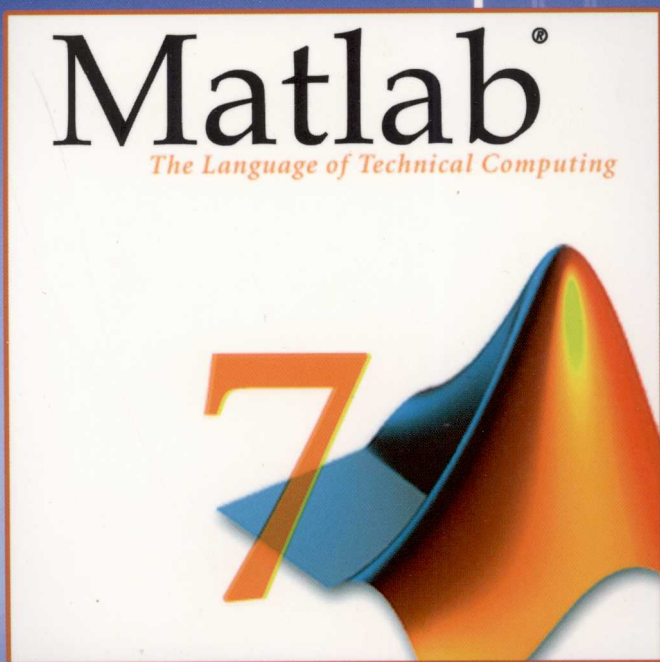


Matlab 7.0/Simulink 6.0 应用实例仿真与高效算法开发

黄永安 李文成 高小科 编著



清华大学出版社

内 容 简 介

本书以科学计算和工程仿真为背景,利用 Matlab/Simulink 工具结合实际工程问题进行讲解,全书图文并茂,突出应用,并配有丰富的计算程序和仿真框图。

全书共 18 章,分为两部分:第一部分为 Matlab 计算篇(含第 1~9 章),这部分主要利用 Matlab 实现各种高效的计算方法,如微积分求解、线性方程组求解、非线性方程组求解、数据建模、微分方程求解、边值问题求解和优化问题求解等计算领域经常遇到的问题。第二部分为 Simulink 仿真篇(含第 10~18 章),利用 Simulink 进行系统建模和工程仿真,如 Simulink 的数值计算、离散和连续混合系统建模、Stateflow、控制系统设计、神经网络、机构仿真和实时工具箱等。本书配有丰富的工程实例,掌握本书之后可以非常容易地利用 Matlab/Simulink 进行常规的科学计算和仿真。

本书内容丰富,涉及多个专业领域,是一本难得的系统的工程书籍,能够帮助读者更好地解决问题,可以作为在校大学生、研究生、教师、工程师和科研人员的参考手册,亦可作为广大工程技术人员的参考用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Matlab 7.0/Simulink 6.0 应用实例仿真与高效算法开发/黄永安,李文成,高小科编著. —北京:清华大学出版社,2008.6

ISBN 978-7-302-17541-4

I. M… II. ①黄… ②李… ③高… III. 计算机辅助计算—软件包, Matlab 7.0/Simulink 6.0 IV. TP391.75

中国版本图书馆 CIP 数据核字(2008)第 063060 号

责任编辑:刘建龙 桑任松

封面设计:杨玉兰

版式设计:北京东方人华科技有限公司

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京宏伟双华印刷有限公司

经 销:全国新华书店

开 本:185×260 印 张:31.75 字 数:765 千字

版 次:2008 年 6 月第 1 版 印 次:2008 年 6 月第 1 次印刷

印 数:1~4000

定 价:48.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:026872-01

前 言

计算机仿真已经成为解决工程实际的必要手段, Matlab\Simulink 软件是功能最强大的仿真软件之一, 可以非常容易地与其他编程语言或仿真软件进行结合, 实现复杂的科学计算和仿真。仿真重点是建立模型, 在模型建立以后, 再设计合理的算法对模型进行计算。Matlab 计算较以往的编程语言更为灵活, 功能更加强大, 上手快, 调试方便, 可以节省大量的编程时间。Simulink 建模较一般程序建模更为直观, 操作也更为简单, 不用死记各种参数即命令的用法, 只需要用鼠标就能够完成非常复杂的工作。Simulink 不但能够进行线性系统仿真, 还支持非线性系统仿真; 不但支持连续系统仿真, 还支持离散系统甚至混合系统仿真; 不但本身功能非常强大, 而且还是一个开放性体系, 用户可以自己开发模块来增强 Simulink 的功能。对于同一个系统模型, 利用 Simulink 可以采用多个采样速率。不但能够实时地显示计算结果, 还能够显示模型所表示实物的实际运动形式。

