

MATLAB6实用教程

内容提要

MATLAB 是美国 MathWorks 公司自 1984 年开始推出的数学软件,它优秀的数值计算能力和卓越的数据可 视化能力使其很快在数学软件中脱颖而出。MATLAB 6 是 MathWorks 公司于 2000 年 11 月推出的最新版本。本 书全面而深入地介绍了 MATLAB 6 的各项主要功能,包括 MATLAB 简介、MATLAB 的安装和基本操作、 MATLAB 语法基础、数值计算、MATLAB 程序设计、数据可视化、符号运算功能、创建用户界面等内容。 本书本着实用的原则,由浅入深、循序渐进,力求让初学者真正学会使用 MATLAB。同时,也希望能够让 老用户体会到 MATLAB 新版本中更多更强的功能,进一步提高使用本软件的能力。

本书适合从事理工科学习和研究的各行各业的工程人员和研究

人员、研究人员、科学家、教师和学生等等。

前言

MATLAB 是美国 MathWorks 公司自 1984 年开始推出的一种使用简便的工程计算语言, MathWorks 公司于 2000 年 11 月推出了该语言的最新版本 MATLAB 6。

经过 MathWorks 公司的不断完善,时至今日,MATLAB 已经发展成为适合多学科、多种工作平台的功能 强劲的大型软件。在国外,MATLAB 已经经受了多年考验。在欧美等高校,MATLAB 已经成为线性代数、自 动控制理论、数理统计、数字信号处理、时间序列分析、动态系统仿真等高级课程的基本教学工具,成为攻读 学位的大学生、硕士生、博士生必须掌握的基本技能。在设计研究单位和工业部门,MATLAB 被广泛地用于研 究和解决各种具体问题。在国内也有越来越多的科学技术工作者参加到学习和倡导这种语言的行列中来,近一 年来的发展尤为迅速。可以说,无论读者从事工程方面的哪个学科,都能在 MATLAB 里找到合适的功能。

一种语言之所以能如此迅速地普及,显示出如此旺盛的生命力,MATLAB有其不同于其他语言的特点。正 如同 FORTRAN 和 C 等高级语言使人们摆脱了需要直接对计算机硬件资源进行操作一样,被称作为第四代计算 机语言的 MATLAB,利用其丰富的函数资源,使编程人员从繁琐的程序代码中解放出来。MATLAB的简洁, 给用户带来的是最直观、最简洁的程序开发环境。

MATLAB 6 以矩阵运算为基础,把计算、可视化、程序设计融合到了一个交互的工作环境中。在这里可以 实现工程计算、算法研究、建模和仿真、数据分析及可视化、科学和工程绘图、应用程序开发(包括图形用户 界面设计)等功能。而且,MATLAB 提供的工具箱为各行各业的用户提供了丰富而实用的资源。

本书内容广泛,涵盖了 MATLAB 的核心内容,适合从事理工科学习和研究的各行各业的工程人员、研究 人员、科学家、教师和学生等。

本书由张晋策划,张伟、贾砚宾等编著。在本书的编写过程中,得到了许多同行的支持和帮助,在此向他 们表示由衷的感谢!

由于时间仓促,作者水平有限,书中错误在所难免,如果读者能够对本书的错误和疏漏之处给予批评指正, 那将是对我们最大的帮助!

