频和视频文件的操作	156
6.3.1	检查音频和视频文件的相关信息	157
6.3.2	读取音频和视频文件	158
6.3.3	保存音频和视频文件	159
6.4	对 ASCII 文件的操作	160
6.4.1	读取 ASCII 文件	161
6.4.2	写入 ASCII 文件	165
6.5	对 MAT 文件的操作	166
6.5.1	读取 MAT 文件	166
6.5.2	写入 MAT 文件	167
6.6	用低端的 I/O 函数操作文件	167
6.6.1	打开和关闭文件	168
6.6.2	逐行读取文本文件	169
6.6.3	读取二进制文件	170
6.6.4	写入二进制文件	171
6.6.5	控制文件句柄	171

第 7 章 MATLAB 的绘图 /173

7.1	基本二维绘图	173
7.2	多图绘制与图形修饰	177
7.2.1	多图绘制	177
7.2.2	图形属性的设置	181
7.2.3	图形的辅助说明	183
7.2.4	设置坐标轴网格与坐标框	187
7.3	特殊二维绘图命令	189
7.3.1	绘制区域图	190
7.3.2	绘制条形图	192



7.3.3	绘制罗盘图	193
7.3.4	绘制彗星图	194
7.3.5	绘制等高线图	194
7.3.6	绘制误差棒图	195
7.3.7	绘制速度向量图和向量场图	196
7.3.8	绘制直方图和饼图	197
7.3.9	绘制填充图和伪彩色图	198
7.3.10	绘制火柴杆图和阶梯图	199
7.3.11	绘制极坐标图和对数坐标图	200
7.3.12	ezplot 和 fplot 绘图	201
7.4	三维绘图命令	204
7.4.1	派生的三维绘图命令	204
7.4.2	mesh 三维网格绘图	205
7.4.3	surf 三维曲面绘图	207
7.4.4	标准三维曲面命令	209
7.5	三维绘图的修饰	210
7.5.1	改变视角	210
7.5.2	设置光源	213
7.5.3	设置色图	214

第 8 章 图形用户界面的设计与实现 /217

8.1	图形对象基础	218
8.1.1	图形对象	218
8.1.2	图形对象句柄	219
8.1.3	图形对象属性的获取与设置	220
8.2	GUI 基础	224
8.2.1	启动 GUI	224
8.2.2	Layout 编辑器	224
8.2.3	保存并运行 GUI	226
8.3	设计 GUI: 曲线绘制及属性修改系统	226
8.3.1	任务 1-1: 使用菜单编辑器进行 GUI 设计	226
8.3.2	任务 1-2: 为设计好的 GUI 编写回调函数	229
8.3.3	任务 1-3: 使用文本框、编辑框和按钮实现用户输入参数	234
8.3.4	任务 1-4: 使用 Slider 实现用户输入参数	237
8.3.5	任务 1-5: 在不同的坐标轴上绘图	237
8.3.6	任务 1-6: 利用 Check Box 控件实现加网格和去边框功能	240
8.4	设计 GUI: 学生成绩查询系统	242
8.4.1	MATLAB 预定义的对话框	242



8.4.2	任务 2-1: 使用 uigetfile 获取学生成绩表的路径和文件名	242
8.4.3	任务 2-2: 使用 Listbox 显示学生姓名	244
8.4.4	任务 2-3: 使用 Pop-up Menu 显示课程名单	245
8.4.5	任务 2-4: 从列表框中选择要查看的学生姓名	246
8.4.6	任务 2-5: 在表控件中显示查询结果	248
8.4.7	任务 2-6: 使用图标设计器设计图标	250
8.4.8	任务 2-7: 使用工具栏实现保存文件功能	251
8.5	设计 GUI: 使用 ActiveX 控件	252

第 9 章 Simulink 概述 /255

9.1	启动 Simulink	255
9.2	模型窗口介绍	256
9.3	建立模型	257
9.3.1	选择模块	257
9.3.2	模块的操作	257
9.3.3	信号线的操作	261
9.3.4	对模型进行注释	263
9.3.5	模型的缩放	263
9.4	运行模型	264
9.5	查看仿真结果	264
9.5.1	示波器参数的设置	264
9.5.2	信号的组合	265
9.6	向量和矩阵	268
9.7	保存和打开已有模型	269
9.8	处理大的模型	269
9.9	常用的仿真参数配置	270
9.9.1	Solver 求解器	271
9.9.2	Solver 选项卡的参数设置	272
9.9.3	Data Import/Export 选项卡的参数设置	275
9.9.4	Optimization 选项卡的参数设置	276
9.9.5	Diagnostics 选项卡的参数设置	277