但是较为完整系统介绍 Matlab\Simulink 强大仿真功能的教材还非常少, 多数偏重于某一方面的应用, 或者没有涉及 Simulink 框图仿真和 Matlab 数值计算。为此, 本书详细地介绍了 Matlab\Simulink 的使用, 其中不但介绍了 Matlab 求解各种复杂科学问题的方法, 而且还介绍了 Simulink 的高级应用和开发, 并配以实例进行说明。

本书的最大特色是介绍了科学计算仿真过程中用到的 Matlab/Simulink 高级仿真功能, 列举了丰富的实例, 讲解深入浅出, 在某些比较难以理解的章节, 都是伴随工程实例的仿真过程讲解的, 使读者能够轻松入门, 学以致用。在欧美各国的高等院校中, 掌握 Matlab 的应用已经成为大学生、研究生和教师的必备技能, 其中 Simulink 更是其中较为深层次的应用。国内的高等院校也在开设 Matlab 课程, 网络社区也非常繁荣。在这种情况下, 作者结合本人开设论坛的经验和实际中遇到较多的问题, 以及在学习 Matlab 中的经验和体会编写了此书, 主要向读者展示 Simulink 的强大仿真功能。

本书内容分为两篇:

第一篇为 Matlab 计算篇, 包括第 1~9 章。介绍 Matlab 的使用方法, 如微积分求解、非线性方程组求解数据建模、微分方程求解, 边值问题求解和优化问题求解有助于分析一般的科学问题。

第二篇为 Simulink 仿真篇, 包括第 10~18 章。介绍 Simulink 的高级使用方法, 讲解 Simulink 的开发和工程计算, S-函数的开发、函数的回调、动画输出, Stateflow 原理与使用技巧, Simulink 控制设计、控制系统仿真, 神经网络控制, Real-Time Workshop, SimMechanics 机构系统应用。

使用读者对象: 本书内容专业, 是一本难得的系统的工程书籍, 能够帮助读者更好地解决问题, 可以作为在校大学生、研究生、教师、工程师和科研人员的参考手册, 亦可作为广大工程技术人员的参考用书。

在本书完成之际, 我们感到万分欣慰。由于本书涉及面较宽, 涉及人员较多, 但是经

过不懈的努力才使这本书能够顺利完成。本书主要由黄永安(华中科技大学)、李文成(西北工业大学)和高小科(西北工业大学)负责,其中黄永安完成第 10~18 章,李文成完成第 6~9 章,高小科完成第 1~5 章。在此一并感谢韩爱红(华北水利水电大学)、魏麟欢(西北工业大学)对本书完成付出的大量劳动。衷心感谢国家自然科学基金(50705035)和国家博士后基金(2007041076)。

由于能力限制及时间仓促,虽然在多个 PC 机上经过反复验算,但书中难免有疏漏之处,希望广大读者批评指正。

目 录

第一篇 Matlab 计算篇

第 1 章 Matlab 语言程序基础..... 1	
1.1 Matlab 的数据类型..... 1	
1.1.1 常量与变量..... 1	
1.1.2 运算符..... 3	
1.1.3 结构数组..... 5	
1.1.4 数组与矩阵的生成..... 8	
1.2 基本数学运算..... 9	
1.2.1 向量及其运算..... 9	
1.2.2 矩阵及其运算..... 11	
1.2.3 多项式的化简与变换..... 14	
1.3 基本符号运算..... 15	
1.3.1 符号对象和表达式..... 15	
1.3.2 符号函数的操作..... 16	
1.3.3 符号对象和其他数据的转换... 17	
1.3.4 符号矩阵的生成..... 18	
1.3.5 符号表达式的操作..... 19	
1.4 基本的流程结构..... 21	
1.4.1 循环结构..... 21	
1.4.2 转移结构..... 23	
1.4.3 开关结构..... 23	
1.4.4 试探结构..... 25	
1.5 函数编写与调试..... 25	
1.5.1 函数的基本结构..... 26	
1.5.2 变量的检测传递..... 27	
1.5.3 串演算函数..... 31	
1.6 图形的绘制..... 33	
1.6.1 二维图形绘制基本语句..... 33	
1.6.2 特殊二维图形绘制语句..... 38	
1.6.3 二维图形的精细控制..... 40	
1.6.4 三维图形绘制..... 42	
1.6.5 三维图形的可视化..... 43	
1.6.6 三维图形的精细控制..... 44	
第 2 章 Matlab 与 Microsoft Office 的 连接..... 46	
2.1 Matlab 与 Microsoft Word 的连接..... 46	
2.1.1 安装 Matlab Notebook..... 46	
2.1.2 使用 Matlab Notebook..... 49	
2.1.3 Notebook 使用时需要 注意的问题..... 51	
2.2 Matlab 与 Microsoft PowerPoint 的 连接..... 51	
第 3 章 微积分问题的数值实验..... 54	
3.1 微积分问题的解析解..... 54	
3.1.1 极限问题的解析解..... 54	
3.1.2 函数导数的解析解..... 56	
3.1.3 积分问题的解析解..... 60	
3.2 函数的级数展开与求和..... 63	
3.2.1 Taylor 幂级数展开..... 63	
3.2.2 Fourier 级数展开..... 66	
3.2.3 级数求和的计算..... 69	
3.3 数值微分问题..... 71	
3.3.1 数值微分算法..... 71	
3.3.2 中心差分方法..... 72	
3.3.3 二元函数的梯度计算..... 74	
3.4 数值积分问题..... 76	
3.4.1 由给定数据进行梯形求积..... 77	
3.4.2 单变量数值积分求解..... 79	