目 录	
-----	--

第−	-章	MAT	LAB 简介	1
	1.1	MAT	TLAB 的产生与发展	1
	1.2	MAT	TLAB 的功能与特点	2
		1.2.1	MATLAB 的功能	2
		1.2.2	MATLAB 的特点	3
	1.3	MAT	TLAB 系统的组成	3
	1.4	MAT	TLAB 工具箱及应用简介	4
		1.4.1	MATLAB 工具箱	4
		1.4.2	MATLAB 主要工具箱简介	4
	1.5	MA	TLAB 6 的新特性	7
	1.6	其他	数学类科技软件	7
		1.6.1	Maple	8
		1.6.2	MathCAD	8
		1.6.3	Mathmatica	8
		1.6.4	MATLAB	8
	1.7	小结		9
笜-	- <u>-</u>	МАТ	I A D 的空港和其木晶作	10
~~~	-무	MAI	LAD 的女衣和李平探旧····································	. IC
	2.1	MAT	TLAB 6 的安装	. 10
		2.1.1	系统要求	. 10
		2.1.2	MATLAB 6 的安装过程	. 10
	2.2	MAT	TLAB 6 的启动与退出	. 13
		2.2.1	MATLAB 6 的启动	. 13
		2.2.2	退出 MATLAB 6	. 14
	2.3	卸载	MATLAB 6	. 14
	2.4	MAT	TLAB 的目录结构	. 15
	2.5	MAT	TLAB 环境	. 18
		2.5.1	MATLAB 集成桌面	. 18
		2.5.2	MATLAB 的命令窗口	. 24
		2.5.3	命令历史窗口	. 31
		2.5.4	工作空间	. 32
		2.5.5	发射台	. 35
		2.5.6	图形窗口	. 36
		2.5.7	编辑/调试窗口	. 37
		2.5.8	路径浏览窗口	. 38
	2.6	帮助	系统	. 39
		2.6.1	帮助命令	. 39
		2.6.2	帮助窗口	.41
		2.6.3	在线帮助	. 42
	2.7	小	结	. 42

第三章	MAT	LAB 语法基础	.43
3.1	矩阵		.43
	3.1.1	MATLAB 中的矩阵	.43
	3.1.2	矩阵的输入	.43
	3.1.3	下标的引用	. 48
3.2	基本	运算符	. 49
	3.2.1	算术运算符	. 49
	3.2.2	关系运算符	. 49
	3.2.3	逻辑运算符	. 50
	3.2.4	操作符	. 50
3.3	变量		. 54
	3.3.1	变量命名规则	. 54
	3.3.2	固定变量	. 54
	3.3.3	————— 全局变量	. 55
3.4	数值		. 56
3.5	数据	类型	.56
3.6	数组		. 57
3.7	字符	串	. 59
3.8	М文	件	. 60
010	3.8.1	命令文件	.60
	3.8.2	函数文件	. 60
3.9	流程	—————————————————————————————————————	. 69
• • •	3.9.1	if-else 语句	.70
	3.9.2	switch-case 语句	.71
	3.9.3	for 循环	.71
	3.9.4	while 循环	.72
3.1	0 小约		.73
空口中	米山(古之		74
<b></b>	到且「	「昇	. 74
4.1	简单	运算	.74
	4.1.1	MATLAB 变量的代数运算	.74
	4.1.2	逻辑运算	.75
	4.1.3	关系表达式与表达式函数	.76
	4.1.4	其他运算	.76
4.2	矩阵	运算	. 77
	4.2.1	矩阵结构操作	. 77
	4.2.2	向量和子矩阵的生成	. 80
	4.2.3	空矩阵	. 82
	4.2.4	MATLAB 中的特殊矩阵	. 83
	4.2.5	矩阵的逆与伪逆	. 84
	4.2.6	矩阵的四则运算	. 85
	4.2.7	矩阵的幂和指数函数	. 88
	4.2.8	转置和共轭	. 88
	4.2.9	矩阵函数	. 89
	4.2.10	特征值与特征向量	. 90
	4.2.11	上海森堡形式、QR 和 QZ 分解	. 92

	4.2.12	舒尔分解和奇异值分解	93
	4.3 多项	〕式	96
	4.3.1	多项式的表达和创建	96
	4.3.2	多项式的运算	97
	4.4 稀疏	矩阵	101
	4.4.1	稀疏矩阵的存储	101
	4.4.2	创建稀疏矩阵	101
	4.4.3	稀疏矩阵运算	
	444	稀商矩阵的特例	103
	445	系数阵为稀疏矩阵的线性方程组	105
	45 多维		105
	451		105
	4.5.2	シーズメーロリスパー	105
	4.5.2	日名社教幻工作	107
	4.5.5	のジェ奴紀工IF 2.维数组的运算	110
	4.5.4	シェ奴ュロリピ弁	111
	4.0 = 14)	奴纪马马和尼奴纪	111
	4.0.1	5日19935旦	1 1 1
	4.0.∠ 4.7 ₩5+₽	细胞数组 ?/\≠⊂和体社	114
	4./ 女义加古	[刀机机统]	119
	4.7.1	取入旧和取小旧	120
	4.7.2	水仙、乘枳和左方	120
	4.7.3	吊用犹江師文	121
	4.8 小ピョ		123
第	五章 MAT	LAB 程序设计	124
	5.1 M文	7件	124
	5.1.1	命令文件	124
	5.1.2	函数文件	
	5.1.3	文件的执行	127
	5.1.4	输入参数和输出参数的检验	128
	5.1.5	子函数	129
	5.1.6	シニーズ 外売函数(Shell Escape Function)	
	52 文件	·操作	130
	5.2 21	文件的打开和关闭	131
	522		132
	523	—————————————————————————————————————	133
	53 类和	1日20人11	135
	5.5 <del>×</del> 1H	*17/1多##注	135
	532	关他们 <b>动</b> 阀应	130
	5.3.2	⋈免伏失级	1/2
	5.2.4	^] ╗ /レレフレーヌ 米和対免的继承	142
	3.3.4 54 程序	大 (μ/) え μ) 近 小	1.42
	J.4 作王/予	^山 小儿口	143
	5.4.1	调小吗吗里心	144
	5.4.2	蚁油□┘у火仁壮	140
	5.4.3	内行的目廷	146
	3.3 时间	コロ	147

		5.5.1	日期的获取和表示	. 147
		5.5.2	日期的转换	. 148
		5.5.3	几个特殊的日期函数	. 150
		5.5.4	用日期格式标注坐标轴	. 151
		5.5.5	计算程序执行时间	. 152
	5.6	文件	调试	. 153
		5.6.1	用调试器调试	. 153
		5.6.2	在命令窗口中调试	. 157
		5.6.3	文件评述	. 159
	5.7	小结		. 159
第六	章	数据⋷	可视化	. 160
	61	一维	图形	160
	0.11	6.1.1		. 161
		612	基本绘图控制	167
		613	图形的标注	171
		614		176
	6.2	三维	ペニンペロル//	. 177
	0.2	6.2.1	→/// 基本绘图命令简介	. 177
		622	三维图形的建立	178
		6.2.3	调整视角	. 186
		624	光昭控制	190
	63	特殊	图形	193
	0.2	6.3.1	条形图	. 193
		632		195
		6.3.3	面积图	. 197
		6.3.4	□ //□ 饼图	. 199
		635	火柴杆图	200
		6.3.6	阶梯图	. 202
		637	ケ量图	203
		638	(人工) 等高线图	205
		639		209
	64	6.5.5	2	2.09
	0	6.4.1	MATLAB 颜色表示法	.210
		6.4.2	着色原理	.211
		6.4.3	色谱矩阵分析	.212
	6.5	图形	过 4 2 4 7 5 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	.216
		6.5.1	MATLAB 图形对象结构	.217
		6.5.2	通用函数	.217
		6.5.3	共有属性	. 223
		6.5.4	根对象	. 227
		6.5.5	图形窗口对象	. 227
		6.5.6	轴对象	. 231
		6.5.7	线段对象	. 235
		6.5.8	曲面对象	. 236
		6.5.9	区域片对象	. 239

	6.5.10 图像对象	
	6.5.11 文字对象	
	6.5.12 光源对象	
	6.5.13 缺省属性及其设置	
6.6	5 图像操作	
	6.6.1 图像的读写	
	6.6.2 图像显示	
	6.6.3 图形窗口的编辑	
	6.6.4 图像打印	
	6.6.5 在不同应用程序间拷贝图形	
6.7	7 动画制作	
6.8	3 小结	
第七章	符号运算功能	
7.1	· 符号计算入门	
	7.1.1 符号变量和符号表达式的生成	
	7.1.2 符号和数值间的转换	
	7.1.3 突变量、复变量和抽象函数	
	7.1.4 默认符号变量	
	7.1.5 替代	
	7.1.6 指定精度计算	
7.2	2 符号矩阵运算	
	7.2.1 符号矩阵的创立	
	7.2.2 符号矩阵的基本运算	
	7.2.3 符号矩阵分解	
	7.2.4 符号矩阵的简化	
7.3	3 符号微积分	
	7.3.1 符号极限	
	7.3.2 符号积分	
	7.3.3 级数求和	
	7.3.4 符号微分和差分	
7.4	4 符号代数方程求解	
	7.4.1 solve 函数	
	7.4.2 线性方程组的解析解法	
	7.4.3 非线性方程的解析解法	
7.5	5 符号微分方程求解	
7.6	5 符号函数的二维图	
	7.6.1 符号函数的简易绘图函数	
	7.6.2 绘制函数图函数	
7.7	7 访问 Maple 函数	
	7.7.1 利用 sym 访问 Maple 函数	
	7.7.2 利用 maple 访问 Maple 函数	
7.8	3 特殊数学函数	
7.9	0 小结	
第八章	创建用户界面	
8.1	界面设计原则	

8.1.1	简单性	
8.1.2	一致性	
8.1.3	熟悉性	
8.1.4	动态交互	
8.2 图册	/编辑工具	
8.2.1	布局设计编辑器	
8.2.2	对齐工具	
8.2.3	属性编辑器	
8.2.4	对象浏览器	
8.2.5	菜单编辑器	
8.2.6	图形中的控件	
8.3 编程	创建用户界面	
8.3.1	编程建立菜单	
8.3.2	编程建立控件	
8.4 小结	<u> </u>	
		205
VIAIL	AD 函奴叩マぶリ化	

# 第一章 MATLAB 简介

作为全书的开始,本章将介绍以下几方面的内容:

- MATLAB 的产生与发展
- MATLAB 的功能与特点
- MATLAB 系统的组成
- MATLAB 工具箱及其应用
- MATLAB6的新特性
- MATLAB 与其他数学软件的比较

## 1.1 MATLAB 的产生与发展

MATLAB 为矩阵(matrix)和实验室(laboratory)两个英文单词的前三个字母的组合,是 MATLAB 的首 创者 Cleve Moler 教授在开发初期命名的。Cleve Moler 教授在数值分析,特别是在数值线性代数的领域中很有 影响,他曾在密西根大学、斯坦福大学和新墨西哥大学任数学与计算机科学教授。

20 世纪 70 年代中期, Cleve Moler 教授和其同事在美国国家科学基金的资助下开发了调用 EISPACK 和 LINPACK 的 FORTRAN 子程序库。EISPACK 是特征值求解的 FORTRAN 程序库, LINPACK 是解线性方程的 程序库。在当时,这两个程序库代表矩阵运算的最高水平。

1980年前后,身为美国 NewMexico 大学计算机系系主任的 CleveMoler,在给学生讲授线性代数课程时, 想教学生使用 EISPACK 和 LINPACK 程序库,但他发现学生用其他高级语言编程极为不便,于是他开始自己动 手构思并开发了 MATLAB (MATrix LABoratory,矩阵实验室),这一软件利用了当时数值线性代数领域最高水 平的 EISPACK 和 LINPACK 两大软件包中可靠的子程序,用 Fortran 语言编写了集命令翻译、科学计算于一身 的一套交互式软件系统。在以后的数年里,MATLAB 在多所大学里作为教学辅助软件使用,并作为面向大众的 免费软件广为流传。

所谓交互式语言,是指人们给出一条命令,立即就可以得出该命令的结果。该语言无需像 C 和 Fortran 语言那样,首先要求使用者去编写源程序,然后对之进行编译、连接,最终形成可执行文件。这无疑会给使用者带来了极大的方便。早期的 MATLAB 是用 Fortran 语言编写的,只能作矩阵运算;绘图也只能用极其原始的方法,即用星号描点的形式画图;内部函数也只提供了几十个。但即使其当时的功能十分简单,当它作为免费软件,还是吸引了大批的使用者。

1983 年春, Cleve Moler 到 Stanford 大学讲学, MATLAB 深深地吸引了工程师 John Little。John Little 敏锐 地觉察到 MATLAB 在工程领域的广阔前景。同年,他和 CleveMoler、Steve Bangert 一起,用 C 语言开发了第 二代专业版。这一代的 MATLAB 语言同时具备了数值计算和数据图示化的功能。

1984 年, Cleve Moler 和 John Little 成立了 MathWorks 公司,该公司正式推出了第一个 MATLAB 的商业版 本 MATLAB 3.0,并继续进行 MATLAB 的研究和开发。当时的 MATLAB 版本已经用 C 语言作了完全的改写, 其后又增添了丰富多彩的图形图像处理、多媒体功能、符号运算和它与其他流行软件的接口功能,使得 MATLAB 的功能越来越强大。

在 MATLAB 进入市场前,国际上的许多应用软件包都是直接以 FORTRAN 和 C 语言等编程语言开发的。 这种软件的缺点是使用面窄、接口简陋、程序结构不开放以及没有标准的基库,很难适应各学科的最新发展, 因而很难推广。MATLAB 的出现,为各国科学家开发学科软件提供了新的基础。在 MATLAB 问世不久的 80 年代中期,原先控制领域里的一些软件包纷纷被淘汰或在 MATLAB 上重建。

1992 年, MathWorks 公司推出了具有划时代意义的 MATLAB 4.0, 并于 1993 年推出了其微机版, 可以配

合 Microsoft Windows 一起使用,使之应用范围越来越广。1994 年推出的 4.2 版本扩充了 4.0 版本的功能,尤其 在图形界面设计方面更提供了新的方法。

1997 年推出的 MATLAB 5.0 版允许了更多的数据结构,如单元数据、数据结构体、多维矩阵、对象与类等, 使其成为一种更方便编程的语言。1999 年初推出的 MATLAB 5.3 版在很多方面又进一步改进了 MATLAB 语言 的功能。

2000 年 10 月底推出了其全新的 MATLAB 6.0 正式版,在核心数值算法、界面设计、外部接口、应用桌面 等诸多方面有了极大的改进。

目前,经过 MathWorks 公司的不断完善,MATLAB 已经成为国际上最流行的科学与工程计算的软件工具, 现在的 MATLAB 已经不仅仅是一个"矩阵实验室"了,它已经成为了一种具有广泛应用前景的全新的计算机 高级编程语言了,它在国内外高校和研究部门正扮演着重要的角色。MATLAB 已经成为线性代数、自动控制理 论、数理统计、数字信号处理、时间序列分析、动态系统仿真等高级课程的基本教学工具;成为攻读学位的大 学生、硕士生、博士生必须掌握的基本技能。在设计研究单位和工业部门,MATLAB 被广泛用于科学研究和解 决各种具体问题。MATLAB 语言的功能也越来越强大,不断适应新的要求提出新的解决方法。MATLAB 所具 有的强大的数学运算能力、方便实用的绘图功能及语言的高度集成性,使它在其他科学与工程领域的应用越来 越广,并且有着更广阔的应用前景和无穷无尽的潜能。可以预见,在科学运算、自动控制与科学绘图领域 MATLAB 语言将长期保持其独一无二的地位。

### 1.2 MATLAB 的功能与特点

MATLAB 集成了数学计算、数据可视化,以及强大的编程语言为科技计算提供了灵活的环境。开放的体系 结构使得使用 MATLAB 和它的相关产品可以非常容易的分析数据、创建算法、创造方便实用用户工具。这些 特点使得 MATLAB 越来越广泛被应用于从个人计算机到超级计算机范围内的各种计算机上。

1.2.1 MATLAB 的功能

经过 Mathworks 公司的不断完善, MATLAB 已具备越来越多的功能:

1.数值计算功能

MATLAB 作为世界顶尖的数学应用软件,其出色的数值计算能力是使之优于其他数值计算软件的决定性因素之一。

2. 符号计算功能

数学计算有数值计算和符号计算之分,仅有优异的数值计算功能并不能满足解决数学计算问题时的需要。 在数学、应用科学和工程计算领域,常常会遇到符号计算问题。1993年,Mathworks公司从加拿大滑铁卢大学 购买了 Maple 的使用权,并以 Maple 的"内核"作为符号计算功能的"引擎",依靠 Maple 已有的库函数,实 现了 MATLAB 的符号计算功

3.数据分析和可视化功能

在科学计算和研究工作中,技术人员经常会遇到大量的原始数据,而对数据的分析往庄难于入手,如果能 将这些数据以图形的方式显示出来,不仅使数据间的关系清晰明了,而且对于揭示其内在本质往往有着非常重 要的作用。

4. 文字处理功能

MATLAB Notbook 为用户提供了强大的文字处理功能。MATLAB Notbook 允许用户从一个文字处理程序 (Microsoft Word)访问 MATLAB 的数值计算和可视化结果。

通过使用 MATLAB Notebook,用户可以方便地创建包含文本、MATLAB 命令及使用 MATLAB 得到的结果的文档。

通常情况下, MATLAB Notebook 可以用于创建:

(1) MATLAB 程序文档;

(2) 有关 MATLAB 的教科书;

(3) MATLAB 手册;

(4) MATLAB 技术报告;

(5) MATLAB 注释文档。

5. SIMULINK 动态仿真功能

SIMULINK 是 MATLAB 为模拟动态系统而提供的一个交互式程序。SIMULINK 允许用户在屏幕上绘制框 图来模拟一个系统,并能够动态地控制该系统。SIMULINK 采用鼠标驱动方式,能够处理线性、非线形、连续、 离散等多种系统。

此外, SIMULINK 还为用户提供了两个应用程序扩展集,包括 Simulink Extensions 和 Blocksets。

Simulink Extensions 是支持在 Simulink 环境中进行系统开发的一些可选择的工具类应用程序,包括了以下 几个工具:

(1) Simulink Accelerator;

(2) Real-TimeWorkshop;

(3) Stateflow.

Blocksets 是在特殊应用领域中设计的 Simulink 程序的集合。Blocksets 包括以下几个领域的应用程序集:

(1) DSP (数字信号处理);

(2) Fixed-Point (定点);

(3) NonlinearControlDesign (非线性控制设计);

(4) Communications (通信)。

1.2.2 MATLAB 的特点

1. 功能强大

MATLAB 不但在数值计算和符号计算方面具有强大的功能,而且在计算结果的分析和数据可视化方面也有 着其他类似软件难以匹敌的优势。此外,MATLAB 的 Notebook 为用户提供了把数学和文字进行统一处理的功能,而 MATLAB 的 Simulink 功能则将其应用扩展到更广的领域。

不仅如此, Mathworks 公司更推出了针对各专业应用的 MATLAB 工具箱。不计算第三方对 MATLAB 的支持软件,仅 Mathworks 公司本身就推出了 30 多个针对不同领域的科学应用的 MATLAB 工具箱。

2. 界面友好、编程效率高

MATLAB 是一种以矩阵计算为基础的程序设计语言,其指令表达方式与标准教科书的数学表达式非常接近。用户不需要有较高的计算机编程基础,只要按照计算要求输入表达式,MATLAB 将为用户计算出结果。

此外,使用 MATLAB 语言设计的程序,其编译和执行速度都远远超过了传统的 C 和 Fortran 语言设计的程序,可以说,MATLAB 在工程计算方面的编程效率远远高于其他编程语言。

3. 扩展性强

MATLAB 的最重要的特点之一就是其可扩展性,这个特点使得用户能够自由地开发自己的应用程序,这些 年来,许多使用 MATLAB 的数学家、工程师和科学家已经开发出相当多的不同应用领域的应用程序。

MATLAB 的这些特点使它获得了对应用学科,特别是对边缘学科和交叉学科的极强的适应能力,并很快成为应用学科计算机辅助分析、设计、仿真以至教学等不可缺少的基础软件。MATLAB 在国外的高校已成为大学 生、硕士生、博士生必须掌握的基本程序设计语言。而在国外的研究设计单位和工业部门,更是早已成为研究 和解决工程计算问题的一种标准软件。

### 1.3 MATLAB 系统的组成

MATLAB 系统主要由 5 个部分组成:

1.开发环境

这是帮助用户管理和使用 MATLAB 功能和文件的设备集合。这些工具的许多是面向用户的图形界面。它 包括了 MATLAB 桌面、命令窗口,命令历史,工作空间,帮助系统,以及文件浏览器和搜索路径浏览器。

### 2. MATLAB 数学函数库

MATLAB 的数学函数库极其庞大,既包括了象求和(sum),正弦函数(sin),余弦函数(cos)一样的基本函数,也包括了象矩阵求逆,矩阵求本征值,Bessel函数,Fourier 变换一样复杂的算法。

3. MATLAB 语言

MATLAB 语言是以矩阵/数组为基本数据单位的一种高级语言,具有控制流语句、函数、数据结构、输入输出及面向对象等编程特点。它既适用于编写简短的小程序,也适用于开发复杂的大型程序。

4. 句柄图形

这是 MATLAB 图形系统。它既包括了二维和三维的数据的可视化、图象处理、动画制作等高级绘图命令, 也包括了可以完全修改和编制部分或完整图形界面的低层次绘图命令。

5. MATLAB 应用程序接口(API)

这是允许用户在 C 和 Fortran 语言程序与 MATLAB 之间相互调用,交流数据的一个库。它使用户即能从 C 和 Fortran 程序中调用 MATLAB 程序,也能 MATLAB 把当作计算引擎在 MATLAB 里读写 MAT 文件。

### 1.4 MATLAB 工具箱及应用简介

#### 1.4.1 MATLAB 工具箱

在本章第一节已经介绍过,MATLAB 由主包和各种工具箱组成。主包是 MATLAB 的核心,工具箱实际上 是扩展的有专门功能的函数。

不同的工具箱给不同领域的用户提供了丰富且强大的功能。如为控制领域提供的线性矩阵不等式控制工具 箱(LMI Control Toolbox),为信号处理领域提供的小波工具箱(Wavelet Toolbox),为神经网络领域提供的神经 网络工具箱(Neural Network Toolbox),为模糊逻辑领域提供的模糊逻辑工具箱(Fuzzy Toolbox)等。目前, 已有涉及自动控制、信号处理、图像处理、经济、数学等多种学科的 30 多种 MATLAB 工具箱投入应用。 应用 MATLAB 的各种工具箱可以在很大程度上减小用户编程时的复杂程度,而 Mathworks 公司也一直致力于 追踪各学科的最新进展,并及时推出相应功能的工具箱。毫无疑问,MATLAB 能在数学应用软件中成为主流是 离不开各种功能强大的工具箱的。

1.4.2 MATLAB 主要工具箱简介

1. 控制系统工具箱

控制领域的计算机辅助设计自产生以来就一直受到控制界的重视。而 MATLAB 正是控制领域进行计算及 辅助设计的一种非常好的工具语言。

MATLAB 的控制系统工具箱(Control System Toolbox)为用户提供了许多控制领域的专用函数,实际上, 这个工具箱就是一个关于控制系统的算法的集合。通过使用这些专用函数,月户可以方便地实现控制系统的部 分应用。此外,使用 MATLAB 的控制系统工具箱还可以方便地进行模型间的转换。下面列出了该工具箱在控 制领域的主要应用:

- (1) 连续系统设计和离散系统设计;
- (2)传递函数和状态空间;
- (3) 模型转换;
- (4)频域响应;
- (5)时域响应;
- (6) 根轨迹和极点配置。

2. 小波工具箱

小波工具箱(Wavelet Toolbox)在信号处理领域的主要应用包括:

(1) 基于小波的分析和综合;

(2)图形界面和命令行接口;

4

- (3) 连续和离散小波变换及小波包;
- (4)一维、二维小波;
- (5) 自适应去噪和压缩。
- 3. 模糊逻辑工具箱

模糊逻辑工具箱 (FuzzyLogicToolbox) 是 MATLAB 用于解决模糊逻辑问题的工具箱。其主要应用包括:

- (1)友好的交互设计界面;
- (2) 自适应神经——模糊学习、聚类以及 Sugeno 推理;
- (3) 支持 SIMULINK 动态仿真;
- (4)可生成 C 语言源代码用于实时应用。
- 4. 神经网络工具箱

神经网络工具箱(NeuralNetworkToolbox)的主要应用包括:

- (1) BP 网络;
- (2) Hopfield, Kohonen 网络:
- (3) 径向基函数网络:
- (4) 竞争、线性、Sigmoidal 等传递函数;
- (5)前馈、递归等网络结构;
- (6)性能分析及应用;
- (7) 感知器:
- (8) 自组织网络。
- 5. 通信工具箱

通信工具箱(Communication Toolbox)提供了 100 多个函数和 150 多个 SIMULINK 模块用于通信系统的仿 真和分析,其主要应用包括:

- (1)信号编码;
- (2) 调制解调;
- (3) 滤波器和均衡器设计;
- (4)通道模型;
- (5)同步:
- (6) 多路访问;
- (7)错误控制编码。
- 6.线性矩阵不等式控制工具箱

线性矩阵不等式控制工具箱 (LMI Control Toolbox)的主要应用包括:

- (1) LMI 的基本用途;
- (2) 基于 GUI 的 LMI 编辑器 LMI 问题的有效解法;
- (3) LMI 问题解决方案。
- 7.图像处理工具箱
- 图像处理工具箱(Image Processing Toolbox)的主要应用包括:
- (1) 二维滤波器设计和滤波输入;
- (2)图像恢复增强;
- (3) 色彩、集合及形态操作;.
- (4) 二维变换;
- (5)图像分析和统计。
- 8.优化工具箱
- 优化工具箱 (Optimization Toolbox)的主要应用包括:
- (1)线性规划和二次规划;
- (2) 求函数的最大值和最小值;
- (3)多目标优化;

- (4)约束条件下的优化;
- (5) 非线性方程求解。
- 9. m分析与综合工具箱
- m分析与综合工具箱(m-Analysis and Synthesis Toolbox)主要应用包括:
- (1) m分析与综合;
- (2) H₂和 H∞最优综合;
- (3) 模型降阶;
- (4) 连续和离散系统;
- (5) m分析与综合理论。
- 10. 偏微分方程工具箱

#### 偏微分方程工具箱 (Partial Differential Equation Toolbox)的主要应用包括:

- (1) 二维偏微分方程的图形处理;
- (2) 几何表示;
- (3) 自适应曲面绘制;
- (4)有限元方法。
- 11. 高阶谱分析工具箱
- 高阶谱分析工具箱 (Higher-Order Spectral Analysis Toolbox)的主要应用包括:
- (1) 高阶谱估计;
- (2) 信号中非线性特征的检测和刻画;
- (3) 延时估计;
- (4)幅值和相位重构;
- (5) 阵列信号处理;
- (6) 谐波重构。
- 12. 频率域系统辩识工具箱

频率域系统辩识工具箱 (Frequency Domain System Identification Toolbox)的主要应用包括:

- (1)辨识具有未知延迟的连续和离散系统;
- (2) 计算幅值 / 相位、零点 / 极点的置信区间;
- (3)设计周期激励信号、最小峰值、最优能量谱等。
- 13.财政金融工具箱

财政金融工具箱 (Financial Toolbox)的主要应用包括:

- (1) 成本、利润分析;
- (2)市场灵敏度分析;
- (3) 业务量及分析优化;
- (4) 偏差分析;
- (5)资金流量估算;
- (6)财务报表。
- 14. 模型预测控制工具箱
- 模型预测控制工具箱 (Model Predictive Control Toolbox)的主要应用包括:
- (1)建模、辨识及验证;
- (2) 支持 MISO 模型和 MIMO 模型;
- (3) 阶跃响应和状态空间模型。
- 15. 样条工具箱
- 样条工具箱 (Spline Toolbox)的主要应用包括:
- (1) 分段多项式和 B 样条;
- (2) 样条的构造;
- (3)曲线拟合及平滑;

- (4) 函数微分、积分。
- 16. 统计工具箱

统计工具箱 (Statistics Toolbox)的主要应用包括:

- (1) 概率分布和随机数生成;
- (2) 多变量分析;
- (3)回归分析;
- (4) 主殛分析;
- (5) 假设检验。
- 17. 信号处理工具箱

信号处理工具箱 (Signal Processing Toolbox)的主要应用包括:

(1) 数字和模拟滤波器设计、应用及仿真;

- (2)参数化模型:
- (3) 谱分析和估计;
- (4) FFT 变换;
- (5) DCT 变换。

### 1.5 MATLAB 6 的新特性

与 MATLAB 的其他版本相比, MATLAB 6 提供的新桌面前端使用户可以更方便快捷的访问 MATLAB 的代码、变量、数据文件、图形, 演示和联机帮助。另外, MATLAB 6 提供的众多新的交互工具使 MATLAB 图形的导入、导出、绘制更为简便。

在 MATLAB 6 中,数学计算、外部数据的存取以及编码得到了重大的改进,GUI 也得到了进一步发展。 MATLAB 6 的新特点包括了:

● 提供了一个崭新的桌面前端,包含命令窗口,命令历史窗口,工作区浏览器,数组编辑器等等,使 MATLAB 环境的管理更加灵活。

为交互地图形编辑及注释提供了新的编辑工具,使需要记忆的编码和图形命令、属性大大减少。

● 改进了数学计算和算法:优化了 LAPACK 库,使矩阵计算更快捷;提供了新的 FFTW 库,提升了基于 FFT 的功能表现;增加了基于 Qhull 的功能;提供了新的微分的方程解法,以及众多更精确的算法。

为数据图的快速分析提供了新的数据统计分析和拟和工具。

● 提供了更先进的数据可视化手段:2-D 图象、表面以及作为透明对象的卷的显示;提供了为控制前景的交互式照相机工具菜单。

- 为调用 Java 程序和在 MATLAB 直接使用 Java 对象提供了接口。
- 提供了实现 MATLAB 与外部的仪器交流的串口通讯接口。
- 加强了用以开发当前的接口和界面的 GUI 设计工具 (GUIDE)。
- 集成了微软 Visual Studio 的扩展程序。

### 1.6 其他数学类科技软件

在当今众多数学类科技应用软件中,就软件数学处理的原始内核而言,可分为两大类。一类是数值计算型 软件,如 MATLAB、Xmath、Gauss 等,这类软件长于数值计算,能够高效率的处理大批数据;另一类是数学 分析型软件,如 Mathematica、Maple 等,这类软件以符号计算见长,能给出解析解和任意精度解,其缺点是处 理大量数据时效率较低。

目前在科技和工程界上比较流行和著名的数学软件主要有四个,分别是 Maple V、MATLAB、MathCAD 和 Mathematica。它们在各自针对的目标都有不同的特色。

### 1.6.1 Maple

Maple 是由 Waterloo 大学开发的强大的交互式数学系统软件,它不但具有精确的数值处理功能,而且具有 无以伦比的符号计算功能。它提供的进行符号代数表达式处理的数学环境,可表示任意精度的数字和进行二维 及三维作图、设计。Maple V 的符号计算能力还是 MathCAD 和 MATLAB 等软件的符号处理的核心。Maple 提 供了 2700 余种数学函数,涉及范围包括:普通数学、高等数学、线性代数、数论、离散数学、图形学。它还提 供了一套内置的编程语言,用户可以开发自己的应用程序,而且 Maple 自身的 2700 多种函数,基本上是用此 语言开发的。

Maple 采用字符行输入方式,输入时需要按照规定的格式输入,虽然与一般常见的数学格式不同,但灵活 方便,也很容易理解。输出则可以选择字符方式和图形方式,产生的图形结果可以很方便地剪贴到 Windows 应 用程序内。

Maple 提供的超过 2700 个函数,可运用于多种科技和工程应用软件,并且能应用于多种不同的计算机系统。 Maple 为 UNIX 系统和 Macintosh 系统用户增加了多项额外的功能,并且支持用户使用 URL 超级连接功能,来 直接连接到 WEB 页上的 Maple 工作表。

### 1.6.2 MathCAD

MathCAD 是美国 Mathsoft 公司推出的一个交互式的数学系统软件。从早期的 DOS 下的 1.0 和 Windows 下的 4.0 版本,到现在的 8.0 版本,功能也从简单的数值计算,直至引用 Maple 强大的符号计算能力,使得它发生了一个质的飞跃。

MathCAD 是集文本编辑、数学计算、程序编辑和仿真于一体的软件。MathCAD 的主要特点是输入格式与 人们习惯的数学书写格式很近似,采用 WYSWYG(所见即所得)界面,特别适合一般无须进行复杂编程或要 求比较特殊的计算。MathCAD 还带有一个程序编辑器,对于一般比较短小,或者要求计算速度比较低时,也是 可以采用它的。这个程序编辑器的优点是语法特别简单。

MathCAD 可以看作是一个功能强大的计算器,没有很复杂的规则;同时它也可以和 Word、Lotus、WPS2000 等字处理软件很好地配合使用,可以把它当作一个出色的全屏幕数学公式编辑器。Mathcad 8 还内嵌了 IE,以 促进网络间的协作。该版本使你能够将数字符号、图表和文本结合在一个独特的交互式工作表中,并且增强了 计算能力,使用户能通过它自动解决线形和非线形计算问题。具有更先进的图形界面和可视化功能,对用户来 说更加易用了。

#### 1.6.3 Mathmatica

Mathematica 是由美国物理学家 Stephen Wolfram 领导 Wolfram 研究所开发的数学系统软件。它拥有强大的数值计算和符号计算能力,在这一方面与 Maple 类似,但它的符号计算不是基于 Maple 上的,而是自主开发的。

Mathematica 的基本系统主要是用 C 语言开发的,因而可以比较容易地移植到各种平台上,Mathematica 是 一个交互式的计算系统,计算是在用户和 Mathematica 互相交换、传递信息数据的过程中完成的。Mathematica 系统所接受的命令都被称作表达式,系统在接受了一个表达式之后就对它进行处理,然后再把计算结果返回。 Mathematica 对于输入形式有比较严格的规定,用户必须按照系统规定的数学格式输入,系统才能正确地处理, 因此以前版本输入指令时需要。

在 Mathematica 3.0 中有了更多的新特性和更强大的功能。在数值计算中,能够控制计算结果的精度,增强 和精简优化了代数表达式,增强了函数功能。对于图形,能够自动转换成包括 EPS、TIFF、GIF 等的多种格式。 支持"所见即所得",提供了 700 种应用于数学和其他领域的特殊符号。加快了计算的执行速度,降低了对存储 器的要求。引入了输入面板,并且可以修改、重组输入面板,改善了不断切换大小写字符的繁琐方式。

### 1.6.4 MATLAB

如前所述,MATLAB 原是矩阵实验室(Matrix Laboratory)在70年代用来提供Linpack和 Eispack软件包的接口程序,采用 C 语言编写。从80年代出现3.0的 DOS 版本,逐渐成为科技计算、视图交互系统和程序语言。MATLAB 可以运行在十几个操作平台上,比较常见的有基于 Windows 9X/NT、OS/2、Macintosh、Sun、Unix、Linux 等平台的系统。

8

MATLAB 程序主要由主程序和各种工具包组成,其中主程序包含数百个内部核心函数,工具包则包括复杂 系统仿真、信号处理工具包、系统识别工具包、优化工具包、神经网络工具包、控制系统工具包、 µ 分析和综 合工具包、样条工具包、符号数学工具包、图像处理工具包、统计工具包等。

MATLAB 是数值计算的先锋,它以矩阵作为基本数据单位,在应用线性代数、数理统计、自动控制、数字 信号处理、动态系统仿真方面已经成为首选工具,同时也是科研工作人员和大学生、研究生进行科学研究的得 力工具。MATLAB 在输入方面也很方便,可以使用内部的 Editor 或者其他任何字符处理器,同时它还可以与 Word 结合在一起,在 Word 的页面里直接调用 MATLAB 的大部分功能,使 Word 具有特殊的计算能力。

如果仅是要求一般的计算或是普通用户日常使用,首选 MathCAD,它在高等数学方面所具有的能力,足够一般客户的要求,它的输入界面也特别友好。如果要求计算精度、符号计算和编程方面的话,最好同时使用 Maple 和 Mathematica,它们在符号处理方面各具特色,可以优势互补。如果要求进行矩阵方面或图形方面的处理,则选择 MATLAB,矩阵计算和图形处理是它的强项。

### 1.7 小结

在本章中,简单地介绍了 MATLAB 的产生和发展历程、MATLAB 的主要组成和工具箱的功能,介绍了 MATLAB 的主要特点、MATLAB6 的新特性以及 MATLAB 和 Maple、MahCAD、Mathmatica 的比较。通过本 章的学习,可以对 MATLAB 的用途和特点有一个基本的了解。

# 第二章 MATLAB 的安装和基本操作

要顺利地使用 MATLAB 6, 首先要做的工作就是安装 MATLAB 6, 这是正常使用 MATLAB 的重要保证。 本章主要介绍以下内容:

- MATLAB 6 的安装
- MATLAB6的启动与退出
- MATLAB 6 的卸载
- MATLAB 的目录结构
- MATLAB 6 的开发环境
- MATLAB 的帮助系统

# 2.1 MATLAB 6 的安装

### 2.1.1 系统要求

MATLAB 6 比以前的版本提供了更强大的功能和更完善的帮助系统,当然,它也对系统硬件提出了更高的要求。以下是在 PC 机上安装 MATLAB 6 的资源要求:

- 基于 Intel Pentium, Pentium Pro, Pentium II, Pentium III 或更高主频的 PC 机
- Microsoft Windows 95, Windows 98, Windows NT 4.0 或 Windows 2000 的操作系统
- 光驱(MATLAB 也能够通过 TCP/IP 协议进行网络安装)
- 内存要求:至少要 64MB 内存,推荐 128 MB 内存
- 至少8位显示适配器和至少能显示 256 色彩色显示器

● 磁盘空间的大小取决于采用的分区形式及是否安装联机的帮助文件。MATLAB 安装程序将通知你对安装分区硬盘空间的要求。

其它推荐的项目还包括:

- 微软 Windows 支持的图形加速器卡
- 微软 Windows 支持的打印机
- 微软 Windows 支持的声卡
- 如果用户要运行 MATLAB Notebook,则需要预先安装 Microsoft Word 8.0 (Office 97),或 Office 2000。 如果用户想创建自己的 MEX 文件则需要安装下列之一的编译器:
- Compaq 的 Visual Fortran 5.0 或 6.1
- 微软的 Visual C/C++ 5.0 或 6.0
- Borland C/C++ 5.0 , 5.02
- Borland C++ Builder 3.0, 4.0, 或 5.0
- Lcc 2.4 (已与 MATLAB 捆绑)

用户还需要安装 Adobe Acrobat Reader 来阅读和打印 PDF 格式的 MATLAB 帮助文档。

- 国际 用户可以登陆 MathWorks 的网站 www.mathworks.com 察看最新相关信息。
- 2.1.2 MATLAB 6 的安装过程

随着 MATLAB 版本的更新,安装也越来越简便。对于 MATLAB 5.3,用户只要按照安装界面的提示逐步进

行。

将 MATLAB 6 光盘放入光驱,对于微软 Windows 系列操作系统,安装程序都会自动开始运行。

如果安装程序不启动,用户还可以打开一个 Windows 资源管理器窗口,单击你的 CDROM 驱动器, 并且在顶层目录双击 setup.exe 程序,从而启动安装程序。

首先出现的如图 2-1 所示的初始画面,随后在安装窗口中出现"Welcome to the MathWorks Installer"(欢迎 安装 MATLAB)对话框,如图 2-2 所示。



图 2-1 安装初始画面



图 2-2 欢迎安装对话框

单击"Next"按钮,将继续安装,如果单击"Cancel"按钮,将中断中断安装程序。下一步出现的是用户账号的输入窗口,如图 2-3 所示,用户应在相应的位置输入 32 位的用户账号,单击"Next"按钮,如果密码输入正确,安装过程将继续,这时出现如图 2-4 所示的"Software License Agreement"(软件许可协议)对话框,选择"No"不接受协议,将中断安装;单击"Back"按钮,将返回上一步;选择"Yes"接受协议内容,将出现"Customer Information"(用户信息)对话框,如图 2-5 所示。在"Name"和"Comply"中分别输入自己的姓名和公司名字,单击"Next"按钮,继续安装程序。



图 2-3 用户账号输入窗口

	The MathWorks, Inc. Software License Agreement	
	<pre>iead the terms and conditions of this Agreement carefully before nstalling the Frograms. This License Agreement represents the ndividual or an entity) and The MatHWorks [("MatHWorks") oncerning the Frograms and the accompanying user Documentation. by installing the Programs, you accept the terms of the Agreement. f you are not villing to do so, do not continue with the mstallation.</pre>	-
I	o you accept the terms of the license agreement? If these terms are agreemble to you, please indicate so by clicking es button below. Installation will not continue if you do not click the Yes button.	the 1

图 2-4 软件许可协议对话框

Customer Information	
KETEASE	Please enter your name and company below: Name: Company:
	Toth Total Total

图 2-5 用户信息对话框

下一步出现的是"Product List"(产品列表)对话框,如图 2-6 所示,在此对话框中包含了四项对话框内容:

(1)选择安装路径,安装程序默认路径是"c:\matlabR12",用户可以根据自己系统的具体情况,自己指定 安装目录,但要保证安装路径上有足够的剩余硬盘空间。在"Product List"对话框的右端,安装程序根据用户 安装要求给出了安装所需的硬盘空间,以及相应硬盘分区的剩余空间。

(2)选择是否安装产品文档。如果用户选择安装文档,安装程序将把 HTML 文件拷贝进 MATLAB 安装目录。用户也可以选择从 CD 上阅读文档、在 MATLAB 安装目录以外的路径安装,或者是选择以后再安装。

🔉 安装程序并不安装文档的 PDF 版本,用户可以在安装 CD 或在 MathWorks 网站上获得 PDF 文档。

(3) 如果安装产品文档,选择文档语言设置。

(4)从产品列表中选择用户想要安装的项目。安装程序默认是准许安装所有产品,如果用户不想安装某一 个产品,可以单击它的名字前复选框中的标记取消选中。对与安装过程中取消安装的任一部件,用户都能够在 以后安装它。



图 2-6 产品列表对话框

单击"Next"按钮,将出现 Matlab 6 的安装进度条,如图 2-7 所示。

MATLAB 6° provides new tools that make it easy to access the power of MATLAB and interface with external programs and data. These tools include:	
<ul> <li>The MATLAB develop, a user interface for managing data, Hist, and applications</li> <li>Easy-to-use plating tools for interactive plot creation</li> <li>Data Statistics and Base Tiriting instructions for guide analysis</li> <li>Import Witzard for previewing and importing data files in a range of formatis</li> <li>Area interface for calling Jerva and loading Jeva objects from MATLAB</li> <li>Senial port interface to communicate with devices from MATLAB</li> </ul>	
% complete: 1%	Cancal

图 2-7 文件安装进度

文件安装结束后,将出现安装结束提示,用户要重新启动系统才可以使用 Matlab 6。

# 2.2 MATLAB 6 的启动与退出

#### 2.2.1 MATLAB 6 的启动

用户可以使用以下几种方法启动 MATLAB:

(1)使用安装程序在桌面上创建的快捷方式,如图 2-8 所示,在此快捷方式上双击即可开始运行 MATLAB 6。

(2) 单击"开始"按钮,在"程序"菜单中找到"MATLAB Release 12",如图 2-9 所示,单击"MATLAB R12"就可启动 MATLAB 6。

(3) 单击"开始"按钮, 然后单击"运行"按钮, 在"运行"对话框中输入"matlab"也可启动 MATLAB 6, 如图 2-10 所示。

A
MATLAB R12



图 2-8 MATLAB 的快捷方式

图 2-9 "开始"中的 MATLAB 菜单



图 2-10 "运行"程序对话框

使用上述几种方法都可以启动 MATLAB 6,在出现欢迎画面后,出现 MATLAB 6的工作环境,如图 2-11 所示。



图 2-11 MATLAB 6 的工作环境

2.2.2 退出 MATLAB 6

退出 MATLAB 的方式也有很多种:

- (1) 用鼠标单击工作环境窗口右上角的关闭按钮"X"。
- (2) 用鼠标双击工作环境窗口左上角的 MATLAB 图标。
- (3) 使用快捷键 "Ctrl+Q"。
- (4) 在命令窗口 (command window) 输入"quit"命令。
- (5) 在命令窗口输入" exit" 命令。
- (6)在 MATLAB 命令窗口的 "File"菜单下选择 "Exit MATLAB", 如图 2-12 所示。

•
+0
. <b>+</b> ₩
.+S
+Q

图 2-12 MATLAB 6 的 "File " 菜单

## 2.3 卸载 MATLAB 6

当需要卸载 MATLAB 时,单击"开始"按钮,依次指向"程序"、"MATLAB Release 12"、"R12 uninstaller", 如图 2-9 所示,单击"R12 uninstaller",将出现"Uninstall Product List"(卸载产品列表)对话框,如图 2-13 所示,单击"Cancel"出现让用户确认是否终止卸载程序的对话框,如图 2-14 所示;单击"Next"将开始进行卸载,显示如图 2-15 所示的卸载进度条。卸载完成后,由用户单击"Finish"(结束)退出卸载程序。

#### MATLAB 6 实用教程



图 2-13 "Uninstall Product List"对话框



图 2-14 确认终止卸载对话框

📣 Uninstall status	
	Removing files for Motorola ISP Developer's Kit
ELEASE	Emist. Cancel
~	

图 2-15 卸载进度

▶ 用户还可以使用"开始"、"设置"、"控制面板"、"添加/删除程序",选中"MATLAB Family of Products Release 12"运行卸载程序,或者在 MATLAB 6 的安装路径下的"uninstall"子目录中, 双击"uninstall.exe"启动卸载程序。

卸载程序不会自动删除安装路径下的"uninstall"和"work"子目录,要完全卸载 MATLAB 6,用户需要 自己手工删除遗留的 MATLAB 目录。

# 2.4 MATLAB 的目录结构

MATLAB 在安装盘上以压缩形式分布,安装程序把这些文件解压缩并安装到 Windows 环境下。由于 MATLAB 文件众多,对于 MATLAB 的用户,了解 MATLAB 安装后的目录结构是很必要的。安装完成后 MATLAB 安装目录下将包括表 2-1 所示的一些文件和子目录(以 MATLAB 在 IBM-PC 及其兼容机上安装后的目录结构为例)。

王	2 1
18	2-1

### MATLAB 的主要目录结构

文件及子目录	所包含文件的功能
MATLAB R12.exe	启动 MATLAB 的快捷方式
License.txt	包含 MATLAB 软件许可协议
\TOOLBOX	工具箱
\EXTERN	建立外部软件和 MATLAB 的接口工具
\HELP	帮助系统
\GHOSTSCRIPT	解释和运行页面描述文件(PostScript)的共享软件组
\DEMOS	示范程序
\JA	国际化文件
\JAVA	Java 库文件
\UNINSTALL	卸载文件目录
\SYS	运行 MATLAB 需要的工具及操作系统相关库文件
\WORK	缺省的当前目录
\BIN\WIN32	二进制代码和相关文件

在\TOOLBOX\MATLAB 目录下的内容为 MATLAB 核心部分工具包,这些子目录如表 2-2 所示。

±	~ ~	
衣	2-2	

### \TOOLBOX\MATLAB 目录下的工具包

子目录	所包含文件的功能
AUDIO	声音控制函数
\DATAFUN	数据分析和 Fourier 变换函数
\DATATYPES	数据类型和结构
\DEMOS	示例
\ELFUN	基本的数学函数
\FUNFUN	功能函数
\ELMAT	基本矩阵和矩阵操作函数
\GENERAL	通用命令
\GRAPH2D	绘制二维图形的函数
\GRAPH3D	绘制三维图形的函数
\GRAPHICS	通用绘图命令
\IOFUN	低级文件 I/O 函数
\MATFUN	矩阵函数——数值线性代数
\LANG	语言结构设计和调试函数
\SPARFUN	稀疏矩阵函数
\SPECGRAPH	特殊图形函数
\SPECFUN	特殊数学函数
\TIMEFUN	时间、日期和日历函数
\STRFUN	字符串函数
\POLYFUN	多项式和插值函数
\UITOOLS	GUI设计工具
WINFUN	视窗系统接口函数
\OPS	运算符和特殊符号
VERCTRL	版本控制函数

表 2-3 列出了 MATLAB 中其他重要的工具箱。

=	0	2
রন্থ		٠٦
~	_	~

\TOOLBOX 目录下的其他工具包

子目录	所包含文件的功能
\SIMULINK	SIMULINK 工具箱
\STATEFLOW	STATEFLOW 工具箱
\CDMA	CDMA工具箱
\COMM	通信工具箱
\CONTROL	控制系统工具箱
\DSPBLKS	数字信号处理模块集
\DAQ	数据采集工具箱
\DATABASE	数据库工具箱
\DATAFEED	数据流工具箱
\DIALS	计量仪表模块集
\EXLINK	EXCEL 接口工具
\FILTERDESIGN	筛选设计工具箱
\FINANCE	金融工具箱
\FIXPOINT	固定点模块集
\FUZZY	模糊逻辑工具箱
\GARCH	GARCH 工具箱
\IMAGES	图像处理工具箱
\INSTRUMENT	仪器控制工具箱
\LMI	线性矩阵不等式工具箱
\COMPILER	MATLAB 编译工具箱
\RPTGEN	MATLAB 报告发生器
\MAP	地图绘制工具箱
\MPC	模型预测控制工具箱
\MOTDSP	摩托罗拉 DSP 开发工具集
\MUTOOLS	μ分析与综合工具箱
\NNET	神经网络工具箱
\NCD	非线性控制系统设计模块集
\OPTIM	最优化工具箱
\PDE	偏微分方程工具箱
\POWERSYS	动力系统模块集
\RTW	Real-Time Workshop 工具箱
\ROBUST	鲁棒控制工具箱
\SB2SL	SYSTEMBUILD 到 SIMULINK 的转换器
\SIGNAL	信号处理工具箱
\RPTGENEXT	SIMULINK 报告发生器
\SPLINES	样条工具箱
\STATS	统计工具箱
\SYMBOLIC	符号数学工具箱
\IDENT	系统识别工具箱
\WAVELET	小波工具箱

\EXTERN 目录下的是建立 MATLAB 外部程序接口所需的工具,该目录包括了四个子目录,如表 2-4 所示。

表 2-4

\EXTERN 的子目录

子目录	所包含文件的功能
\EXAMPLES	应用 C 语言和 FORTRAN 语言的 API 程序实例
\LIB	外部接口程序的目标连接库
\INCLUDE	外部接口程序库的头文件
\SRC	建立 MEX 文件的 C 源程序举例

\GHOSTSCRIPT 目录下是解释和运行页面描述文件(PostScript)的共享软件组,如表 25 所示。

表 2-5	MATLAB\GHOSTSCRIPT 子目录
子目录	所包含文件的功能
\BIN	关于 PostScript 的二进制文件
\FONTS	PostScript 的字体文件
\PS_FILES	PostScript 初始化文件

### 2.5 MATLAB 环境

MATLAB6 与其它版本的一大区别就是其灵活、方便的桌面。新的 MATLAB 6 桌面为用户与 MATLAB 的 交互提供了灵活,直观的环境,使用户能够快速方便的存取数据文件、程序、图形以及联机帮助。这个新的桌 面配置包括了命令窗口(Command Window),命令历史窗口(Command History),发射台(Launch Pad),工作 区(Workspace),和数组编辑器(Array Editor),路径浏览器,帮助,调试与编译器等桌面工具。

### 2.5.1 MATLAB 集成桌面

启动 MATLAB 后,桌面显示如图 2-11 所示,最上面显示"MATLAB"字样的一栏为标题栏,标题栏的右端依次为窗口最小化、窗口缩放和关闭窗口的按钮。标题栏下面的菜单包含了"File(文件)"、"Edit(编辑)" "Web(网页)"、"View(视图)"、"Window(窗口)"、"Help(帮助)"六项。菜单栏下面的工具栏显示了九个 工具按钮和路径浏览器,熟练的使用工具按钮可以使工作更快捷、方便。

### 1. 桌面的菜单栏

MATLAB 的桌面菜单由六项主菜单组成,下面分别介绍各选项的作用。

(1) 单击菜单栏上的"File"选项,就会显示下拉式子菜单,如图 2-16 所示。



图 2-16 "File"选项的下拉式菜单

> New

File 菜单下的子菜单 New 有四个选项:

" M-file " 新建一个 M 文件,该命令将打开 MATLAB 的 M 文件编辑/调试器。通过 M 文件编辑/调试器, 用户可以创建自己的 M 文件,也可以编辑已有的 M 文件,并可以调试 MATLAB 程序。

"Figure"新建一个空白图形窗口。

"Model" 自动打开仿真库并新建一个 Simulink 模型窗口。

"GUI"打开一个新的图形界面控制面板。在图形界面控制面板里,用户可以通过选中面板里的图形模板, 然后在打开的图形窗口中绘制就可以生成精美的图形界面。

> Open

弹出打开文件对话框,用户可以搜寻要打开的 MATLAB 文件所在目录,选中该文件,然后单击"打开"按钮,MATLAB 将用指定的编辑器打开已经建立的 M 文件。

Close Command Window

关闭命令窗口。

Import Data

单击将会弹出打开文件对话框,用户可以搜寻要导入的数据文件所在目录,选中该文件,然后单击"打开" 按钮对话框引入数据。

Save Workspace As

该项的作用是把当前工作空间的所有变量用后缀为 .mat 的文件保存起来。单击该选项将弹出一个保存文件 对话框 ,用户通过该对话框选择文件的存储目录和名字。

Set Path

单击该选项,将打开 MATLAB 的路径浏览器,如图 2-17 所示。用户通过路径浏览器可以更改 MATLAB 执行命令时的搜索路径。单击"Add Folder"、"Add with Subfolders"都将达目录搜索对话框,用户可以选中相应目录,添加 MATLAB 执行命令时的搜索路径。选中已经存在的 MATLAB 搜索路径,用户可以使用"Remove"删除它,或使用"Move to Top"、"Move Up"、"Move Down"、"Move to Bottom"更改 MATLAB 的搜索次序。

📣 Set Path		
All changes take effect im	mediately.	
	MATLAB search path:	
Add Folder	📄 D:\MATLABR12\toolbox\matlab\general	
Add with Out Officer	🛅 D:\MATLABR12\toolbox\matlab\ops	
Add with Suptoiders	📄 D:\MATLABR12\toolbox\matlab\lang	
	📄 D:\MATLABR12\toolbox\matlab\elmat	
Move to Top	📄 D:\MATLABR12\toolbox\matlab\elfun	
	📄 D:\MATLABR12\toolbox\matlab\specfun	
Mave Up	📄 D:\MATLABR12\toolbox\matlab\matfun	
Romovo	📄 D:\MATLABR12\toolbox\matlab\datafun	
Remove	📄 D:\MATLABR12\toolbox\matlab\audio	
Move Down	📄 D:\MATLABR12\toolbox\matlab\polyfun	
	D:\MATLABR12\toolbox\matlab\funfun	
Move to Bottom	📄 D:\MATLABR12\toolbox\matlab\sparfun	
	DiMATLABD12iteelboximetlebiareeb2d	<b>⊻</b>
Save Close	e Revert Default	Help

图 2-17 "Set Path"对话框

> Print

该选项用于打印命令窗口中的内容,也可以设置一些打印参数,如打印机的类型、纸张大小、送纸方向、 图形的打印质量等等。

Print Selection

当选中命令编辑区的一部分后,单击该选项即可按照设置好的打印机的参数打印被选中部分。

Exit MATLAB

单击"Exit MATLAB"选项,将关闭 MATLAB。关闭 MATLAB 的快捷键为 Ctrl+Q。

➢ 最近使用的四个文件显示在"Print Selection"选项和"ExitMATLAB"选项之间,单击文件名,即 可打开选中的文件。

Preferences

单击该选项,将打开如图 2-18 所示的命令窗口的显示格式选项卡。通过设置这张卡的选项,可以调整显示格式。

A Preferences	
⊡– General	Command Window Preferences
Command Window     Editor/Debugger     Gurrent Directory     Workspace     Array Editor     GUIDE     Figure Copy Template     Simulink	Text display          Numeric format:       short         Numeric display:       loose         Spaces per tab:       4         Display       Echo on         Limit matrix display width to eighty columns         Image: Enable up to 100       tab completions         Command session scroll buffer size:       Max
	OK Cancel Help

图 2-18 "Preferences"对话框

"Prefences"对话框主要包括了"General"、"Command Window"、"Editor/Debugger"、"Help"、"Current Directory"、"Workspace"、"Array Editor"、"GUIDE"、"Figure Copy Template"和"Simulink"等选项。

"General"选项卡主要用于控制桌面输出结果的字体、色彩等显示格式,打印机设置等。

"Command Window"选项卡选项卡主要用于控制命令编辑区的输出结果的显示格式,选择程序编辑器, 指定帮助目录,命令执行时是否回显,命令与命令之间以及变量名和变量的值之间是否留空行等。图 2-19 显示 了"Command Window"选项卡。上左边的"Numeric Format"框中,从"Short (default)"项到"Rational"项 是互斥的,通过选择这些选项可以控制命令窗口中计算结果的显示格式,这 里的数据输出的格式只影响数据 的显示格式,并不影响数据的存储和计算精度。"Echo On"选项可以选择运行 MATLAB 的程序时是否在命令 窗口中显示正在运行的命令。当选中时,显示运行的每条命令。值得注意的是,当选择显示运行的命令时,程 序的运行速度有所降低。本书在命令窗口一节中详细讲解了"Command window"选项卡中各选项意义。

"Figure Copy Template"选项卡如图 2-20 所示。该选项卡的功能是选择拷贝 MATLAB 所绘制的图形时的 形式。选中图 2-20 的"Clipboard Format"框中的"Metafile"选项,则图形拷贝时采用矢量图形格式:当选 中"Bitmap"选项,图形将以Windows的位图格式拷贝。选中图 2-20 中的"Size"选项,MATLAB 会利用其图 形的"Paper Position"属性,根据需要粘贴的纸张的大小,自动调整图形的大小;否则,将以图形的实际大小粘 贴。选中图 2-20 中的"Force White background"选项,粘贴时将选用白色背景。

### MATLAB 6 实用教程

APreferences	
🖅 General	Command Window Preferences
Command Window     Font & Colors     Editor/Debugger     Current Directory     Workspace     Array Editor     GUIDE     Figure Copy Template     Simulink	Text display          Numeric format.       short       Image: Short       Image: Short       Image: Short       Image: Short       Image: Short       Short       Image: Short       <
	OK Cancel Help

图 2-19 "Command Window"选项卡

📣 Preferences	
<ul> <li>General</li> <li>Command Window</li> <li>Font &amp; Colors</li> <li>Editor/Debugger</li> <li>Help</li> <li>Current Directory</li> <li>Workspace</li> <li>Array Editor</li> <li>GUIDE</li> <li>Figure Copy Template</li> <li>Copy Options</li> <li>Simulink</li> </ul>	Figure Copy Template Copy Options Preferences  Clipboard format  Metafile (may lose information)  Preserve information (metafile if possible)  Bitmap  Figure background color  Cuse figure color  Force white background  Transparent background  Size  Match figure screen size Select this option to copy the figure as it appears on screen, or leave it unchecked to use the Page Setup settings to determine its size.
	OK Cancel Help

图 2-20 "Figure Copy Template"选项卡

(2) 单击菜单栏上的"Edit"选项,就会显示编辑的下拉式子菜单,如图 2-21 所示,该下拉式菜单从上到下为:"Undo"、"Redo"、"Cut"、"Copy"、"Paste"、"Past Special"、"Select All"、"Delete"、"Clear Command Window"、"Clear Command History"、"Clear Workspace"共11项。其中:

- ▶ "Undo"、"Redo"的作用分别为撤销上一次操作和重复上一次操作。
- ➢ "Cut"、"Copy"、"Past"三项的作用分别为剪切、复制、粘贴,这四项和其他应用程序的相应菜 单没什么差别。

"Past Special"选项将启动 MATLAB 的导入数据对话框,如图 2-22 所示。"Column separator"选项要求用 户选择数据文件中列的分隔符。

### MATLAB 6 实用教程

		🥠 Import Wizar	1: E:/HOST/Fzm10	. dat					
		What column se	parator does you	r data use?					
		Column separ	ator						
Undo	Ctrl+Z	🔿 Comma	Space	C Semicolon	🔿 Tab	Other	T	Fext header line	es:
Redo		Preview of E:/HC	ST/Fzm10.dat						
		Time	R	CO	C1 🔺 💷	🖌 data 🗸 textdata 🔪			
Cut	CtrltX	19.98	28441.6	8.097	14 - C	Preview	truncated t	o 10x10	
		39.98	27324.7	7.81739					r
Copy	Ctr1+C	79.98	25925.4	7.46315		1	2	3	L
Posto	C+x1+V	99.98	25491.8	7.3525		1 0	30000	0.0000	L
Lasce	041.7	119.98	25171.1	7.27038		2 0		) 0	
Paste Sp <u>e</u> cial.		139.98	24932.4	7.20911		3 0	0	0 0	L
		179.90	24/55.0	7 12844		4 0	0	0 0	
Select All		199.98	24517	7.10214		5 0	C	0 0	
Detect MT		219.98	24439.4	7.08212		6 0	C	0 0	F
Delete		239.98	24380.1	7.06681		7 0	 (	1 0	t
-		259.98	24334.6	7.05506				0	⊢
C1 C	W 2 - 1	2/9.90	24299.6	7.04602		0		, ,	┝
crear command	<u>m</u> indow	319.98	24251.7	7.03364		91 U	L	JI II	ŀ
Clear Command	History								
				Cancol		lost » Einich	1		
Clear W <u>o</u> rkspac	:e			Cancer «	Back r	rimsh			

图 2-21 "Edit"的下拉菜单

- 图 2-22 导入数据对话框
- ➢ "SelectAll"选项用于选定文本编辑区的所有内容,以便进一步复制。
- ▶ "Delete"的作用是删除选中的内容。
- ➢ "Clear Command Window"、"Clear Command History"、"Clear Workspace"三项的作用分别是当 命令窗口、命令历史窗口、工作区窗口的内容很乱时,清除显示的全部内容。
- ➤ "Clear Command Window"项只是让视觉效果更好一点,但并不删除命令窗口、命令历史窗口、 工作区窗口的变量内容。
- (3) 单击"View"选项,将打开如图 2-23 所示的视图下拉菜单。

A MATLAB			_ 🗆 🗵
<u>File Edit View Web Window Help</u>			
🗋 😅 🧷 Desktop Layout	Default	work	
Launch Pad	Command Window Only		Z X
F MATI Command Window	Simple Short History		A
Command History	Tall History	l, select "MATLAB Help" from the Help menu.	
← Cont     ✓ Current Directory     ✓ Torkspace	<u>F</u> ive Panel		
🕂 📣 Date 🖌 Launch Pad			
H- Date Help			
H- Filt Current Directory Filter			
🕂 📣 Fine 🛛 Torkspace View Options 🔸			
🖅 📣 Financial Time Series Toolbox			
E- I Financial Toolbox	<b>T</b>		
Lounsh Bod     Worksnare			
Callen Paul Wonopuce			
Command History	N X		
% 10:04 PM 6/30/01%			
Command History Current Directo	ny 🔳		
Ready			

图 2-23 "View"的下拉菜单

MATLAB 6 的视图是非常灵活方便的,在 MATLAB 的桌面环境中包含了"Lanch Pad"、"Command History"、 "Command Window"、"Workspace"和"Current Directory"五个窗口,在视图菜单栏上"Undock Workspace" 和"Help"之间,用户可以分别单击这五个桌面工具的标签,去掉其前面的符号"",这样就可以关闭该窗口, 再次选中就可以打开该窗口。

"Desktop Layout"选单中有"Default"、"Command Window Only"、"Simple"、"Short History"、"Tall History"和"Five Panel"六个选项,分别预设了六种桌面的视图形式,用户可以通过单击选项更改桌面的视图形式:

- "Default (缺省)"包含命令窗口,在一个表单框中的命令历史和当前的目录浏览器窗口,和在一 个表单框中发射台和工作区浏览器窗口。可以单击窗口标签,切换同一表单框中的浏览窗口。
- ➢ "Command Window Only (仅有命令窗口)"仅仅包含命令窗口,这使 MATLAB 看起来显得类似 于先前的版本。

- ➤ "Simple (简单形式)"只包含命令历史和命令窗口。
- "Short History"包含命令窗口和缩小的命令历史窗口,上面是在一个表单框中的当前的目录浏览器和工作区浏览器。
- ➤ "Tall History"在左边包含了放大的命令历史窗口,命令窗口和在一个表单框中的当前目录浏览器和工作区浏览器,如图 2-24 所示。

AMATLAB	
<u>F</u> ile <u>E</u> dit <u>V</u> iew We <u>b</u> <u>W</u> indow <u>H</u> elp	
🗅 😅 🕺 🏦 🛍 🗠 🗠 🎁 🤶	Current Directory: D:\matlabR12\work
Command History	Current Directory
% 10:04 PH 6/30/01%	D:\matlabR12\work 💌 🔜 🛍 🚧
	All files File Type Last Modified Description
	Current Directory Workspace
	Command Window
	To get started, select "MATLAB Help" from the Help menu.
	۲ ۲
Ready	

图 2-24 "Tall History"视图

- ▶ "Five Panel"五个桌面工具将会同时出现,如图 2-25 所示。
- 用户可以使用鼠标拖动任何一个桌面工具的标题栏,自由改变桌面工具的位置和大小,把桌面更 改为自己喜欢的视图方式,
- Sear "Undock Workspace"选项将"Workspace"窗口提出 MATLAB 桌面,成为独立的一个 Windows 窗口。
- 用户还可以单击桌面工具相应窗口标题栏右边的箭头按钮,或者拖动标题栏移出 MATLAB 桌面窗口,把任何一个桌面工具提出 MATLAB 桌面系统,成为一个独立的 Windows 窗口。

A MATLAB		
<u>F</u> ile <u>E</u> dit <u>V</u> iew We <u>b</u> <u>W</u> indow <u>H</u> elp	p	
🗋 😂 🕺 🖻 🛍 ဟာ က 🎁	Current Directory: D:\matlabR12\work	
Launch Pad 7	Workspace Command Window	7 ×
MATLAB     ACommunications Toolbox     AControl System Toolbox     Anta Acquisition Toolbox     Antabase Toolbox     Antabase Toolbox	To get started, select "MATLAB Help" from the	a He
Command History 🔀 🗙	Currant. Directory       D: \matlably       D: \matlably       The ryr	
Ready		

图 2-25 "Five Panel" 视图

(4) 单击"Web"项,将打开 MATLAB 的网络下拉菜单,如图 2-26 所示。单击选项将会打开网络浏览器, 连接 MathWorks 的相应网络站点。

(5)单击菜单栏的"Window(窗口)"选项,可以查看目前 MATLAB 打开的所有窗口,并可选中某个窗口来实现在不同窗口之间的转换。"Close All"选项将关闭所有打开的 MATLAB 工作窗口。

(6) 单击菜单栏的"Help"选项,可以打开 MATLAB 的帮助下拉菜单,如图 2-27 所示。

	<u>F</u> ull Product Family Help <u>M</u> ATLAB Help
The <u>M</u> athWorks Web Site	Using the Desktop Using the Command Window
<u>T</u> echnical Support Knowledge Base <u>P</u> roducts Member <u>s</u> hip	Dem <u>o</u> s
	<u>A</u> bout MATLAB

图 2-26 "Web"的下拉菜单

图 2-27 "Help"的下拉菜单

- "Full Product Family Help"、"MATLAB Help"、"Using the Desktop"、"Using the Command Window"项,都将打开 MATLAB 的帮助窗口,并首先显示 MATLAB 的帮助系统的分类。
- ▶ "Demos"项可以通过演示 MATLAB 提供的例子来熟悉相关部分的用法。
- ▶ "About MATLAB"项将打开关于 MATLAB 的版本和版权等信息。

#### 2. 桌面的工具栏

命令窗口中菜单栏下面是工具栏,工具栏提供了一些方便的按钮,如图 2-28 所示。下面将分别介绍各个按 钮的功能。



图 2-28 桌面的工具栏

□ 按钮:相当于菜单栏中"File"选项中的"New"命令;
 ☞ 按钮:相当于菜单栏中"File"选项中的"Open"命令;
 ◎ 按钮:相当于菜单栏中"Edit"选项中的"Cut"命令;
 ◎ 按钮:相当于菜单栏中"Edit"选项中的"Copy"命令;
 ◎ 按钮:相当于菜单栏中"Edit"选项中的"Paste"命令;
 ◎ 按钮:相当于菜单栏中"Edit"选项中的"Undo"命令;
 ◎ 按钮:相当于菜单栏中"Edit"选项中的"Redo"命令;
 ◎ 按钮:相当于菜单栏中"Edit"选项中的"Redo"命令;

? 按钮:帮助按钮。

MATLAB 的桌面系统是由命令窗口(Command Window),命令历史窗口(Command History),发射台(Launch Pad),工作区(Workspace),和数组编辑器(Array Editor),路径浏览器,帮助,调试与编译器等桌面工具组成的,这个环境方便了用户管理变量、输入输出数据以及生成和管理 M 文件,下面将分别介绍 MATLAB 的这些桌面工具。

2.5.2 MATLAB 的命令窗口

命令窗口是用户与 MATLAB 主要的交互工具,同时也是 MATLAB 实现各种功能的窗口,MATLAB 的各种 操作都是从命令窗口开始的,因此有必要对命令窗口有深入的了解。启动 MATLAB,在 MATLAB 桌面中将出 现命令窗口。用户可以根据前面所述更改桌 MATLAB 面视图方式,为方便陈述起见,在讲述各桌面工具时,更 改桌面视图为此工具的单一桌面,如这里的命令窗口,已使用"View"菜单中"Desktop Layout"下的"Command Window Only"。

在图 2-29 所示的命令窗口中,除标题栏、菜单栏、工具栏以外的窗口,用于输入和显示计算结果,称为命 令编辑区。如前所述,选择菜单栏中"File"的"Preference"选项,可以设置命令编辑区的显示风格。



图 2-29 "Command Windows"命令窗口

每次启动 MATLAB,命令编辑区都会显示一行帮助信息 To get strarted, select "MATLAB Help" from the Help menu,提示用户可以使用"Help"菜单下"MATLAB Help"获得帮助。

1. 基本输入与函数运行

帮助信息后将显示提示符号 ">>"。符号 ">>"表示 MATLAB 已准备好,正等待用户输入命令。这时,用 户就可以在 ">>"提示符后面键入命令,按下回车键后,MATLAB 就会解释执行所输入的命令,并在命令后面 给出计算结果。

例如,当输入生成3×3矩阵的命令

A=[1 2 3;4 5 6;7 8 10]

回车,将在命令行的下面显示:

A =

1	2	3
4	5	6
7	8	10

要执行函数运算,在提示符后面输入包括所有参数的函数,然后回车

例如,

magic(2)

MATLAB 将返回

ans =

1 3

4 2

➢ 用户可以在任何时候使用 Ctrl+C 或 Ctrl+Break 终止正在运行的 MATLAB 程序。

如果用户想要在运行输入多行命令,可以在每行的末尾使用 Shift+Enter 或 Shift+Return 进行断行,直到输入结束,然后按回车键运行跑所有命令。在使用编辑区时,用户还可以象使用 Word 一样,把选中的内容拖动到需要的位置。

对于命令窗口的函数,用户还可以使用鼠标的右键快捷菜单对被选中的函数进行操作,如图 2-30 所示。

A MATLAB	- 🗆 ×
<u>File Edit View Web</u> Mindow Help	
🗋 🖙 🐰 🖻 🛍 🕫 🕬 🥵 🎁 ? Current Directory: D:VnatlabR12twork 🔄 🛄	
To get started select "NATLAD Mels" from the Mels serve	<b></b>
To get started, serect mariano nerp irom the nerp ment.	- 1
Evaluate Selection	- 1
A = Open Selection	- 1
1 2 Help on Selection	- 1
4 5 7 8 1 Cut	- 1
Copy	- 1
Paste	- 1
	- 11
	- 1
	- 1
	- 1
	- 11
Rd	<b>▼</b>
Ready	

图 2-30 命令窗口的鼠标右键菜单

(1)" Evaluate Selection"对被选择项求值,在命令窗户中运用鼠标选择函数,然后单击鼠标右键并从上下 文菜单选择" Evaluate Selection"。当 MATLAB 非常繁忙时,用户不能对被选择项求值。

(2) "Open Selection"打开被选择函数的 M 文件,在命令窗户中选择函数,然后单击鼠标右键并且从上下 文菜单选择开选择"Open Selection", MATLAB 将在编辑/调试窗口打开此函数的 M 文件。

➢ MATLAB 的运算是单进程的,当 MATLAB 繁忙的时候,用户的任何命令都被放在堆栈中,由 MATLAB 依次进行,先前的一个命令完成时,下一个命令才会被运行。

2. 输入输出控制

在命令窗口中用户可以使用多种方法控制和解释 MATLAB 的输入和输出:

(1) 大小写敏感

MATLAB 是区分大小写的。例如,用户在运用绘图函数是,就不能把"plot"写成"Plot"。同样,变量"a"和变辆"A"是完全不同的两个变量。需要注意的是,如果用户使用函数的帮助时,在命令窗口中枢如"help+函数",MATLAB 返回的函数名字的所有字母都将被大写显示;而在用户运用函数时,所用的函数名都需要小写,不能够大写,这是需要区分的。

(2) 在一行上输入多个函数

在 MATLAB 的命令窗口的编辑区,一个命令行上可以输入多个命令函数,韩属于函数之间用分号隔开。例 如,在一个命令行上输入三个函数

format short; x = (1:10)'; logs = [x log10(x)]

回车,MATLAB将按照从左到右的顺序依次执行三个函数。

(3) 长函数的输入

如果在使用 MATLAB 中出现一个命令语句不适合在一行上输入,可以在该命令行的末尾使用一个省略号(3 个点...),表示命令语句在下一命令行上继续,按回车键,那么就可以继续进入语句。例如,

 $s = 1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + 1/7 \dots$ 

 $- \frac{1}{8} + \frac{1}{9} - \frac{1}{10} + \frac{1}{11} - \frac{1}{12};$ 

在 MATLAB 中, 单独一个命令行上所允许的最多的字符数是 4096。

(4) 语句结构的色彩设置

使用不同的颜色表示不同的语句结构,可以帮助用户更好发现个语句单元,例如对 else/if 陈述的匹配。以下是编辑区不同色彩类别,如图 2-31 所示。

- ▶ 正在输入的字符串是紫色的。
- ▶ 已输入的字符串是粟色的。
- ▶ 命令关键词,例如流控制函数"for"等,是蓝色的。
- 双击开放或关闭标志对 "()"、"[]"、"{}",标志对间的字符被高亮选中
- ▶ 键入标志对的关闭标志时如")",其匹配的打开标志如"("将被简短的加亮显示。
- 注释标志"%"后的注释语句是绿色的。
- ▶ 输入的系统命令,例如"!",是金色的。
- ▶ 错误信息以红色出现。



图 2-31 命令编辑区的色彩显示

用户可以使用前面所述的 "File "、 "Perefence "、 "Command Window "选项卡,更改命令窗口的字体和色彩 设置。

(5)命令行的编辑

使用以下这些命令编辑方式,将使用户在 MATLAB 命令窗口的使用上节约大量的时间。

▶ 使用剪贴板

使用 MATLAB 的时候,可以经常使用工具栏上 "Cut"、"Copy"、"Paste"、"Undo"和 "Redo"编辑命令, 这些命令也可以从菜单栏里 "Edit"的下拉菜单中获得。

▶ 命令行回想

在使用命令窗口进行编辑时,用户可以使用方向键、"Tab"键和组合的快捷键显示、编辑和再次使用以前 使用的命令行。例如,如果用户错误的输入了

rho = (1 + sqt(5))/2

MATLAB 将返回:

Undefined function or variable 'sqt'

由于用户错误的将" sqrt " 写作了" sqt ", MATLAB 认为这是一个变量或者没有定义的函数。在这种情况下, 用户没有必要重新输入全部命令行,按一下方向键"",在 MATLAB 提示符后将出现上一次输入的命令行, 这时只需将光标移动到" sqt "并增加遗漏的" r"进行修改。重复使用向上方向箭"", 可以在 MATLAB 提示 符后出现更早以前输入的命令行。

在用户使用 MATLAB 命令窗口的过程中,用户所输入的每一个命令行都被存储在一个缓冲区内。对于已经 使用过的函数,用户可以先输入该函数的少数前几个字母,使用"smart recall"(快捷回想)自动补齐函数命令。 例如,用户可以输入"plo"然后按下向上方向键,MATLAB 将记起最近使用过的以"plo"开始的函数,最为 贴近的函数是绘图命令"plot"。 表 2-6 列出了 MATLAB 命令编辑区中可以使用的方向键和快捷键,经常使用 Emacs 编辑器的用户会比较熟悉这些功能键。

表 2-6

#### MATLAB 命令编辑区的编辑快捷键

快捷键	组合键	功能
	Ctrl+p	显示上一条命令行
	Ctrl+n	显示下一条命令行
	Ctrl+b	后退一个字符
	Ctrl+f	前进一个字符
Ctrl+	Ctrl+r	右移一个词
Ctrl+	Ctrl+l	左移一个词
Home	Ctrl+a	光标移到行首
End	Ctrl+e	光标移到行尾
Esc	Ctrl+u	清除命令行
Delete	Ctrl+d	删除光标后的字符
Backspace	Ctrl+h	删除光标签的字符
	Ctrl+k	删除光标到命令行末尾的内容
Shift+home		高亮选中光标到命令行开始的内容
Shift+end		高亮选中光标到命令行末尾的内容

#### ▶ 使用命令补齐

用户在使用 MATLAB 的过程中,可以只输入函数名、变量、文件名、句柄图形的前几个字符,然后按下" Tab"键, MATLAB 将自动为用户正确的完成剩余部分。例如,如果用户已经建立了变量" costs_march",在使用中,用户只用输入

costs

然后按下"Tab"键,MATLAB会自动完成变量名,显示

costs_march

按下回车键, MATLAB 开始运行命令。在这个例子里, MATLAB 将显示 costs_march 的内容。

如果 MATLAB 环境中有超过一个的函数、变量名都以用户输入的字符开始,用户可以按两次"Tab"键, MATLAB 将为用户提供一张表,列出所有可能的命令。例如用户输入

cos

并按"Tab"键, MATLAB不会显示任何内容, 说明有多个命名以"cos"开始。 再次按下"Tab"键, MATLAB将显示

cos cos_tr cosh cosint costfun

MATLAB显示的可能命令列表中包括以"cos"开始的函数,也包括了用户创建的变量。

(6)清除命令窗口

用户可以菜单栏中"Edit"选择"Clear Command Window"项清除命令窗口的内容。正如前面讲过的,清 除命令窗口的内容,只是为了视觉上美观,并不会清除工作区,用户仍然可以使用定义的所有变量,也可以使 用命令行回想重复使用先前的函数命令。

∞ 用户可以在 MATLAB 的命令窗口输入 " clc ", 同样可以清除命令窗口。

(7)禁止显示结果

在使用 MATLAB 的过程中,用户可以在命令行结尾输入分号";"结束,按回车键后,MATLAB 运行语句 时不会显示返回内容。在产生大矩阵时,这一点特别有用。例如,用户要产生一个 100 维的矩阵,输入

A=magic(100);

然后按回车键,MATLAB将创造一个大矩阵但是并不在命令窗口中显示结果。

(8)分页显示输出结果

如果在命令窗口中的输出如果过长,它可能不适合在屏幕以内显示,或者显示过快,使用户无法看到它, 这时,用户可以使用" more "函数控制命令窗口的输出结果以分页显示。默认情况下," more "函数是关闭的" more off "。如果用户输入 MATLAB 将每次一页显示命令行的输出结果。在第一页显示之后,用户可以使用表 2-7 所示的功能键控制 输出结果的显示。

表 2-7