第 10 章 常用模块库及数据的导入导出 /279

10.1	Simulink 基本库简介	279
10.2	Sources 信源库与外部数据的导入	280
10.2.1	Sources 信源库介绍	280
10.2.2	外部数据的导入	282
10.2.3	从工作区导入模块参数	286



10.3	Sinks 信宿库与仿真数据的导出	288
10.3.1	Sinks 信宿库介绍	288
10.3.2	Floating Scope 悬浮示波器	289
10.3.3	仿真数据的导出	291
10.4	User-Defined Functions 库	293
第 11 章 动态系统的仿真与分析 /296		
11.1	连续系统	296
11.1.1	连续模块库介绍	296
11.1.2	Integrator 积分器	297
11.1.3	采用积分器进行连续系统的仿真	300
11.1.4	采用传递函数进行连续系统的仿真	302
11.1.5	采用零-极点模块进行连续系统的仿真	304
11.1.6	采用状态方程进行连续系统的仿真	305
11.1.7	4 种连续系统仿真方法的比较	306
11.2	离散系统	307
11.2.1	离散模块库介绍	307
11.2.2	Unit Delay 单位延迟模块	308
11.2.3	采用 Unit Delay 进行离散系统的仿真	309
11.2.4	采用传递函数进行离散系统的仿真	310
11.2.5	采用状态空间进行离散系统的仿真	311
11.2.6	零阶保持器	311
11.2.7	多速率的离散系统	312
11.3	命令行仿真	313
11.3.1	使用 sim 命令对系统进行仿真	313
11.3.2	获取和设置仿真参数	316
11.3.3	确定模型的状态	318
11.3.4	寻找模型的平衡点	319
11.3.5	模型的线性化	322
第 12 章 子系统 /325		
12.1	子系统的建立与基本操作	325
12.1.1	建立子系统	325
12.1.2	子系统的基本操作	328
12.2	封装子系统	328
12.2.1	封装子系统	328
12.2.2	使用封装子系统	337
12.2.3	修改封装子系统	338



12.3	模块库技术	338
12.3.1	建立模块库	338
12.3.2	库模块与引用块的关联	339
12.3.3	在库浏览器中加入自建模块库	342
12.4	Simulink 自带的子系统模块库	344
12.4.1	Enable 使能子系统	344
12.4.2	Trigger 触发子系统	345
12.4.3	触发使能子系统	346
12.4.4	Switch Case 子系统和 Switch Case Action Subsystem 子系统	349
12.4.5	If 系统和 If Action Subsystem 子系统	352

MATLAB 是一种用于科学计算的高级语言,它将高性能的数值计算和方便的可视化集成在一起,并提供大量的专业领域函数,使其在诸如数字信号处理、系统辨识、自动控制、优化设计、神经网络、图像处理、符号计算、动态仿真、统计学、通信等领域表现出一般高级语言难以比拟的优势,因此广受科研人员、教师和学生的喜爱。

本章将对 MATLAB 进行概述,使读者对 MATLAB 有一个初步的认识,为深入学习打下基础。

1.1 MATLAB 简介

1.1.1 MATLAB 的由来

MATLAB 这个名字是由 Matrix(矩阵)和 Laboratory(实验室)两词的前三个字母组合而成的,含有矩阵实验室的意思。20 世纪 80 年代前后,时任美国新墨西哥大学计算机系主任的 Cleve Moler 博士在讲授线性代数课时发现,即使是用当时最先进的 EISPACK 和 LINPACK 软件包求解线性代数问题,其求解过程也很烦琐。为减轻学生的编程负担,他和同事用 FORTRAN 语言编写了一组调用 EISPACK 和 LINPACK 库程序的“通俗易懂”的接口,这就是处于萌芽状态的 MATLAB。以后几年,MATLAB 作为免费软件在大学里被广泛使用,深受师生欢迎。

1.1.2 MATLAB 的发展

1984 年,Cleve Moler 教授与其曾经的学生 John Little 等人联合起来成立了 MathWorks 公司,专门从事 MATLAB 软件的开发,并把 MATLAB 正式推向市场。从此时起,MATLAB 的内核开始采用 C 语言进行编写。通过不断推陈出新,MATLAB 已从开始时期的简单数值计算工具,逐步发展成为具有强大的科学计算能力、能够图视交互的高效率的高级程序设计语言。以下是关于 MATLAB 的一些重要时间点:

1984 年,推出 MATLAB 1.0 版。

1993 年,推出 MATLAB 4.0 版,从此告别 DOS 版,并新推出 Simulink。

1997 年,推出 MATLAB 5.0 版。

2000 年,推出 MATLAB 6.0 版。

2002 年,推出 MATLAB 6.5 版,从此拥有强大的、成体系的交互式界面。

2004 年,推出 MATLAB 7.0 版,Simulink 发展到 6.0 版。

2006 年,推出 MATLAB 7.2(R2006a),在技术层面上又一次实现飞跃,从此产品的发布模式也发生了改变,即分别在每年的 3 月和 9 月进行两次产品发布,版本的命令方式为“R+年份+代码”,其中对应于上、下半年的代码分别是 a 和 b。每次产品发布都会包含所有的产品模块,如模块的更新升级、Bug 修订以及推出新产品等。

2008 年,增强了面向对象的编程功能,能以更快的速度开发复杂的科学计算应用程序。

2009 年,从 R2009a 开始,引入了 License Center——在线 License 管理工具, MATLAB 和 Simulink 产品家族软件需要在安装后进行激活才能使用。

2015 年, MATLAB 的最新版本是 R2015a。

值得一提的是,在学习和使用 MATLAB 时,并不一定需要安装最新版本。事实上, MATLAB 软件在操作性、交互性等方面每隔三四年才会有一次质的飞跃,其余时段更多的是在技术上和算法上增加或修正一些现有的工具箱,版本之间的差异并不大。虽然本教材使用的是 MATLAB2014a 的中文版,但考虑到大多数的 MATLAB 版本以英文版居多,故为了方便读者学习,本教材在功能介绍上仍以英文版为基础,在不影响读者理解的情况下,对命令和功能的介绍仍以其英文名字为主。例如,用 Desktop 代替“布局”、用 Editor 代替“编辑器”、用 Command Window 代替“命令行窗口”等。

1.1.3 MATLAB 语言的特点

MATLAB 之所以能被迅速地普及、显示出强大的生命力,是因为它有着不同于其他语言的特点:

(1) MATLAB 规则简单,更贴近人的思维方式。

(2) MATLAB 以矩阵(数组)为基本单位,其命令形式与数学、工程中常用的习惯形式十分相似,因此被称为“演算纸式的”科学计算语言。

(3) MATLAB 提供大量的专业工具箱函数,把编程人员从烦琐的程序代码中解放出来,以便让他们更专注于系统设计和算法优化。

1. 高效方便的矩阵和数组运算

与 C/C++、Java 等基于数值运算的编程语言不同, MATLAB 是基于矩阵(数组)进行运算的,一次运算就可以完成其他语言需要重复多次才能完成的工作。由于提供了丰富的矩阵函数,使之在解决数字信号处理、系统建模、自动控制、统计学等领域的问题时显得简洁、高效,具有其他高级语言不可比拟的优势。

假设有一个矩阵 A,要求计算它的行列式值和逆矩阵。由于 C/C++ 或 Java 并没有直接提供相应的函数,再加上计算行列式值和逆矩阵需要一定的数学功底,因此,即使是优秀的编程人员,也需要花费相当的时间和精力才能完成工作。而在 MATLAB 中,求解上述问题则变得非常简单,只需要三个语句就可以实现。

【例 1-1】 本例的功能是展示 MATLAB 在矩阵运算中的优势。

```
>>A=[1 3 5;2 6 8;2 5 1];
```

```
%第一步:为矩阵 A 赋值,此处 A 为 3×3 的矩阵
```

```
%注意本语句以";"结尾,表示不显示本语句的执行结果
```

```

>>B=det(A)                                %第二步:用MATLAB提供的求行列式函数det计算行列式值
                                           %注意本语句没有以";"结尾,表示要显示本语句的执行结果

B=
    -2

>>C=inv(A)                                %第三步:用MATLAB提供的求逆函数inv计算逆矩阵

C=
    17.0000   -11.0000    3.0000
    -7.0000    4.5000   -1.0000
     1.0000   -0.5000    0

```

事实上,许多复杂的数学运算,诸如求特征值、矩阵分解、函数积分、曲线拟合、目标优化等,都有现成的库函数可以调用。更重要的是, MATLAB 具有严格的解题规则,能够根据不同的应用情况采用不同的优化算法,从而保证了结果的可靠性和求解的快速性。