3.4.3 双重积分问题的数值解.....	81	5.3.1 迭代法.....	146
3.4.4 三重定积分的数值求解.....	83	5.3.2 塞德尔迭代法.....	147
3.5 曲线积分与曲面积分的计算.....	83	5.3.3 牛顿迭代法.....	149
3.5.1 曲线积分及 Matlab 语言 求解.....	83	第 6 章 数据建模问题的数值实验	152
3.5.2 曲面积分与 Matlab 语言 求解.....	87	6.1 插值与数据拟合.....	152
第 4 章 线性代数方程组的数值实验	90	6.1.1 一维数据的插值问题.....	153
4.1 特殊矩阵的输入.....	90	6.1.2 二维网格数据的插值问题.....	155
4.1.1 数值矩阵的输入.....	90	6.2 样条插值与数值微积分.....	158
4.1.2 符号矩阵的输入.....	94	6.2.1 样条插值的 Matlab 表示.....	158
4.2 矩阵基本分析.....	94	6.2.2 基于样条插值的数值微 积分运算.....	162
4.2.1 矩阵的特征值与特征向量.....	94	6.3 由已知数据拟合数学模型.....	164
4.2.2 矩阵的 LU 分解.....	95	6.3.1 多项式拟合.....	164
4.2.3 矩阵的 QR 分解.....	97	6.3.2 函数线性组合的 曲线拟合方法.....	166
4.2.4 矩阵的奇异值分解.....	97	6.3.3 最小二乘曲线拟合.....	169
4.3 线性代数方程的求解.....	100	第 7 章 微分方程问题的数值实验	172
4.3.1 线性代数方程组.....	100	7.1 常系数线性微分方程的符号解.....	172
4.3.2 Lyapunov 方程.....	108	7.1.1 线性常系数微分方程.....	172
4.3.3 Sylvester 方程.....	109	7.1.2 特殊非线性微分方程.....	175
4.3.4 Riccati 方程.....	110	7.2 微分方程问题的数值解法.....	176
4.4 稀疏矩阵的线性方程.....	110	7.2.1 微分方程问题算法概述.....	176
4.4.1 稀疏矩阵.....	111	7.2.2 Runge-Kutta 算法及 Matlab 实现.....	177
4.4.2 稀疏矩阵的运算.....	116	7.2.3 一阶微分方程组的 数值解.....	178
4.4.3 稀疏矩阵的分解.....	118	7.2.4 单个高阶微分方程的 转换.....	183
4.4.4 稀疏矩阵的特征值分解.....	119	7.3 特殊微分方程的数值解.....	185
4.4.5 稀疏矩阵方程的求解.....	120	7.3.1 刚性微分方程组的求解.....	185
第 5 章 非线性方程解法	130	7.3.2 隐式微分方程求解.....	190
5.1 非线性方程的数值解法.....	130	7.3.3 微分代数方程的求解.....	192
5.1.1 二分法.....	130	7.3.4 时滞微分方程求解.....	193
5.1.2 迭代法.....	132	7.4 微分方程初值问题的精细 积分方法.....	197
5.1.3 牛顿迭代法.....	136	7.4.1 齐次方程和指数矩阵的 精细积分算法.....	197
5.2 非线性方程(组)的符号解法.....	142		
5.2.1 solve()函数.....	142		
5.2.2 fzero()函数.....	143		
5.2.3 fsolve()函数.....	143		
5.3 非线性方程组的数值解法.....	146		

7.4.2 非齐次方程的精细 积分算法	198	8.2.2 偏微分工具箱(PDE ToolBox) 简介	223
7.4.3 一般结构动力方程的精细 积分方法	199	8.2.3 求解偏微分方程典型函数 介绍及其应用举例	226
7.4.4 增维精细积分方法	203	8.2.4 PDE 图形界面简介	231
附录 常微分方程数值求解器	207		
第 8 章 微分方程的边值问题和 偏微分方程工具箱	211	第 9 章 Matlab 的最优化问题的 计算机求解	240
8.1 常微分方程边值问题的 计算机求解	211	9.1 无约束最优化问题方程求解	240
8.1.1 线性方程边值问题的 打靶算法	212	9.1.1 基本理论	240
8.1.2 线性微分方程边值问题的 有限差分算法	216	9.1.2 解析解法和图解法	244
8.1.3 常微分方程边值问题的 求解器	220	9.1.3 基于 Matlab 的数值解法	245
8.2 偏微分方程求解	222	9.2 有约束最优化问题计算机求解	252
8.2.1 偏微分方程的有限元法	222	9.2.1 基本理论	252
		9.2.2 基于 Matlab 的数值 解法简介	255
		9.3 整数规划问题计算机求解	256
		9.3.1 基本理论	256
		9.3.2 基于 Matlab 的数值 解法简介	259