```
分页显示控制
```

功能键	功能
Enter 或 Return	显示下一行
Space	显示下一页
q	终止显示

(9) 数值显示样式的控制

在 MATLAB 的命令窗口中,默认情况下,命令窗口中都是以 5 位数显示包括了小数点在内的数字。用户可 以通过修改"文本显示属性"更改 MATLAB 命令窗口中数字的显示格式和间距。需要注意的时,改变"文本显 示属性产量"仅仅是影响了数字的显示格式,并不影响 MATLAB 对它们的计算和保存。

> "format"函数

用户可以使用"format"函数控制数字的值在 MATLAB 命令窗口的输出格式。

🔉 用户指定的输出显示格式只对当前的任务适用。

以下是使用"format"的一个例子,使用不同的格式显示2个元素的矢量 x

format short

```
x = [4/3 \ 1.2345e-6]
x =
```

```
1.3333 0.0000
```

format short e

```
x = [4/3 \ 1.2345e-6]
```

```
x =
```

1.3333e+000 1.2345e-006

```
format +
```

```
x = [4/3 1.2345e-6]
x =
```

++

```
> 控制间距
```

在 MATLAB 的命令窗口中,用户可以使用

format compact

```
命令禁止显示中的空白行,以使用户可以看到更多的输出内容;使用
```

format loose

命令取消对空白行的显示禁止。

3.运行程序

```
(1)运行 M 文件
```

M 文件是包含 MATLAB 语言编码文件,运行 M 文件和运行其他 MATLAB 函数的方法是一样的。在命令窗口的编辑区中输入 M 文件名,然后按回车键,将开始运行 M 文件。

在执行文件过程中,如果用户想显示文件中的每一个函数的执行情况,可以在显示属性中选中"Echo on" 选项,或者使用"echo"函数,在命令窗口中输入

echo on

用户可以使用"Ctrl+c"或者"Ctrl+Break"终止正在运行的程序。

(2)运行外部程序

惊叹号"!"是一个推出"shell"的标志命令,它表示后面输入命令行是一个对操作系统操作的命令行。使用它用户可以在不停止 MATLAB 运行的情况下调用其他应用程序。例如,在 UNIX 系统中,输入命令行

将调用 vi 编辑器编写一个名为 darwin.m 的文件。当用户退出编辑器后,操作系统将把控制全交还给 MATLAB。

4. 命令窗口的属性设置

用户可以使用属性设置"Command Window"选项卡定义命令编辑区的数值输出结果的显示格式,选择程序 编辑器,指定帮助目录,命令执行时是否回显,命令与命令之间以及变量名和变量的值之间是否留空行以及设 置文字的类型、风格、大小和色彩等。用户可以从菜单栏中"File"下拉菜单种选择"Peferences"启动属性对 话框。选择展开"Command Window",如图 2-32 所示,下面依次阐述个选项的作用。

📣 Preferences	
🖅 - General	Command Window Preferences
Command Window     Font & Colors     Editor/Debugger     Workspace     Array Editor     GUIDE     Figure Copy Template     Copy Options     Simulink	Text display         Numeric format:         Spaces per tab:         Spaces per tab:         Display         Echo on         Limit matrix display width to eighty columns         Image: Enable up to 100         tab completions         Command session scroll buffer size:         Min         Image: Heat table
	OK Cancel Help

图 2-32 "Command Window"选项卡

(1) 文本显示属性

指定产量在命令窗户中显得怎么样:

- Numeric format 数值的输出在命令窗口中显示的格式。这仅仅影响数字怎么被显示,并不影响 MATLAB 对它们的计算与存储。
- ➢ Numeric display 数值的输出命令窗户中显示时的间距,是否显示空白行。" compact " 禁止显示空白行,紧缩显示;" loose " 准许显示空白的行。
- ➢ Spaces per tab Tab 键所占的空格数,默认值为 4。

(2)显示属性

- ▶ Echo on 选中该选项, MATLAB 在执行 M 文件期间, 命令窗口将回显每一个命令。
- Limit matrix display width to eighty columns 选中该选项,即便命令窗口的宽度足够大, MATLAB 命令窗口也只能显示 80 列宽度的矩阵。不选该选项,表示命令窗口可以输出显示比 80 列宽的矩阵,直到充满命令窗口的宽度
- ▶ Enable up to n tab completions 允许命令补齐。
- Command session scroll buffer size 设置任务缓存区大小。缓存区的对 MATLAB 执行命令时, 所支持的命令行回想功能影响很大。
- (3) 命令窗口的字体和色彩属性

Font

为命令窗户指定使用的字体。用户可以使用桌面字体,选择"Use desktop font"。用户也可以自己设置自己 喜欢的字体,选择"Use custom font",并指定字体的"Type"、"Style"、"Size",当完成选择后,"Sample area" 将显示样例。

如果你在命令窗户中想要字体与那一样为一般的字体与颜色偏爱指定了,选择使用。 如果你想要命令窗户字体不同,选择使用各种字体并且为命令窗户指定字体特征:

> Color

用户还指定在命令窗户中使用的颜色:

"Text color"设置一般文本的颜色; "Syntax highlighting"设置特殊文本的颜色; "Backgroung color"设置 背景颜色。选中后,单击"Set Colors" 指定颜色。

## 2.5.3 命令历史窗口

启动 MATLAB 后,在 MATLAB 环境中将出现命令历史窗口,如图 2-33 所示。命令历史窗口的功能的是记录最近在命令窗口中运行的命令历史,命令历史窗口中列出了最近在命令窗口中运行的命令列表。

用户可以通过选择 View 菜单中 Command History 项来显示出或隐蔽命令历史窗口。如图 2-34 所示。

在命令历史窗户中的任何函数、命令上双击或者单击鼠标右键并且选择上下文菜单中选择 Evaluate Selection,如图 2-35 所示,都将在命令窗口中重新执行该命令。



图 2-33 MATLAB 环境中的命令历史窗口

图 2-34 View 下拉菜单

♥#ATLAB Rila Rdit View Wab Window Halm	_ 🗆 ×
□     ⇒     S     B     B     ∞     ○     T     ?     Current Directory:     D:\matlabR12\work	
Launch Pad	Command Window 🛛 🛪 🗙
🕀 📣 MATLAB	
🕀 📣 Communications Toolbox	lo get started, sel
🕀 📣 Control System Toolbox	>> elseif A <b< td=""></b<>
🗄 📣 Data Acquisition Toolbox 📃 💌	??? elseif A <b< td=""></b<>
Launch Pad Workspace	Error: "End of Input"
Command History	≫ elseif A⊲B
′great′	??? elseif A <b< td=""></b<>
elseif A <b< td=""><td>Error: "End of Input"</td></b<>	Error: "End of Input"
less'	1
elseit A=B	>>> A=[1 2 3;4 5 6;7 8
A==B	A =
if A>B	
'great'	1 2 3
elseif A <b< td=""><td>4 5 6</td></b<>	4 5 6
alroif A==B	
A=[1 2 3;4 5 6:7 8 10]	>>
A=[1 2 3;4 5 Copy	
A=[1 2 3:4 5 Evaluate Selection	
A=L1 2 3:4 5 A=F1 2 3:4 5 Create M-File	
A=[1 2 3:4 5	-11
% 11:35 PM Delete Selection	
elseif A/B Delete to Selection	
Delete Entire History	≝ ∠
Command History Conence/rectory	

使用复制、粘贴命令将命令历史窗口的函数、命令复制到命令窗口,或者将选中的函数、命令直接又命令 历史窗口拖动到命令窗口,都将在命令窗口重新执行该命令。

图 2-35 命令历史窗口中命令的调用

在命令历史窗口中,对于选中的命令,用户可以进行如下各种操作:

- > 复制命令到命令窗口
- ▶ 重新执行命令
- ▶ 拖动选择的命令到命令窗口
- ▶ 删除命令
- ▶ 创建包含命令的 M 文件

### 2.5.4 工作空间

启动 MATLAB 后,在 MATLAB 环境中将出现工作空间窗口,如图 2-36 所示。用户可以通过选择 View 菜 单中 Command History 项来显示出或隐藏工作空间。

4	MATLAB					<u>_ 0 ×</u>
Fi	le <u>E</u> dit <u>V</u> iew We <u>b</u>	<u>M</u> indow <u>H</u> elp				
C	) 🛩   X 🖻 🛍 ·	ට ලං 🛛 🞁	? Curr	ent Directory: D:\matlabR12\work	<b>_</b>	
W	orkspace				× 5	Command Window 🛛 🔊 🗙
C	🍰 🔚 🛛 🗊 🖓 Sta	<b>ck:</b> Base	<b>-</b>			started, select "MATL
F	Name	Size 3x3	Bytes 72	Class double array		A≺B
						t ASB
						nd of Input" expected
						A <b £ A <b< td=""></b<></b 
						nd of Input" expected
						3;4 5 6;7 8 10]
						2 3
						8 10
						3;4 5 6;7 8 10]
ŀ	Launch Pad	Workoposo				
H		workspace	<u> </u>	· · · · · · · · · · · · · · · · · · ·		
Co	ommand History				× s	
e.] (1	.seit A\B .ess'					
R	Command Hist	ory Curre	nt Directory			
Ľ-		<u> </u>		2		

图 2-36 MATLAB 环境中的工作空间窗口

在 MATLAB 中,工作空间(Workplace)是一个重要的概念。工作空间指运行 MATLAB 的程序或命令所生 成的所有变量和 MATLAB 提供的常量构成的空间。工作空间是一个比较抽象的概念每次启动 MATLAB, MATLAB 会自动建立一个工作空间,工作空间在 MATLAB 运行期间一直存在,关闭 MATLAB 后工作空间自动 消失。刚打开的 MATLAB 的工作空间中只有 MATLAB 提供的几个常量,如 pi(3.1415926...)、虚数单位 i 等。 运行 MATLAB 的程序时,程序中的变量被加入到工作空间。除非用特殊的命令删除某变量,否则该变量在关闭 MATLAB 之前一直存在。由此可见,在一个程序中的运算结果以变量的形式保存在工作空间中,又可被别的程 序继续利用。可以随时查看工作空间中的变量名及变量的值。某个时刻的工作空间中的所有变量可以保存到一 个文件中,这样,当关闭 MATLAB 后,所有的变量的值仍然存在,当下次启动 MATLAB 后,又可用相关的命 令把保存的工作空间的所有变量调入到当前工作空间。

在命令窗口键入 who 和 whos 命令可以看到目前工作空间里的所有变量。who 命令所给的信息比较简短,只 给出变量名; whos 还给出变量的大小、数据类型等具体内容。以下为一个包含有八个变量的工作空间用 whos 命令查询的显示结果。命令的 clear 可删除工作空间里的变量。

例如,对于执行过以下命令的工作空间里,使用 who 和 whos 查询变量的结果为

C=[ 9 4; 2 8; 6 7]; X=pinv(C), Q=X*C, P=C*X ; who Your variables are:

A C P Q X

32

Name	Size	Bytes	Class
А	3x3	72	double array
С	3x2	48	double array
Р	3x3	72	double array
Q	2x2	32	double array
Х	2x3	48	double array

Grand total is 34 elements using 272 bytes

工作空间窗口用于显示工作空间里变量的图形方式。其功能大致相当于 whos 命令加上 clear 命令,但更为 直观、方便。如图 2-37 所示。另外,在工作空间窗口里可以更改变量名。

📣 Workspace			
<u>F</u> ile <u>E</u> dit <u>V</u> iew We <u>b</u> <u>W</u>	indow <u>H</u> elp		
🖆 🔚 🔯 箭 Stack	Base	<u>-</u>	
Name	Size	Bytes	Class
A A	3x3	72	double array
<b>I</b> II C	3x2	48	double array
P P	3x3	72	double array
<b>⊞</b> Q	2x2	32	double array
X	2x3	48	double array
Ready			

图 2-37 工作空间窗口中显示的变量

对于工作空间中被选中的变量,用户可以单击鼠标唤出其上下文菜单,如图 2-38 所示。

选中上下文菜单中的 Open Selection 项,将激活数组编辑窗口,可以对选中的变量进行修改,如图 2-39 所示。

📣 Workspace						
<u>F</u> ile <u>E</u> dit <u>V</u> iew We <u>b</u> <u>W</u> i	indo	w <u>H</u> elp				
🖙 🔚 🔯 🛱 Stack	Ва	se 🔳	3			
Name	• • •	Size	Bytes	• • • • • •	Class	
A E	4x4		128	double	array	
<b>⊞</b> c	3x2		48	double	array	
Open Selection			72	double	array	
Graph Selection	Þ	plot		•	array	
Select All		surf 2-D Graphics				
Import Data		3-D Graphics 🔹 🕨 mesh				
Save Selection As		Special 2-D Graphics 🕨 surf				
Save Workspace As		Special 2.5-D Graphics				
Delete Selection Clear Workspace						
Ready						

图 2-38 工作空间中的上下文菜单

Numeric Ionn	all follorio		(izo: 1/	by M	
			126.  4	Dy H	
	1	2	3		
1	1	2	3		
2	4	5	6		
3	7	8	10		
			10		
4					

图 2-39 数组编辑器

选中上下文菜单中的 Graph Selection 项,将在图形窗口中,以用户选择的方式对变量作图。例如,对于上例中

C=[

- 9 4;
- 2 8;
- 6 7];
- X=pinv(C), Q=X*C, P=C*X ;

产生的变量 C, 以选项 Plot 和选项 Surf 分别作图,结果如图 2-40、2-41 所示。



图 2-40 Plot 选项的作图结果



图 2-41 Surf 选项的作图结果

选中上下文菜单中的 Import Data 项,将出现如图 2-42 所示的打开文件窗口,允许用户选择数据文件为所选 变量赋值。

Open						?×
搜寻(I):	🔄 work		-	<b>E</b>	2 🖻	
计研究的计	<b>I</b>				+7.77	~
	J			_	3121 (	U I
文件类型 (I):	MAT-files (*	(mat)		-	取消	i
a sa						

图 2-42 打开文件窗口

## 2.5.5 发射台

每次启动 MATLAB,在 MATLAB环境中将显示 MATLAB发射台。如图 2-43 所示。

A MATLAB		
<u>F</u> ile <u>E</u> dit <u>V</u> iew We <u>b</u> <u>W</u> indow <u>H</u> elp		
🗋 🚔 🖟 🛅 💼 🖘 🖘 🎁 🥐 Current Directory: D:\matlabR12\work	<b>_</b>	
Launch Pad 7 🗙	Command Window	×۲
E MATLAB		<u> </u>
	Q =	
Help	1.0000 0.0000	
- 😿 Demos	0.0000 1.0000	
Product Page (Web)		
🗄 🛋 Control System Toolbox	P =	
🕀 🐗 Data Acquisition Toolbox	0.0000 0.1050	0.0010
🛨 🐗 Database Toolbox	-0.1958 0.7754	0.3685
🗄 🔿 📣 Datafeed Toolbox	0.3213 0.3685	0.3952
🗄 🔿 👫 Filter Design Toolbox	N	
🕀 📣 Financial Derivatives Toolbox	// wito	
🕀 📣 Financial Time Series Toolbox	Your variables are:	1.1
A Financial Toolbox		
	ACIQA	
	>> whos	
+- 1 Image Processing Toolbox	Name Size	Bytes Class
+	A 3x3	72 double ar
	C 3x2	48 double ar
Workspace Launch Pad	P 3x3	72 double ar
Command History	X 2x3	48 double ar
elseif A==B	Grand total is 34 elem	ents using 272 byte
	>>_	
Command History Current Directory		
Ready		

图 2-43 MATLAB 环境中的发射台

MATLAB 的发射台为用户迅速访问 MATLAB 提供工具、演示以及所有 MathWorks 产品文档提供了方便。 在发射台中列出了所有系统上被安装了的所有的产品,单击选定产品左边的"+"号,将显示该产品的目录;对 于已显示目录的产品,单击该产品单击选定产品左边的"-"号,将折叠该目录。

打开一个产品目录后,双击其中的选定项目,或单击鼠标右键并且从上下文菜单选择 Open,将运行该项目。 例如,双击发射台窗口中 MATLAB 选项下的 Demos,将运行 MATLAB 的演示程序,如图 2-44 所示。



图 2-44 从发射台启动演示程序

用户可以通过创建 info.xml 文件在发射台窗口中加入自己的接口程序。在发射台窗口中选择任意一个项目, 单击鼠标右键,并且选择从上下文菜单选择 Edit Source,这是将在编辑/调试窗口中显示该产品的 info.xml。如 图 2-45 所示。

D:	\MATLA	BR12\t	oolbox	\matlab	\general\inf	o.xml							- 🗆 🗵
File	<u>E</u> dit	<u>V</u> iew	<u>T</u> ext	Debug	Brea <u>k</u> point:	s We <u>b</u>	<u>W</u> indow	Help					
Ľ	🖻	1 🖨	*	<b>b</b> 🛍	က က <b>ဖြ</b>	<b>i</b>		<b>e</b>   43	) tê B	1	Stack: B	ase 🔻	] 🗵
23	labe	1×urr	ent Di	rectory	<pre></pre>								
24	<call< td=""><td>back&gt;f</td><th>ilebro</th><th>wser√c</th><th>allback&gt;</th><th></th><td></td><td></td><td></td><td></td><td></td><td></td><td>1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -</td></call<>	back>f	ilebro	wser√c	allback>								1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -
25	<100D	.≫\$too⊥	.box/ma	tlab/ge:	neral/figure	1 con. g	11/1001	~					
20	1112	titem/											
28	list	item>											
29	labe	1>Work	space<	/label>									
30	<cal1< td=""><td>back&gt;</td><th>orkspa</th><th>ce√cal</th><th>lback&gt;</th><th></th><td></td><td></td><td></td><td></td><td></td><td></td><td></td></cal1<>	back>	orkspa	ce√cal	lback>								
31	<i con<="" td=""><td>.≫tool</td><th>.box/ma</th><th>tlab/ge</th><th>neral/figure</th><th>icon.g</th><td>if∜icor</td><td>2</td><td></td><td></td><td></td><td></td><td></td></i>	.≫tool	.box/ma	tlab/ge	neral/figure	icon.g	if∜icor	2					
32	√lis	titem)	·										1 A A A A A A A A A A A A A A A A A A A
33													
34	dist	item≻	112.2	• •									
00 00	Labe	1./Fath	<pre>\Labe</pre>	1//11	11								
30	Call	Dack∕p \\$t+ool	athtoo how/mo	t∨call +lob/ao	Dack∕ novol∕fizevo			5					
38	1is</td <td>titem)</td> <th>, DOX/IIIa &gt;</th> <th>crab/ge.</th> <th>nerar, rigure</th> <th>r con. g</th> <td>11 \ 1001</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	titem)	, DOX/IIIa >	crab/ge.	nerar, rigure	r con. g	11 \ 1001						
39	1 1 1 1												
40	⊲list	item>											
41	labe	1×30II	E (GUI	Builde	r)								
42	<call< td=""><td>back&gt;g</td><th>;ui de√/</th><th>callbac</th><th>k&gt;</th><th></th><td></td><td></td><td></td><td></td><td></td><td></td><td></td></call<>	back>g	;ui de√/	callbac	k>								
43	<i con<="" td=""><td>¦≫\$tool</td><th>.box/ma</th><th>tlab/ge</th><th>neral/figure</th><th>icon.g</th><td>if∜icor</td><td>2</td><td></td><td></td><td></td><td></td><td></td></i>	¦≫\$tool	.box/ma	tlab/ge	neral/figure	icon.g	if∜icor	2					
44	≪lis	titem>	•										
45	a · .												
40	(11st	.1tem∕ 1∖p	P.	(W.L	1/1-1-1								
47	 	1		ge (пер //	) ( raner -	/	3	.1.17 -	<b>.</b>	//	1-\		
													►
Read	зy												

图 2-45 info.xml 文件示例

用户可以为自己的应用程序创造一个类似的 info.xml 文件,并且把它放在在搜索路径上的一个文件夹中。 然后在发射台窗口单击鼠标右键,并且从上下文菜单选择 Refresh,刷新发射台窗口,就可以发射台窗口显示自 己的产品接口。

2.5.6 图形窗口

在 MATLAB 中,图形窗口和命令窗口是相互独立的窗口,用户可以通过选择 File 菜单下 New 中的 Figure, 或者在命令窗口中输入

figure

启动如图 2-46 所示的图形窗口。

🚽 Figure No. 1
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>I</u> nsert <u>T</u> ools <u>W</u> indow <u>H</u> elp
D 😂 🖬 🎒 🖎 A フノ 🔎 🔎 つ

图 2-46 图形窗口

MATLAB 的图形命令自动将图形画在图形窗口上,当 MATLAB 上没有图形窗口时,图形命令将新建一个 图形窗口;当 MATLAB 已经存在一个或多个图形窗口时,MATLAB 一般指定最后一个图形窗口作为当前图形 命令的输出窗口。

图形窗口不仅是 MATLAB 图形的显示窗口,也是图形的编辑窗口。这将在后面的章节中详细阐述。 在图形窗口的工具栏如图 2-47 所示。



图 2-47 图形窗口的工具栏

其中

▶图标	图形编辑图标,	选择该图标后,	可以对图形进行修改

 A 图标
 插入文本

- ▶ 图标 绘制带箭头的线段
- 图标 绘制线段
   会制线段
   会制线
   会制
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
   会
- ▶ 图标 放大图形
- ₽图标 缩小图形
- ◎ 图形 三位图形的旋转

选择 View 菜单下的 Camera Toolbar,将增加一个新的工具栏,如图 2-48 所示。



图 2-48 照相机工具栏

该工具栏的作用主要是对图形的视点、光源设置进行修改,感兴趣的读者可以参阅 MATLAB 帮助获得更多信息。

2.5.7 编辑/调试窗口

MATLAB 提供了一个内建的具有编辑和调试功能的程序编辑器。在 MATLAB 的命令窗口中输入

edit

或者选择 File 菜单下的 New 中的 M-file、单击工具栏中的新建图标,都将显示如图 2-49 所示的编辑/调试 窗口。



图 2-49 编辑/调试窗口

在程序编辑器里,不向的文本内容分别用不同颜色的字体显示,这样更加醒目,便于调试。图 2-50 为程序 编辑器界面上的工具按钮。

图 2-50 编辑窗口工具栏

更	P	
6.	( <b>万</b> 1+二	

- ♪ 图标 显示函数
- ●图标 设置断点
- ▲图标 取消断点
- 图标 执行当前行
- 🖻 图标 执行当前行,如果需要调用函数,运行调用的函数
- ◎图标 执行当前行到文件结束
- 1 图标 执行程序到下一个断点活文件结束
- 🙆 图标 退出调试模式

在后面的章节中将详细的介绍着协调时命令的用法。

2.5.8 路径浏览窗口

路径浏览器能够修改 MATLAB 的搜索路径并查看任一路径下的所有文件。选择 File 菜单条下的 Set Path 项,即可打开路径浏览器。图 2-51 为路径浏览器的界面。

📣 Set Path		
All changes take effect imr	nediately.	
	MATLAB search path:	
Add Folder	D:\MATLABR12\toolbox\matlab\general	
And a with Out Calabora	🛅 D:\MATLABR12\toolbox\matlab\ops	
Add with Subfolders	D:\MATLABR12\toolbox\matlab\lang	- 11
	🛅 D:\MATLABR12\toolbox\matlab\elmat	- 11
Move to Top	D:\MATLABR12\toolbox\matlab\elfun	
	💼 D:\MATLABR12\toolbox\matlab\specfun	
Move Up	🛅 D:\MATLABR12\toolbox\matlab\matfun	- 11
Remove	🛅 D:\MATLABR12\toolbox\matlab\datafun	- 11
	D:\MATLABR12\toolbox\matlab\audio	- 11
Move Down	🛅 D:\MATLABR12\toolbox\matlab\polyfun	- 11
Mous to Dottom	D:\MATLABR12\toolbox\matlab\funfun	- 11
	D:\MATLABR12\toolbox\matlab\sparfun	
All the second states	D'MATLABB12)toolhov(motlob)groph2d	
Save Close	Revert Default	Help

图 2-51 路径浏览器

用户可通过拖动右窗口中的目录到相应位置或者使用窗口左边的5个移动按钮来改变 MATLAB 的搜索先后顺序。

用户只有将自己的目录加入到搜索范围内,在 MATLAB 的命令窗口中键入文件名后才能执行。加目录到 搜索范围的方法为:

- ▶ 单击图 2-51 中的 Add to Path 按钮;
- 在出现的浏览窗口中选中相应目录;
- ▶ 单击 Refresh 按钮,修改后的搜索路径将起作用;
- ▶ 如果用户想永远保存当前搜索路径,单击 Save 按钮。

# 2.6 帮助系统

MATLAB 里有以下几种方法可获得帮助:帮助命令、帮助窗口、MATLAB 帮助台、在线帮助页或直接链接 到 MathWorks 公司。下面分别介绍如何用帮助命令、帮助窗口、MATLAB 帮助台、在线帮助页等方式获得帮助。

2.6.1 帮助命令

帮助命令是查询函数语法的最基本方法,查询信息直接显示在命令窗口。例如命令

help magic

将显示如下信息:

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.

🔉 MATLAB 命令窗口里显示的帮助信息用大写来突出函数名,但在使用函数时,应用小写

MATLAB 按照函数的不同用途分别存在不同的子目录下,用相应的帮助命令可显示某一类的所有函数。例 如,所有的线性代数函数均收在 matfun 子目录下,使用

Help matfun 回车将显示所有线性代数函数命令

help matfun

Matrix functions - numerical linear algebra.

Matrix analysis.

norm	- Matrix or vector norm.
normest	- Estimate the matrix 2-norm.
rank	- Matrix rank.
det	- Determinant.
trace	- Sum of diagonal elements.
null	- Null space.
orth	- Orthogonalization.
rref	- Reduced row echelon form.
subspace	- Angle between two subspaces.

## Linear equations.

$\setminus$ and /	- Linear equation solution; use "help slash".
inv	- Matrix inverse.
rcond	- LAPACK reciprocal condition estimator
cond	- Condition number with respect to inversion.
condest	- 1-norm condition number estimate.
normest1	- 1-norm estimate.
chol	- Cholesky factorization.
cholinc	- Incomplete Cholesky factorization.

# 命令 help 将显示所有的子目录,例如

help

...

. ...

TIDI D	
HELP	topics:

matlab\general	- General purpose commands.	
matlab\ops	- Operators and special characters.	
matlab\lang	- Programming language constructs.	
matlab\elmat	- Elementary matrices and matrix manipulation.	
matlab\elfun	- Elementary math functions.	
matlab\specfun	- Specialized math functions.	
matlab\matfun	- Matrix functions - numerical linear algebra.	
matlab\datafun	- Data analysis and Fourier transforms.	
matlab\audio	- Audio support.	
matlab\polyfun	- Interpolation and polynomials.	
matlab\funfun	- Function functions and ODE solvers.	
matlab\sparfun	- Sparse matrices.	
matlab\graph2d	- Two dimensional graphs.	
matlab\graph3d	- Three dimensional graphs.	
matlab\specgraph	- Specialized graphs.	

lookfor 命令和 help 命令相似,它们都只对 M 文件的第一行进行关键字搜索。help 只搜索出那些关键字完全 匹配的结果,lookfor 对搜索范围内的 M 文件进行关键字搜索,条件比较宽松。例如,因为不存在 inverse 函数,

使用 help inverse 命令的搜索结果为:

inverse.m not found.

而 lookforinverse 命令得到如下结果

lookfor inverse

INVHILB Inverse Hilbert matrix.

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.
- ASINH Inverse hyperbolic sine.
- ATAN Inverse tangent.
- ATAN2 Four quadrant inverse tangent.

对 lookfor 命令加上-all 选项,可对 M 文件进行全文搜索。

2.6.2 帮助窗口

帮助窗口给出的帮助信息和帮助命令给出的信息内容一样,但在帮助窗口更容易浏览与之相关的其他函数。 在 MATLAB 命令窗口中有三种方法进入帮助窗口:

- 单击菜单条上的问号按钮;
- ▶ 键入 helpwin 命令;
- ▶ 选取帮助菜单里的 "MATLAB Help" 项。

帮助窗口如图 2-52 所示。



图 2-52 帮助窗口

用户可以自由调整帮助窗口,关闭导航窗口的帮助窗口如图 2-53 所示,单击图 2-53 中左上角的"?"图标, 将恢复导航窗口。



图 2-53 关闭导航的帮助窗口

帮助窗口的帮助信息都是用超文本标记语言(HTML)写成,可以通过 Netscape 或 Microsoft 的 InternetExplorer 阅读。在 MATLAB 的帮助窗口中,只要在导航窗口中单击相关的内容逐级查找就可以找到相应 的帮助信息。帮助台上特别有用的是 Search 功能。用户在"Find in Page"按钮左边的输入框里键入关键字,即 可得到及时的帮助。

2.6.3 在线帮助

帮助窗口的所有文件均有相应的 PDF 格式文件,用 Adobe Acrobat Reader 可以阅读,称为在线帮助页。在 线帮助页翻页和查找都相当方便。用户选中帮助窗口上关于 PDF 格式文件的选项,或是在命令窗口中键入命令 doc,便会自动打开 Adobe Acrobat Reader。命令 doc 后可加上关键字,MATLAB 会自动定位到相关的页码。在 线帮助页包括所有的字体、图形和图像,打印在线帮助页可以得到精美的帮助硬拷贝。

对于联网用户,还可以通过帮助台很方便地访问 MathWorks 公司的主页,向 MathWorks 公司询问问题、建议及指出错误。

# 2.7 小 结

在本章中,介绍了 MATLAB 6 的安装、卸载过程以及 MATLAB 的各个组成部分,其中包括:命令窗口、 命令历史窗口、程序编辑/调试器、图形窗口、路径浏览器及联机帮助等。通过本章的学习,可以熟悉 MATLAB 的工作环境。

# 第三章 MATLAB 语法基础

MATLAB 不同于其他编程语言,是一种简洁直观的交互式开发语言,本章将介绍 MATLAB 语法的基础知识,包括以下内容:

- ▶ 矩阵
  - ▶ 基本运算符
  - ▶ 变量
  - ▶ 数值与数据类型
  - ▶ 字符串与数组
  - ➢ M 文件
  - ➢ 流程控制语句

# 3.1 矩阵

MATLAB 号称"矩阵实验室", MATLAB 中最重要也是最基本的功能就是矩阵的运算, 用户学习 MATLAB 也应该从矩阵开始。

3.1.1 MATLAB 中的矩阵

在 MATLAB 中,一个矩阵既可以是普通数学意义上的矩阵,也可以是标量或向量。对于标量(一个数)可以 将之作为 1x1 矩阵,而向量(一行或一列数)则可以认为是1×n或n×1矩阵。另外,一个0×0矩阵在 MATLAB 中被称为空矩阵。

使用 MATLAB,不仅能够进行实数矩阵的运算,而且能够进行复数矩阵的运算。用户 在对 MATLAB 的学习过程中,应特别留意对矩阵运算的学习。

3.1.2 矩阵的输入

在 MATLAB 中输入矩阵需要遵循以下基本常规:

- (1) 把矩阵元素列入方括号中;
- (2) 每行内的元素间用逗号或空格分开;
- (3) 行与行之间用分号隔开。

在 MATLAB 中, 可以用以下几种方式输入矩阵:

1. 通过显式元素列表输入

对一个简单矩阵来说,最好的输入方式是直接通过显式元素列表输入。例如直接输入:

x = [1 2 3; 4 5 6; 7 8 10]

或

x=[1,2,3;4,5,6;7,8,10]

或者混合输入

 $x=[1\ 2\ 3\ ;\ 4\ ,\ 5\ ,\ 6\ ;\ 7\ 8\ 10]$ 

会得到同样的输出结果:

 $\mathbf{x} =$ 

( 或者

	4	5	10						
	オモ较り		10 矩阵,为 ⁻	了使其输入方	式更符合用	1户习惯,	可以将矩阵元	素分行输并用回车键	
" S	hift+Enter	") 代替分	,,, }号。 例如	输入一个较力	、 矩阵 v:	5, 5, 5, 5, 7	5 7715721170		
	y=[1.0 1.	1 1.2 1.3	1.4 1.5 1.6	1.7 1.8 1.9					
	2.0 2.1 2.	.2 2.3 2.4	2.5 2.6 2.7	2.8 2.9					
	3.0 3.1 3.	.2 3.3 3.4	3.5 3.6 3.7	3.8 3.9					
	4.0 4.1 4	.2 4.3 4.4	4.5 4.6 4.7	4.8 4.9					
	5.0 5.1 5.	.2 5.3 5.4	5.5 5.6 5.7	5.8 5.9					
	6.0 6.1 6.	.2 6.3 6.4	6.5 6.6 6.7	6.8 6.9]					
	当用户键	建入回车后	后,显示如	下:					
	y =								
	Columns	1 through	7						
	1.0000	1.1000	1.200	0 1.3000	1.4000	1.5000	1.6000		
	2.0000	2.1000	2.200	0 2.3000	2.4000	2.5000	2.6000		
	3.0000	3.1000	3.200	0 3.3000	3.4000	3.5000	3.6000		
	4.0000	4.1000	4.200	0 4.3000	4.4000	4.5000	4.6000		
	5.0000	5.1000	5.200	0 5.3000	5.4000	5.5000	5.6000		
	6.0000	6.1000	6.200	0 6.3000	6.4000	6.5000	6.6000		
	Columns	8 through	10						
	1.7000	1.8000	1.900	0					
	2.7000	2.8000	2.900	0					
	3.7000	3.8000	3.900	0					
	4.7000	4.8000	4.900	0					
	5.7000	5.8000	5.900	0					
	6.7000	6.8000	6.900 ۲۰۰۰ ۲۰۰۰ ۲۰۰۰	0					
	2. 通过:	外部数据	又1十加報 ッ・ 今く		ᄝᅕᄲᄉᆘᅽᆂᄹ				
	用户可以	A使用 L	0ad 中气	了加载外部数划 1立步— 人名刘	古义什切建大	ᡛ᠙╪。 ᡷᡤᠫᡪ᠊ᠵᠯ᠊ᡵᠧᢧ᠘	4		
	例如,用户可以使用编辑器产生一个全部由数据组成的文本文件,								
	1.0 5.0	2.0	5.0	4.0 8.0					
	9.0	10 0	11 0	12 0					
	13 0	10.0	15 0	16 0					
	13.0 田戸司以	」 ↓将该文オ	12.0	10.0 5为数据文件 fi	emn dat _	后在 МАТІ	AB 的命今窗口	□由输入如下命今・	
	load temr	n dat	~~~~		inp.uu ₆ m				
		.AB 会自	动创建一个	个变量名为 ter	mp的矩阵,	输入 temp	并回车,则显着	큤	
	temp =				-p Hover (	int, comp			
	1	2	3 4	Ļ					
	5	6	7 8	3					
	9	10	11 12	)					
	13	14	15 16	5					
	3.使用	M 文件仓	J建矩阵						
	用户也可	J以通过 N	A 文件创函	建矩阵。M 文·	件实际上是	一种包含 M	IATLAB 代码的	的文本文	
如果	用户已经	E在 MATI	LAB 中创	建了一个文件	:				
	B=[								
	1.	0 5.	0 9.0	13.0					
	2.0	0 6.	0 10.	0 14.0					

44

3.0 7.0 11.0 15.0 4.0 8.0 12.0 16.0];将该文件存为 sample.m 后,在 MATLAB 命令窗口内输入 sample 回车 MATLAB 将会创建变量名为 B 的矩阵,用户再输入 B 后回车,将显示该矩阵 B =1 5 9 13 2 14 6 10 7 3 11 15 4 8 12 16 如果编写 M 文件 sample.m 时,没有在后加上";",输入 sample 将直接显示矩阵。 4. 使用 MATLAB 函数产生矩阵 MATLAB 为用户提供了四个能够产生基本矩阵的函数。 (1) zeros 函数 zeros 函数能够产生元素全为0的矩阵,如在 MATLAB 中输入: C = zeros(5,7)MATLAB 将会产生一个 5 行 7 列的空矩阵: C =0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (2) ones 函数 ones 函数能够产生元素全为1的矩阵,如在 MATLAB 中输入: D=ones(5,6)MATLAB 将会产生一个 5 行 6 列的元素全为 1 的矩阵: D = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 (3) rand 函数 rand 函数能够产生均匀分布的随机元素矩阵,如在 MATLAB 中输入: E=rand(7,9)MATLA 将会产生 7 行 8 列矩阵 E,每个矩阵元素的数值范围都是(0.0,1.0)。 E =Columns 1 through 7 0.9501 0.0185 0.1763 0.3529 0.2722 0.4186 0.6813 0.2311 0.8214 0.4057 0.8132 0.1988 0.8462 0.3795 0.5252 0.6068 0.4447 0.9355 0.0099 0.0153 0.8318 0.4860 0.6154 0.9169 0.1389 0.7468 0.2026 0.5028

0.8913

0.7621

0.4565

0.7919

0.9218

0.7382

0.4103

0.8936

0.0579

0.2028

0.1987

0.6038

0.4451

0.9318

0.4660

0.6721

0.8381

0.0196

0.7095

0.4289

0.3046

Columns 8 through 9

0.1897	0.3784
0.1934	0.8600
0.6822	0.8537
0.3028	0.5936
0.5417	0.4966

0.1509 0.8998

0.6979 0.8216

(4) randn 函数

### randn 函数能够产生正态分布的随机元素矩阵,如在 MATLAB 中输入:

F=randn(7, 7)

将会产生矩阵

 $\mathbf{F} =$ 

0.0000	0.5779	-1.4751	0.7990	1.0823	-0.9499	0.9863
-0.3179	0.0403	-0.2340	0.9409	-0.1315	0.7812	-0.5186
1.0950	0.6771	0.1184	-0.9921	0.3899	0.5690	0.3274
-1.8740	0.5689	0.3148	0.2120	0.0880	-0.8217	0.2341
0.4282	-0.2556	1.4435	0.2379	-0.6355	-0.2656	0.0215
0.8956	-0.3775	-0.3510	-1.0078	-0.5596	-1.1878	-1.0039
0.7310	-0.2959	0.6232	-0.7420	0.4437	-2.2023	-0.9471
		<b>NTN</b>				

除了以上几个函数, MATLAB 还为用户提供了一些产生特殊矩阵的函数。

(5) magic 函数

magic 函数是 MATLAB 中一个具有非常奇妙特性的函数,该函数能够自动产生一种特 殊的 N 阶方阵(其中, N=1, 3, 4, 5.....,可由用户指定),这些 N 阶方阵具有一个共同的特性,即矩阵的行、列和对角线上的

元素之和都相等。

例如,使用 magic 函数产生一个 6 阶方阵,在 MATLAB 命令窗口内输入

x=magic(6)

回车将显示矩阵

x =

35	1	6	26	19	24
3	32	7	21	23	25
31	9	2	22	27	20
8	28	33	17	10	15
30	5	34	12	14	16
4	36	29	13	18	11

不难发现,矩阵 x 的任意行、列和对角线上的元素之和都是 111。

(6) diag 函数

diag 函数可以产生对角矩阵。例如对于前面刚定义矩阵 x,输入

diag(x)

结果将显示矩阵 x 的对角矩阵

ans =

35

32

- 2
- 17
- 14

(9) eye 函数和 size 函数

size 函数用来返回指定矩阵的行数和列数, eye 函数可以产生指定行数和列数的单位矩阵。eye 函数和 size 函数常常在一起使用,格式为 eye(size(A)),用以产生与指定矩阵 A 同阶的单位矩阵。

例如,在MATLAB命令窗口内输入:

eye(size(x))

结果会产生一个6×6的单位矩阵:

ans=

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

5. 生成大矩阵

 $B = [x \quad x + 32; x + 48 \quad x + 16]$ 

在 MATLAB 中,可以将小矩阵联接起来生成一个较大的矩阵。事实上,前面直接输入生成矩阵的方法就是 将单个元素联接起来生成一个矩阵。矩阵的联接算子是方括号"[]"。

例如,可以使用前面例子中的矩阵 x 联接起来生成大矩阵 B。

B =	:											
	35	1	6	26	19	24	67	33	38	58	51	56
	3	32	7	21	23	25	35	64	39	53	55	57
	31	9	2	22	27	20	63	41	34	54	59	52
	8	28	33	17	10	15	40	60	65	49	42	47
	30	5	34	12	14	16	62	37	66	44	46	48

4	36	29	13	18	11	36	68	61	45	50	43
83	49	54	74	67	72	51	17	22	42	35	40
51	80	55	69	71	73	19	48	23	37	39	41
79	57	50	70	75	68	47	25	18	38	43	36
56	76	81	65	58	63	24	44	49	33	26	31
78	53	82	60	62	64	46	21	50	28	30	32
52	84	77	61	66	59	20	52	45	29	34	27

使用四个6×6的子矩阵组合产生了一个12×12的大矩阵。

在实际应用中,用户可以利用上述不同方法创建自己需要的矩阵。

6. 删除矩阵的行和列

在 MATLAB 里,可通过把矩阵的列或行赋值为空矩阵来删除某行或某列。例如命令

x=(:,2)=[]

将删除矩阵 x 的第二列。在 MATLAB 里,不能使用下面将要讲道德双下标来删除矩阵的某个元素。例如, 当矩阵 x 不为标量时,运行命令

x(1,2) = []

将给出错误信息。但可用单下标来删除矩阵的某个或某些元素,MATLAB 将删除这些元素,并将剩下的元 素按列编成一个长行向量。

例如对于一个 2×2 矩阵 A A=[ 1 2 3: 4 5 6]; 在命令窗口输入命令行 A(2:2:6)=[] 回车显示结果 A =1 2 3

3.1.3 下标的引用

在 MATLAB 中,矩阵元素的引用可用两个下标来表示,如在矩阵 A 中,第 i 行第 i 列的元素用 A(i,j)表示; 也可用一个下标来表示。用单个下标表示元素并不只限于向量。对于矩阵,由于 MATLAB 的运算基本上都是对 列操作的,矩阵可以认为是按列优先排列的一个长的列向量,从而可用单下标引用。例如2×2的矩阵 A,A(1) 表示第一列的第一个元素,A(2)表示第一列的第二个元素,A(3)表示第二列的第一个元素,A(4)表示第二列的第 二个元素。

当矩阵的下标超出矩阵的实际元素的下标时,将给出错误信息。但是,当某个值被赋给矩阵的—个新的元 素时,MATLAB会自动增加矩阵的维数大小。例如4×4的矩阵A,执行命令A(4,5)=17后,矩阵将变成4× 5 的新矩阵。

对于矩阵 A =[ 16 3 2 13; 5 10 11 7 9 6

15 4 14 1]; 在命令窗口输入

8;

12;

A(4,5)=17

回车显示结果

A =

16	3	2	13	0
5	10	11	8	0
9	6	7	12	0
4	15	14	1	17

在下标的表达式里使用冒号表示矩阵的一部分。例如 A(1:k,j)表示矩阵 A 的第 j 列的前 k 个元素。A(:,j)表示 矩阵 A 的第 j 的所有元素。由于有了冒号运算符,在 MATLAB 里编程变为特别简洁。例如,如果要求上例中矩 阵 A 的第 3 列元素之和, 只需在命令窗口输入命令行

sum(A(:,3))

回车显示结果

ans =

34

# 3.2 基本运算符

MATLAB 的运算符可分为四类:算术运算符、关系运算符、逻辑运算符和操作符。其中操作符和算术运算 符的优先级最高,其次是关系运算符,再其次是逻辑运算符。下面将分别介绍这几类运算符。

3.2.1 算术运算符

MATLAB 的算术运算符如表 3-1 所示。

表 3-1

#### 算术运算符

操作符	说明	
+	ከባ	
-	减	
*	矩阵乘	
*	数组乘	
^	矩阵乘方	
.^	数组乘方	
1	矩阵左除	
.\	数组左除	
/	矩阵右除	
./	数右除	

大部分的算术运算符只对相同维数酌数组的对应运算作运算。对于矩阵和向量,算术运算符连接的两个运 算数必须同维或者两个中有一个是标量。当一个运算数是标量时,运算符将招标量和另一个运算数的每一个运 算进行运算。

3.2.2 关系运算符

关系运算符对于程序的流程控制非常有用,在 MATLAB 的循环和条件控制中经常使用。MATLAB 的关系运算符如表 3-2 所示。

Ŧ	2		2
বহ	э	-	2

#### 关系运算符

操作符	说明
==	等于
~=	不等于
>	大于
<	小于
>=	大于等于
<=	小于等于

除了长方形矩阵和向量外,所有关系运算符连接的两个运算数必须是同维、每维同大小的。对于长方形矩

阵和向量,两个运算数或者同维、同大小,或者其中有一个运算数是标量。当有一个运算数是标量时,关系运 算符把标量和另一个运算数的每个运算进行比较。关系运算符将生成一个 0~1 矩阵,当运算数相应元素对于关 系运算符为真时,对应位置上生成 1,否则为 0。切记,MATLAB 的所有关系运算符都是对运算数中对应的元 素逐个计算的。下面是一个关系运算符的例子。第三句命令 A<B 将对矩阵 A 和 B 对应的元素逐个对比 / 当 A 中的元素比 B 中同位置的元素小时,结果矩阵中对应元素为1,否则为0。

在命令窗口中输入

A=[							
1	2	3					
6	9	7					
10	5	7]	;				
В=	[						
23	3	5					
0	8	3					
2	45	9]	;				
A <b< td=""><td></td><td></td><td></td></b<>							
回车显示结果							
ans =							
1	1	1					

0 0 0 1 0 1

当一个矩陈 A 和空矩阵用关系运算符比较时,除非矩阵 A 为主矩阵或标量,MATLAB 将给出错误信息。要 判断一个矩阵是否为空矩阵,用户可以使用函数 isempty()进行检验。例如当矩阵 A 为空矩阵时,在命令窗口中 输入

isempty(A) 回车显示结果

ans =

1

3.2.3 逻辑运算符

MATLAB 的逻辑运算符如表 3-3 所示。

表 3-3

逻辑运算符

操作符	说明
&	逻辑和
!	逻辑或
~	逻辑非

MATLAB 里中逻辑运算符和关系运算符一样,除了逻辑运算符的一个运算数为长方阵或向量时,另一个运 算数可为标量,其余情况下均要求矩阵同维。 逻辑运算符也对运算数的相应元素——运算,最后结果为一个 0~1 矩阵。当逻辑表达式的值为真时,赋值1,否则为0。

在 MARAB 里, 对于数值矩阵, 当元素为0时, 逻辑上为假; 当元素为非0时, 逻辑上为真。逻辑运算符 "与(&)"的运算数都为真时,表达式的值为真,其他情况(一真一假或两个都假)下为假。逻辑运算符"或()" 的两个运算数都为假时,表达式的值为假,其他情况下(一真一假或两个都真)下为真。逻辑运算符"非(~)"为 单目运算符(只有一个运算数),当运算数为真时,表达式的值为假;当运算数的值为假时,表达式的值为真。

### 3.2.4 操作符

MATLAB 中将会用到的部分操作符如表 3-4 所示。

表 3-4

4. 操作	乍符
符号	说明
:	冒号
;	分号,用于分隔行
1	逗号,用于分隔列
()	圆括号,指出在算术表达式中先后次序
[]	方括号,用于构成数组和矩阵
{}	大扩号,用于构成单元数组
	小数点或域访问符
	父目录
	用于语句行尾表示该行未完
%	用于注释
!	用于调用操作系统命令
=	用于赋值

上表中,只对这些符号做了简单说明,下面将对几个较为重要的符号做进一步介绍。

1.冒号

冒号是 MATLAB 最重要的运算符之一,也是 MATLAB 最常用的运算符之一。冒号可以用于以下几个方面: (1)冒号可以用来输入行数组。

例如,当用户在 MATLAB 命令窗口内输入

a=2:8

输入回车后,显示如下

a =

2 3 4 5 6 7 8

除了输入增量为1的行数组,冒号也可以指定一个增量,如输入

a=10:15:100

输入回车后,显示如下:

a=

10 25 40 55 70 85 100 用户还可以为行数组指定一个负增量,如输入

a=1:-1/8:1/16

输入回车后,显示如下:

a =

Columns 1 through 7