2. 面向对象进行编程

具有面向对象编程的特点,尤其在制作图形用户界面(GUI)方面简单快捷。

3. 功能强大的工具箱

MATLAB 工具箱包括两个部分:核心工具箱和可选工具箱。其中,核心工具箱含有数百个内部函数,它们又可分为功能性工具箱和学科性工具箱。功能性工具箱主要用来扩充符号计算、Simulink 仿真、图形处理和实时交互等功能,其函数可用于多个学科;学科性工具箱的专业性更强,如图像处理工具箱(Image Processing Toolbox)、控制系统工具箱(Control System Toolbox)、金融工具箱(Financial Toolbox)、系统辨识工具箱(System Identification Toolbox)等,而且这些工具箱都是由该领域内的高水平专家编写的,用户可以直接使用它们进行研究。

4. 方便的绘图功能

使用 MATLAB 进行绘图非常方便,它的高层绘图命令简单明了、容易掌握;底层绘图命令能更灵活地控制和显示数据图形。

【例 1-2】 本例的功能是利用 plot 函数绘制曲线。

```

>>x=0:0.1:10;                            %x 从 0 到 10 均匀取值,间隔为 0.1
>>y=sin(x);                               %sin 正弦函数
>>z=cos(x);                               %cos 余弦函数
>>plot(x,y,'r-',x,z,'g--');              %利用 plot 命令绘制曲线,并设置两条曲线的参数
>>title('绘制正弦和余弦图');             %为图形设置标题
>>legend('sinx','cosx');                  %为图形设置图例
>>grid on;                                %加网格

```

程序的最终绘图结果如图 1-1 所示。

另外,利用曲面绘制函数 surf 和罗盘绘制函数 compass 可以快速地绘制曲面或罗盘箭头图形(如图 1-2 和图 1-3 所示)。

以上只是初步展示了 MATLAB 的绘图功能,在后面的章节中还会对其进行详细的介绍。


```

Undefined function 'cod' for input arguments of type 'double'.
>>C=cos(A)           %显示本语句的执行结果
C=
-0.9900

```

6. 强大的扩充能力

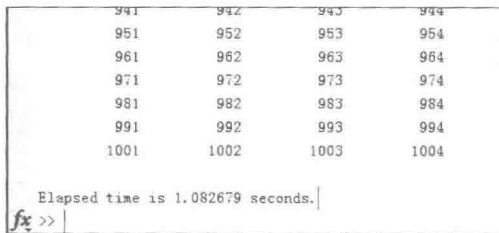
MATLAB有丰富的工具箱函数,而这些函数的源码都是开放的,用户可以对源文件进行修改,也可以将其加入自己的文件中。

此外,用户自己编写的函数与MATLAB的库函数在形式上是一样的,因此用户函数可以作为MATLAB的库函数来调用。用户完全可以根据自己的需要,方便地建立新的库函数或扩充原有的库函数,以提高使用MATLAB的效率。

最后,MATLAB可以通过建立MEX文件的方式与C/C++、Java、FORTRAN等进行混合编程,而这种编程方式已逐渐成为流行趋势。

7. 较弱的执行效率

MATLAB的执行效率要比一般高级语言低。究其原因,既有用户的编程习惯问题也有MATLAB自身的问题。MATLAB提供了灵活的数据使用方式、交互式编程方式、中间结果的查看方式,而这些必然会为解释和执行工作带来挑战。例1-4-1是一个双重循环的小程序,如果利用C语言编写,只需要微秒级的执行时间,但MATLAB却执行了1s的时间。仅针对本例而言,最主要的原因是C语言的程序只能通过调用输出函数显示运行结果,而MATLAB只要在语句后面没有“;”,就会在命令窗口中显示该语句的执行结果。程序循环了1000次,就要显示1000次中间结果,整个程序的运行时间当然会变长(如图1-4所示)。为验证此假设,例1-4-2取消了中间结果的显示,运行时间果然大大缩短(如图1-5所示)。



```

951    952    953    954
951    952    953    954
961    962    963    964
971    972    973    974
981    982    983    984
991    992    993    994
1001   1002   1003   1004

Elapsed time is 1.082679 seconds.
fx >>

```

图 1-4 例 1-4-1 的程序运行结果截图



```

>> chl_4
Elapsed time is 0.000368 seconds.
fx >>

```

图 1-5 例 1-4-2 的程序运行结果截图

【例 1-4-1】 本例的功能是测试 MATLAB 的执行效率低的原因之一。

```

clear all           %本语句是清除工作空间中的所有变量的常用语句之一
tic                %计时开始,tic与toc结合使用,可对它们之间的程序运行时间进行计时
for i=1:100        %外层for循环,i从1变到100,步长为1,即循环100次
    for j=1:10     %内层for循环,j从1变到10,步长为1,即循环10次
        A(i,j)=10*i+j %此处没有";",表示要显示中间结果
    end           %结束内层的for循环
end              %结束外层的for循环
toc              %计时结束