第二篇 Simulink 仿真篇

第 10 章 Simulink 分析工具与 模型调试	261	10.4.4 显示积分信息	273
10.1 打开 Simulink 调试器	261	10.5 显示模型信息	273
10.1.1 窗口调试方式	261	10.5.1 显示模型中模块的执行 顺序	273
10.1.2 命令行调试	263	10.5.2 显示模型中的非虚拟 系统	274
10.2 进行模型仿真与调试	264	10.5.3 显示具有过零点的潜 在模块	275
10.3 断点设置	268	10.5.4 显示代数环	275
10.3.1 无条件中断	269	10.5.5 显示调试器状态	276
10.3.2 条件中断	270		
10.4 显示仿真信息	270	第 11 章 Simulink 进行数值计算	277
10.4.1 显示模块输入输出 信息	271	11.1 微分方程求解器 Solver	277
10.4.2 显示代数环信息	272	11.2 刚性方程求解实例	278
10.4.3 显示系统状态	272	11.3 Simulink 仿真中的代数环问题	282

第 12 章 Simulink 连续系统、离散系统和混合系统，以及状态系统建模

12.1 连续系统建模.....	287
12.1.1 线性系统.....	287
12.1.2 非线性系统.....	297
12.2 离散系统建模.....	297
12.2.1 模块介绍.....	297
12.2.2 离散系统实例.....	298
12.3 混合系统建模.....	303
12.4 状态模块仿真.....	306
12.4.1 状态空间.....	306
12.4.2 连续系统的状态空间表示.....	307
12.4.3 离散系统的状态空间表示.....	307
12.5 实例分析.....	307
12.5.1 状态系统实例.....	307
12.5.2 振动系统实例.....	311

第 13 章 S-函数的建立与应用

13.1 何为 S-Function.....	315
13.2 在模型中使用 S-函数.....	316
13.2.1 初识 S-函数.....	316
13.2.2 S-Function 所起的作用.....	319
13.3 S-函数工作原理.....	319
13.3.1 模型的数学描述.....	320
13.3.2 仿真过程.....	320
13.3.3 S-函数回调方法.....	321
13.4 M 文件 S-函数的编写.....	322
13.5 M 文件 S-函数模板.....	323
13.6 M 文件 S-函数简单实例.....	326
13.7 连续、离散和混合系统 M 文件 S-函数.....	333
13.7.1 连续系统.....	333
13.7.2 离散系统.....	336

第 14 章 Stateflow 使用技巧与实例应用

14.1 Stateflow 基础.....	340
------------------------	-----

14.1.1 初识 Stateflow.....	340
14.1.2 状态转移图.....	341
14.1.3 状态与转移.....	342
14.1.4 事件.....	342
14.2 运行 Stateflow	343
14.2.1 将 Stateflow 嵌入 Simulink 中的一个简单范例.....	343
14.2.2 利用 Stateflow 来表示模型的控制部分.....	345
14.2.3 通过迁移来改变 Stateflow 状态.....	347
14.2.4 通过事件来激发 Stateflow.....	348
14.2.5 Stateflow 使用数据变量.....	348
14.3 为目标生成 C 代码.....	349
14.4 利用状态和迁移进行控制.....	350
14.4.1 创建一个 on-off 模型.....	350
14.4.2 在 Stateflow 图表中绘制迁移.....	352
14.4.3 为 Stateflow 图表添加一个触发事件.....	354
14.4.4 向 Stateflow 图表传递一个触发事件.....	355
14.5 进行 Stateflow 图表仿真.....	355
14.5.1 定义模型仿真参数.....	356
14.5.2 Stateflow 图表仿真的基本步骤.....	357
14.6 仿真过程中的调试.....	360
14.7 Stateflow 常用命令.....	363
14.8 Stateflow 仿真实例.....	364

第 15 章 控制系统设计与仿真.....