```
1.00000.87500.75000.62500.50000.37500.2500
```

Column 8

0.1250

(2) 冒号能够从矩阵和数组中挑选出指定的元素、行和列。

例如,对于前面例子中的矩阵 x

x = 35 26 19 24 1 6 3 32 21 23 25 7 31 9 2 22 27 20 8 28 17 10 15 33 30 5 34 12 14 16 4 36 29 13 18 11 在 MATLAB 命令窗口内输入:

x(1:4,5)

回车后显示

ans =

19 23

27

10

该命令指定从矩阵 x 中提取第一行到第四行每行的第五个元素。

同理,使用冒号能够从一个大矩阵中抽取小矩阵。例如,对于上例中的矩阵 x,可以使用冒号从 x 中抽出小矩阵 y,在命令窗口输入

y=x(1:3,:)

回车显示结果 v =

35	1	6	26	19	24
3	32	7	21	23	25
31	9	2	22	27	20

很明显,矩阵 y 是由矩阵 x 的前三行组成的。

(3) 冒号还可以用来产生简单的表格

先请读者观察下面的一小段程序

x=(-1:0.1:1);

- y=sin(x);
- z=[x,y];

这段程序首先利用冒号运算产生行数组,接着将该行数组赋给变量 x;计算变量 x 中每个元素的正弦值,得 到另一个列数组并赋给变量 y,将 x 和 y 合并,组成一个有两列的矩阵 z。这段程序的输出结果如下:

```
z =
```

Columns 1 t	hrough 7							
-1.0000	-0.9000	-0.8000	-0.7000	-0.6000	-0.5000	-0.4000		
Columns 8 t	hrough 14							
-0.3000	-0.2000	-0.1000	0	0.1000	0.2000	0.3000		
Columns 15 through 21								
0.4000	0.5000	0.6000	0.7000	0.8000	0.9000	1.0000		
Columns 22	through 28							
-0.8415	-0.7833	-0.7174	-0.6442	-0.5646	-0.4794	-0.3894		
Columns 29	through 35							
-0.2955	-0.1987	-0.0998	0	0.0998	0.1987	0.2955		
Columns 36	through 42							
0.3894	0.4794	0.5646	0.6442	0.7174	0.7833	0.8415		

.

2.分号

分号除了在矩阵中用来分隔行以外,如果出现在一条语句的末尾,则说明除了这条语句外,还有语句等待 输入。这时,MATLAB将不给出运行的中间结果,当所有语句输入完毕,回车后,才显示最终的运行结果。如 上例小程序的输入,又例如,在 MATLAB的命令窗口内输入

A=eye(3,4)'; B=[1 2 3; 4 5 6; 7 8 9; 11 12 13]; 在上面的一小段程序中,可以找到分号的两种作用。如果用户输入第一条语句时,末尾不加分号,MATLAB 将直接输出单位矩阵

A=

 $\begin{array}{cccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$ 

0

0 0

正因为在输入前两个语句时在末尾都加上了分号, MATLAB 并不输出中间结果, 而是当输入语句 A+B 并回 车后, 才显示最终结果

ans= 2 2 3 4 6 6 7 8 10

11 12 13

矩阵 A 种 " ' " 表示矩阵转置,分号 " ;" 的作用是不显示中间结果。在矩阵 B 中,分号用来分隔行,这也 是分号的最基本作用。 、

3.方括号

方扩号"[]"可以用来输入矩阵,还是矩阵的联接算子。除词以外,方括号还可以用来删除矩阵的行或列。 例如,在 MATLAB 命令窗口内输入下列语句

x=magic(5);

x(:,3)=[];

Х

回车显示结果

x =

17	24	8	15
23	5	14	16
4	6	20	22
10	12	21	3
11	18	2	9

可以发现,这个5阶魔术方阵的第三列元素被删除了。

需要注意的是,使用方括号删除了矩阵的行或列后,该矩阵仍应当为一个符合定义的矩阵。在使用方括号 对矩阵元素的删除操作时,如果剩余的元素不能够构成符合定义的矩阵,这样的操作将产生错误。

例如,删除矩阵 x 中的一个元素,在 MATLAB 的命令窗口内输入

x(1,3)=[]

MATLAB 将产生一个错误信息:

??? Indexed empty matrix assignment is not allowed.

4."…"

在前面已经讲到过,在输入程序时,如果遇到较长的命令行而在一行中无法输入该命令行时,可以在未完的语句末端输入三个点"..."来表示将在下一行继续输入。

例如,输入如下命令:

+1/256+1/512+1/1024;

可以注意到,在第一行的末尾,添加了"...",这样,MATLAB 认为这一命令行还没有结束,用户可以在 下一行接着输入程序语句。

a=1+1/2+1/4+1/8+1/16+1/32+1/64+1/128...

# 3.3 变量

与一般的编程语言不同,MATLAB并不要求用户在输入变量的时候进行声明,也不需要指定其阶数。当用 户在 MATLAB 工作空间内输入一个新的变量时,MATLAB 会自动给该变量分配适当的内存,若用户输入的变 量已经存在,则 MATLAB 将使用新输入的变量替换原有的变量。

### 3.3.1 变量命名规则

MATLAB 的变量命名规则如下:

- (1) 变量名应以字母开头
- (2) 变量名可以由字母、数字和下划线混合组成
- (3) 组成变量名的字符长度应不大于 31 个
- (4) MATLAB 区分大小写

例如,在MATLAB 命令窗口内输入

Record_l=234

则创建了一个变量名为 Record_1, 仅有一个元素的矩阵, 读者可以注意到, 改变量名是由字母、数字和下 划线混合组成的。

一旦知道变量名,用户就可以查询该变量的构成。

例如,存在已经定义过的变量 b

b=[1 2 3 ; 4 5 6 ; 7 8 9]

如果用户已经忘记改变量的值,则可在 MATLAB 命令窗口内直接键入变量名

b

回车后, MATLAB 会显示变量 b 的值:

b=

1 2 3 4 5 6

7 8 9

3.3.2 固定变量

在 MATLAB 中,存在一些固定的变量,如表 3-5 所示。

表 3-5

固定变量

变量名	说明
1	虚数单位,定义为 $\sqrt{-1}$
j	虚数单位,作用同 i
pi	圆周率
eps	浮点相对精度
realmin	最小的正浮点数
realmax	最大的正浮点数
inf	无穷大
NaN	Not-a-number,表示不定值

在 MATLAB 中,圆周率 p 的值可以用以下两个语句计算得出

4*atan(1)

或

imag(log(-1))

其结果都为

ans=

3.1416

(1) 变量 eps 在决定奇异性和秩时,被作为一个容许误差,其值为: eps=2.2204e-016

(2) realmin 是 MATLAB 提供的最小的正浮点数。

(3) realmax 是 MATLAB 提供的最大的正浮点数。

(4)在 MATLAB中,用 inf 表示无穷大,变量 inf 是当一个非零数除以零时产生的。

例如,在 MATLAB 命令窗口内输入:1/0

结果如下,用户可以发现,MATLAB将并不产生错误,而是产生一个警告信息和一个特殊值 inf。

Warning: Divide by zero.

ans =

Inf

(5)固定变量 NaN 用来表示不定值,当出现 0/0 等在数学上难以定义的表达式时,MATLAB 将用变量 NaN 来表示计算结果。

例如,在命令窗口中输入

inf/inf

回车显示结果

ans =

NaN

3.3.3 全局变量

同其他编程语言一样,在使用 MATLAB 进行编程的过程中,有时会需要某个变量作用在多个函数上。这时 只要将该变量声明为全局变量即可,即在该变量前添加关键字 "global "加以定义。

定义后的全局变量可以被所有定义该变量为全局变量的函数使用,如果想在命令行中使用全局变量,也要 定义该变量为全局变量。

如果函数里的变量为局部变量,则只能被本函数调用,在其他函数里和基本的工作空间中不能调用。和在 命令窗口中一样,在 M 文件中,变量在使用前无需定义其维数和每维的大小。但是在把一个变量付给另一个变 量时,要求等号右边的变量有值。

在 MAILAB 里,变量名区分大小写。而且尽管不是必须的,但作为一个约定俗成的规则,MATLAB 程序员 应尽量用大写字母书写全局变量。

下面为一个使用全局变量的例子。函数 weight_add 将输入的参数加权相加,建立文件 weight_add.m,在第 三行定义全局变量 ALPHA 和 BETA。文件内容为:

function all=weight_add(x,y)

%add two variable with differrent weight.

Globle ALPHA BETA

all=ALPHA*x+BETA*y;

在命令窗口中输入

global ALPHA BETA

ALPHA =1

BETA=2

all=weight_add(1,1);

将在工作空间生成变量 all,其值为 3。由于在函数和工作空间中都把 ALPHA 和 BETA 两个变量定义为全 局变量,所以只要在命令窗口中改变 ALPHA 和 BETA 的值,就可改变加权值,而无需修改 weight_add.m 文件。

上例只在函数 weight_add 和命令窗口中把 ALPHA 和 BETA 变量定义为全局变量,在编程时,可在所有需 要调用全局变量的函数里定义全局变量,这样就可实现变量共享。为了在工作空间中使用全局变量,也要定义 全局变量。

在函数文件里,全局变量在使用前必须先声明,全局变量的定义语句应放在变量使用以前,为了便于了解 所有的全局变量,一般把全局变量的定义语句放在文件的前部。

# 3.4 数值

在 MATLAB 中数值的表示方法有很多。MATLAB 可以使用传统的十进制计数法来表示一个数,也可以使 用科学计数法表示一个数。

MATLAB 可以对复数进行运算。在 MATLAB 中,复数单位 i=sqrt(-1),在命令窗口中显示其值为:

i =

0 + 1.0000i

用户可以利用如下语句复生成数

z=a+bi

或

z=r*exp(**q** *i)

其中 r 是复数的模, q 是复数幅角的弧度数。

在 MATLAB 中,所有数值在内存的存储格式都是使用 IEEE(电器和电子工程师协会)的 浮点标准规定的长 型格式,其数值的有效范围约为从到 10⁻³⁰⁸到 10⁺³⁰⁸。

下面列出的数值表示方法在 MATLAB 中都是合法的:

1	87.685489	28i
-1999	5.075868e-33	-3.14159i
0.123456	7.15263e58	13e6i

## 3.5 数据类型

在 MATLAB 里共有六种基本的数据类型,每一种类型都可以是一维、二维和多维的。这六种是:双精度型 (double)、字符型(char)、稀疏型(sparse),8 位型(uint8)、细胞型(cell)和结构型(struct)。一般把这六种类型的二维 变量称为矩阵,其中最常用的是双精度矩阵和字符型数组。MATLAB 的计算都采用双精度,MAIL4B 提供的绝 大部分函数都是对双精度矩阵和字符串操作的,其他几种数据类型用于特殊的场合。比如8 位型用于图像处理。 稀疏型用于稀疏矩阵,细胞型和结构型一般用于编写大型软件。这六种数据类型的关系可用图 3-22 表示。

在图 3-22 中,数组在最上一层,表明 MARAB 的所有数据类型都是向量。从大类止数组可分为字符型、细胞型、结构型和数值型;数值型又可分为双精度型和 8 位型;稀疏型是双精度型的一种,因为稀疏矩阵的每一个元素都是以双精度存储和运算的。表 3-6 列出了这些数据类型的一些例子。

表 3-6

#### 数据类型

数据类型	举例	说明
doub l e	[1 2;3 4]	双精度数值类型,是最常用的类型
char	'Hello'	字符数组,每个字符占16位
sparse	speye(5)	双精度稀疏矩阵,只存储矩阵中的非0元素
cell	<pre>{17'hello'eye(2)}</pre>	细胞数组,数组中的每个元素可为不同类型、不同维数
struct	a.day=12;	结构数组相当于数据库的记录,把相关的数据列在一
	a.color='Red';	起,称为属性,不同属性的数据类型可以不同
	a.mat=magic(3);	
unit8	unit8(magic(3));	8 位型,为无符号整数,最大可表示 255,不可进行数学运算

变量的数据类型可用函数 isa 来查看,其调用格式为:

isa(变量名,数据类型)

比如当矩阵 s 为稀疏矩阵时,下面三条命令都将返回 1。

isa(s, 'sparse')

isa(s, 'doublel')

isa(s,' numeic ')

MARAB的六种数据类型都支持一定的函数和运算(方法),子一层的数据类型支持其父一层的所有运算,例

如双精度型数据支持所有数组一层的计算。表 3-7 将列出所有的数据类型支持的方法。

表 3-7	数据类型及支持方法								
_	数据类型方法								
	数组	多维下标、组合(如[a,b])、转置、行列初等变换、数组变形(reshap)、求维数、各							
	维的大小(size,length,ndims)								
	细胞型 各元素用{}引用。								
_	字符型 字符函数(strcmp,lower),计算时自动转换成双精度型								
	数值型 Find 函数、复数元素、冒号算符(1:10)								
		(sin、cos、sum,prod、sort等)。							
_	8 位型	存储特性							
	稀疏型	稀疏函数和算符(/ , .* , splu , spchol 等)							
	结构型	属性引用							

和其他应用软件不同之处在于,在 MATLAB 里,不能用 double、char 来定义变量。之所以把数据类型分为 六种,只是为了把具有相同性质的数组归纳分类。

在 MATLAB 里,有些数据如矩阵下标等并不需要采用双精度,如果采用双精度存储,将占用大量的空间。 8 位型数据正是为了弥补双精度数据的不足。8 位型数据不能做任何计算,在计算之前,必须用函数 double 转换 为双精度数据。

在 MATLAB 里,可以创建自己的数据类型,也可以为已有的数据类型增加新的使用方

法。自己定义的数据类型可以和 MATLM 已有的数据类型一样使用。

# 3.6 数组

数组是 MATLAB 中一个较重要的概念,实际上,在线性代数的范围以外,矩阵可以被当作二阶的数值型数组。但是 MATLAB 中,数组和矩阵的运算却有较大的区别。在本小节中,将重点讨论关于数组运算的有关内容。

1.数组的加减运算

数组的加减运算和矩阵的加减运算的规则是相同的,数组的加减运算也就是数组的相应元素的加减运算。 例如,对于已知数组

a =				
	1	2	3	
	4	5	6	
	7	8	9	
b =				
	1	2	3	
	4	5	6	
	7	8	9	
输入	∖a+b	,然后回	回车,	显示结果为
ans	=			
	2	4	6	
	8	10	12	

14 16 18

可见,数组的加法与矩阵的加法遵循的相同的运算规则,就是数组的相应元素的加减运算。

2. 数组的乘除运算

数组的乘法运算符是".*",即在乘号前加一个点来表示是数组的乘法运算。数 组的乘法运算必须在具 有相同阶数的数组间进行,其结果是数组的对应元素间相乘的结果组 成的新数组。

例如,对于数组 a 和 b,在 MATLAB 命令窗口中输入

a.*b

回车显示结果为

ans =		
1	4	9
16	25	36
49	64	81

由此可见,数组的乘法与矩阵的乘法遵循的运算规则是不同的。但是在 MATLAB 中,可以对数组使用矩阵的运算符号,也可以对矩阵使用数组的运算符号。当对数组使用矩阵的运算符号时,实际上是把数阻挡左矩阵进行运算;同样,对矩阵使用数组的运算符号,实际上是把矩阵当作数组进行运算。

数组的除法与矩阵的除法其运算法则也是不相同的。

同样对于上面的数组 a 和 b,在 MATLAB 命令窗口中输入

a.∖b

回车显示结果为

ans =		
1	1	1
1	1	1
1	1	1

可见,数组的除法也是对应元素间的相除。

数组的除法没有左除和右除,即符号".\"和"./"表示的作用是一致的,这一点也是与矩 阵除法不一样的无影响。

顺便提一下,数组与一个标量间的乘除运算就是指该数组的每个元素与这个标量分别进行乘除运算。

3.数组的乘方

数组的乘方使用运算符".^"进行表示。为了更清晰地说明数组的乘方运算,这里给出了一个较简单的例子。 例如,对于已知数组 x 和 y

- x= 6
- 4
- 2
- -
- y= 1
- 2

3

在 MATLAB 命令窗口中输入

x.^y

回车显示结果为

ans =

6

16 8

可以将结果写成如下形式,也许能更方便地说明

ans =

 $6=6^{1}$ 

 $16=4^2$ 

 $8=2^{3}$ 

由上式可以清晰的看到,数组的乘方运算实际上是以前一个数组为底,后一个数组为指数,其对应的元素 分别进行指数运算得到的结果。显然,数组间的乘方运算只有在阶数相同的数组间才能进行。

对于数组和标量进行乘方运算,可以分为两种情况。一种是数组为底的情况,另一种是数组为指数的情况。

58

对于前一种情况,例如输入

x.^3

回车显示结果为

ans =

 $216=6^{3}$ 

 $64=4^3$ 

 $8=2^{3}$ 

其结果实际上是以数组中的每个元素为底,分别与作为指数的标量进行乘方运算得到的一个新的数组。 对于后一种情况,例如输入

3.^y

回车显示结果为

ans =

 $3=3^{1}$ 

 $9=3^{2}$ 

 $27=3^{3}$ 

其结果实际上是以给定的标量为底,分别以数组中的每个元素作为指数进行乘方运算,而得到的一个新的 数组。

3.7 字符串

在 MATLAB 中,字符串是作为字符数组用单引号输入到程序中的。

例如,用户可以在命令窗口中输入

t='hello the world'

≥ 字符数组是计算空格的,所以变量t实际上是一个1×15的字符数组,而不是一个1×13字符数组。

在 MATLAB 内部,字符是以数值的格式存储的。用户可以使用如下命令查看变量 t 在 MATLAB 内部的存储格式:

x=double(t) 回车后结果如下:  $\mathbf{x} =$ Columns 1 through 13 101 108 108 111 32 116 104 101 32 119 111 114 104 Columns 14 through 15 108 100 可以看到,变量t中的每个字符元素都被转化成ASCII码的相应数值。 用户还可以使用函数 char()将 ASCII 码的相应数字转化成字符,在 MATLAB 命令窗口内输入 char(x)回车后,显示结果如下: ans =hello the world 通过字符与数值的互相转变,我们就可以研究计算机使用的不同字体。在 ASCII 码基础字符集中,用整数 32 至 127 表示可印刷的字符(32 以下的整数被用做表示不可印刷的控制符号),在 MATLAB 中,这 96 个数字被

放置在一个6×16的数组中,在 MATLAB 命令窗口中输入如下命令:

F=reshape(32:127,16,6)' 回车后,将显示这个数组 F=

	Colu	ımns 1 t	through	13									
	32	33	34	35	36	37	38	39	40	41	42	43	44
	48	49	50	51	52	53	54	55	56	57	58	59	60
	64	65	66	67	68	69	70	71	72	73	74	75	76
	80	81	82	83	84	85	86	87	88	89	90	91	92
	96	97	98	99	100	101	102	103	104	105	106	107	108
	112	113	114	115	116	117	118	119	120	121	122	123	124
	Colu	ımns 14	through	n 16									
	45	46	47										
	61	62	63										
	77	78	79										
	93	94	95										
	109	110	111										
	125	126	127										
	可以使	用函数	char()	将数组	F中的	数值转	化成相	应的字	符,输	入			
	char(F)	)											
	回车后	;,结果	?如下:										
ans	=												
	!"#\$%。	&'()*+,-	/										
	012345	56789:;•	<=>?										
	@ABC	CDEFG	HIJKLN	ИNO									
	PQRST	ruvw2	XYZ[\]^	`									
	`abcdef	fghijkln	nno										
	pqrstuv	/wxyz{	}~●										

3.8 M文件

M 文件是 MATLAB 所特有的使用该语言编写的磁盘文件。所有的 M 文件都是以".m"作为文件的扩展名。 MATLAB 中的 M 文件可以分为两种类型,一种是命令文件,另一种是函数文件。

3.8.1 命令文件

在 MATLAB 中,既不接受输入参数也不返回输出参数的 M 文件称之为脚本。这种 M 文件是在 MATLAB 的工作空间内对数据进行操作的。

当用户在 MATLAB 中调用一个脚本文件, MATLAB 将执行在该脚本文件中所有可识别的命令。脚本文件 不仅能够对工作空间内已经存在的变量进行操作,还能够使用这些变量创建新的数据。

尽管脚本文件不能返回输出参数,但其建立的新的变量却能够保存在 MATLAB 的工作空间之中,并且能够 在之后的计算中被使用。除此之外,脚本文件还能够使用 MATLAB 的绘图函数来产生图形输出结果。

### 3.8.2 函数文件

在前面的有些章节的内容里,读者实际上已经接触了一些函数。

与脚本不同, MATLAB 中的函数是指那些能够接收并输出参数的 M 文件。在 MATLAB 中, 函数名和 M 文件 件名可以是相同的, 但函数的对变量的操作是在其函数体内, 这种操作与 MATLAB 在工作空间内对变量的操作 性质是不同的。

下面的这个例子可以帮助读者更好的理解这一点。

例如,对于求矩阵的秩的函数"rank",可以查看该函数的代码。

在 MATLAB 的命令窗口中输入:

type rank

将显示函数" rank' '的代码如下:

function r = rank(A,tol)

%RANK Matrix rank.

% RANK(A) provides an estimate of the number of linearly

% independent rows or columns of a matrix A.

% RANK(A,tol) is the number of singular values of A

% that are larger than tol.

% RANK(A) uses the default tol = max(size(A)) * norm(A) * eps.

% Copyright 1984-2000 The MathWorks, Inc.

% \$Revision: 5.9 \$ \$Date: 2000/06/01 02:04:15 \$

s = svd(A);

if nargin==1

tol = max(size(A)') * max(s) * eps;

end

r = sum(s > tol);

在 MATLAB 中,函数代码的第一行总是以关键字"function"开始,同时,该行也给出了函数名和参数的状态。

在上例中,函数代码的第一行给出了函数名"rank",还显示该函数有两个输入参数及一个输出参数。

读者可以看到,在该函数代码的随后几行都是以"%"号开头,以"%"号开头的语句行在 MATLAB 中被 用来表示该行是对该函数的注释,这些注释行将在用户使用帮助命令查 询该函数时显示。

函数代码的最后一部分,就是 MATLAB 中实现该函数功能的执行语句。这部分代码实际上才是函数代码的 主体。函数体中的所有参数如 A, tol, r 都是函数内部变量,这些变量不同用户的定义变量,与 MATLAB 工作 空间内的变量是分离的。

MATLAB 中为用户提供了功能各异的函数。按照函数类型将之分为基本数学函数、专用数学函数、矩阵函数、绘图函数、多项式函数、数据分析函数、各种功能的工具箱函数等几大类。由于 MATLAB 中函数数量巨大、功能各异,不可能对每个函数的功能和特点都进行详细介绍,本小节中仅对上述类型函数的前几种加以简单说明和举例。用户可以参阅 MATLAB 帮助,对某些函数进行更详细的了解。在本书的后继章节中,随着内容的不断深入,也将遇到更多的函数。

1. 基本数学函数

MATLAB 基本数学函数可以实现较为简单的数学计算,如求绝对值、平方根等。由于 MATLAB 中支持复数运算,所以对负数进行开方和作对数运算,在 MATLAB 中不会发生错误,而且将给出适当的复数结果。 MATLAB 的基础数学函数包括以下几个方面:

(1) 三角函数

- (2) 指数函数
- (3) 复数函数
- (4) 舍入函数

下面将举出几个例子来简要说明 MATLAB 的基础数学函数的用法。

例(1), 在 MATLAB 的命令窗口输入

a=0:0.1:1 ; b=sin(a) ;

 $0=\sin(a)$ 

b t n - t

回车显示结果为

b =

Columns 1 through 7

0 0.0998 0.1987 0.2955 0.3894 0.4794 0.5646 Columns 8 through 11

0.6442 0.7174 0.7833 0.8415 使用 MATLAB 的三角函数可以很容易地计算出用户要求的任何三角函数值。 例(2),在MATLAB的命令窗口输入 A=1:10: B=sqrt(A); B 回车显示结果为 B =Columns 1 through 7 1.0000 1.4142 1.7321 2.0000 2.2361 2.4495 2.6458 Columns 8 through 10 2.8284 3.0000 3.1623 在本例中,方便地求出了一个一阶矩阵中的每个元素的平方根。 例(3),在 MATLAB 的命令窗口输入 imag(3-5i)回车显示结果为 ans = -5 该函数用来求复数的虚部,MATLAB 提供的复数方面的函数可以对复数及复数矩阵进行运算。 例(4),对于矩阵  $\mathbf{x} =$ [0.0123 -0.0123 1.7958 -1.7958 5.6328 -5.6328] 在 MATLAB 的命令窗口输入: fix(x) 回车显示结果为 ans = 0 0 -1 1 5 -5 该函数用于将矩阵中的元素向零舍入。 从以上几个例子不难发现,使用 MATLAB 的函数进行简单数学运算实在是非常方便快 捷的。 🗻 用户可以在命令窗口内输入如下命令,用以查看基本函数列表: help elfun 2. 专用数学函数 MATLAB 在为用户提供较为基本的数学函数的同时,也为用户提供了更多、功能更强大的专用数学函数, 如 MATLAB 提供的专用数学函数贝塞尔函数等。几乎全部的 MATLAB 数学函数都能进行复数运算。 专用数学函数包括以下几类: (1) 特殊数学函数 (2) 数论函数 (3) 坐标变换函数

下面的几个例子简要说明了专用数学函数的用法。

例(1),对于矩阵

4

x=
[1

62
3 7 5 9] 若需要计算矩阵 x 的每个元素的 gamma 函数,用户可以在 MATLAB 的命令窗口输入 gamma(x) 回车显示结果 ans = 1 6 2 720 24 40320 在数论函数中,函数"lcm()"用于求两个矩阵的相应元素的最小公倍数, (2),对于已定义的矩阵 X 和 Y X=[15 7 9 3 2 11] Y=[6 28 7 21 19 7] 在 MATLAB 的命令窗口输入 lcm(X, Y)回车显示结果 ans =30 28 21 63 38 77 用户可以在命令窗口内输入如下命令,用以查看专用函数列表: help specfun 3. 矩阵函数 MATLAB 的矩阵函数主要包括以下几类: (1) 矩阵分析函数 (2) 线性方程函数 (3) 奇异值分解函数 (4) 本征值函数 下面分别举例对 MATLAB 的矩阵函数加以简单说明。 在矩阵分析函数中,主要是关于秩与条件的函数。 例如,对于矩阵 A= [17 24 1 8 15 7 23 5 14 16 4 6 13 20 22 10 12 19 21 3 11 18 25 2 9] 使用"rank()"求该矩阵的满秩,在MATLAB的命令窗口输入: rank(A) 回车显示结果

ans =

5

在 MATLAB 中, 计算秩有几种不同的方法, 如果采用不同的方法其舍入规则不同, 对同一矩阵可能得到完 全不同的值。

在线性方程函数中,最基本的三角分解函数可以将一个方阵表示成两个基本三角阵的积,其中一个三角矩 阵为上三角矩阵,另一个为下三角矩阵。这种分解被称为"lu"分解,其算法是高斯变量消去法。

例如,对于方阵 B

	B=						
	16	2	3	13			
	5	11	10	8			
	9	7	6	12			
	4	14	15	1			
使月	<b></b> ¶ " lu	" ' 分	≧解,在	MA	TLAB 的i	命令窗口输	ì入
[1,u	]=1u(1	B)					
回3	车显示	结果					
1 =							
	1.00	00		0	0	(	)
	0.31	25	0.7685	5	1.0000	0	
	0.56	25	0.4352	2	1.0000	1.0000	
	0.25	00	1.0000	)	0	0	
u =							
	16.00	00	2.0000	)	3.0000	13.0000	
		0	13.500	0	14.2500	-2.2500	
		0		0	-1.8889	5.6667	

0 0 0 0.0000

用户可以使用该运算的逆运算来证明"1u"分解的正确性。即,在MATLAB的命令窗 口输入

l*u

回车显示结果

ans =

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

显然,使用"1u()"函数能精确地将一个方阵分解为两个基本三角阵,其中,用"1"表 示下三角矩阵,而用"u"表示上三角矩阵。

在 MATLAB 中,矩阵的正交变换能够通过 MATLAB 提供的 "qr()"函数方便地实现。 例如,对于上例中的矩阵 A,在命令窗口中输入:

[q,r]=qr(A)

回车显示结果

q =

-0.5234	0.5058	0.6735	-0.1215	-0.0441
-0.7081	-0.6966	-0.0177	0.0815	-0.0800
-0.1231	0.1367	-0.3558	-0.6307	-0.6646
-0.3079	0.1911	-0.4122	-0.4247	0.7200
-0.3387	0.4514	-0.4996	0.6328	-0.1774

1	r =							
	-32.4808	-26.6311 -	21.3973 -	23.7063 -2	25.8615			
	0	19.8943	12.3234	1.9439	4.0856			
	0	0	-24.3985	-11.6316	-3.7415			
	0	0	0	-20.0982	-9.9739			
	0	0	0	0	-16.0005			
	用 q*r 能够很	民容易地验证	E其结果为	A,在命令管	窗口输入			
	q*r							
	回车显示结算	果						
;	ans =							
	17.0000	24.0000	1.0000	8.0000	15.0000			
	23.0000	5.0000	7.0000	14.0000	16.0000			
	4.0000	6.0000	13.0000	20.0000	22.0000			
	10.0000	12.0000	19.0000	21.0000	3.0000			
	11.0000	18.0000	25.0000	2.0000	9.0000			
:	在 MATLAB	中,对矩阵	进行奇异值	i分解是使用	3 " svd() "	函数来完成的。	在奇异值分解中	产生三个因数矩阵
" u "	、" S "、" V "。	其中 " u " 矩	E阵和" v "	矩阵是正交	矩阵 ," s '	'矩阵是对角矩	阵,对于被进行者	奇异值分解的矩阵,
假设	为 X,则 svo	d(X)恰好返回	回矩阵"s"	的对角元素	₹,即 X 的	奇异值。		
	例如,对于,	上例中的矩网	车 B,将其ì	进行奇异值	分解,在奇	命令窗口内输入	:	
	[u,q,v]=svd(E	3)						

回车显示结果

u =

	-0.5000	0.6708	0.5000	-0.2236
	-0.5000	-0.2236	-0.5000	-0.6708
	-0.5000	0.2236	-0.5000	0.6708
	-0.5000	-0.6708	0.5000	0.2236
q =	=			
	34.0000	0	0	0
	0	17.8885	0	0
	0	0	4.4721	0
	0	0	0	0.0000
v =	:			
	-0.5000	0.5000	0.6708	0.2236
	-0.5000	-0.5000	-0.2236	0.6708
	-0.5000	-0.5000	0.2236	-0.6708
	-0.5000	0.5000	-0.6708	-0.2236

在 MATLAB 中,还有几种函数也可以被进行奇异值分解,包括欧几里德矩阵范数 norm()、条件数 cond() 和秩 rank()函数等。

一般来说, 对于 n 阶矩阵 A, 如果有 n 个 l 值满足  $Ax=l x \parallel b$  A 的本征值 x 为 A 的本征向量。MATLAB 中计算本征值使用函数 " eig(A) ", 并以列数组的形式返回本征值。

如果矩阵 A 为实对称矩阵,其本征值为实数;如果阵 A 为非对称矩阵,其本征值为复数。

例如,对于实对称矩阵 X 和非对称矩阵 Y

X =

[0 0.5000 0.5000 0.5000 2.0000 -1.5000 0.5000 -1.5000 [0

Y=
[-1 0
0 1]
在 MATLAB 的命令窗口内输入
eig(X)
回车显示结果
ans =
-1.1908
0.3724
2.8184
在 MATLAB 的命令窗口内输入
eig(Y)
回车显示结果
ans =
-1

1

4. 绘图函数

MATLAB 的绘图功能非常强大, MATLAB 为用户提供了许多绘图函数, 用于绘制二维图形、三维图形以及特殊图形。

- (1)用于绘制二维图形的函数包括以下几种类型:
  - ▶ 绘制 X-Y 坐标图的函数
  - ▶ 坐标控制函数
  - ▶ 用于给图形添加注解的函数
  - > 复制和打印图形的函数

MATLAB 能够绘制线性坐标、全对数坐标、半对数坐标和极坐标下的二维图形,并能够给图形添加标注。 MATLAB 最基本的绘图函数是 " plot() "。

例如,对于一个列数组

23]

若需要绘制出与其下标对应的图形,可在 MATLAB 的命令窗口内输入:plot(y) MATLAB 将打开一个图形窗口,如图 3-1 所示。

66



图 3-1 向量 y 的图形表示