```

程序在命令窗口的运行结果截图如图 1-4 所示,运行时间为 1.08s 左右。

【例 1-4-2】 本例的功能是验证 MATLAB 执行效率低的原因之一。

```
clear all
tic                %计时开始
for i=1:100
    for j=1:10
        A(i,j)=10 * i+j; %不显示中间结果
    end
end
toc                %计时结束
```

与例 1-4-1 相比,例 1-4-2 仅多了一个“;”,即不显示语句的执行结果。由此可见,显示中间运行结果是极耗时间的,因此除非有必要,否则尽量不要显示中间结果,尤其是在循环过程中显示中间结果,避免增加程序运行时间。

当然,除此之外,还有其他一些需要注意的问题,相关内容将在后面详细介绍。

1.1.4 MATLAB 的组成

一般认为 MATLAB 主要由 5 大部分组成,分别是 MATLAB 语言、桌面工具与开发环境、数学函数库、图形系统和应用程序接口。笔者认为还要增加一部分,就是众多的专业工具箱,正是它们让编程工作变得更加简单和高效。

1. MATLAB 语言

MATLAB 语言是一种基于矩阵(数组)的高级语言,具有面向对象、程序流控制、函数、数据结构和输入/输出等功能。小到单纯的算法设计,大到复杂的应用系统开发, MATLAB 语言都能轻松地胜任。

2. 桌面工具与开发环境

简单地说,桌面工具和工作环境就是一系列实用工具的集合,例如命令窗口、编辑器、调试器、代码分析器、帮助浏览器等,它们能帮助用户更快地编写函数或文件。而且这些工具大多是基于图形用户界面(GUI)的,使用起来非常方便。

3. 数学函数库

数学函数库由大量的科学计算函数组成,不仅包含了最基本的初等函数,如 sum、sine 和复数计算等,还包含了复杂的高级函数,如矩阵求逆、特征值计算和快速傅里叶变换等。以 MATLAB2014a 版为例,数学函数库按类别分别存放在 R2014a\toolbox\matlab 目录下的 8 个子目录中,如表 1-1 所示。

4. 图形系统

图形系统不仅包含了大量高级的 2D 或 3D 的数据可视化、图像处理、动画生成和图形显示等命令,还包含了许多低级的图形命令,能让用户按照意愿定制图形的样式或者为应用程序定制图形用户界面。具体的函数分为 5 类,分别放置在 R2014a\toolbox\matlab 目录下的 5 个子目录中,如表 1-2 所示。

表 1-1 MATLAB 数学函数库的分类

目录名	函数功能	目录名	函数功能
datafun	数值分析和傅里叶变换等	matfun	矩阵函数、数值线性函数
elfun	初等数学函数	polyfun	插值和多边形近似
elmat	对矩阵和矩阵元素的操作	sparfun	稀疏矩阵函数
funfun	功能函数和 ODE 求解	specfun	专门数学函数

表 1-2 MATLAB 图形系统的分类

目录名	函数功能	目录名	函数功能
graph2d	二维图形函数	specgraph	专门图形函数
graph3d	三维图形函数	uitools	图形用户界面工具
graphics	图形句柄函数		

5. 应用程序接口

应用程序接口可以让 MATLAB 与外部设备或程序(例如 C/C++、Java、FORTRAN 程序等)进行数据交换和程序移植,既能弥补其执行效率较低的弱点,也能增强其他应用程序进行软件开发的能力,从而提高软件开发效率。

6. 专业工具箱

MATLAB 提供四十多个专业工具箱,分别涵盖了数据获取、科学计算、控制系统设计与分析、数字信号处理、数字图像处理、金融财务分析以及生物遗传工程等专业领域。表 1-3 是一些常用的工具箱。

表 1-3 MATLAB 的常用工具箱

工具箱英文名称	工具箱中文名称
Matlab Main Toolbox	MATLAB 主工具箱
Communication Toolbox	通信工具箱
Control System Toolbox	控制系统工具箱
Financial Toolbox	财政金融工具箱
Fuzzy Logic Toolbox	模糊逻辑工具箱
Higher-Order Spectral Analysis Toolbox	高阶谱分析工具箱
Image Processing Toolbox	图像处理工具箱
LMI Control Toolbox	线性矩阵不等式工具箱
Model predictive Control Toolbox	模型预测控制工具箱
μ -Analysis and Synthesis Toolbox	μ 分析工具箱
Neural Network Toolbox	神经网络工具箱