15.1 何为 Simulink 控制系统设计.....	371
15.2 线性化模型.....	372
15.3 磁力球模型线性化实例.....	372
15.3.1 磁力球模型示意图.....	372
15.3.2 磁力球模型方程.....	373

15.3.3 创建或打开一个 Simulink 模型	373	16.2 神经网络模型预测控制实例分析 ...	425
15.3.4 开始线性化工程	374	16.2.1 模型预测理论	425
15.3.5 配置一个线性化模型	375	16.2.2 问题的描述	426
15.3.6 确定工作点	378	16.2.3 建立模型	427
15.3.7 线性化模型	385	16.2.4 系统辨识	428
15.3.8 线性化模块	387	16.2.5 系统仿真	431
15.3.9 分析结果	388	16.3 NARMA_L2(反馈线性化)控制实例分析	432
15.3.10 导出并保存工程	392	16.3.1 反馈线性化控制理论	432
15.4 离散系统建模	393	16.3.2 问题的描述	434
15.4.1 离散系统建模的基本概念	393	16.3.3 建立模型	435
15.4.2 不同采样速率的彩色显示	394	16.3.4 系统辨识	436
15.4.3 混合系统建模	394	16.3.5 系统仿真	439
15.5 经典控制系统的设计与仿真	395	16.4 神经网络模型参考控制实例分析 ...	440
15.5.1 时域分析	395	16.4.1 模型参考控制理论	440
15.5.2 频域分析	396	16.4.2 问题描述	441
15.5.3 根轨迹分析	397	16.4.3 建立模型	441
15.5.4 常用控制器的设计与仿真 ...	398	16.4.4 系统辨识	443
15.6 现代控制系统的设计与仿真	402	16.4.5 系统仿真	448
15.6.1 现代控制系统的特点和研究内容	402	第 17 章 SimMechanics 机构系统应用	450
15.6.2 模型参考自适应控制系统	403	17.1 SimMechanics 介绍	450
15.6.3 实例分析	404	17.1.1 初识 SimMechanics	450
15.7 滑模变结构控制系统设计与分析	406	17.1.2 SimMechanics 能够做什么 ...	451
15.7.1 滑模变结构控制理论	406	17.2 SimMechanics 模块介绍	451
15.7.2 连续时间滑模控制	407	17.3 建立一个简单的机构实例	454
15.7.3 离散时间滑模控制	407	17.3.1 创建 SimMechanics 模型 ...	455
15.7.4 滑模变结构控制的应用	408	17.3.2 建立一个单摆模型	456
第 16 章 Simulink 神经网络工具箱	417	17.4 单摆运动可视化	465
16.1 Neural Network Blockset 模块库	417	17.5 四连杆结构仿真实例	469
16.1.1 神经网络模块介绍	417	第 18 章 Real-Time Workshop	480
16.1.2 神经网络模块的生成及模型设计	420	18.1 Real-Time Workshop 简介	480
		18.2 生成普通的实时程序	483
		18.2.1 打开演示程序	483
		18.2.2 实例演示	485
		参考文献	494

第一篇 Matlab 计算篇

第 1 章 Matlab 语言程序基础

本章旨在介绍关于 Matlab 的基本知识，使读者可以更好地理解后续章节中的内容。其实，对于 Matlab 的基本知识的介绍可以在市面上流行的各类 Matlab 书籍中看到，那么本书中对于此部分的介绍又有哪些自己的特点呢？首先，省去了 Matlab 中一些与高效率计算编程无关的赘述，使读者在能力理解和运用本书中心部分对于高效算法的介绍前提下，用最少的的时间获得必需的预备知识；其次是一些后续章节中经常用到的基本概念和方法作了更为详细的介绍，这样读者在阅读完本章节以后可以更轻松地理解本章中对于算法介绍的内容。

本章主要包括：

- Matlab 的数据类型
- 基本数学运算
- 基本符号运算
- 基本的流程结构
- 函数编写与调试
- 图形的绘制

1.1 Matlab 的数据类型

本节主要介绍 Matlab 中的数据类型。不同类型的对象在计算机语言中可用不同类型的变量来描述。为使读者能够更好地理解后续章节的算法程序，本节专门对 Matlab 的数据类型进行介绍。

1.1.1 常量与变量

1. 常量

Matlab 中，常量和变量是基本的语言元素。Matlab 的数值采用传统的十进制表示，可带负号或小数点。举例如下：

5 -87 0.001 3.1415 1.05457e-34 6.02e23

Matlab 还提供了一些内部常量，也可以理解为 Matlab 默认的预定义变量。这些常量定

义了编程和应用中经常用到的数据,如圆周率、虚数单位等。随着 Matlab 启动,这些常数就被产生。

表 1-1 Matlab 中的一些内部常量

内部常量	基本意义
ans	默认变量名或默认变量名, Matlab 会自动将无指定输出变量的计算结果默认保存到 ans 变量中
eps	机器零阈值
Inf 或 inf	表示无穷大的结算结果,如 1/0
i 或 j	虚数单位,有 $i=j=\sqrt{-1}$
pi	圆周率
NaN 或 nan	非数值,通常表示不合法数值的值。如, 0/0 或 inf/inf 所得到的结果
realmax(realmin)	最大(小)正实数
nargin(nargout)	函数输入(输出)变量数目
computer	计算机类型
version	Matlab 版本字符串