```
使用 MATLAB 的坐标控制函数能对图形进行控制,如添加网格线、标注等。
例如,在 MATLAB 命令窗口的以上语句后输入
```

grid title('sample'); xlabel('x'); ylabel('y');

在图形窗口中被自动添加了网格线及标注,如图 3-2 所示。





(2)除了绘制二维图形, MATLAB 还为用户提供的绘制三维图形的函数, 使得其绘图功能可以应用到更 广泛的领域。用于绘制三维图形的函数包括以下几种类型:

- > 绘制基础三维图形的函数
- ▶ 颜色控制函数
- ▶ 光照函数
- ▶ 颜色映象函数
- > 视点控制函数

本书将在后续章节中详细讲解 MATLAB 的三维绘图功能,这里仅举出一个例子来简要先是一下 MATLAB 的三维绘图功能。

三维绘图函数中,"mesh(Z)"函数可以绘制矩阵 Z 中元素的三维网格图。在 MATLAB 的命令窗口输入: mesh(eye(18))

MATLAB 将在绘图窗口中绘制如图 3-3 所示的图形。

(3) 此外, MATLAB 的用于绘制特殊图形的特殊绘图函数包括以下几种类型:

- > 绘制特殊二维图形的函数
- ▶ 绘制特殊三维图形的函数
- ▶ 图象显示
- ▶ 产生动画效果的函数
- > 实体模型函数



图 3-3 "mesh()"函数应用举例

MATLAB 可以用不同的方式绘制图形,如填充图、条状图及饼状图等。例如,对于矩阵 X=

[1 2

4 5 6

7 8 9]

在 MATLAB 命令窗口中输入: area(X);

3

则 MATLAB 将在打开的图形窗口线是如图 3-4 所示的图形。



图 3-4 矩阵 X 的填充图

在 MATLAB 命令窗口中输入

bar(X);

则 MATLAB 将在打开的图形窗口线是如图 3-5 所示的图形。





本小节仅对 MATLAB 的绘图函数作了概括介绍,在本书的后面的章节中将对如何在 MATLAB 中绘制图 形作更详细地介绍。

MATLAB 中的函数是以两种不同的方式提供给用户的。一种是内置在 MATLAB 的核心 中的函数,如 sqrt 函数和 sin 函数,另一种是通过 M 文件实现的函数,如 gamma 函数。其中,由于前一种函数内置在 MATLAB 的核心中,所以执行效率非常高。而通过 M 文件实现的函数极大地扩展了 MATLAB 的功能。

# 3.9 流程控制语句

MATLAB 提供了简明的流程控制语句以便用户使用。本小节将对 MATLAB 的流程控制 语句进行简要的讨论。表 3-6 列出了流程控制语句的关键字:

表 3-6		流程控制语句关键字
-	关键字	说明
_	if	条件执行语句
_	elseif	if 语句条件
_	else	语句条件
_	end	终止作用域
_	for	指定次数的循环
	while	不指定次数的循环
_	break	终止循环
_	switch	开关语句
	case	列出不同情况
	otherwise	否则语句
	return	返回到调用函数

### 3.9.1 if-else 语句

if 语句将计算所跟逻辑表达式的值,若该逻辑表达式的值为真,将继续执行后面的语句。关键字 elseif 和 else 提供了 if 语句的条件,关键字 end 用来终止 if 语句的作用域。

#### if 语句的最简单的格式是

if logical_expression

statements

end

如果该逻辑表达式为真,MATLAB 将执行" if "和" end "之间的所有语句;如果该逻辑表达式为假,MATLAB 将跳过" if "和" end "之间的所有语句,并且继续执行下面的语句。

例如

```
if rem(x,2)==0
disp('x is even')
y=x/2 ;
```

end

如果该逻辑表达式是非标量的矩阵,则该矩阵的所有元素必须都是非零元素。例如,当 A 为矩阵时,下面 的语句

if A

statements

end

实际上相当于

if all(A(:))

statements

end

又如,对如下语句

if a==b

当 a 和 b 都是标量时,该语句是合法的;但当 a 和 b 是矩阵时,语句 " a==b "将不再是判断 a 和 b 是否相等,而将计算矩阵 a 和 b 何处相等。实际上,如果 a 和 b 是不同阶数的矩阵时,语句 " a==b "将产生错误。

在 MATLAB 中,更恰当的判断相等的方法是使用 MATLAB 提供的函数"isequal",用户可以将上面的语句 改为

isequal(a,b) 这样将会减少出现意外错误的可能性。 if 语句的更一般的格式是

if expression

statementsl elseif expression statements2 else statements3 end 上述语句首先判断 if 语句后的表达式,当为真时,执行 statementsl;否则跳到 elseif 语句,判断其后的表达 式,若为真,执行 statements2:若为假,执行 statements3。 如下例: if n<0 disp('Input must be positive'); elseif rem(n,2) == 0X=n/2; else X=(n+1)/2; end 3.9.2 switch-case 语句 switch 语句将根据关键字 "switch"后的表达式的值来执行确定的语句。 switch-case 语句的一般格式为 switch switch_expr case ease_expr, statement, ...., statement case {case_exprl, case_expr2, case_expr3,...) statement, ...., Statement otherwise, statements end switch-case 语句同 if-else 语句类似,都是属于决策控制语句。switch_expr 可以是任何合法的 MATLAB 表达 式,当 switch_expr 与 case 语句后的任何一个分支相匹配,则相应执行该 case 语句后的程序代码;如果与所有 的分支都不匹配,则执行 OTHERWISE 后的程序代码。 如下例: switch input_num case -1 disp('negative one'); case 0 disp('zero');

casel
disp('positive one');

otherwise

disp('other value');

end

3.9.3 for 循环

for 循环语句的一般格式为:

for variable=expr,

statements

end

for 循环可以执行指定次数的循环,该语句首先将一个表达式赋予一个变量,表达式的通常格式为 X=Y,然 后按照表达式的列来执行 for 和 end 之间的语句。

例如,

- for m=1:6
- $S(m)=m^{m+l}$

end

这一小段程序依次将表达式 1:6 的列赋予变量 m,并循环执行语句 S(m)=m*m+l,该程序的执行结果如下 S =

```
2
                      9
          5
                7
                           11
S =
    2
          5
                7
                      9
                           11
S =
                      9
     2
          5
               10
                           11
S =
     2
          5
               10
                     17
                           11
S =
     2
          5
               10
                     17
                           26
S =
     2
          5
               10
                     17
                           26
                                 37
for 循环也可以用于嵌套使用,例如:
    for i=1:m
```

```
for j=l:n A(i,j)=1/(i+j-1) \text{ ;} \\ end \\ end \\ \end{array}
```

```
3.9.4 while 循环
```

while 循环语句的一般格式为

while expression

statements

end

与 for 循环不同的是, while 循环不能指定循环的次数, 当表达式里的所有元素为真时, 就执行 while 和 end 语句之间的程序代码。

例如

n=1;

```
while prod(1:n)<le50
```

n=n+1

end

回车显示结果为

n =

42

# 3.10 小结

本章主要介绍了 MATLAB 使用中的语言基础,包括了 MATLAB 中的矩阵、数组、字符串、变量、数值和数据类型等基本概念,还介绍了 MATLAB 的基本运算符、M 文件以及 M 文件编写中的流程控制语句。通过本章的学习,可以熟悉 MATLAB 的语法和使用基础。

# 第四章 数值计算

数值计算是 MATLAB 的最基本、最重要的功能, MATLAB 能够成为世界上最优秀的教学软件之一, 和它 出色的数值计算能力是分不开的。本章主要讲述的内容包括:

- ➢ MATLAB 的基本数学运算
- ➢ MATLAB 的矩阵运算
- ▶ 多项式与多维数组
- ▶ 稀疏矩阵
- ▶ 结构数组与细胞数组
- ▶ 数据的分析与统计

### 4.1 简单运算

我们先从 MATLAB 的数学的简单运算开始说明。就像计算器一样,数学式的计算是直接了当的。如果要计

算

1+2+3

或

 $1 \times 10 + 2 \times 20 + 3 \times 30$ 

这样的简单算式,可以在命令窗口中提示符号>>之后直接输入要计算的算式,MATLAB 将计算的结果以 "an="显示。如果算式是

x = 1 + 2 + 3

MATLAB 将计算的结果以"x="显示。例如在命令窗口中输入

1+2+3

回车显示结果

ans =

6

```
如果输入的是
```

x=1+2+3

回车显示结果

x =

6

如果在上述的例子结尾加上";",则计算结果不会显示在命令窗口上,要得知计算结果只须键入该变量名即 可。

4.1.1 MATLAB 变量的代数运算

如果给定两个变量 A 和 B,则我们可以用 A+B,A-B,A*B 可以立即得出其加、减和乘运算的结果。若这 两个变量是两个矩阵,而且这两个矩阵数学上不可以这样运算,则将得出错误信息,并终止正在运行的程序。 在 MATLAB 下,如果 A 和 B 中有一个是标量,则可以无条件地进行这样的运算。MATLAB 不介意这些变量是 纯实数还是含有虚部的复数。

以下的例子,显示了 MATLAB 对变量使用的灵活性

apple=5

```
apples =
5
orange=10
orange =
10
total_cost=apple*2+orange*4
total_cost =
50
average_cost=total_cost/(apple+orange)
average_cost =
```

3.33334

用户可以将上述指令打在同一行,以","或是";"分开;如果命令窗口上一个指令过长可以在结尾加上 "…"(代表此行指令与下一行连续)。

按照运算符的优先级可把算术运算中运算符分为五级,在每一级里的运算符有同样的优先级,在运算时自 左向右结合。各级包含的运算符为:

(1)优先级最高的是转置符(.')、幂符(.^)、复共轭转置(')、矩阵幂符(^);

(2)标星加(+)、标量减(-);

(3)向量乘法(*)、向量右除(./)、向量左除(.)、矩阵乘法(*)、矩阵右除(/)、矩阵左除(\);

- (4)加法(+)、减法(-);
- (5) 冒号运算符(:)。

MATLAB 默认的运算优先等级可用括号强制改变。例如下面两个算式的计算先后顺序不一样。

c=a./b.^2

c=(a./b).^2

亦即 A 矩阵开 10 次方得出的主根。

数组的点运算是相当重要的。所谓点运算即两个数组相应元素的运算,如 A.*B 得出的是 A 和 B 对应元素的积,故一般情况下 A*B 不等于 A.*B。矩阵的点乘又称为其 Hadamard 积。点运算的概念又可以容易地用到点乘方上,例如

A.^2, A.^A

等都是可以接受的运算式子。

用户可以将上述指令打在同一行,以","或是";"分开;如果命令窗口上一个指令过长可以在结尾加上 "…"(代表此行指令与下一行连续)。

4.1.2 逻辑运算

MATLAB 并没有单独定义逻辑变量。在 MATLAB 中,数值只有0和"非0"的区分。非0往往被认为是逻辑真,或逻辑1。MATLAB 除了支持单独两个数值的逻辑运算外,还支持矩阵的逻辑运算,如 A&B,A|B,和 ~A 分别表示逻辑与、或、非的运算。

MATLAB 的逻辑运算符如表 4-2 所示。

表 4-2

```
逻辑运算符
```

操作符	说旧
<u>ا¥الالا</u>	
<u> </u>	
	逻辑或 逻辑或
~	逻辑非正常的问题,这个问题,这个问题,这个问题,我们就能能能能能能能。

75

例如,下面A和B矩阵与运算 A=[0234;1350]; B=[1053;1505];

A&B 回车将得出如下结果 ans =

#### 4.1.3 关系表达式与表达式函数

MATLAB 的大于、小于和等于等关系分别由>、<和=表示。MATLAB 的关系运算符如表 4-3 所示。MATLAB 的判定方法不完全等同于 C、Frotran 等只能处理单个标量的语言。MATLAB 关系表达式返回的是整个矩阵。

表 4-3

关系运算符

操作符	说明
==	等于
~=	不等于
 >	大于
<	小于
 >=	大于等于
<=	小于等于

例如,比较两个矩阵 A 和 B 是否相等,则可以给出如下命令,并得出相应的结果

A=[0 2 3 4;1 3 5 0];

B = [1 0 5 3; 1 5 0 5];

A == B

回车显示结果

ans =

0 0 0 0

 $1 \quad 0 \quad 0 \quad 0$ 

确实使得 A 和 B 对应元素相等的位将返回 1, 否则返回 0。

MATLAB 还可以用 >= 和 <= 这样的符号来比较矩阵对应元素的大小。

另外,MATLAB 还提供了 all()和 any()两个函数来对矩阵参数作逻辑判定。all()函数在矩阵中全部元素非 0 时返回 1,而 any()函数在矩阵有非零元素返回 1。find()函数将返回逻辑关系全部满足时的矩阵下标值,这个函数在编程中是相当常用。用户还可以使用 isnan()类函数来判定矩阵中是否含有 NaN 型数据。如果有则返回这样参数的下标。此类函数还有 isfinite(), isclass(), ishandle()等。

4.1.4 其他运算

MATLAB 还支持其他运算,如取整、求余数等,这些功能可以使用 round(), fix(), rem() 等函数来实现。

圭	1 1	
18	4-4	

取整命令和相关命令

函数	说明
round()	求最接近自变量的整数。
fix()	求0 方向最接近自变量的整数。
floor()	求小于或等于的最接近的整数
ceil()	求大于或等于的最接近的整数
rem()	求整除的余数
gcd()	求整数的最大公因子
lcm()	求正整数的最小公倍数 , 也能用于决定最小公因子
rat()	求连续的分数表达式

## 4.2 矩阵运算

#### 4.2.1 矩阵结构操作

在前面已经讨论了用户可以使用显示输入、通过外部数据加载、使用 M 文件以及使用 MATLAB 函数生成 等方法产生矩阵。

用户可以使用 eye()建立二维单位矩阵,使用 zeros()建立空矩阵,使用 ones()建立矩阵元全为为 1 的矩阵, 使用 rand()建立矩阵元是 0~1 之间均匀分布的随机数的矩阵,使用 randn()建立矩阵元服从零均值、单位方差正 态分布的随机数的矩阵,

1. MATLAB 中随机数的产生

MATLAB 使用的新的随机数发生器,可以设置几个随机数的种子。它能够产生在闭区间[2⁻⁵³,1-2⁻⁵³]上所 有的浮点数。理论上它能够产生 2¹⁴⁹² 10⁴⁴⁹ 个不重复的数。

表 4-5

MATLAB 的随机数种子

随机数种子	说明
rand('state')	返回一个有 35 个元素的向量,包含随机状态发生器的当前状态
rand('state',s)	设置随机种子发生器的状态为 s
rand('state',0)	设置随机种子发生器为它的原始状态
rand('state',j)	设置随机种子发生器为它的第 j 种状态 , j 为整数
rand('state',sum(100*clock))	使得随机种子发生器在每个不同时刻都设置为一种不同的状态
randn('state')	返回一个有两个元素的向量 , 其中包含正态随机种子发生器

为了避免总是从相同的随机种子开始而得到相同的随机数序列,可以使用 MATLAB 中的 clock 函数。例如 在命令窗口输入

rand('state',sum(100*clock));R=rand('state');R(1:5) 回车显示结果

ans =

0.8700 0.5435 0.1949

0.3009

0.0243

#### 2. 利用已有矩阵建立新矩阵

(1)在 MATLAB 中还有利用已存在的矩阵建立新矩阵的函数 diag()。假设矩阵 A 是  $m \times n$  的矩阵, x 是一 个有 n 个元素的向量,例如

A= [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]; x=[-5 -10 -15]; 在命令窗口中输入 diag_element=diag(A), Diag_elemet=diag(A,2) 回车显示结果 diag_element =



表 4-6

diag()命令集

命令	说明
diag(A)	生成一个由矩阵 A 主对角线元素组成的列向量
diag(x)	生成主对角线元素值取自向量 x,其余元素为 0 的方阵
diag(A,k)	生成一个由矩阵 A 第 k 条对角线的元素组成的列向量 , k=0 为主对角线
diag(x,k)	生成第 k 条对角线元素值取自向量 x , 其余元素为 0 的矩阵

(2) 用户在 MATLAB 中可以使用 triu()和 tril()建立三角矩阵。

表 4-7

triu()和 tril()命令集

命令	说明
triu(A)	生成一个主对角线及以上元素取自 A 中相应元素, 其余元素都零,和 A 大小
	相同的上三角矩阵,
triu (A,k)	生成一个第 k 条对角线及以上元素取自 A 中相应元素,其余元素为零,和 A
	大小相同的上三角矩阵
tril(A)	生成一个主对角线及以下元素取自 A 中相应元素,其余元素为零,和 A 小相
	同的下三角矩阵
tril(A,k)	生成一个第k条对角线及以下元素取自A中相应元素,其余元素为零,和A大
	小相同的下三角矩阵。

对于每一个方阵 A 都有下列关系:

A = triu(A, 1) + tril(A, -1) + diag(diag(A))例如上例中的矩阵 A, 在命令窗口输入 triu(A,1)+tril(A,-1)+diag(diag(A)) 回车显示结果 ans =1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

3.矩阵结构变换

在 MATLAB 中可以通过增加元素、行和列将一个矩阵或者向量进行扩展。由于 MATLAB 可以自动地改变 矩阵的大小,所以使用已存在的矩阵的一部分来创建一个新矩阵是很容易的,这在许多应用中都很有用。从已 存在的矩阵中建立一个矩阵就和定义一个新矩阵一样。元素用空格或逗号分隔,行用分号或回车分隔;在大矩 阵中定义子矩阵是其相反过程。

(1)扩展向量

用户可以使用以下几种方式扩展已知向量 x。假设用户想通过已有定义过的向量

x=(9 10)

获得新的向量

newx=(9 10 0 5)

可以在命令窗口中输入

newx=x; newx(3)=0; newx=(4)=15