表 1-1 中所列常量可不必声明,直接调用。这里值得指出的是表 1-1 中所列常量的意义是在表中变量名未被用户赋值的情况下才成立的。假如用户对表中任何一个内部常量的字符串进行了赋值,那么该常量的默认意义或者默认值已被用户的新赋值所覆盖,只不过这种“覆盖”是临时的,如果用户运行了 clear 指令将 Matlab 内存清空,或者 Matlab 指令窗被重新启动,那么所有的内部常量又会恢复到系统默认的意义。

【例 1-1】 系统内部常量的调用和赋值。

```

pi                %查看系统的默认值
ans =
    3.1416
pi = ans + 1      %pi 被重新赋值,不再是原系统默认值
pi =
    4.1416
clear            %清空内存中的变量
pi              %重新查看 pi 的值
ans =
    3.1416
  
```

可见,在无指定输出变量时候,系统自动把值赋给了 ans。通过对 pi 进行重新赋值,原先的默认值(圆周率)被覆盖,清空内存之后,pi 又恢复到系统默认值。

2. 变量

所有 Matlab 中定义的变量都以数组或者矩阵形式保存,它提供的数据类型多达十余种,如逻辑型、字符型、数值型、单元数组、结构数组、函数句柄等。下面简单介绍一下几种数据类型。

- 逻辑型 此类数据类型只能包含 1、0 或者 true、false。例如：

```
A = [1 3 5; 4 1 7; 2 6 9];
```

```
B = A>2
```

```
B =
```

```
0     1     1
1     0     1
0     1     1
```

- 字符型 又称字符串数组，如果没有串数组及其相应的操作，数据的可视化和构建 Matlab 的宏指令都会遇到困难。字符串的创建也有值得注意的地方，就是必须首先将待建的字符置于英文状态下的“单引号对”中。例如：

```
clear %清除所有内存变量
```

```
A = 'Hello, everyone!' %对 A 进行赋值
```

```
A =
```

```
Hello, everyone!
```

- 数值型 数值数组及其运算始终是 Matlab 的核心内容。数值型又可分为单精度数值型数组(single)、双精度数值型数组(double)和整型数组。Matlab 进行数值计算时，默认所有变量都为 double 型。使用 format 命令可以设置变量的输出格式，例如：

```
a = 2.3609 %系统默认为 double 型
```

```
a =
```

```
2.3609
```

```
format long %设置系统为 15 位定点格式输出
```

```
a
```

```
a =
```

```
2.3609000000000000
```

```
format long e %设置系统为 15 位浮点格式输出
```

```
a
```

```
a=
```

```
2.3609000000000000e+000
```

在 Matlab 中，变量的命名是要遵守一定规则的。首先，对变量命名要注意区分大小写，比如，Myname 和 myname 表示两个不同的变量。pi 代表系统默认的圆周率，但 Pi、PI 和 pI 都不是；其次，在 Matlab 中对变量进行命名时，第一个字符必须是英文字母，如 0abc、_abc 等都不是合法的；另外，Matlab 中在变量名的中间不能包含空格(backspace)、标点和括号等，不过可以包含下划线，比如，ab_c 是可以的，ab c 或者 ab,c 等都不是合法的。

1.1.2 运算符

运算符从其功能来分大致有三种：算术运算符、比较运算符和逻辑运算符。现在分别介绍一下它们的构成和使用。

1. 算术运算符

Matlab 中的运算符不只是完成传统意义上的算术所需的四则运算，另外还有幂、转置等运算。由于 Matlab 具有强大的矩阵运算功能，所以它的很多运算符都是针对矩阵操作的。

表达式所采用的运算符如表 1-2 所示。

表 1-2 算术运算符及其功能

运 算 符	功 能	运 算 符	功 能
+	加	-	减
*	乘	./	右除
.\	左除	^	幂
'	转置	'	复数共轭转置
*	矩阵乘法	/	矩阵除法
^	矩阵的幂	:	冒号操作符

要说明的是，一维的运算可以看作是矩阵运算的一种特殊形式，其运算符是一样的，例如一维运算中没有的转置，也可以看作是一维矩阵的转置，只是它转置后的结果不变罢了。此外，Matlab 中针对向量的算术操作专门设计了一类近似的运算符，就是在一般的矩阵运算符前加“.”，它代表所参与运算的向量对应元素之间的运算。