```
依次指定向量扩展的每一个元素,或者在命令窗口中输入
  newx = [x \ 0 \ 15]
  或者在命令窗口中输入
  temp=[0 15]; newx=[x temp]
  使用"[]"对向量的每"列"进行操作。以上三种方式,回车都显示想要的结果向量
  newx=
  91005
  (2)扩展矩阵
  用户可以使用"П"对已定义的矩阵进行扩展,即可以扩展矩阵的行和列,也可以是用两个矩阵扩展生成一个
新的矩阵。
  如对已知矩阵
  A=[
  1
    2
  3 4];
  用户以下两种方法可以对矩阵扩展一个新行,在命令窗口输入
  newA=[A;[13 14]]
  或者在命令窗口输入
  z=[13 14]; newA=[A;z]
  回车都将显示结果
  newA=
        2
     1
     3
        4
     13
        14
  有时还可以对矩阵添加多个新行,可以在命令窗口输入
  newA=[A;[13 14];[5 6];[7 8]]
  回车将显示结果
  newA=
     1
        2
     3 4
       14
     1
  5
     6
  7
     8
  如果要对矩阵扩展一个新列,在命令窗口输入
  y=[11;12]; newA=[A y]
  或者在命令窗口输入
  newA=[A [11; 12]]
  回车都将显示结果
  newA =
          2
      1
            11
      3
         4
             12
  同样也可以对矩阵扩展多列,在命令窗口输入
  newA=[A [5;6] [7;8]]
  回车显示结果如下
  newA =
      1
          2 5
                  7
      3
         4 6
                 8
```

使用两个已知矩阵扩展生成一个新矩阵的操作是和扩展矩阵的行列操作是相似的。对于已知矩阵 A=[ 1 2 3 4]; B=[ 5 6 7 8]: 在命令窗口输入命令 ABvert=[A;B], ABhoriz=[A B] 回车就可以得到 ABvert =1 2 3 4 5 6 7 8 ABhoriz =

对于 ABvert 来说, 它的列数一定等于矩阵 A 和 B 的列数; 而对于 ABhoriz 来说, 它的行数一定等于矩阵 A 和 B 的行数。

(3) 其他矩阵变构命令

在 MATALB 中, 用户还可以使用一些命令来变换矩阵的结构, 表 4-8 列出了部分函数。

表 4-8

矩阵结构变换函数

命令	说明
fliplr(A)	通过二维矩阵 A 的行元素按照交换位置生成一个新矩阵
flipud(A)	通过二维矩阵 A 的列元素按照交换位置生成一个新矩阵
flipdim(A,dim)	生成一个在 dim 维矩阵 A 内的元素交换位置的多维矩阵
rot90(A)	生成一个由矩阵 A 逆时针旋转 90°而得的新阵
rot90(A,k)	生成一个由矩阵 A 逆时针旋转 k ×90°而得到的新阵

#### 4.2.2 向量和子矩阵的生成

1. 冒号在 MATLAB 中的作用

在 MATLAB 中可以使用冒号":"来代表一系列数值。有时也使用它来定义一个子矩阵。

(1)使用冒号":"定义向量

表中先给出用冒号":"来定义向量的方法。

表 4-9

数字序列

命令	说明
I:k	创建从 <i>i</i> 开始、步长为 1 、到 <i>k</i> 结束的数字序列 , 即 i,i+1, i+2, , k。如果
	i>k , MATLAB 则返回一个空矩阵。数字 i 和 k 不必是整数 , 该序列的最后一个数
	是小于或等于 k
I:j:k	创建从 i 开始、步长为 1、到 k 结束的数字序列 ,即 i, i+j, i+ 2j,, k。如果 j=0 ,
	则返回一个空矩阵。数字 i、j 和 k 不必是整数 , 该序列的最后一个数是小于或等于
	k

如果输入在命令窗口中输入

v=2:7

或者

v=2:7.7

回车 MATLAB 将返回相同结果  $\mathbf{v} =$ 2 5 6 3 4 7 MATLAB 中可以使用负步长,定义向量,如在命令窗口中输入 v=6: - 1:1 回车显示结果为  $\mathbf{v} =$ 5 4 3 2 1 6 MATLAB 中也可以使用非整数的步长,定义向量,如在命令窗口中输入 r=1.2:-0.8: - 3.2 回车显示结果为 r = 1.2000 0.4000 -0.4000 -1.2000 -2.0000 -2.8000 注意最后一个数值为-2.8。 (2)使用冒号":"定义矩阵 在 MATLAB 中可以使用冒号定义矩阵,这相当于先使用冒号定义了一些向量,再使用向量生成矩阵。例如 在命令窗口中输入 M=[2:4 0.1:1:2.1; 1:6] 回车显示结果为 M =2.0000 3.0000 4.0000 0.1000 1.1000 2.1000 1.0000 2.0000 3.0000 4.0000 5.0000 6.0000 (3)使用冒号":"生成函数表 冒号能够用来生成函数表,比如 sine,在命令窗口中输入 a=0.0; b=2*pi;n=11;x=(a:(b-a)/(n-1):b)'; y=sin(x);Ftable=[x y] 回车显示结果为 Ftable = 0 0 0.6283 0.5878 1.2566 0.9511 1.8850 0.9511 2.5133 0.5878 3.1416 0.0000 3.7699 -0.5878 4.3982 -0.9511 5.0265 -0.9511 5.6549 -0.5878 还有其他的一些预定义函数也可以用来创建线性序列和逻辑序列,用户可以查看帮助户的更多信息。 2. 子矩阵 如果从矩阵中抽取行和/或列组成新的矩阵,那么这个新的就称为原矩阵的子矩阵。矩阵中的行和列也可以

称为矩阵的子矩阵。所以一个矩阵可以有许多子矩阵。这可以推广到多维数组中去。在表 4-10 中列出了对二维

#### 数组操作的命令。

表 4-10

定义子矩阵

命令	说明
A(i,j,,k)	返回多维数组 A 中下标为(i,j,,k)的元素值
A(:,j)	返回二维矩阵 A 中第 j 列列向量
A (i,:)	返回二维矩阵 A 中第 i 行行向量
A(:,j:k)	返回由二维矩阵 A 中的第 j 列 , 第 j+1 列 , 直到第 k 列列向量组
	成的子阵
A (i:k,:)	返回由二维矩阵 A 中的第 i 行 , 第 i+1 行 , 直到第 k 行行向量组
	成的子阵
A (i:k,j:l)	返回由二维矩阵 A 中的第 i 行到第 k 行行向量和第 j 列到第 l 列
	列向量组成的子阵
A(j:k)	返回一个行向量 , 其中的元素为 A(:)中的从第 j 个元素到第 k 个
	元素
A([j1 j2] )	返回一个行向量,其中的元素为 A(:)中的第 j1、j2…元素
A(:,[j1 j2])	返回矩阵 A 的第 j1 列、第 j2 列等的列向量
A([i1 i2]:,)	返回矩阵 A 的第 i1 行、第 i2 行等的行向量
A([i1 i2] [j1	返回矩阵第 i1 行、第 i2 行等和第 j1 列、第 j2 列等的元素
j2])	
A(:,:,,:)	返回矩阵 A 本身
A (:)	将矩阵 A 中的每列合并成一个长的列向量

例如上面的矩阵 Ftable,可以定义子矩阵

Submatrix = Ftable(:, 2:4)

回车显示结果

Submatirx =

0.6283 0.5878

1.2566 0.9511

1.8850 0.9511

也就是新矩阵的每一列是从矩阵 Ftable 的第 2 到第 4 行。

要取出其中一维的最后一个元素值,可以用值 end 来取。如果 A 是一个 4 ×3 ×2 的数组, A(end,2,1)就可以 得到元素  $a_{421}$  的值, A(end,end)得到元素  $a_{432}$  的值。

3. 冒号表达式能够与关系运算符一起使用。

利用冒号可以写出复合表达式,例如通过下面简短的命令可以挑选出矩阵 Ftable 第 2 列中大于 0 的元素所在的行向量,在命令窗口中输入

Selected=Ftable(Ftable(:,2)>0 ,:)

回车显示所得的结果

Selected =

0.6283	0.5878
1.2566	0.9511
1.8850	0.9511
2.5133	0.5878
3.1416	0.0000

#### 4.2.3 空矩阵

1.空矩阵

在 MATLAB 中对空矩阵的定义是

A =[]

用户可通过 whos 命令察看在内存中的驻留变量的详细信息。一些函数对空矩阵操作返回一个常量,在编写 程序时这常常是有用的。表中列出了这些常用命令,表 4-11 中的矩阵 E 是一个空矩阵,而 A 是一个有空维的矩 阵,可以使用命令 squeese()清除矩阵中的空维。

表 4-11

空矩阵函数

命令	说明
squeeze(A)	返回没有空维的矩阵
sum(E)	返回 0
prod(E)	返回1
max(E)	返回E
min(E)	返回 E

2. 有空维的矩阵

在使用 MATLAB 的过程中,有时需要创建一个多维的矩阵,但是这个矩阵中可能有几维是空的,这时可以 使用用命令 zeros 来定义空的行向量和列向量。比如要创建一个大小为 1-2-0-0-2 的矩阵,可以使用命令

zeros( 1,2,0,0,2)

回车显示结果为

ans =

Empty array: 1-by-2-by-0-by-2

#### 4.2.4 MATLAB 中的特殊矩阵

除了以前提到过的零矩阵、单位矩阵和1矩阵、随机矩阵等特殊矩阵,MATLAB中还有一些命令用于生成 试验矩阵。

1.希尔伯特矩阵

希尔伯特(Hilbert)矩阵,也称 H 阵,其元素为  $h_{ij} = 1 / (i+j-1)$ 。由于它是一个条件数差的矩阵,所以将它用 来作为试验矩阵。

表 4-12

Hilbert 矩阵

命令	说明
hilb(n)	生成一个 n×n 的希尔伯特矩阵
invhilb(n)	生成一个 n×n 的希尔伯特矩阵的逆矩阵,其元素都为整数

例如果在命令窗口输入

H=hilb(3), Hinv=invhilb(3)

回车 MATLAB 就会相应地显示结果

H =

1.0000	0.5000	0.3333
0.5000	0.3333	0.2500
0.3333	0.2500	0.2000

Hinv =

9 -36 30

-36 192 -180

30 -180 180

可以看出希尔伯特矩阵和它的逆矩阵都是对称矩阵。

2.托普利兹矩阵

托普利兹(Toeplitz)矩阵是由两个向量来定义的,一个行向量和一个列向量。对称的托普利兹矩阵由单一向量来定义。

Toeplitz 矩阵

命令	说明
toeplitz(k,r)	生成一个非对称的托普利兹矩阵,将 k 作为第 1 列,将 r 作为第 1 行。其余
	的元素与左上角相邻元素相等
toeplitz(c)	用向量 c 生成一个对称的托普利兹矩阵

例如	1,已	知		
x=[1	234	4]; y=[	9876]	;
那么	在命	令窗口	中输入	
A=to	peplitz	z(x,y),E	B=toeplitz	z(x)
回车	显示	结果		
A =				
	1	8	7	6
	2	1	8	7
	3	2	1	8
	4	3	2	1
B =				
	1	2	3	4
	2	1	2	3
	3	2	1	2

#### 3. 其它特殊矩阵

下面的表 4-14 中给出了 MATLAB 中其他特殊矩阵。

表 4-14

其他特殊矩阵

命令	说明	
compan(p)	生成一个p多项式的友矩阵,也就是它的特征多项式是p 。p 是	
	一个包含多项式系数的向量	
gallery(n)	生成一个在数字分析中有名的 n×n 试验矩阵	
gallery	从 family 族中返回一个矩阵	
family		
hadamard(k)	返回一个阶数为 n=2k 的 Hadamard 矩阵	
hankel(x)	返回一个由向量 x 定义的 Hankel 方阵	
rosser	给出 Rosser 矩阵	
vander(x)	回一个倒数第2列为向量 x的 Vandermonde 矩阵	
magic(n)	给出一个 n×n 的魔方矩阵	
pascal(n)	返回一个 n×n 的 Pascal 矩阵,它是对称、正定的矩阵,它的元素由 pascal 三角	
	组成,它的逆矩阵的所有元素是整数	
hankel(x,y)	返回一个 m×n 的 Hankel 矩阵,它的第 1 列为向量 x ,最后一行为向量 y	

例如为了能够得到一个 Householder 矩阵,可以运行命令 help private/house,这将给出指定参数的信息。使用下面的语句可用来创建一个 Householder 矩阵 H

x=[2;5;3]; [V,Beta]=gallery('house',x); H=eye(3,3)-Beta*V*V' 回车显示结果 H= ______-0.3244 _-0.8111 _-0.4867

-0.8111 0.5033 -0.2980 -0.4867 -0.2980 0.8212

#### 4.2.5 矩阵的逆与伪逆

如果矩阵 A 为方阵且非奇异,方程 AX = I 和 XA = I 的解称为矩阵 A 的逆,用 A⁻¹,表示。用户可以使用函数 inv()求矩阵的逆,使用函数 det()求矩阵的行列式。

例如在命令窗口中输入命令

d = det(A), x=inv(A)

可以分别求的矩阵 A 的行列式 d 和逆矩阵 x。

如果忽略舍入误差,命令 x = inv(A)*B 和命令 x = A\B 应该得到相同的结果。但计算时,应尽量使用后者,因为斜杠和反斜杠运算符的数值计算更准确,占用内存更小,算得更快。

当矩阵 A 为长方阵时,方程 AX = I 和 XA = I 至少有一个无解。这时矩阵的伪逆能在某种程度上代表矩阵的 逆。下面为求矩阵伪逆的例子

C=[

- 9 4;2 8;
- 6 7]:
- · , ,
- X=pinv(C), Q=X*C, P=C*X 回车显示结果

 $\begin{array}{cccccc} X = & & & & \\ & 0.1159 & 0.0729 & 0.0171 \\ & 0.0534 & 0.1152 & 0.0418 \\ Q = & & & \\ & 1.00 & 0.00 \\ & 0.00 & 1.00 \\ P = & & \\ & & \\ & 0.8293 & -0.1958 & 0.3213 \\ & -0.1958 & 0.7754 & 0.3685 \\ & 0.3213 & 0.3685 & 0.3952 \\ \end{array}$ 

但 P*C=C, X*P=X 其性质类似单位矩阵。

4.2.6 矩阵的四则运算

MATLAB 中的大多数运算可以直接对矩阵应用。除了在前面章节中讨论的算术运算"+"、"-"、"*"、"^"、 "/"、"\"、逻辑运算符外,还有用于转置和共轭的运算符、有理数运算符等。

MATLAB 的用户应该知道矩阵中的元素总数极限是 16348。

此外,矩阵运算还有算术函数和逻辑函数,需要注意的是,有些函数仅能在二维矩阵中使用。

1.加减法

如果矩阵 A 和 B 具有相同的维数,那么就可以定义两个矩阵的和

A+B

和两个矩阵的差

A-B

矩阵加减 A±B,实际上是每个对应矩阵元素进行加减运算,即元素  $a_{ij,n}$  ±  $b_{ij,n}$ 。在 MATLAB中,一个 m

 $\times n$  矩阵 A 和一个标量,即一个  $1 \times 1$  矩阵 s 之间也能进行加减运算。矩阵运算

A+s

得到与 A 相同维数的矩阵,相应的元素为 $a_i + s$ 。

#### 如果已有定义矩阵

A=

- [1 2
- 3 4];
- B=
- [5 6
- 7 8];

在 MATLAB 命令窗口中输入命令

```
Add=A+B, Sub=A-B, Add100=A+100
回车得到结果
Add =
        6
              8
        10
              12
Sub =
        -4
             -4
             -4
        -4
Add100 =
              102
        101
        103
              104
```

2. 乘法

如果矩阵 A 的列数等于矩阵 B 的行数,那么可以进行矩阵乘法运算,即

C=A*B

如果矩阵 A 的列数和矩阵 B 的行数不相等,除非这两个矩阵中有一个是 1 ×1 的矩阵,即是一个标量,否 则 ATLAB 将返回一个错误信息,不接受的这种情况的运算。在 MATLAB 中,乘法的运算符是"*"。乘法运算得 到的矩阵 C 有与矩阵 A 相同的行数和与矩阵 B 相同的列数,矩阵元素  $c_{ij}$ 是矩阵 A 的第 i 行和矩阵 B 的第 j 列的

点积。

```
对于方阵,也可定义矩阵乘积积 B*A,但其结果通常与 A*B 不同。
例如上例中的矩阵 A 和矩阵 B,在命令窗口中输入命令
MultAB=A*B, MultBA=B*A
回车后在屏幕上显示如下的结果
MultAB =
19
    22
43
    50
MultBA =
  23
       34
   31
       46
又例如取向量 x 和 y 为
x=[1 2 3];
y=[1;10;100];
在命令窗口输入命令
s=x*y, M=y*x
回车显示结果为
s =
  321
M =
   1
        2
            3
   10
       20
            30
  100
      200
           300
MATLAB 还包含了其他一些乘积形式。
(1) dot()命令
```

如果 x 和 y 是两个向量,命令 dot(x,y)得到具有相同元素数量的两个向量的点积,也称为标量积或内积。如 果点积为零,表明两个向量是正交的。假设 A 和 B 具有相同的维数的矩阵,则 dot(A,B)定义了这两个矩阵的点 积,其结果是一个行向量,其元素是第1列、第2列等矩阵 A 和矩阵 B 对应列的点积。

```
(2) cross()命令
   对于各具三个元素的两个向量 x 和 y, 命令 cross(x,y)给出了向量积或叉积, 即
   x \times y = (x_2y_3 - x_3y_2 \quad x_3y_1 - x_1y_3 \quad x_1y_2 - x_2y_1)
   结果仍是一个向量,而且与向量 x、向量 y 都是正交的。
   cross()命令也可以应用于 3×n 矩阵,其结果仍是一个 3×n 矩阵,这里的第 i 列是矩阵 A 和 B 中的第 i 列的叉
积。
   例如,可以假设向量
   x = [1 0 0];
   y = [0 \ 1 \ 0];
   在命令窗口中输入命令
   Crossprod=cross(x,y)
   回车将得到结果
   Crossprod =
       0
           0
                1
   对于向量 x 和 y, 它是正交向量, 在命令窗口中输入
   scalar1=dot(x,Crossprod), scalar2=dot(y,Crossprod)
   回车显示结果
   scalar1 =
       0
   scalar2 =
       0
   (3) kron()命令
   Kronecker 张量积可以用于创建大的矩阵,它由命令 kron(A, B)得到。如果 A 是一个 m×n 矩阵, B 是一个 k×r
矩阵,那么这个命令就返回一个 m*k×r*n 的矩阵。
   例如使用前例中的矩阵 A 和矩阵 B,
   A =
            2
       [1
       3
            4];
   B =
       [5
            6
       7
            8];
   在命令窗口中输入
   K=kron(A, B)
   回车显示的结果为
   K =
       5
                10
                     12
            6
       7
            8
                14
                     16
      15
           18
                20
                     24
      21
           24
                28
                     32
   3. 除法
   在 MATLAB 中,有两个矩阵除法的符号,左除"\"和右除"?"。矩阵的除法实际上就是线性方程的求解,如
Ax=B 这一线性方程的解即为
   x=inv(A)*B,
   或更简单地
```

 $x = A \setminus B$ 

这又称为矩阵的左除,而

x=B/A

称为矩阵的右除。如果 A 是一个非奇异方阵,那么 A\B 和 B/A 对应 A 的逆与 B 的左乘和右乘,即分别等 价于命令 inv(A)*B 和 B*inv(A)。例如,有矩阵

A=[			
	1	2	
	3	4];	
B=[			
	5	6	
	7	8];	
在命	令匿	司输入	<
Rigl	nt=B/	A, Left	=A\B
回车	三显示	结果	
Rigl	nt=		
	-1	2	
	-2	3	
Left	=		
	-3.0	000	-4.0000
	4.00	000	5.0000

4.2.7 矩阵的幂和指数函数

对于二维方阵, A 的 p 次乘方可以用命令 A ^p 实现。如果矩阵 A 为方阵, p 是一个正整数, 那么这个幂可以由许多矩阵乘法运算定义,即 A^p 表示 A 自乘 p 次; 如果 p=0, 将得到一个与矩阵 A 维数相同的一个矩阵; 如果矩阵 A 为方阵且非奇异, p<0时, 如果 A⁻¹存在,则可定义 A ^-(p), 它是与 inv(A)^ p 相同的, A^(-p) 表示矩阵 A⁻¹自乘 p 次。

在 MATLAB 中,幂运算是和 exp(A)、sqrt(A)那样的表达式一样的数组运算,即它们是对 A 中每个矩阵元素作幂运算。例如,对于矩阵

在 MATLAB 中,表达式 A^(2/3)是合法的,其计算结果和特征值的分布有关。函数 sqrtm(A)计算出比 A^(1/2) 更精确的结果。线性常微分方程组可以表示为 dx/dk=Ax,其中 x 为标量 t 的函数组成的向量。方程组的解可表 示为:x(t) =  $e^{At}x(0)$ ,其中, $e^{At}$ 可用命令 expm(A)计算。

4.2.8 转置和共轭

MATLAB 中一个重要的运算是矩阵的转置和共轭转置,它在 M AT L A B 中用撇""表示。如果 A 是一个实数矩阵,那么当它被转置时,矩阵的第1行将变成新矩阵第1列,第2行变成新矩阵的第2列,依此类推,一个 *m×n* 矩阵经过转置将变为一个 *n×m* 矩阵。如果矩阵是方阵,那么这个矩阵将是以主对角线为轴发生旋转。

如果矩阵 A 的元素 a_{ii}是复数,那么转置运算还使所有元素发生了共轭变换,即矩阵 A'的第(i,j)项上的元素

含有是 a_{ii}的共轭。如果希望复数矩阵仅发生转置,那么可以使用点远算,在撇号"'"之前输入一点"."。A.'表 示转置,其结果与 conj(A')相同。如果 A 是实数矩阵,那么 A'与 A.'的作用是相同的。

			- J. /	
对于	-已決	口矩阵	≰Α和矩阵 Β	
A=[				
1	3	5;		
1	2	4;		
0	5	1];		
B=[	22; 1	7; 13	];	
在命	令逐	副口中	「输入	
T=A	A', R=	=B'		
回车显示结果				
T=				
	1	1	0	
	3	2	5	
	5	4	1	
R=				
	22	17	13	

4.2.9 矩阵函数

在 M AT L A B 中预定义的数学标准函数是基于矩阵对元素的运算的。如果 f 是这样一个预定义的数学标准 函数,A是一个矩阵,那么f(A);j=f(aij),函数对每一个矩阵元素进行操作。如果元素是复数,那么根据这个函数 产生的矩阵也可以是复数,矩阵的维数没有改变。

例如,对于矩阵

a=[ 0 -3; 5 1; 4 -6]; 在命令窗口输入 abs(a) 回车显示结果 ans= 0 3

5 1

6 4

MATLAB也能处理方阵的指数函数,例如A^{1/2} (A的平方根)或e^A。表 4-15 列出了常用的矩阵函数

表 4	-15
-----	-----

矩阵函数

命令	说明
expm(A)	使用 Pade 近似法计算 e ^A
expm1(A)	使用一个 M 文件和与内部函数相同的算法计算 e ^A
expm2(A)	使用泰勒级数计算 e ^A
expm3(A)	使用特征值和特征向量计算 e ^A
funm(A,fcn)	计算由字符串 fcn 指定的 A 的矩阵函数
[F,E]=funm(A,fcn)	计算如上矩阵函数,但返回结果矩阵 F 和剩余近似值矩阵 E
polyvalm(p,A)	估算矩阵 A 的一个多项式
logm(A)	计算 A 的对数
sqrtm(A)	计算 A ^{1/2}

#### 4.2.10 特征值与特征向量

使用 MATLAB 中的命令计算矩阵的特征值和特征向量很方便的,可以得到不同的子结果和分解,这在处理 线性代数问题是很有用。值得注意的是,下面要讨论的的命令只能对二维矩阵进行操作。

如果 A 是一个 m×n 的矩阵,求 A 的特征值问题就是求方程组

Ax= x

的解,其中 是一个标量,x 是一个长度为 n 的列向量。标量 是矩阵 A 的特征值,向量 x 是相对应的特征向量。即使矩阵 A 是实数矩阵,它的特征值和特征向量仍有可能是复数。一个  $n \times n$  的矩阵应当有 n 个特征值,分别表示为 _1, _2, ....., _n。

在 MATLAB 中使用函数命令 eig()来确定矩阵 A 的特征值和特征向量。特征向量的归一化,就是指每个特征向量的欧几里得范数为 1。

表 4-16 列出了求矩阵特征值和特征向量的函数。

表 4-17

特征值与特征向量

命令	说明
eig(A)	求包含矩阵 A 的特征值的向量
[X,D]=eig(A)	产生一个矩阵 A 的特征值在对角线上的对角矩阵 D 和矩阵 X ,它们的列是
	相应的特征向量满足 AX=XD。为了得到有更好条件特征值的矩阵要进行
	相似变换
[X,D]= eig(A,'nobalance')	不经过平衡处理求得矩阵 A 的特征值和特征向量也就是不进行平衡相似变
	换
eigs(A)	返回一个由矩阵 A 的部分特征值组成的向量 , 和命令 eig()一样 , 但是不返
	回全部的特征值。如果不带有参量,则计算出最大的特征值。当计算所有
	特征值时,如果矩阵 A 的秩不小于 6,则计算出 6 个特征值来
eigs(f,n)	求出矩阵 A 的部分特征值。在使用一个矩阵列的线性运算符时,字符串 f
	中包含的是 M 文件的文件名 , n 指定问题的 阶次
eigs(A,B,k,sigma)	求矩阵 A 的部分特征值
condeig(A)	返回一个由矩阵 A 的特征值条件数组成的向量
balance(A)	求平衡矩阵
[T,B]=balance(A)	找到一个相似变换矩阵 T 和矩阵 B , 使得它们满足 B=T ⁻¹ AT
	B 是用命令 balance 求得的平衡矩阵

#### 例如,对于已知矩阵

```
A=[
-9 -3 -16;
13 7 16;
3 3 10];
在命令窗口输入
[Eve, Eva] = eig(A)
回车显示结果
Eve=
   -0.7071 -0.5774 -0.5774
   0.7071 0.5774 -0.5774
   0.0000 0.5774 0.5774
Eva=
   -6.0000
            0
                    0
   0
         10.000
                    0
   0
            0 4.0000
可见特征值都是非零数。
而对于矩阵
A=[
   3 -2 -0.9 2*eps;
```

	-2 4	-1 -eps	3;	
	-eps/4	eps/2	-1 0;	
	-0.5	-0.5	0.1 1];	
由于	-矩阵中有	前元素与	截断误差	<b>[相当,求特征值时将产生错误,在命令窗口中输入命令</b>
[V1	,D1]=eig(.	A), x=A*V	/1-V1*D1	, [V2,D2]=eig(A,'nobalance'), y= A*V1-V1*D1
回车	显示结果	Ę		
x=[				
	0.00	0	0.00	0.00
	0.00	0	0.00	0.00
	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	1.0066
y=				
	1.0e-015	*		
	0.8882	0	0.0295	-0.5551
	-0.8882	0.0555	0.2254	0.4996
	-0.0172	-0.0015	0.0066	0

0.1943 -0.2220 -0.2220 0.5829

从上例可以看出,有时若不选用 nobalance 选项,计算结果将完全错误。这是由于在调用 eig()时,首先要调用使各矩阵元素大致相当的平衡程序,该程序将把原矩阵中本可以忽略的元素放大。当 A 矩阵来自程序的中间 计算结果时,就有可能发生本例的情况,但一般说来,采用 nobalance 选项将有效的减小计算误差。

#### 满足条件 yA= y

的非零行向量 y 被称作左特征向量。如果使用命令 eig()对 A'作用,可以计算出矩阵 A 的左特征向量,因为:

 $A' y' = \overline{I} y'$ 

这里的撇号"'"代表矩阵的转置和共轭复数, 上的短杠表示共轭。矩阵特征值的集合称为矩阵的谱,谱 半径 **r** (A)定义为 max(abs(eig(A)))。矩阵 A 的特征值的乘积等于 det(A),和等于 trace(A),这是矩阵 A 主对角线 上元素的和。

如果 X 是一个矩阵如果 X' X = I,则特征向量是正交的,这对于对称矩阵是成立的。

如果矩阵 A 是实数矩阵,但是有复数特征值,那么这些特征值是以共轭复数的形式出现的。如果 [X,D]=eig(A),可以用命令 cdf2rdf 将矩阵 D 转换为一个实数块对角矩阵。在对角线上用一个 2×2 实数块代替共 轭复数对。

例如对于已知矩阵

0;

A=[ 0 1

-1 0 0;

0 0 31;

在命令窗口输入

```
[X, D]=eig(A), [Y, E]=cdf2rdf(X, D)
```

```
回车显示结果
```

X=

· <b>x</b> -			
	0.7071	0.7071	0
	0+0.7071i	0-0.7071i	0
	0	0	1.0000
D=			
	0+1,0000i	0	0

	0	0-1.0000i	0
	0	0	3.0000
Y=			
	0.7071	0	0
	0	0.7071	0
	0	0	1.0000
E=			
	0	1	0
	-1	0	0
	0	0	3
		_ / / /	

所的矩阵 E 正好是矩阵 A。

➢ 特征向量是特征多项式 det( Ⅰ I - A)=0 的根,其中 I 是单位矩阵。用命令 poly()来求特征多项式。 roots()求得特征值,但是用命令 eig()的特征值更准确,精度更高。

4.2.11 上海森堡形式、QR 和 QZ 分解

1. 海森堡形式

如果只求特征值和特征向量,推荐用上一节中提到的方法。然而,有时要求更详细了解计算过程,可用在 这一节和下一节中定义的命令来满足这样的要求。

如果矩阵 H 的第一子对角线下元素都是零,则它是一个上海森堡(Hessenberg)矩阵。如果矩阵是对称矩阵,则它的海森堡形式是对角三角阵。MATLAB 可以通过相似变换将矩阵变换成这种形式。

表 4-18

海森堡形式

命令	说明
hess(A)	返回矩阵 A 的上海森伯形式
[P,H]=hess(A)	返回一个酉矩阵 P 和上海森伯矩阵 H , 使 A=P H P和 PP'=I

2.QR 算法

在 MATLAB 中,除了可以用命令 eig()来求矩阵特征值,QR 算法也是计算矩阵所有特征值的一种有效的数 学方法。在用这种方法时,建议首先将矩阵转换成相似的上海森伯形式。

QR 算法是基于 QR 因式分解的一种算法,每个 m×n 的矩阵 A 都可以表示成

A=QR

其中 Q 是一个 m×m 的酉矩阵, R 是一个 m×n 的上三角矩阵。如果 A 是一个方阵, R 也还是这样的一个矩阵。当用命令 qr()时, 会返回矩阵 Q 和 R。如果 A 是上海森伯矩阵,则 Q 也是一样。对于 QR 算法,下面给出一些简短的描述:

QR 算法:

- 1) 令 A0=A, k=0;
- 2) 找到 A_k的分解: A_k=Q_kR_k;
- 3) 迭代计算下一个矩阵: A_{k+1}=Q_kR_k, 令 k=k+1;

4) 返回到 2。

这种方法也称为不移位的 QR 方法,就是在某种约定下逼近于上三角矩阵。因为所有的矩阵 A_k和 A₀=A 相 似,所以有和原始矩阵相同的特征值,即最后的上三角矩阵的对角线元素就是 A 的特征值。

如果矩阵一开始就转换成有接近一半元素是零的上海森伯形式,就可以减少可观的计算步骤。QR 方法作为 MATLAB 的一个内建函数,早运算过程中为了加快逼近速度也可进行移位。

表 4-19

QR 因式分解

命令	况明
[Q,R]=qr(A)	产生一个 m×m 的酉矩阵 Q 和一个 m×n 的上三角矩阵 R , 使得 A=QR
[Q,R,P]=qr(A)	产生一个大小为 m×m、列正交的酉矩阵 Q , 一个对角线元素递减的 m×n 的
	上三角矩阵 R 和一个置换矩阵 P , 使得 AP=QR

	命令		说明
[Q,R]=qr insert(Q,R,j,b)		r insert(Q,R,j,b)	由于在矩阵 A 的 j 列后插入一个额外的列 b 而得到新的 QR 因式分解 ,Q 和 R 是对矩阵 A 进行 QR 因式分解得到的矩阵。如果 j=n+1 , 那么 b 就插入在 短底 A 的是E— 列
	[Q,R]= qrdelete(Q,R,j)		
	[Q1,R1]	= qrupdate(Q,R,x,y	) 给出 A+xy的 QR 分解,也就是用秩为1的矩阵改变 A 的 QR 分解
	这田经中————————————————————————————————————	N田不移位的 O	D.因式公解管注进行计算的例子。对拓陈
			【四八刀 解异/公伍门 们 异时 [7] 1 , 7] 尼叶
	A = [ 0 3 16		
	13 7 16·	,	
	13 7 10, 3 3 10]		
	进行因式分解	, 足	
	A0 = hess(A):	" [00. R0]=ar(A0)	A1=R0*O0, $[O1, R1]=ar(A1)$ ; $A2=R1*O1$
	回车显示结果	₹	
	A1=		
	1.79922	26.8770	-12.6126
	2.3625	4.5085	-0.1434
	0	4.9518	1.6923
	A2=		
	17.6077	11.3432	5.0128
	-15.3516	-13.6557	-5.8721
	0	1.0748	4.0480
	在整个计算过	过程中一直保留	着上海森伯形式。
	3.QZ 算法		
	QZ 算法是用	来计算复数矩队	车的复数特征向量对和广义特征值的。在 MATLAB 中,依据下面的命令形式来
调月	目命令 qz()。		
	[C, D, Q, Z, V	$\frac{1}{2} = qz(A,B)$	
<u> </u>	得到 <b>灯用线</b> 刀	T系是/义特征	值的上二用矩阵 C、D 和厂又符征问重矩阵 V。矩阵 Q 和 Z 是受换矩阵,使得
QA	Z=C 和 QBZ=I	U。 ་ᆂ ᇬᅎᄼᄵᅇᅝᅻ	->+
	QL 力法定基	丁Q乙万解的方	)/ <b>ズ</b> 。
4.2.	12 舒尔分解	和奇异值分解	
	1. 舒尔分解		
	如果 A 是一~	个方阵,则有一	个这样的酉矩阵 U, 使得
	U ⁻¹ AU=U'AU	=T	
	其中 T 是上王	三角矩阵。这是	一个相似变换,因此矩阵 A 和 T 有相同的特征值。因为 T 是一个对角矩阵,因
此其	其对角线元素就	就是它的特征值	5
	如果 A 是实物	数矩阵日对称	那么 T 有对角形式,U 的列就是 A 的特征向量。

如果 A 是实数矩阵,但是有复数特征值,那么 T 就是一个复数矩阵。为了避免复杂的计算,用 2×2 的实数 矩阵来代表每一对共轭复数特征值。这样 T 就是一个块三角实数矩阵。MATLAB 中用命令 schur()来进行实数和 复数矩阵 A 的舒尔分解。

如果 A 是实数矩阵,那么 schur(A)返回实数的舒尔形式,但是如果 A 是复数矩阵,则给出复数形式。它们的差别在于实数形式中在对角线上用 2×2 的矩阵块来表示共轭复数特征值对,而复数形式给出的对角线元素是 复数。函数 rsf2csf 可以将实数形式转换成复数形式。 表 4-20

舒尔分解

命令	说明
schur(A)	给出矩阵 A 的舒尔分解,也就是如上所述的矩阵 T
[U,T]=schur(A)	给出矩阵 A 的舒尔分解和一个酉矩阵 U,使得 A=UTU'
[V,S]=rsf2csf(U,T)	将实数舒尔形式矩阵 U 和 T 转变成复数舒尔形式矩阵 V 和 S

#### 例如对于给定的矩阵

A1=[ 2 1; 1 2]; A2=[ 2 1; 0 2];A3= [0 1; -1 0]; 在命令窗口输入命令 Sch1=schur(A1), Sch2=schur(), ... [U, Sch3]=schru(A3) 回车显示结果为 Sch1= 0 1 0 3 Sch2= 2 1 2 0 Sch3= 0 1 -1 0 U= 1 0 0 1 因为有复数特征值,所以可以看出矩阵 Sch3 不是一个上三角矩阵。为了验证一下可以输入 [V,S]=rsf2csf(U,Sch3) 回车显示结果 V= 0+0.7071i 0.7071 -0.7071 0-0.7071i S=0+1.0000i 0 0 0-1.0000i 这里的特征值是 i 和 - i。 2. 奇异值分解 MATLAB 还可以计算奇异值分解,即 SVD 和矩阵的奇异值。这些数都是非负数,在某种特定情况下,它

们和矩阵的特征值相同。命令 svds()和命令 eigs()相似,也返回一些奇异值。

表 4-21

奇异值分解

命令	说明	
svd(A)	返回一个包含矩阵 A 奇异值的向量	
[U,S,V]=svd(A)	返回一个对角矩阵 S 和大小分别为 m×m 和 n×n 的酉矩阵 U 和 V , 矩阵 S	
	的大小和 A 相同 , 也是 m×n 的 , 而且 奇异值在矩阵的对角线上。奇异值	
	是非负数且按降序排列。这些矩阵满足 A=USV'和 U'AV=S	
[U,S,V]=svd(A,0)	得到一个"有效大小"的分解,只计算出矩阵 U 的前 n 列	
svds(A,k,0)	计算出 k 个最大的奇异值和相应矩阵 A 的向量。如果 k 没有给出 , 则缺省	
	值为 5。如果 0 作为最后一个参量值,则计算最小值;否则计算最大值	
gsvd(A)	给出广义奇异分解	

在 MATLAB 中,矩阵 A 的伪逆可以用命令 pinv(A)来求,也可用 SVD 分解来求矩阵的伪逆。 对于矩阵

A=[			
1 1;			
1 2;			
1 3];			
B=[			
1 -1	0;		
1 2	0;		
1 3	0];		
在命令	窗口输入商	令	
[Ua, Sa,	Va]=svd(A	A), [Ub, Sb, Vb]=svd(B)	
回车显	示结果		
Ua=			
0.3231	-0.8538	0.4082	
0.5475	-0.1832	-0.8165	
0.7719	0.4873	0.4082	
Sa=			
4.0791	0		
0	0.6005		
0	0		
Va=			
0.4027	-0.9153		
0.9153	0.4027		
Ub=			
-0.1641	0.9668	0.1961	
0.5653	0.2551	-0.7845	
0.8084	0.0178	0.5883	
Sb=			
3.9116	0	0	
0	1.3036	0	
0	0	0	
Vb=			
0.3092	0.9510	0	
0.9510	-0.3092	0	
0	0	1.0000	
可以看	出矩阵Aヲ	和 B 有两个非零的奇异值,	也就是它们的秩为 2.

这些矩阵的逆是没有办法求的,但是伪逆可以用 PseudoA=pinv(A) 和 PseudoB=pinv(B) 来求,回车得到结果 PseudoA= 1.3333 0.3333 -0.6667 -0.5000 0.0000 0.50000 PseudoB= 0.6923 0.2308 0.0769 -0.2692 0.0769 0.1923 0 0 0 0

### 4.3 多项式

MATLAB 提供了标准多项式运算的函数,如多项式求根、求值和维分。另外,还提供了一些用于更高级运算的函数,如曲线拟合和部分分式展开等。

4.3.1 多项式的表达和创建

在 MATLAB 里, 多项式是用行向量表示的。多项式

 $p(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n$ 

用以下系数向量表示

 $P=[a_0 \quad a_1 \quad a_2 \quad \dots a_n]$ 

除了上述的直接输入法,多项式行向量还可由命令 ploy 创建。如果 A 为矩阵,则 ploy 将创建 A 矩阵的 特征多项式;如果 A 为向量[b₁ b₂ b₃ ... b_{n-1} b_n],则创建(x-b₁)(x-b₂)...(x-b_{n-1})(x-b_n)生成的多项式的系数向量。这样 就把多项式问题转化为向量问题。

1. 系数向量的直接输入法

由于在 MATLAB 中的多项式是以向量形式储存的,因此最简单的多项式输入即为直 接的向量输入, MATLAB 将自动将向量元素按降幂顺序分配给各系数值。向量可以为行 向量,也可以是列向量。

例如,在命令窗口输入

p=[1-56-33];

回车显示结果

poly2sym(p)

ans =

x^3-5*x^2+6*x-33

🔉 其中的 poly2sym 是符号工具箱中的函数,可将多项式向量表示成为符号形式

2. 特征多项式

多项式创建的另一个途径是从矩阵求其特征多项式获得,由函数 poly 实现。例如在命令窗口输入

a=[1 2 3;2 3 4;3 4 5];

p1=poly(a), poly2sym(p1)

回车显示结果

p1 =

1.0000 -9.0000 -6.0000 -0.0000

ans =

x^3-9*x^2-6*x-7343508239050119/2535301200456458802993406410752

由特征多项式生成的多项式的首项系数一定是1

n 阶矩阵则一般产生 n 次多项式。

3. 由根创建多项式

由给定的根也可产生其对应的多项式,此功能也由函数 poly 实现。例如在命令窗口输入

root=[-5 -3+4i -3-4i];

p=poly(root), poly2sym(p)

回车显示结果

p =

1 11 55 125

ans =

x^3+11*x^2+55*x+125

🕿 若要生成实系数多项式,则根中的复数必定对应共轭

有时生成的多项式向量包含虚部小量,可用 real 函数将其滤掉

4.3.2 多项式的运算

#### 1. 多项式求根

函数 roots 用来求多项式的根。它的指令形式为:

r=roots(p)

函数 poly 由给定的根创建多项式,将返回多项式的系数。它的指令形式为:

p2=poly(r)

- 说明:
  - ▶ 在 MATLAB 种,按约定将根以列向量的形式保存。
  - 函数 roots 给定多项式,返回多项式的根;而函数 poly 给定根,返回多项式的系数。roots 和 poly 互为逆运算。

例如,对于向量

p=[2 -5 6 -1 9]

在命令窗口输入

roots(p)

#### 回车显示结果

ans =

1.6024 + 1.2709i 1.6024 - 1.2709i -0.3524 + 0.9755i -0.3524 - 0.9755i

2. 特征多项式

poly 函数也可以用来计算矩阵的特征多项式的系数。

A=[1.2 3 -0.9;5 1.75 6;9 0 1];

poly(A)

ans =

1.0000 -3.9500 -1.8500 -163.2750

这个多项式的根,是矩阵 A 的特征根或特征值,利用 roots 函数宋计算。

3. 多项式求值

polyval 函数用来计算多项式在给定变量时的值。也可以采用矩阵形式计算多项式的值。在这种情况下, p(x)=x³-2x-5 变为 p(X)=X³⁻2X-5I,其中 X 是一个方阵,I 是单位阵。 例如,求多项式p的值: p=[11155125]; b=[11;11]; polyval(p,b) ans =

192 192 192 192

polyvalm(p, b)

ans =

206 8181 206

4. 卷积和解卷

多项式的乘法和除法运算分别对应于卷积运算和解卷运算。函数 conv 和 deconv 用来实现这些运算。 长度为 m 的向量 a 和长度为 n 的向量 b 的卷积 c 定义为

$$c(k) = \sum_{j=1}^{k} a(j)b(k+1-j)$$

式中, c 向量的长度为(m-n+1)

解卷是卷积的逆运算,更具体地说向量 a 对向量 c 进行解卷将得到商向量 q 和余量 r,并且满足

$$c(k) - r(k) = \sum_{j=1}^{k} a(j)q(k+1-j)$$

下例中计算了两多项式的卷积和解卷积。

```
p=[2 -5 6 -1 9];
poly2sym(p)
ans =
2*x^4-5*x^3+6*x^2-x+9
d=[3 -90 -18];
poly2sym(d)
ans =
3*x^2-90*x-18
pd=conv(p,d)
pd =
     6 -195 432 -453 9 -792 -162
poly2sym(pd)
ans =
6*x^6-195*x^5+432*x^4-453*x^3+9*x^2-792*x-162
pl=deconv(pd, d)
pl =
                             9
     2
          -5
               6 -1
```
5. 多项式的导数

polyder 函数用来计算任何多项式的导数。它不但可以计算单个多项式的导数,也可以用来计算两个多项式的乘法和除法的导数。

例如,在命令窗口输入

a=[1 3 5];

b=[2 4 6];

p=polyder(a), q=polyder(b), c=polyder(a, b), [q,d]=polyder(a,b)

回车显示结果

p = 2 3 q =4 4  $c \equiv$ 38 8 30 56 q =-2 -8 -2 d = 4 16 40 48 36

说明:

> 调用函数 polyder 只有一个输入参数和一个输出参数是计算单个多项式的导数。

> 调用函数 polyder 有两个输入参数,而只有一个输出参数的是计算两个多项式乘积的导数。

> 调用函数 polyder 有两个输入参数,有两个输出参数是计算两个多项式除法的导数。

6. 多项式拟合

多项式拟合是多项式运算的一个重要组成部分,在工程应用及科研工作中都得到了 广泛的应用。其实现 一方面可以由矩阵的除法求解超定方程来进行;另一方面在 MATLAB 中还提供了专用的拟合函数 poiynto 其调 用格式为:

- ➢ polyfit(X,Y,n) 其中X、Y为拟合数据,n为拟合多项式的阶数。
  - ➢ [p, s]=polyfit(X, Y, n) 其中 p 为拟合多项式系数向量, s 为拟合多项式系数向量的结构 信息。

例如在命令窗口输入