2. 比较运算符

比较运算符的作用主要是用来比较两个数值的大小，它的写法与一般的手写比较运算符有所区别。具体如表 1-3 所示。

表 1-3 比较运算符

运 算 符	功 能	运 算 符	功 能
>	大于	>=	大于等于
<	小于	<=	小于等于
==	等于	~=	不等于

值得指出的是，一般的表示等于的运算符在 Matlab 中写为“==”，传统意义上的等号“=”在 Matlab 运算中用于赋值运算。此外，进行比较操作时注意以下两点：①标量可与任何维数组进行比较。比较在此标量与数组每个元素之间进行，比较结果与参与比较的数组同维。②参与比较的没有标量时，比较运算符两端的数组必须维数相同，比较在两数组对应位置上的元素间进行，因此比较结果将与参与比较的数组相同。下面举例说明。

【例 1-2】比较运算。

```
A = [1 2 3 4 5];
B = [5 4 3 2 1];
C = (A<3)
C =
    1    1    0    0    0
D = (A == B)
D =
    0    0    1    0    0
```

3. 逻辑运算符

逻辑运算符的主要功能是判断参与比较的对象之间的某种逻辑关系。表 1-4 列出了各种逻辑运算符及其对应的功能。

表 1-4 逻辑运算符及其对应功能

运 算 符	函 数	含 义	举 例
&	and	与、和	$A \& B \Rightarrow [0,0,1,0,0]$
	or	或	$A = [1,1,1,0,0]$ $A B \Rightarrow [1,1,1,1,1]$
~	not	否、非	$B = [0,0,1,1,1]$ $\sim A = [0,0,0,1,1]$
	xor	元素相异返回 1, 元素相同返回 0	$\text{xor}(A,B) = [1,1,0,1,1]$
	bitand	二进制数字位逻辑操作符, 表示与、和	$\text{bitand}(A,B) = 0100 = 4$
	bitor	二进制数字逻辑操作符, 表示或	$A = 20 = 10100$ $B = 12 = 1100$ $\text{bitor}(A,B) = 11100 = 28$
	bitcmp	二进制数字逻辑操作符, 表示否、非	$\sim B = 0011 = 3$
	bitxor	二进制数字逻辑操作符, 相异数返回 1, 相同数返回 0	$\text{bitxor}(A,B) = 11000 = 24$
&&		如符号两端表达式皆为真, 返回 true(1), 否则返回 0	$(a \sim = 0) \&\& (a/b > 3)$
		如符号两端表达式有一式为真, 返回 true(1), 否则返回 0	$(a == 0) (a/b > 0)$

这里说明一下, 二进制数字位逻辑操作时, 往往首先把运算量转换为二进制表示, 然后逻辑运算在两个二进制数的自右到左相应数位间进行, 输出的结果为运算所得的二进制数所对应的十进制数。当逻辑运算中没有标量时, 参与运算的数组必须维数相同, 运算在两数组的对应位置元素间进行。

此外, “&&” 和 “||” 操作都有其特殊的性质, 两者都属于先决逻辑操作符。“&&” 操作时, 先观察运算符左侧的参与运算的表达式是否为“假”; 若是, 则马上给出运算结果为“假”, 而不必再观察运算符右侧的参与运算量。当左侧的运算量为“真”时, 才接着对右侧运算量进行计算或者判断, 进而执行“与”逻辑运算。“||”操作时, 首先判断左侧的运算量是否为“真”; 若是, 立即给出计算结果“真”, 而不必观察右侧参与运算量; 若不是, 则对右侧运算量进行观察, 进而执行“与”操作。

1.1.3 结构数组

结构数组(也有人称为构架数组)能在一个数组里存放任何类型的数据。结构数组的基本单元(element)是结构。数组中的每个单元以下标区分, 它们之间是平等的。结构数组维数不受限制, 可以为一维、二维或更高维。

结构数组的创建有两种方式，下面我们就看一个例子。

【例 1-3】下面创建一个 `my_software` 结构数组，里面存储的是一个软件系统的信息。首先看第一种方法，直接使用赋值语句构造结构数组。

(1) 直接对域进行赋值

```
my_software.name = 'Matlab' ;
%可以直接将字符串赋值给结构数组的域
my_software.version = '7.0.0.19920(R14)' ;
my_software.computer = 'PCWIN' ;
my_software.testdata.test1= [3.74 3.27 ; 3.56 3.41];
%可以直接将数组赋值给结构数组的域
my_software.testdata.test2= [65.23 68.75 ; 63.06 61.18];
```

(2) 查看结果

```
my_software %显示结构数组的内容
my_software =
name: 'Matlab'
version: '7.0.0.19920(R14)'
system: 'PCWIN'
computer: 'PCWIN'
testdata: [1x1 struct]
my_software.testdata %显示 testdata 域中的内容
ans =
    test1: [2x2 double] ,
    test2: [2x2 double]
my_software.testdata.test1 %显示 test1 域中的内容,系统默认为 double 型
ans =
    3.7400    3.2700
    3.5600    3.4100
```

另外，还可通过 `Matlab` 的一个专门的构造函数 `struct` 来创建结构数组。

(1) 使用 `struct` 函数构造结构数组

```
my_software =
struct('name','Matlab','version','7.0.0.19920(R14)','computer','PCWIN',
'testdata',[]); %用'[]'来预定义空域'testdata'
my_software.testdata = struct('test1',[ 3.74 3.27 ; 3.56 3.41],
'test2',[ 65.23 68.75 ; 63.06 61.18]); %再次调用'struct'命令来对
% 'testdata'进行赋值
```

(2) 查看结果

```
my_software %显示结构数组的内容
my_software =
name: 'Matlab'
version: '7.0.0.19920(R14)'
system: 'PCWIN'
computer: 'PCWIN'
testdata: [1x1 struct]
```

事实上结构数组内的其他内容也是一样的,本例就是简单介绍了两种不同的创建方法。这里还要说明几点:

(1) `struct` 函数指令中的输入参数,第 2、4、6 等偶数项如为字符串需用单引号,或者为数组,数值数组或者字符数组均可。

(2) 空数组 “[]” 可以用来创建新的空域,并且可以在接下来的语句中直接对其进行赋值操作。

(3) `struct` 命令不能创建多重的结构数组,如有需要可以多次调用 `struct` 命令或者嵌套调用 `struct` 函数。

接下来介绍结构数组中内容的提取和编辑。

【例 1-4】结构数组内容的提取和编辑。

首先建立一个简单的统计个人计算机信息的结构数组。

```
mycomputer.manufacturer = 'Lenovo';
mycomputer.type = '补天系列商用机';
mycomputer.cpu = 'Pentium(R) 4 3.000GHz';
mycomputer.system = 'window XP English version';
```

查看结果:

```
mycomputer
mycomputer =
manufacturer: 'Lenovo'
type: '补天系列商用机'
cpu: 'Pentium(R) 4 3.000GHz'
system: 'window XP English version'
```

如果要获得电脑是何系列的信息,并且将该信息单独另外存储,可以在结构名 `mycomputer` 后面跟一个小数点,接下来是要获取信息的字段名。如用 Matlab 语句来表达,如下:

```
Pctype_informations = { mycomputer.manufacturer, mycomputer.type };
%将电脑的部分信息存入一个数组中
Pctype_informations %查看新的数组
Pctype_informations =
'Lenovo' '补天系列商用机'
```

下面给读者演示如何在结构数组 `mycomputer` 上进行域的增添和删除操作。

【例 1-5】对结构数据进行域的增添和删除。

继续对例 1-4 中的结构数组进行操作。

```
mycomputer(2).user.usertype = 'administrator';
mycompute %查看 mycomputer 信息
mycomputer =
1x2 struct array with fields:
manufacturer
type
cpu
system
```

```
user
```

可见, 对 `mycomputer(2)` 的赋值操作会影响到整个结构数组 `mycomputer` 的总体结构。在此类数组中任何一个结构上的域增添操作, 都会对整个结构产生影响。

再查看结构数组 `mycomputer(1).user` 的内容。

```
mycomputer(1).user
```

```
ans =
```

```
[ ]
```

当我们再查看 `mycomputer(1).user` 的结构时, 它并没有 `usertype` 这一项子域。可见对 `mycomputer(2)` 中的域 `user` 的子域 `usertype` 的赋值操作并没有对 `mycomputer(1).user` 子域产生影响, 所以在结构数组中任何一个结构上的子域增添操作, 对整个结构中的同级别的域并没有影响。

下面通过实例来总结结构数组的删除操作规律。

```
mycomputer(1).user.username = 'Guest';
```

```
mycomputer(1).user.usertype = 'guest';
```

```
mycomputer(2).user.username = 'admin'
```

```
mycomputer(2).user = rmfield (mycomputer(2).user, 'usertype');
```

```
%删除 usertype 子域
```

```
mycomputer(2).user
```

```
%查看 mycomputer(2).user 的内容
```

```
ans =
```

```
username: 'admin'
```

```
mycomputer(1).user
```

```
%查看 mycomputer(1).user 的内容
```

```
ans =
```

```
username: 'Guest'
```

```
usertype: 'guest'
```

通过以上操作可以看出删除子域的操作也只是影响到被操作的那个具体结构。

1.1.4 数组与矩阵的生成

Matlab 中, 数值数组(numeric array)和数组运算(array operation)构成了其核心内容。本节主要介绍数组和矩阵的生成。

这里主要介绍一维数组的创建。大致可以分三种方法: 直接构造法、增量法、函数构造法。

1. 直接构造法

直接构造法又称逐个元素输入法, 这种方法最直接也最简单。下面举例说明。

【例 1-6】 数组元素的直接构造法。

```
a = [40 6.3 1/2 1e-2 sqrt(5)]
```

```
%采用逐个元素输入法构造数组
```

```
a =
```

```
40.0000 6.3000 0.5000 0.0100 2.2361
```

这就构造了有五个元素的一维数组元素 `a`。