```
x=0:pi/20:pi/2;
```

```
y=sin(x);
```

a=polyfit(x, y, 5);

x1=0:pi/30:pi*2;

y1=sin(x1);

y2=a(1)*x1.^5+a(2)*x1.^4+a(3)*x1.^3+a(4)*x1.^2+a(5)*x1+a(6);

plot(x1, y1, 'b-', x1, y2, 'r*')

legend('原曲线', '拟和曲线')

axis([0, 7, -1.2, 4])

回车显示结果如图 4-1 所示,对 5 阶多项式对[0, p/2]上的正弦函数值进行最小二乘拟合。



图 4-1 正弦函数的最小二乘拟合图

由于拟合是在[0, p/2]进行的,故所得曲线在此区间内与原曲线符合得很好。而在区间外,两曲线差别较大。

7. 部分分式展开

residue 函数用来求解两个多项式之比的部分分式展开。这个指令在应用中非常有用。在控制中,将系统表示成传递函数的形式。

对于多项式 b 和 a, 如果没有重根,

 $\frac{b(s)}{a(s)} = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + \dots + \frac{r_n}{s - p_n} + k_s$ 

其中 r 是留数的列向量, p 是极点位置列向量, k 是直项行向量。例如, 考虑传递函数

```
-4-8s^{-1} <sup>-2</sup>
1 + 6s^{-1} + 8s
在命令窗口输入
b=[-4 8];
a=[1 6 8];
[r, p, k]=residue(b, a)
回车显示结果
r =
   -12
     8
p =
    -4
    -2
k =
     []
给定了三个输入参数(r,p和k),residue 函数将给出多项式形式结果
[b2, a2]=residue(r, p, k)
b2 =
    -4
           8
a2 =
     1
           6
                 8
```

# 4.4 稀疏矩阵

稀疏矩阵是指含有大量 0 元素的矩阵。求解普通方程或者部分微分方程是的数值解,就是一种稀疏矩阵。 为了节省存储空间和计算时间,考虑到矩阵的稀疏性,MATLAB 会使用特殊的命令对它运算。

4.4.1 稀疏矩阵的存储

一个稀疏矩阵中有许多元素等于零,这便于矩阵的计算和保存。如果 MATLAB 把一个矩阵当作稀疏矩阵, 那么只需在 m×3 的矩阵中存储 m 个非零项。第1列是非零元素的行下标,第2列是非零元素的列下标,第3列 是非零元素值,不必保存零元素。如果存储一个浮点数要8个字节,存储每个下标要4个字节,那么整个矩阵 在内存中存储需要16×m个字节。例如使用命令

A=eye(1000);

将得到一个 1000×1000 的单位矩阵,存储它需要 8 Mb 空间。但是如果使用命令

B=speye(1000);

用一个 1000×3 的矩阵来代表,每行包含有一个行下标、列下标和元素本身。现在只需 16Kb 的空间就可以存储这个 1000×1000 的单位矩阵,它只需要满单位矩阵的 0.2%存储空间。

因为 MATLAB 只对矩阵的非零元素进行操作,所以稀疏矩阵的计算速度更快,这是稀疏矩阵的第二个突出的优点。

例如使用上面创建的矩阵 A 和矩阵,计算 2*A 需要一百万次的浮点运算,而计算 2*B 只需要 2000 次浮点运算,大大加快了运算速度。

#### 4.4.2 创建稀疏矩阵

MATLAB 不能自动创建稀疏矩阵,所以要用特殊的命令来得到稀疏矩阵,在表 4-22 中列出了常用的这些命 令。

### 1. 创建稀疏矩阵

在 MATLAB 中, 用命令 sparse()来创建一个稀疏矩阵。 例如在命令窗口输入命令

表 4-22

```
稀疏矩阵命令
```

命令	说明
sparse(A)	由非零元素和下标建立稀疏矩阵 A。如果 A 已是一个稀疏矩阵,则返回 A 本身
sparse(m,n)	生成一个 m×n 的所有元素都是 0 的稀疏矩阵
sparse(u,v,a)	生成一个由长度相同的向量 u , v 和 a 定义的稀疏矩阵。其中 u 和 v 是整数向量 , a
	是一个实数或者复数向量。(ui, vi)对应值 ai , 如果 a 中有零元素 , 则将这个元素排
	除在外。稀疏矩阵的大小为 max(u)×max(v)
sparse(u,v,a,m,n)	生成一个 m×n 的稀疏矩阵, (ui, vi)对应值 ai。向量 u, v 和 a 必须长度相同
sparse(u,v,a,m,n,	生成一个 m×n 的含有 nzmax 个非零元素的稀疏矩阵。(ui, vi)对应值 ai。nzmax 的值
nzmax)	必须大于或者等于向量 u 和 v 的长度
find(x)	返回向量 x 中非零元素的下标。 如果 x=X 是一个矩阵 , 那么 X 的向量就作为一个长
	向量来考虑
[u,v]=find(A)	返回矩阵 A 中非零元素的下标
[u,v,s]=find(A)	返回矩阵 A 中非零元素的下标。用向量 s 中元素的值及 u 和 v 中相应的下标 , 实际
	上就是向量 u、v 和 s 作为命令 sparse 的参数
full(S)	将稀疏矩阵 S 转换成一个满矩阵
spconvert(D)	将一个有三列的矩阵转换成一个稀疏矩阵。D 中的第 1 列作为行的下标, 第 2 列作
	为列的下标,最后一列作为元素值

A=eye(5); B=sparse(A)

回车显示结果

B=

(1,1) 1

(2,2) 1

(3,3) 1

(4,4) 1

(5,5) 1

如果给定向量

ind1=[1 2 3 3 4 2];

ind2=[1 2 1 4 5 3];

nu=[0 1 2 3 0 5];

这里用的是行向量,同样也可以使用列向量。输入命令行

m=sparse(ind1, ind2, nu)

回车显示结果

m=

- (3,1) 2
- (2,2) 1
- (2,3) 5

(3,4) 3

2.矩阵转换

在 MATLAB 中可以使用命令 full()将稀疏矩阵转换成一个满矩阵。例如对于上面创建的的矩阵 m,使用命

<del></del>

Fm=full(m) 回车显示结果

Fm=

0	0	0	0	0
0	1	5	0	0
2	0	0	3	0
0	0	0	0	0

输入命令

whos

可以查看矩阵的大小及存储所占空间,能够看出虽然两个矩阵的大小相同,但是其中稀疏矩阵需要的存储 空间更小些。

### 4.4.3 稀疏矩阵运算

1.稀疏矩阵的运算规则

前面章节中的算术和逻辑运算都适用于稀疏矩阵。MATLAB 中对满矩阵的运算和函数同样可用在稀疏矩阵 中。结果是稀疏矩阵还是满矩阵,将取决于运算符或者函数及下列的操作函数:

- 当函数用一个矩阵作为输入参数,输出参数为一个标量或者一个给定大小的向量时,输出参数的格式总是返回一个满阵形式,如命令 size。
- 当函数用一个标量或者一个向量作为输入参数,输出参数为一个矩阵时,输出参数的格式也总是 返回一个满矩阵,如命令 eye。还有一些特殊的命令可以得到稀疏矩阵,如命令 speye。
- > 对于单参数的其他函数来说,通常返回的结果和参数的形式是一样的,如 diag。
- 对于双参数的运算或者函数来说,如果两个参数的形式一样,那么也返回同样形式的结果。在两个参数形式不一样的情况下,除非运算的需要,均以满矩阵的形式给出结果。
- ▶ 两个矩阵的组和[A B],如果 A 或 B 中至少有一个是满矩阵,则得到的结果就是满矩阵。
- > 表达式右边的冒号是要求一个参数的运算符,遵守这些运算规则。
- ▶ 表达式左边的冒号不改变矩阵的形式。

例如,对于已定义的向量

A=eye(5); B=sparse(A); h=[1; 2; 0; 4; 5]; C=5*B, D=A+B, x=B\h

102

回3	车显着	示结婚	果		
C=					
	(1,1	1)	5		
	(2,2	2)	5		
	(3,3	3)	5		
	(4,4	4)	5		
	(5,5	5)	5		
D=					
	2	0	0	0	0
	0	2	0	0	0
	0	0	2	0	0
	0	0	0	2	0
	0	0	0	0	2
x=					
	1				
	2				
	0				
	4				
_	5				
СĘ	<u></u> €—∕	个稀正	<b>流矩</b> 隊	车,I	) 是一个满矩阵,x 这是一个满向量。
在	MAT	LAB	甲辽	2有很	&多的命令可以对非零元素进行操作,表 4-23 种列出了这些命令。

#### 表 4-23

矩阵非零元素的操作

命令	说明
nnz(A)	求矩阵 A 中非零元素的个数。它既可求满矩阵也可求稀疏矩阵
spy(A)	画出稀疏矩阵 A 中非零元素的分布。也可用在满矩阵中,在这种情况下,只给 出非零元素的分布
spy(A,cstr,size)	用指定的颜色 cstr 和在 size 规定的范围内画出稀疏矩阵 A 中非零元素的分布
nonzeros(A)	按照列的顺序找出矩阵 A 中非零的元素
spones(A)	把矩阵 A 中的非零元素全换为 1
spalloc(m,n, nzmax)	产生一个 m×n 阶只有 nzmax 个非零元素的稀疏矩阵。这样可以有效地减少存 储空间和提高运算速度
nzmax(A)	给出为矩阵 A 中非零元素分配的内存数。不一定和 nnz(A)得到的数相同
issparse(A)	如果矩阵 A 是稀疏矩阵,则返回 1;否则返回 0
spfun(fcn,A)	用 A 中所有非零元素对函数 fcn 求值 , 如果函数不是对稀疏矩阵定义的 , 同样 也可以求值
spfun(A)	求稀疏矩阵 A 的结构秩。对于所有的矩阵来说,都有 sprank(A rank(A)

### 2. 稀疏矩阵的初等变换

3. 稀疏矩阵的分解

### 4.4.4 稀疏矩阵的特例

MATLAB 中有四个基本稀疏矩阵,它们是单位矩阵、随机矩阵、对称随机矩阵和对角矩阵。前面已经讨论 过使用 speye()函数产生单位矩阵,下面着重讨论后两种矩阵形式。

1. 随机稀疏矩阵

MATLAB 中可以使用 sprand()命令产生随机稀疏矩阵,表 4-23 列出了常用命令形式。

例如对于已知矩阵

A=[

0 1 0 0;

1 0 0 0;

0 1 0 0;

0 0 1 0];

在命令窗口中输入

r=sprandn(A)

回车显示结果

```
r=
```

(2,1)	-0.4326
-------	---------

- (1,2) -1.6656
- (3,2) 0.1253
- (4,3) 0.2877

### 得到的随机稀疏矩阵 r 和矩阵 A 有着相同结构的,且元素服从均匀分布

表 4-24

随机稀疏矩阵

命令	说明
sprand(A)	生成与 A 有相同结构的随机稀疏矩阵 , 且元素服从均匀分布
sprand(m,n,dens)	生成一个 m×n 的服从均匀分布的随机稀疏矩阵 , 有 dens×m×n 个非零元素 ,
	0 dens 1。参数 dens 是非零元素的分布密度
sprand(m,n,dens, rc)	生成一个近似的条件数为 1/rc、大小为 m×n 的随机稀疏矩阵。如果 rc 是一个
	长度为1 l(min(m, n))的向量,那么矩阵将 rc _i 作为它1 个奇异值的第一个,其
	他的奇异值为0
sprandn(A)	生成与 A 有相同结构的随机稀疏矩阵 , 且元素服从正态分布。
sprandn(m,n,dens, rc)	生成一个 m×n 的服从正态分布的随机稀疏矩阵,和 sprand 一样
sprandsym(S)	生成一个随机对称稀疏矩阵。它的下三角及主对角线部分 与 S 的结构相同,
	矩阵元素服从正态分布
sprandsym(n,dens)	生成一个 m×n 的随机对称稀疏矩阵。矩阵元素服从正态分布,分布密度为 dens
sprandsym(n,dens, rc)	生成一个近似条件数为 1/rc 的随机对称稀疏矩阵。元素以 0 对称分布,但不是
	正态分布。如果 rc 是一个向量,则矩阵有特征值 rc _i 。也就是说,如果 rc 是一
	个正向量,则矩阵是正定矩阵
sprandsym(n,dens, rc,k)	生成一个正定矩阵。如果 k=1,则矩阵是由一正定对称矩阵经随机 Jacobi 旋转
	得到的,其条件数正好等于 1/rc;如果 k=2,则矩阵为外积的换位和,其条件
	数近似等于 1/rc
sprandsym(S,dens,r	生成一个与矩阵 S 结构相同的稀疏矩阵,近似条件数为 1/rc。参数 dens 被忽略,
c,3)	但是这个参数在这个位置以便函数能确认 最后两个参数的正确与否

### 2. 对角稀疏矩阵

用命令 spdiags()可以取出对角线元素,并创建带状对角矩阵。假设矩阵 A 的大小为 m×n,在 p 个对角线上有非零元素。B 的大小为 min(m×n)×p,它的列是矩阵 A 的对角线。向量 d 的长度为 p,其整型分量给定了 A 的对角元:

di<0 主对角线下的对角线。

di=0 用主对角线

di>0 主对角线上的对角线

表 4-25

对角稀疏矩阵

命令	说明
[B,d]=spdiags(A)	对角稀疏矩阵求出 A 中所有的对角元,对角元保存在矩阵 B 中,它们的下标
	保存在向量 d 中
spdiags(A,d)	生成一个矩阵,这个矩阵包含有矩阵 A 中向量 d 规定的对角元
spdiags(B,d,A)	生成矩阵 A, 用矩阵 B 中的列替换 d 定义的对角元
A=spdiags(B,d,m,n	用保存在由 d 定义的 B 中的对角元创建稀疏矩阵 A
)	

### 4.4.5 系数阵为稀疏矩阵的线性方程组

在 MATLAB 中,系数阵为稀疏矩阵曲线性方程组的解法可分为两类:直接法和迭代法。一般来说,直接法都是高斯消去法的变种,直接法需对矩阵的每个元素进行操作。迭代法通过有限步迭代,得出方程组的近似解。 一般情况下直接法比迭代法更快、适用范围更广,但直接法需要更多的内存空间。

当用直接法求解方程组时,一殷不采用 lu、ch01 等分解函数,而使用矩阵的左除和右除运算符。例如当矩阵 A 为方阵时,线性方程组 A*x = B 的解为 x = A\G;当矩阵 A 为长方阵时,采用最小二乘法求解。当矩阵 A 为方阵时,无论是满阵还是稀疏矩阵,A\B 占用的空间和 A 差不多,而计算方法将根据 A 的具体情况采用下面几种方法:

▶ 当 A 为三角阵时,对矩阵 B 的每一列直接计算。

当 A 为三角阵的初等变换阵时,先对 A 进行变换,再对矩阵的每一列进行计算。

▶ 当 A 为对称或共轭对称且对角元素都为正时,采用 Cholesky 分解再对 B 的每列进行求解。

▶ 其他情况时,采用 LU 分解,再对矩阵 B 的每列求解。

MATLAB 提供了六个专门用于求解稀疏线性方程组的函数; bicg, bicgstab, cgs, gmres, pcg和 qmr。这六 个函数都用于解方程组 Ax = b, 其中函数 pcg 只适合对称正定矩阵 A, 其他五个可解非对称方阵。

## 4.5 多维数组

在 MATLAB 中,数组的维数可以超过二维,多维数组可以是数值型、字符型、单元型、或结构型的。多维数组有广泛的用途。例如,多变量数据的表示,或者二维数据的多页表示。MATLAB 提供了一系列的函数直接 支持多维数组运算,用户可以通过创建 M 文件来扩充这些函数。

4.5.1 多维数组的表示

多维数组是通常的二维矩阵在 MATLAB 中的扩展。二维矩阵有两维:行维和列维。矩 阵的元素可以通过 它的两个下标访问,第一个下标代表行索引,第二个下标代表列索引。多维数组采用另外的下标索引。例如一 个三维数组,使用三个下标

▶ 第一个表示数组的第一维:行维:

▶ 第二个表示数组的第二维:列维;

> 第三个表示数组的第三维为页,在此使用了页的概念来表示第三维和更高的维。

为了访问数组第二页的第二行、第三列,采用下标(2,3,2)。当给数组增加一维,同样也需要增加下标。例如, 四维数组有四个—下标,前两个表示行和列,后两个表示数组的第三、第四维。

➢ 通常的多维数组函数位于 datatypes 子目录中

4.5.2 创建多维数组

可以采用和创建二维数组相同的方法创建多维数组。另外,MATLAB 提供了一个有用的函数 cat 来建立多维数组。

1.利用索引生成数组

创建多维数组的方法之一就是先创建一个二维数组,然后将它扩展到多维。例如,首先创建一个简单的二 维数组 A:

A=[578;019;436];A 是一个3x3的数组,它有三行和三列,采用下列指令给A加一个第三维

A(:,:,2)=[1 0 4;3 5 6;9 8 7]

A(:,:,1) =

5	7	8
0	1	9
4	3	6

A(:,:,2) =

1 0 4

3 5 6

9 8 7

可以采用相同的方式继续加入行、列或页。

若要将 A 扩展到任意维, 需要

给数组增加合适的下标,并且赋上所需的数值。

给对应的数组维增加相同的元素个数。对于数值数组,所有的列必须具有相同数目的元素,所有的页必须有相同的行和列。

例如,可以利用 MATLAB 标量扩展的能力,以及冒号算子,用一个值填充数组的整个一维,在命令窗口输

A(:,:,3)=5;

A=(:,:,3)

回车显示结果

ans =

5	5	5
5	5	5
5	5	5

要将 A 变成一个 3x3x3x2 的四维数组,可以在命令窗口输入

A(:,:,1,2)=[1 2 3;4 5 6;7 8 9]

A(:,:,2,2)=[9 8 7;1 3 4;7 8 9]

A(:,:,3,2)=[1 1 0;4 5 6;7 7 9]

🗻 当前两个指令输入以后, MATLAB 用零来填充 A, 以维持所必须的数组对应的维数大小

2.利用函数生成数组

可以使用 MATLAB 函数,如 randn, ones 和 zeros 生成多维数组,其用法和生成二维数组相同。指令中提供的每一个参数代表了结果数组的对应维数的大小。例如,生成一个 4x3x2 正态分布的随机数组指令为:

b=randn(4, 3, 2)

生成一个每个元素都是常数的数组,使用 repmat 函数。repmat 指令通过一个表示数组维数的向量复制一个数组(在此,为 lxl 数组),生成多维数组。例如,在命令窗口输入

b=repmat(5, [3 4 5])

回车显示结果

b(:,:,1) =

U(.,.,1	) –			
	5	5	5	5
	5	5	5	5
	5	5	5	5
b(:,:,2	2) =			
	5	5	5	5
	5	5	5	5
	5	5	5	5
b(:,:,3	3) =			
	5	5	5	5
	5	5	5	5
	5	5	5	5
b(:,:,4	l) =			
	5	5	5	5
	5	5	5	5
	5	5	5	5

λ

b(:,:,5) =

5	5	5	5
5	5	5	5
5	5	5	5

🔉 一个数组的任一维大小可以是零,称之为空数组

3. 利用 cat 函数创建多维数组

cat 函数使用一个简单的方式创建多维数组,它将—系列的数组沿着特定的维连接成—个数组

b=cat(dim, A1, A2,...)

其中 A1, A2 等等是需要连接的数组, dim 表示连接到第几维。

例如,可以利用 cat 创建一个新数组

b=cat(3, [2 8;0 5], [1 3;7 9])

5

b(:,:,1) =

2 8

0

b(:,:,2) =

1 3

7 9

cat 函数使数组中已经存在的元素和新数据合并。另外, cat 函数可以嵌套调用。例如, 使用下面的命令将 创建了一个四维数组

A=cat(3, [92;65], [71; 84]);

B=cat(3, [35;01], [56; 21]);

D=cat(4, A, B,cat(3,[12;34], [43;21]));

如果需要, cat 自动在维数之间加上下标 1。例如:创建 2x2x1x2 的数组, 输入

C=cat(4, [1 2;4 5], [7 8;3 2]);

在前一种情况下, cat 插入了足够多的一维的数组, 用以创建一个四维数组, 它的第四维不是 1。

如果 dim 参数是 5 , 则前一个指令将产生一个 2x2xlxlx2 的数组。这个额外的增加用于索引数组的表达式。

例如,可以使用C(1,2,1,2)来引用四维数组的值为8的元素。

4. 多维数组信息的获取

可以使用 MATLAB 函数和命令得到所创建的多维数组的信息,如图 4-26 所示。

表 4-26

#### 多维数组信息函数

信息	命令	例子
数组大小	size	size(c)
数组维数	ndims	ndims(c)
数组存储和格式	whos	whos

4.5.3 用多维数组工作

许多应用于二维数组的概念都可以扩展到多维数组。

考虑一个 10×5×3 数组 a, 其元素为随机整数

a=fix(8*randn(10, 5, 3));

1.索引

为了访问多维数组的单个元素,使用整数下标。每一个下标表明一维,其中第一个表明行维、第二个表明 列维、第三个表明第一页等等。例如,为了访问 a 的第二页的元素 (3,2),可以使用指令 a(3,2,2)。

可以使用向量作为数组下标。在这种情况,每一个数组元素是一个有效的下标,也就是,位于数组所定义的维数范围内。为了访问 a 的第三页的元素(2,1),(2,3),和(2,4),使用指令 a(2,[1 3 4],3)。

MATLAB 的冒号索引可以扩展到多维数组。例如,为了引用 a 的第二页的整个第三列,使用 a(:, 3,2)。

冒号算子也可用于访问其它的数据的子集。例如, a(2:3, 2:3, 1)返回一个 2x2 的数组, 它是 a 的第一页的数据的子集, 包含了 a 的第一页的第二、三行和第二、三列的数据。

冒号算子可以作为数组下标出现在赋值语句的左右两边。

例如,创建一个4x4的全0数组

C=zeros(4,4)

现在将数组 a 的 2x2 子集赋给数组 C 中间的四个元素

C(2:3, 2:3)=a(2:3, 1:2, 2)

在利用冒号进行多维数组索引时,要避免产生歧义。例如,下面的赋值语句:

A(:,:,2)=1:10

就是歧义的,因为它没有提供足够的关于维数形状的信息来接收数据。在上面的情况中,赋值语句试图将 一个一维向量赋给二维空间,MATLAB将给出错误信息。为了解决这个问题,必须确保提供了足够的空间分派 数据。例如,

A(1,:, 2)=1:10;

2.数组整形

除非改变了数组的形状和大小,MATLAB的数组保持创建时的维数。可以通过增加和删除元素来改变数组的大小。

通过重新确定数组的行、列或页而保持相同的元素可以改变数组的形状。reshape 函数执行了这一操作。对于多维数组,它的指令形式为:

B=reshape(A,m,n)

B=reshape(A, m, n, p...)

B=reshape(A, [m n p ... ])

B=reshape(A,siz)

说明:

B=reshape(A, m, n)返回 m×n 矩阵 B, 其元素按列取自矩阵 A, 如果 A 没有 m*n 个元素, 将会产生错误 结果。

B=reshape(A, m, n, p...)或 B=reshape(A, [m n p....])返回 N 维数组,其元素按列取自矩阵 A,但是维数变为 m×n×p×...; m*n*p...必须等于 prod(size(x))。

B=reshape(A,siz)返回多维数组 B,它的维数由向量 siz 决定,它和 A 具有相同的元素,prod(siz)必须等于 prod(size(A))。

当创建或整形数组时,如果采用显式说明,MATLAB 将创建单一维;或者执行一个计算时,将会产生数组 其中某一维的大小为 1。例如

B=repmat(5, [2 3 1 4]; size(B)

ans =

2 3 1 4

3.数组各维的移动

shiftdim 数组各维的移动

B=shiftdim(X, n)

[B,nshifts]=shiftdim(X)

说明:

- B=shifidim(X,n) n次移动数组 X 的各维。n必须是整数,当 n 是正数时,shiftdim 指令将把数组的各维向左移,同时将前面的 n 维移到后面:当 n 是负数时,shiftdim 指令将各维向右移,同时填补上单一维。
- ▷ [B,nshifis]=shiftdim(x) 返回数组 B,它和 A 具有相同的元素数目,但是它删除掉了 A 中的最前面的所有的单一维。nshifis 返回被删除的单一维的数目。如果 size(A,dim)=1,则 A 的第 dim 维就是单一维。
- ▶ 如果 X 是标量,则 shiftdim 指令不起作用。

```
4. 单一维删除
   squeeze 函数的调用方式为
   B=squeeze(A)
   返回的数组 B 与 A 具有相同的元素,但是删除了其中所有的单一维。
   例如,下面的命令对矩阵的维数进行变换
   a=rand(1, 1, 3, 1, 2);
   [b, n]=shiftdim(a), c=shiftdim(b, -n), d=shiftdim(a, 3), e=squeeze(a)
   回车显示结果
   b(:,:,1) =
       0.4565
       0.0185
       0.8214
   b(:,:,2) =
       0.4447
       0.6154
       0.7919
   n =
        2
   c(:,:,1,1,1) =
       0.4565
   c(:,:,2,1,1) =
       0.0185
   c(:,:,3,1,1) =
       0.8214
   c(:,:,1,1,2) =
       0.4447
   c(:,:,2,1,2) =
       0.6154
   c(:,:,3,1,2) =
       0.7919
   d(:,:,1,1,1) =
       0.4565
                0.4447
   d(:,:,1,1,2) =
       0.0185
                0.6154
   d(:,:,1,1,3) =
       0.8214
               0.7919
   e =
               0.4447
       0.4565
       0.0185
                0.6154
       0.8214
                0.7919
   squeeze 函数并不影响二维数组,行向量仍然是行向量。
   5. 数组维数重新排列及逆排列
   permute 函数将数组地维数重新排列,其命令调用格式为
   B=permute(A, dim)
   dim 是一个向量,用于确定数组 A 的各维的新的顺序,其中 1 对应于第一维(行), 2 对应于第二维(列), 3
对应于第三维(页)等等。例如
```

B=permute(A, [2 1 3])

为进一步了解 permute 函数的细节,考虑一个 5x4x3x2 的四维数组 A,重新排列各维。首先放列维、紧跟着 是第二页维,然后是第一页维,最后是行维,结果产生一个 4x2x3x5 的数组。

可以将 permute 函数看作是 transpose 函数的扩展:交换数组的行维和列维。对于 permute, 输入的维数的顺序表决定了下标的重排顺序。

ipermute 函数是 permute 函数的逆运算。给定一个输入数组 A 和一个向量 V, ipermute 产生一个数组 B 使得 permute(B, V)返回 A。

例如,下列指令创建了数组 E 等于输入数组 C。

D=ipermute(C , [1 4 2 3]);

E=permute(D, [1 4 2 3]);

当把一个数组重新排列以后,可以调用 ipermute 指令并采用相同的维数向量重新获得。

4.5.4 多维数组的运算

许多的 MATLAB 运算和数学函数允许采用多维数组作为参数。这些函数在多维数组的特定维上进行操作。 也就是说,针对单个元素、向量或矩阵进行操作。

1. 向量操作函数

针对向量进行操作的函数,如 sum, mean 等,隐含地对多维数组的第一个非单一维进行运算。但是,也有 例外, cross 函数求两个向量的叉乘,对多维数组的第一个维数长度为3的非单一维进行运算。

查 在许多情况,这些函数对于输入的参数有其它的限制。例如,一些函数接受多个数组要求数组大小相同

2. 函数按元素进行操作

MATLAB 函数对于二维数组按元素进行操作,象 elfun 目录的三角函数和指数函数对于 多维数组采用同样的方式工作。例如, sin 函数返回一个和输入参数大小相同的数组。输出数组的每一个元素是输入数组对应元素的正弦值。

类似的,算术、逻辑和关系算子对多维数组按对应元素进行运算,要求多维数组有相同的大小。如果一个 是标量、而另一个是数组,则将标量和数组的每一个元素进行运算。

3. 矩阵运算函数

针对矩阵进行运算的函数,例如 matfun 目录下的线性代数和矩阵函数,不接受多维数组作为参数。也就是 说,对于多维数组,不能执行 matfun 目录下的函数运算或数组的"*","^","\"或"/"运算。在这种情况下, 采用多维数组作为参数或操作数将会出错。

可以通过索引将多维数组的一部分应用于矩阵函数或算子。例如,要创建一个三维数组,在命令窗口输入 A=cat(3, [1 2 3;9 8 7;4 6 5], [0 3 2;8 8 4;5 3 5],[6 4 7;6 8 5;5 4 3])

回车显示结果

A(:,:,1) =			
1	2	3	
9	8	7	
4	6	5	
A(:,:,2) =			
0	3	2	
8	8	4	
5	3	5	
A(:,:,3) =			
6	4	7	
6	8	5	
5	4	3'	
加甲收敕人	~ 夕 绀	粉细应田	工动动物

如果将整个多维数组应用于 eig 函数,将会出错。例如

??? Error using ==> eig

Input arguments must be 2-D.

然而,可以使用多维数组的一部分用于 eig 函数。例如,利用冒号索引数组的一页

eig(A(:,:,2))

ans =

12.9129

-2.6260

2.7131

如果数组的第一个下标不是冒号,必须使用 squeeze 函数以避免出现错误。例如, eig(A(2,:,:)) 将会产生错误,因为输入数组的大小是[133]。而表达式 eig(squeeze(A(2,:,:))),将会给 eig 函数 传递一个有效的二维数组

# 4.6 结构数组与细胞数组

细胞数组和结构数组是 MATLAB 中比较特殊的两种数据类型。细胞数组的每个元素称为细胞,细胞本身就 是 MATLAB 的数组。在 MATLM 中,正是由于有细胞数组,才可将不同类型、不同维数的数组组合成一个大数 组。

结构数组也把不同的数据类型组合在一起。和细胞数组不同之处在于:结构数组的引用是通过属性名来实 现的。在本节中将分别介绍结构数组和细胞数组。

4.6.1 结构数组

前面已介绍的数据类型有数值数组、字符数组。有时需要将不同的数据类型组合成一个整体,以便于引用。 这些组合在一个整体中的数据是相互联系的。例如,一个学生的学号、姓名、性别、年龄、成绩、家庭地址等 项,这些项都是和某个学生有联系的。如果分别定义为不同的变量,难以反应它们之间的内在联系。应当把它 们组合成一个组合项,在一个组合项里包含若干个类型不同的数据项。MATLAB 提供了这样一种数据类型,称 为结构,它相当于数据库中的"记录"。

一个学生的学号、姓名、性别、年龄、成绩、家庭地址等项为不同的数据类型。其中,学号可以为数值类 型或字符类型,姓名为字符类型,年龄为数值标量,而成绩可以为标量,也可以为数值数组。在 MATLAB 里, 把结构的这些不同的项称为属性,例如上面学号、姓名、性别、年龄、成绩、家庭地址等都为属性。和其他数 据类型一样,结构也是数组,而且可以定义多维的结构数组。

1.结构数组的定义

要定义一个结构数组,可以采用两种方法:

用赋值语句定义;

➤ 用函数 struct 定义。

用赋值语句定义结构时,只要给出结构的属性赋值,MATLAB 就会自动把该属性增加到结构中,赋值时, 结构名和属性名用英文句号分开。例如,下面三条语句将定义一个 1×1 的结构数组,结构名为 student,有三个 属性:name、num、test。该结构数组只有一个元素,在命令窗口中键入结构名 student,将显示该元素所有属性 的属性值的特性。

student.name='John Doe'; student.num=12345; student.test=87; student student = name: 'John Doe' num: 12345 test: 87 再键入以下三行可给该结构数组增加一个元素 student(2).name='Ann Mili'; student(2).num=12346;

student(2).test=56;

现在结构数组 student 的维数为 1X2。当结构数组的元素超过 1 个时, MATLAB 的帮助信息中, 不再显示不同属性的值, 而只显示数组名、属性名和维数大小。fieldnames 函数可用来查看结构的所有属性名, fieldnames 把属性名组成细胞数组。下面为以上有两个元素的 student 的显示信息。

student

student =

1x2 struct array with fields:

- name
- num
- test

fieldnames(student)

ans =

'name'

'num'

'test':

当没有定义某个元素的某些属性值时,MATLAB 会自动把这些属性值赋为空矩阵。在结构数组中不同的元素的属性名必须相同,例如上面结构数组 student 的两个元素都只有三个相同的属性名:name、num、test。但不同元素的同一属性可以有维数不同的值。

函数 struct 也可用来定义结构数组,其调用格式为,

结构致组名=struct(属性列表)

其中,属性列表为结构的所有用性构成的字符数组或细胞数组。利用这种调用格式时,所有的属性值都为 空矩阵。也可在函数 struct 中输入属性值,调用格式如下

结构数组名 = structt( ' 属性 1 '、属性值 1 , ' 属性 2 '、属性值 2 , ...)

2. 结构数组属性的修改、设置和获取

结构数组一旦形成,就可用相应的函数取出数组中的某个元素并修改该元素的某个属性的值。以上面建立的 student 数组为例,命令 sb = sbdent(2).name 可取出第 2 个元素的 name 属性的值。

name2=student(2).name

name2 =

Ann Mili

可用命令

student(2).test=[34 56 89;56 76 87];

修改第二个元素的 test 属性的值。而命令

test1=student(2).test(2,3)

将取出第二个元素的 test 属性值的第二行第三列上的数值

test1 =

87

对于结构数组,也可采用下标来引用数组中的某些元素。例如命令

B=patient(2);

将生成只有一个元素的结构数组。

函数 setfield 和 getfield 分别用于给结构数组的某个元素的属性赋值和取值。函数 getfield 的调用格式为:  $f = getfield(结构数组, {元素下标}, '属性'', {属性值下标})$ 

大括号 "{}"中的参数表示可选。当数组只有一个元素时,元素下标可以省略。例如,要取出 student 数组 中第二个元素的 ' name ' 值,命令为: str=getfield(student,{2},'name')

str =

Ann Mili

该命令等价于 f = student(2).name。

函数 setfield 和 getfield 的调用类似。

和其他数组一样,结构数组可用 slze 函数来求数组的维数。size 函数还可用来求结构数组中某个元素的某个属性值的维数。下面仍以结构数组 student 为例说明 size 的用法。

size(student) ans = 1 2 size(student(2).test) ans =

2

3

当给结构数组中某个元素添加新的属性时,数组中的所有元素都添加上新属性。没有指定局性值的那些元素的属性值为空矩阵。同样,当删除某个元素的一个属性时,数组中所有元素的该属性都被删除。删除属性的命令为 rmfield,其调用格式为

stru2=rmfield{'数组名',属性)

命令 rmfield 并不能删除已有的数组的属性,只是在生成的新的结构数组中,不再包含被删除的属性

3. 结构数组的运算

当结构数组为函数的输入参数时,运算和其他数组一样。例如下列命令将计算 student 的 test 矩阵的每列的 平均值。

```
mean(student(2).test)
```

ans =

45 66

当对数组的某个属性值进行统计时,可采用循环语句。

Total=0;

for j=1:length(student)

total = total+student(j).num ;

88

```
end
```

可用中括号简化上述循环。下面两句中的任何一句都和以上循环等价。

Total=sum(student.num);

total = sum([student(1).num , student(2).num]) ;

对于结构数组,还可以编写函数来处理。下面是一个处理结构数组的例子。函数 concen 的输入参数为结构数组,要求结构数组的每个元素有两个属性:身高(hight)和体重(welght)。

function r=concen(test); a=polyfit([test.height],[test.weight],1); r=[test.weight]-(a(1)*[test.height]+a(2)); hight=[test.hight]; weight=[test.weight]; plot(hight,weight,'r'); hold on; plot(hight,r,'y'); hold off; 4.结构数组中的数据组织 数据在结构数组中的组织形式跟数据的特性、数据的使用方法有关。在创建结构数组时,应该对数组中数 据的性质和使用场合有一定的了解。例如对于一次考试中的成绩表,至少有两种数据组织方法。

第一种方法是把姓名、学号、成绩当作结构的属性。可以建立结构数组

A.name=struct( ' name1 ' , ' name2 ' , ' name3 ' ,..);

A.number=struct(123;124;125;...);

A.score=struct(98;78;56;...);

这时,结构数组中只有一个元素。这种组织方法,可以很方便地从数组中取出所有学生的成绩,进行统计 分析。例如,计算平均成绩,可以使用命令

average=mean(A.score)

也可把每个人的相关信息当作数组的一个元素,可以建立结构数组

B(1).name='name1';

B(1).number=123;

B(1).score=98;

B(2).name='name2';

B(2).number=124;

B(2).score=78;

••••

采用这种组织方法时,求平均成绩的调用命令为

average=mean([B.score])

当采用 A 数组的组织方式时,可以很容易地对成绩进行排序等处理,而对数组 B 中成绩进行排序,就相对 麻烦一些;当统计的人数增加时,采用数组 B 更方便。

5. 结构数组的嵌套

一个结构数组的属性可以包含另一个结构数组,甚至包含许多结构数组组成的数组。既可采用 struct 函数也可通过直接赋值来生成嵌套结构数组。下面分别用两种方法来生成同一嵌套结构数组。

直接赋值的方法

A(1).data=[3 4 7;8 0 1];

A(1).nest.testnum='Test 1';

A(1).nest.xdata=[4 2 8];

A(1).nest.ydata=[7 1 6];

A(2).data=[9 3 2;7 6 5];

A(2).nest.testnum='Test 2';

A(2).nest.xdata=[3 4 2];

A(2).nest.ydata=[5 0 9];

使用 struct 函数

A=struct([1 2],'data',[3 4 7;8 0 1], 'nest'....

struct([1 1],'testnum','Test 1','xdata',[4 2 8],'ydata',[7 1 6]));

```
A(2).data=[9 3 2;7 6 5];
```

A(2).nest.testnum='Test 2';

A(2).nest.xdata=[3 4 2];

A(2).nest.ydata=[5 0 9];

引用嵌套结构数组某个几素的属性时,和基本的结构数组类似,采用圆点符。例如对于上面的结构数组, 用 A(1).nest 引用元素 A(1)的 nest 属性.为一个结构; A(1).nest.ydata(2)指 A(1)元素的 ydata 属性的第二个元素。

4.6.2 细胞数组

细胞数组也是 MATLAB 里的一类特殊的数组。在 MATLAB 里,由于有细胞数组这个数据类型,才能把不同类型、不同维数的数组组成为一个数组。

细胞数组的每一个元素可为类型不向、维数不同的矩阵、向量、标量或多维数组,所有元素用大括号括起 来。如矩阵 A=[1 2 3 4;2 3 4 5 ;3 4 5 6],

则命令

 $C = \{A \quad sum(A) \quad sum(sum(A))\}$ 

得到一个 1 × 3 的细胞矩阵。第一个元素为矩阵 A,第二个元素为矩阵 A 的列向量求和形成的行向量,第三 个元素为第二个元素求和得到的标量。由于前两个元素太大,只给出维数信息,第三个元素给出具体数值。

细胞矩阵可用来存储一些维数不等的矩阵。

1.细胞数组的生成

有两种方法可以生成细胞数组:

用赋值语句直接生成;

▶ 先用 cell 函数预分配数组,然后再对每个元素赋值。

当采用赋值语句直接生成细胞数组时,一次只能为数组中的—个元素赋值。随着元素的增多,MATLAB 会 自动地扩充数组的维数和大小。有两种方法可对元素赋值:一种方法采用数组元素的下标赋值。细胞数组元素 的引用和其他数组无异,只是所赋的值用大括号括起来。例如,下面四句命令将建立一个 2×2 的细胞数组。

 $A(1,1) = \{ [1,2,3;0,5,8;7,2,9] \};$ 

A(1,2)={'Anne cat'};

 $A(2,1)=\{3+7i\};$ 

 $A(2,2)=\{0:pi/10:pi\};$ 

如同空的中括号可把变量赋为空矩阵,符号"{}"(大括号对中无仟何值)可把变量赋为空数组。

另一种方法则把细胞数组的元素用大括号括起来,而所赋的值采用其他数组的形式。

例如下面四句生成的细胞数组和上面所生成的完全一样。

A{1,1}=[1,2,3;0,5,8;7,2,9];

 $A{1,2}='Anne cat';$ 

 $A{2,1}=3+7i;$ 

A{2,2}=0:pi/10:pi;

如同中括号对"[]"用来形成矩阵,大括号对"{}"是细胞数组的标志。两者不同之处在于,大括号可以 嵌套,即细胞数组的元素可以是细胞数组,而中括号不可以。

在大括号中, 逗号或者空格表示每行元素之间的分隔, 分号表示不同行之间的分隔。命令 C = {[12], [3 4]; [5 6], [7 8]};

将牛成一个2×2的细胞数组,数组的每个元素都为1×2的数值矩阵。

生成细胞数组的另一种方法是先用 cell 预定义数组,然后用赋值语句给每个元素赋值。例如命令 B = cell(5,3);

将定义一个 5×2 的细胞数组,数组 B 的所有元素都为空矩阵,然后就可对数组的元素分别赋值了,如 B(1,3) = {1:3};

当把细胞数组取名为工作空间中已经存在的数值数组时,MATLAB 将会给出错误信息。和其他数组一样, 当对细胞数组的某一个数组赋值时,MATLAB 并不删除该数组的其他元素。所以,计算出现莫名其妙的错误时, 可删除工作空间中的变量再试。

2. 细胞数组内容的查看

在 MATLAB 命令窗口中采用变量名查看细胞数组的值时,将显示数组的简要信息。下面为前面创建的矩阵 阿 A 的信息。

A =

[3x3 double] 'Anne cat'

[3.0000+7.0000i] [1x11 double]

函数 celldisp 用来显示细胞数组的每个元素的值。函数 cellplot 将画出细胞数组的每个元素的结构图。图 4-2 为命令 cellplot(A)画出的数组 A 的结构图。



图 4-2 细胞数组的图形表示

图 4-2 中,分别给出了四个元素的结构,元素(1,1)为 3×3 矩阵,在图中用九个方格表示;而元素(2,1) 中为标量,图中显示出了该元素的值。

当给已经定义的细胞数组下标范围外的元素赋值时,MATLAB 将自动扩维,对于没有赋值的元素,赋值为 空矩阵。例如用命令 A(3,3) = (55)给上述数组 A 添加一个元素时,矩阵 4 的维数自动变为 3×3。而元素(1,3)、 (2,3)、(3,1)、(3,2)的值部为空矩阵。

综合使用大括号和中括号可以取出细胞数组的某个元素或某些元素的值,也可取出某个元素值的一部分。 在赋值语句中,每次只能给细胞数组的一个元素赋值,采用形如 "A{1,:}=..."的赋值语句是不合法的。和赋值 语句不同,一次可以取出细胞数组中的多个元素。删除细胞数组的元素也采用赋值为空矩阵的办法,在删除命 令中,不用大括号。下面为元素取值的例子。

n{1,1}=[1 2;4 5]; n{1,2}=name'; n{2,1}=2-i; n{2,2}=7;  $c=n\{1,2\}$ c =name  $d=n\{1,1\}(2,2)$ d =5 3. 细胞数组的变形 和其他数组一样,细胞数组也可用赔 reshap 命令重新排列元素的位置。排列前后数组中的元素个数不变。 例如 a=cell(3,4);

size(a) ans = 3 4

b=reshape(a,6,2);

size (b)

ans = 6 2;

4. 用细胞数组代替变量列表

在函数输入参数列表、函数输出参数列表、变量值显示命令和创建数组时,结合使用冒号运算符和大括号 可以方便地引用细胞数组中的多个元素,并把引用的元素分为不同的变量。采用这种方法,可使程序更加简洁。 例如

输入参数列表  $c(1) = \{ [1 2 3] \};$  $c(2) = \{ [1 \ 0 \ 1] \};$  $c(3) = \{ [9 8 7 4] \};$ 函数输出参数列表  $d=conv(c\{1:2\})$ d = 1 4 2 3 2 变量值显示 c{1:3} ans =1 2 3 ans =1 0 1 ans =9 8 7 4 创建数组  $b = [c{1};c{2}]$ b = 1 2 3 1 0 1

当把函数的多个输出参数定义为细胞数组时,数组的每个元素对应输出参数列表的一个变量。当然,也可 以采用 varargin 和 varargout 把输出列表转换成细胞数组。例如,当 B 为 3 × 3 的数值矩阵时,下面的命令将把 B 的特征值矩阵和特征向量矩阵组合成二个元素的细胞数组。

 $[d{1:2}]=eig(b)$ 

d =

[3x3 double] [3x3 double]

5. 细胞数组的嵌套

细胞数组允许嵌套,即一个细胞数组的元素可以是细胞数组,甚至可以是以细胞数组为元素的细胞数组。 下面将分别介绍两种创建嵌套细胞数组的方法。

下面是一个用嵌套大括号定义的细胞数组。

 $A(1,1) = \{ones(5)\};$ 

 $A(1,2) = \{ \{ [5 3 4; 5 6 8; 9 0 7] 'Name1'; [28 6+9i] \{ 17 [] \} \} \};$ 

在上面的命令中,最外面一对大括号用来定义 A 为细胞数组,第二对大括号表明 A 的第二个元素为细胞数组,最里面一对大括号表示 A(1,2)的第二行第一列上的元素为细胞数组。

也可结合 cell 函数和赋值语句来定义嵌套细胞数组。例如可以使用下面的程序与上面的方法等价的建立细 胞矩阵

A = cell(1,2); $A(1,2) = \{ cell(2,2) \};$  $A(1,1) = \{ones(5)\};$  $A{1,2}(1,1)={[5 2 8;7 3 0;6 7 3]};$  $A\{1,2\}(1,2)=\{'Name1'\}$  $A{1,2}(2,1)={[28 6+9i]};$  $A{1,2}(2,2)={cell(1,2)};$  $A\{1,2\}\{2,2\}(1)=\{17\};$ 当结构的属性的多少和属性名不一样时,不能把这些结构组成结构数组细胞数组。在下面的例子中,细胞 数组的两个元素全为结构,第一个元素有二个元素有二个属性。所有的定义语句为: c = cell(1,2);c{1}.name='Tom'; c{1}.number=123; c{1}.score=78; c{2}.name='Jarry'; c{2}.number=124; .... 6. 细胞数组和数值数组之间的转换 一般采用循环语句来实现细胞数组和数值数组之间的转换。例如对于细胞数组 A  $A{1,1}=[1 2;3 4I;$  $A\{1,2\}=[9\ 0;0\ 1];$  $A{2,1}=[7 8;4 1];$  $A{2,2} = [4i 3+2i;1-6i 5];$ 通过下面的循环语句转换为数值数组; for k=1:4 for i=1:2 for j=1:2 $NUM(i, j, k) = A\{k\}(i, j);$ end end end 同样,也可用循环语句把数值数组转换为细胞数组,例如 g = cell(1, 16);for m=1:16 g(m)=NUM(m);end 7. 结构细胞数组 当不同结构的属性不同时,可组成细胞数组,每个结构为细胞数组中的—个元素。 下面的例子中,第一个结构有三个属性,第二个结构有二个属性,下面的语句将这两个结构组成一个细胞 数组。

m=cell(1,2);

m{1}.xdata=[-.03 0.41 1.98 2.12 17.11];

118

m{1}.ydata=[-3 5 18 9 0];

m{2}.label='12/6/75-6/3/98';

m{2}.obs=[56 32 51 6;12 89 34 22];

# 4.7 数据分析和统计

本章将介绍 MATLAB 对数据处理和统计分析的命令。如果没有特别强调 ,本章中的 A 和 B 将是指 m×n×...×p 的多维矩阵 , 而 x 将是指一个向量。

MAnAB 对数据分析命令采用如下两条假定:

统计数据为行向量或列向量时,函数对整个向量进行计算当成单变量的不同观察值;

> 当数据为矩阵时,命令对列进行运算,即把每一列的数据当成是同一变量的不同观察值。

当用户自己编写统计程序时,应当特别注意"对列运算"的特点。

### 4.7.1 最大值和最小值

在 MATLAB 中使用表 4-27 中列出的命令可以方便的求解最大和最小值。

表	4-27
---	------

最大值和最小值

命令	说明
max(x)	返回 x 中最大的元素值,如果 x 是复数,则返回 max(abs(x))值
max(A)	返回一个含有 A 中第 1 维最大值的 1×n××p 矩阵。对于二维矩阵来说,
	返回一个行向量,它的每个元素是 A 中相应列最大的元素;如果 A 为
	复数时,则返回 max(abs(A))值
[y,ind]=max(A)	返回一个含有 A 中第 1 维最大值的 1×n××p 矩阵 y , 并在行向量 ind
	中保存每列的最大数的行下标
max(A,B)	返回一个和 A、B 相同维数的矩阵 , 每一元素都是在 A 和 B 中的相同
	位置上是最大的元素
C=max(A,[],dim)	给出在指定的 dim 维内 A 的最大分量。如 max(A,[],1), 则给出 A 中最
	大的行向量
min(x)	返回向量 x 中最小的元素。该命令关于矩阵的操作和 max 一样, 如果
	x 是复数,则返回 min(abs(x))值

### 例如对于已知矩阵

A=[ 1 2 3; 2 3 1; 3 2 1]; 在命令窗口输入命令行 max(A) 回车将显示结果 ans= 3 3 3 3

### 4.7.2 求和、乘积和差分

### 1. 求和

在 MATLAB 中使用命令 sum()和 cumsum()可以求得各种不同的和。表 4-27 中列出了常用的求和命令。

表 4-27

求和命令

命令	说明
sum(x)	返回向量 x 所有元素的和
sum(A)	返回一个包含矩阵 A 各列元素之和的 1×n××p 矩阵
cumsum(x)	返回一个 x 中元素累计和的向量,也就是第 2 个元素是 x 中前两个元素之和,以
	此类推
cumsum(A)	返回一个与 A 同样大小的矩阵, 它的列是 A 中列的累计和
cumsum(A,dim)	给出 A 中 dim 维的元素累计和,命令 cumsum(A)和命令 cumsum(A,1)相同

例如对上例中的矩阵 A 进行求和及累积和, 可以在命令窗口中输入

### Tsum=sum(A), Tcsum=cumsum(A)

### 回车显示结果

Tsum=

1

6 7 5 Tcsum=

2	3	
3	5	4
6	7	5

### 2. 乘积

乘积的计算也和求和类似。表 4-28 中列出了常用的乘积命令。

表 4-28

### 乘积命令

命令	说明
prod(x)	返回 x 中各元素乘积
prod(A)	返回一个元素是列乘积的多维矩阵
prod(A,dim)	给出 dim 维内的元素乘积
cumprod(x)	返回一个 x 中各元素累计积的向量 , 也就是第 2 个元素是 x 中前两个元素的累
	计积,以此类推
cumprod(A)	返回一个矩阵,其中列元素是 A 中列元素的累计积
cumprod(A, dim)	给出在 dim 维内的累计积

### 例如使用在上例中定义中的矩阵 A,执行命令

回车显示结果

6	12	3
1	2	3
2	6	3
6	12	3:
	6 1 2 6	6 12 1 2 2 6 6 12

# 3. 差分计算

表 4-29

#### 差分与梯度

<u> </u>	28-00
	[
diff(x)	给出一个长度为 n - 1 的向量 , 它的元素是长度为 n 的向量 x 中相邻的元素的
	差。如果 x=(x ₁ x ₂ x _n ),则 diff(x)=(x ₂ - x ₁ x ₃ - x ₂ x _n - x _{n·1} )
diff(A)	│ 在 A 的第一维内计算相邻元素的差分。对于二维矩阵来说,就是
	diff(A)=A(2:m,:) - A(1:m - 1,:)
diff(x,k)	求出第 k 次差分,diff(x,2)和 diff(diff(x))等价

命令	说明
diff(A,k,dim)	在 dim 维内求出第 k 次差分。
[DAdx,DAdy, DAdz,]=gradi ent(A)	在矩阵 DAdx、DAdy、DAdz 等中返回矩阵 A 的偏导数 ,每个矩阵包含 $\partial A / \partial x$ 、 $\partial A / \partial y$ 、 $\partial A / \partial z$ 等相应的下标
[DAdx,DAdy DAdz,]= gradient(A,h1, h2,h3,)	返回偏导数 $\partial A/\partial x$ 、 $\partial A/\partial y$ 、 $\partial A/\partial z$ 等,如果给出参量 h1,h2,h3,可将它们用作每个变量的步长
del2(A)	返回离散拉普拉斯算子,矩阵中的元素为 A 中元素和它相邻的四个元素的平均 值的差分

在 MATLAB 中可以使用命令 diff 可以进行差分计算,表 4-29 中还列出了一些与 diff 命令相关的其他命令。 差分计算是很容易的,它还可以当作导数的近似值来用。例如在命令窗口中输入

x=[1 4 9 16 25] ; d1=diff(x), d2=diff(d1), d3=diff(d2)

回车显示结果

d1 =3 5 7 9d2 =2 2 2 2d3 =0 0

🔉 如果将计算得到的差分作为导数的近似值来用,必须除以两点之间的距离

### 4.7.3 常用统计命令

除了上面提到的 max、min、sum 、prod 等对矩阵列操作的命令外,MATLAB 还提供了数据统计分析的命 令。

1. 平均值、中值和标准差

表 4-30 列出了 MATLAB 中进行平均值、中值和标准差计算的常用命令。

表	4-30
---	------

平均值、中值和标准差

命令	说明
mean(x)	求出向量 x 的算术平均值
mean(A,dim)	给出一个 1×n××p 的矩阵,它包含 A 中第 1 维的各个平均值。如果给出了 dim,
	就在 dim 维内计算
median(x)	求出向量 x 中元素的中值
median(A,dim)	给出一个 1×n××p 的矩阵, 它包含 A 中第 1 维各列的中值。如果给出了 dim,
	就在 dim 维内计算
std(x)	求出向量 x 中元素的标准差
std(A,dim)	给出一个 1×n××p 的矩阵,它包含 A 中第 1 维的各列标准差。如果给出了 dim,
	就在 dim 维内计算标准差

例如对于矩阵

A =[

1 1; 2 2; 3 3; 4 100;];

执行下列命令

ave=mean(A), med=median(A), dev=std(A)

回车显示结果

2. 协方差

在 MATLAB 中使用命令 cov 和 corrcofe 来求协方差和相关系数。需要注意的是,这些命令只能用在二维矩 阵中。表 4-31 中列出了常用的求协方差和相关系数的命令。

表 4-31

协方差和相关系数

命令	说明
cov(x)	求向量 x 的协方差
cov(A)	求协方差矩阵,对角线元素是 A 中各列的方差
cov(x,y)	等同于 cov([x y]), x 和 y 是列向量
corrcoef(A)	求相关矩阵
corrcoef(x,y)	等同于 corrcoef([x y]), x 和 y 是列向量

### 例如可以定义如下向量

x=[1;1;1]; y=[1;2;2]; z=[0;-1;1]; 使用下列命令求其方差和协方差 varx=cov(x), vary=cov(y), varz=cov(z),... cvxy=cov(x,y),cvxz=cov(x,z),cvyz=cov(y,z) 回车显示结果 varx = 0 vary = 0.3333 varz = 1 cvxy = 0 0 0 0.3333 cvxz = 0 0 0 1 cvyz = 0.3333 0 1.0000 0

还可以使用 corrcoef 命令来求得相关矩阵。

3.排序

在 MATLAB 中可以用命令 sort 来进行数据排序。表 4-32 列出了常用的排序命令。

表 4-32	
--------	--

### 排序

命令	说明
sort(x)	返回一个向量 x 的元素按递增排序的向量。如果元素是复数 , 则使用绝
	对值进行排序,即sort(abs(x))
[y,ind]=sort(x)	返回下标向量 ind。就是 y=x(ind)。另外向量 y 是 x 中元素按递增排序得 到的
sort(A,dim)	对 A 中各列按递增排序,注意矩阵的行已被改变。如果给出了 dim,则 在 dim 维内进行排序

	命令			说明
	[B, Ir	nd]=sort	(A)	返回矩阵 Ind 和矩阵 B , 矩阵 B 的列为矩阵 A 中按递增排序的列 , 矩阵
				Ind 的每列相对应于上面提到的向量中列 ind
	sortro	ows(X, o	col)	对矩阵 A 的各行按递增排序。如果行的元素是复数,它们以 abs(x)为主,
				以 angle(x)为辅进行排序。如果给出 col,则根据指定的列数对行进行排
				序
例如	存在日	己定义	的矩阵	
A =				
	0	4	4	
	2	0	2	
	4	2	0	
在命	令窗[	コ输入	命令行	
[Asc,	ind]=s	sort(A)	, Des=flipud	d(sort(A))
回车	显示约	吉果		
Asc =	=			
	0	0	0	
	2	2	2	
	4	4	4	
ind =				
	1	2	3	
	2	3	2	
	3	1	1	

4

2

0

flipud 命令使矩阵以递减排序。

4

2

0,

Des =

4

2

0

在本章中,介绍了 MATLAB 数值计算的各个方面:矩阵、多项式、细胞数组、结构数组、稀疏矩阵的生成 和运算、数据的分析和统计。通过本章的学习,可以熟悉 MATLAB 强大的数值计算功能。

4.8 小结

# 第五章 MATLAB 程序设计

MathWorks 公司将 MATLAB 称为第四代编程语言,足见其简洁。MATLAB 的编程效率比 BASIC、C、Fortran 等语言要高级得多,而且容易移植和维护。本章中将系统地介绍如何用 MATLAB 语言编程,主要内容包括

- ➢ MATLAB 的 M 文件
- ➢ MATLAB 中的文件操作
- ▶ 类与对象
- ▶ 时间与日期函数
- ▶ M 文件的优化与调试

# 5.1 M文件

用 MATLAB 语言编写的可在 MATLAB 中运行的程序,称为 M 文件。M 文件包含两类:命令文件和函数文件。两者的区别在于;命令文件没有输入参数,也不返回输出参数,而函数文件可以输入参数,也可返回输出参数;命令文件对工作空间中的变量进行操作,而函数文件的变量为局部变量,只有其输入、输出变量保留在工作空间中。

一般来说,命令文件用于把很多需在命令窗口输入的命令放在一起,以便于修改;而函数文件用于把重复的程序段封装起来,使程序更加简洁。事实上,使用任何文本编辑器都可编写 M 文件,编辑后的 M 文件可以在 MATLAB 的命令窗口中运行。无论是函数文件还是命令文件都可被别的程序调用。

5.1.1 命令文件

由于命令文件没有输入参数和输出参数,只是一些命令行的组合,所以命令文件比函数文件简单。命令文件可对工作空间中的变量进行操作,也可生成新的变量。即使命令文件运行结束,命令文件产生的变量仍将保留在工作空间中,直到关闭 MATLAB 或用相关命令加以删除。

调用命令文件时,不用输入参数,也没有输出参数,文件自身建立需要的变量。当文件执行完毕后,用命 令 whos 可以查看工作空间中的变量,变量仍然保留在工作空间中。

5.1.2 函数文件

和命令文件相比,函数文件稍微复杂一些。下面为一个只有两行的函数文件例子

function c=myfile ( a,b )

c=sqrt ( a+b )

将其保存为 t.m , 一旦该函数文件建立 , 在 MATLAB 的命令窗口中或在别的文件里 , 就可以用下列命令调 用

a=3;

b=4;

c=t ( a,b )

回车显示结果

7

c =

函数文件的结构可分为五部分。下面为一个典型的 MATLAB 的函数文件结构。

function out=ctrltbul ( x )

%Ctrltbu controller for the truck backer-upper when distance is far.

distance=norm (x (1:2));

alpha=acos (x (1) /distance) -pi/2;

tmp=x (3) -pi/2-alpha;

该函数文件包含以下几部分:

- > 函数定义行 文件的第一行为函数定义行,该行定义函数名、输入参数和参数的个数。
- H1 行 H1 行的字面意思为帮助信息的第一行,在上述文件中为第二行。当用命令 lookfor 查询该 函数的帮助信息时,将显示该行内容。
- 帮助体 文件中两空行之间的部分。当对该函数单独查询帮助信息时,将显示帮助体和 H1 行的 内容。
- 函数体 函数文件中除开函数定义行、HI行和帮助体之外的文本,这些命令执行计算、赋值等实 质性的工作。
- ▶ 注释部分 在函数体中以符号 "%"开始直到该行结束的部分表示对程序的注释。

在函数定义行 function out= ctrltbul (x)中, function 为函数文件的关键字,表明该文件为函数文件;out 为输出参数;ctrltbul 为函数名;x为输入变量。MATLAB 的所有函数文件都遵从这种定义格式。当函数的输出参数不只一个时,用中括号"[]"把输出参数括起来,参数之间用逗号分开。如果有输入变量时,用小括号括起来,当个数不只一个时,用逗号分开。function [x,y,z]=sphere (theta, phi, rho)是一个多输出参数、多输入参数的例子。当函数无输出参数时,输出参数项空缺(等号也省略)或者用空的中括号表示。如

function printresults (x)

function []=printresults (x)

当调用函数时,输入参数和输出参数可以和定义行的变量名不同,这正如在 Fortran 里函数调用时,形式参数和实际参数可以不同名一样。

在函数文件中紧接函数定义行的是 H1 行。由于 H1 行为帮助信息,该行以符号"%"开头。当在命令窗口 中键入命令"help 函数文件名"后,显示的帮助信息的第一行为 H1 行,当用命令 lookfor 查询时,只显示 H1 行。所以在编写函数文件时,应该在 H1 行里尽可能地描述该函数的功能。

在命令窗口中键入命令"help函数文件名"后,除了显示 H1行,还显示帮助体中的内容,直到遇到空行或执行命令行。函数文件中第一个空行或可执行命令行后的帮助信息将被忽略。例如,当键入命令 help sin 后将显示

SIN Sine.

SIN (X) is the sine of the elements of X.

Overloaded methods

help sym/sin.m

当用户自己编写的程序较多时,可为某一个子目录下面的所有文件编写一个帮助文件。这个帮助文件必须 以 Contents.m 为文件名并保存在相应的目录下。由于 Contents.m 文件的所有内容都是帮助信息,所有行都以符 号"%"开头。当输入命令"help 子目录"时,将显示该文件的所有内容。如果子目录中没有 Contents.m 文件, 当输入命令"help 子目录"后,将显示该目录下所有文件的 H1 行。

函数体是实现函数功能的主体部分。函数体的命令包括:函数调用语句、程序流程控制语句、交互输入助 出语句、赋值语句、计算语句以及注释语句、空行等。

函数体中的注释语句可以出现在任何地方。MATLAB 把从百分号 "%"开始到该行结束的部分当作注释语句。在函数体中的所有空行被忽略。如前所述,空行可以结束帮助体。

MATLAB 对函数名的限制如同变量名:以字母开头;由字母、下划线和数字组成;只识别函数名的前 31 个字符。函数文件名由函数名再加上后缀".m"组成,例如函数文件名 average.m。当函数文件名和函数定义行 的函数名不同时,MATLAB 将忽略函数名而确认文件名。所以,当编写函数文件时,文件名可以和函数名不同, 但调用时应该使用文件名。不过最好把文件名和函数名统一,以免出错。

函数文件一旦编写好之后,就可在命令窗口或别的文件中调用。调用的格式和函数文件的定义格式相同, 只是把定义时的参数名换成相应的变量名。

下面的命令文件和函数文件完成相同的任务;求所有小于2000 且为2 的整数次幂的正整数。下面为命令文 件的生成过程。 (1)用任何文本编辑器编写以下内容。 t.m %求小于删且为2的整数次幂的正整数。 f (1) =2; k=1; while f ( k ) <1000 f(k+1) = f(k) *2;k=k+1; end f.k (2)保存该文件为 t.m。 (3) 在 MATLAB 命令窗口中输入文件名 t.m 运行结束后可在屏幕上看到以下内容 f =Columns 1 through 6 2 8 16 32 64 4 Columns 7 through 10 128 256 512 1024 k =10 说明; ▶ 符号"%"引导的行是注释行。 ➤ 不需要用 "end" 作为 M 文件的结尾。 ▶ 若用户把 t.m 文件放在自己的工作目录下,那么在运行 t.m 之前,应该先使该目录处于 MATLAB 的搜索路径上。可以用 MATLAB 的路径浏览器把该目录永久地保存在 MATLAB 的搜索路径上, 也可在运行该程序前临时让 MATLAB 搜索该目录,键入 cd c:\mypath 命令(假定 t.m 保存在 c 盘 mypath 目录下)。 ▶ t.m 运行后的变量可用 who 或 whos 命令查看 下面为函数文件的生成过程。 (1) 用任何文本编辑器编写以下内容。 function f=t (n) %求小于任何正整数丛为2的整数次幂的正整数。 f (1) =2; k=1; while f ( k ) <ceil ( n/2 ) f(k+1) = f(k) *2;k=k+1;end f (2)保存该文件为 t.m。 (3)在 MATLAB 命令窗口中键入以下命令,便可求所有小于 2000 的 2 的正整数次幂。 t (2000) 回车显示结果 ans =Columns 1 through 6

2	4	8	16	32	64
Columns 7 through	10				
128	256	512	1024		

- 说明:
  - 文件 t.m 中第一行的作用是指明该文件为函数文件:定义函数名、输入参数和输出参数。函数名可以是任何合法的 MATLAB 变量名。输入和输出参数根据实际情况而定,参数类型可以是数值,也可以是字符串。在本例中,输入参数是 n,输出参数是 f。
  - 变量 k 对函数文件 t 来讲是局部的。当该函数调用结束后,该变量不再存在。如果在调用函数 t 前工作空间中就已存在变量 k,函数调用后,它不会受到影响。
  - ▶ 在 M 文件的头几行带符号"%",为注释行。注释行有两个作用:第一,随 M 文件全部显示或打印时,起注释作用;第二,供 help 命令在线查询。
  - 函数的输入和输出参数可选,用 nargin和。nargout命令可以查看函数调用时的输可以查看函数调用时的输入和输出参数的个数。

下面为在线查询 t 函数。

▶ 在 MATLAB 命令窗口中键入 help t 命令,

help t

求小于任何正整数丛为2的整数次幂的正整数。

▶ 利用 lookfor t 命令在 MATLAB 中对关键字进行搜索,获得帮助信息。

用 help 命令运行后所显示的 M 文件的帮助信息为 M 文件中注释语句的第一个连续块,至于其他 注释语句,将被 MATLAB 的在线帮助系统忽略; lookfor 命令只显示出函数文件的第一行注释

一般来说,为了利用 MATLAB 对关键词的搜索功能,用户在编写 M 文件对应尽量在第一行多包含一些该 文件的信息。

#### 5.1.3 文件的执行

当 MATLAB 遇到一个新的名称时(可能是变量名、函数名、子函数名等),将采用如下的先后顺序来辨别: (1)检查该名称是否为目前工作空间中的变量。

(2)检查该名称是否为子函数名(子函数为函数文件的内部函数)。

(3)检查该名称是否为局部函数名(局部函数放在一个单独的目录下,只有其上面一层目录里的 M 文件 可以调用这些函数)。

(4)检查该名称是否为 MATLAB 搜索路径上的函数名。MATLAB 将把按照搜索路径的先后顺序搜索到的 第一个同名函数当作该名称引用的函数。

当在命令窗口或其他的文件中调用函数文件时,MATLAB 将把函数编成伪码,并保存在内存中,在 MATLAB 关闭之前,该函数的伪码一直保存着,以便再次调用该函数时,不用重新编写伪码。当然可以用 clear 命令清除 内存中的函数伪码。表 5-1 列出了相关命令的功能。

表 5-1

clear 命令的使用

命令	说明
clear 函数名	删除内存中的指定函数伪码
clear function	删除内存中所有的函数伪码
clear all	删除所有函数和变量

无论是函数还是命令文件的伪码,都可以用 pcode 命令保存到文件中,以备下次使用时跳过编写伪码这一步,从而提高速度。一般情况下,MATLAB 编写伪码的速度相当快,是否使用预先编写的伪码对 MATLAB 的运行速度影响不大。当有大量的图形界面时,用 pcode 命令预先编写伪码可以大大提高速度。还有一种情况,当想隐藏程序的源代码时,可以用 pcode 命令提供伪码。

调用 MATLAB 的函数时,对于那些需改变值的参数,将按值传递;在调用过程中值不改变,仅仅是被引用的参数按地址传递,因为这样可以提高内存的利用效率。

在 MATLAB 里,除了一个基本的工作空间外,MATLAB 还为每个被调用的函数分配一部分内存,称为函

127

数工作空间。每个函数有自己的工作空间。在函数外部只能调用函数的输出参数,除输出参数以外的变量都是 不可调用的。函数的工作空间和 MATLAB 基本的工作空间是相互独立的,函数内部的变量和基本工作空间中的 变量是可以向名的。为了能够调用函数内部变量,可以在函数里把该变量定义为全局变量。

### 5.1.4 输入参数和输出参数的检验

函数 nargin 和 nargoutt 分别用于检查函数被调用时的输入参数和输出参数的个数。由于有这两个函数,编 写程序时可以对个向的调用情况做不同的处理,使函数调用更加灵活。例如,下面程序段中指定,当用一个输 入参数调用该函数时,求该参数的平方;当用两个输入参数调用时,求出两个输入参数的和。

function c=testarg1 (a,b)

if (nargin==1)

c=a.^2;

elseif (nargin==2)

c=a+b;

end

函数 varargin 和 varargout 允许用户输入任意多的输入参数和输出参数。MATLAB 将把所有的输入参数或输 出参数组成一个细胞数组。细胞数组是 MATLAB 特有的一种数据类型。细胞数组的各个元素可以是不同的数据 类型,比如第一个元素为 3×3 的数值矩阵,而第二个元素是一个字符串。下面是一个 varargin 函数的使用例子。 该函数在图形上把所有输入参数点用直线连接起来。

```
function testvar (varargin) %函数定义行。
```

```
for I=1:length (varargin) % 对输入参数组成的细胞数组的每一个元素作记号 x (I) =varargin{I} (1); % % 取出输入点的横坐标。
```

y(I)=varargin{I}(2); %取出输入点的纵坐标。

```
end
```

```
xmin=min ( 0, min ( x )) ;
```

ymin=min ( 0,min ( y )) ;

```
axis (xmin fix (max (x)) +3 ymin fix (max (y) +3)) %绘制坐标轴。
```

```
plot ( x,y )
```

这样,调用函数 testcar 时的参数个数不限,下面的调用语句是合法的。

```
testvar ([23],[15],[48],[65],[42],[903])
```

在函数 testvar 的定义中细胞数组的引用采用 varargin{i}(1)格式,其中用大括号{}括起来的下标为细胞数 组的第几个元素,用小括号()括起来的下标为大括号对应的元素的分量。

%绘制图形。

函数 varargout 的应用和函数 varargin 相似,由于调用函数时的输出参数的个数不定,在编程中将用到函数 nargout。例如下列函数将输入的横坐标向量和纵坐标向量的相应元素配对,形成点,输出点坐标。

```
function [varargout]=change ( arrayin )
```

8

```
for I=1:nargout
```

varargout{I}=arrayin ( :,I ) ;

end

当输入三个点的横坐标向量和纵坐标向量组成的矩阵

a =

1 2 3

6 7

## 用命令

```
[p1,p2,p3]=change ( a )
```

```
调用函数,回车显示结果
```

p1 =

1

6 p2 = 2 7 p3 = 3 8 得到三个点的坐标:p1 = (1,6), p2 = (2,7), p3 = (3,8)。

事实上,在函数调用时,细胞数组和普通参数是可以混合使用的,这是应该把 varargin 和 varargout 放到参数列表的最后。

5.1.5 子函数

每一个函数文件可以包含多个函数。在函数文件的第一行定义的函数为主函数,当调用文件名时,调用主 函数。在函数文件中,除了第一行定义的函数外,其余的函数称为子函数。子函数只能被该文件里的其他函数 和主函数调用。

在函数文件里,子函数的定义和主函数相同。子函数的顺序是任意的。下面是一个包含多个子函数的函数 文件,主函数调用两个子函数来完成计算。该文件分为独立的三段,包含一个主函数和两个子函数。主函数 mainfunction 包含定义行和注释行,共五行;子函数 subfunction1 为三行;子函数 subfunction2 为最后九行。

```
function [avg, med]=mainfunction (u)
%该主函数用于求向量的中间值和平均值。
n=length (u);
avg=mean ( u,n ) ;
med=median (u,n);
function a=subfunction (v,n)
%该函数用于求向量的平均值。
a=sum (v) /u;
                        %了函数 subfunction2 的定义行。
function m=subfunction2 (v,n)
%该函数用于求向量的中间位,当向量元素个数为奇数时,中间值为位置上
% 居中的一个元素;当元素个数为偶数时,取位置居中的两个元素的平均值。
w=sort (v);
if rem (n,2) ==1
m=w((n+1)/2);
else
m=(w(n/2)+w(n/2+1))/2;
end
```

当在一个函数文件中调用函数时,MATLAB 首先检查该函数是否为该文件的子函数。然后将检查是否为局 部函数,然后再检查该函数是否为搜索路径上的 M 文件名。由于最先搜索的是子函数名,所以于函数的名字可 以和其他的 M 文件名相同。

放在于目录 private 下的函数称为局部函数。只有 private 子目录的父目录里的函数才能调用这些局部函数。 例如在目录 A 下建立于目录 private,子目录 private 中的函数只能被目录 A 中的函数调用。所有 private 中的函 数都称为局部函数。由于 private 中的函数在 A 目录以外不能被调用,所有 private 中的函数和 A 目录以外的函 数可以同名,而不会互相干扰。

### 5.1.6 外壳函数 (Shell Escape Function)

有时为了利用不同软件的优点,可以把不同的工作交给不同的软件去完成。例如,MATLAB 对于循环语句 计算较慢,而 FORTRAN和C语言对循环计算很快,编程时,如果有大量的循环计算,就可把这部分工作交给 C 或 FORTRAN 语言去执行。MATLAB 提供了和 C 或 FORTRAN 语言交流的功能。实现和其他软件交互的函数称为外壳函数。外壳函数把工作空间中的变量存到磁盘上,再调用其他软件的可备执行程序处理磁盘上的变量,最后把处理过的变量调回工作空间。在 MATLAB 里运行 C 或 FORTRAN 语言的可执行文件,应在文件名前加上感叹号"!"。下面为一个外壳函数的例子。

functioin y=loop ( a,b,c,d )

save date a b c d

!forloop

load data

该函数把 a、b、c、d 四个变量存入文件 data.mat 中,再调用 C 或 FORTRAN 语言的可执行文件 forloop 来 处理这些变量,最后把处理后的变量再调回工作空间。

# 5.2 文件操作

文件是程序设计的一个重要概念。所谓"文件"一般指存储在外部介质上的数据的集合。一般数据是以文件的形式存放在外部介质(如硬盘)上的。操作系统是以文件为单位对数据进行管理的,也就是说,如果想找 到外部介质上的数据,必须先按文件名找到所指定的文件,然后再从该文件中读取数据。要向外部介质上存储 数据也必须先建立一个文件(以文件名标识),才能向它输出数据。

在前面有的章节中,输入和输出有时是从命令窗口中完成的,即用键盘输入数据,结果显示在屏幕上。从 操作系统的角度讲,每一个和主机相连的输入输出设备都看作是一个文件。例如键盘是输入文件,而显示器和 打印机是输出文件。

在程序运行时,常常需要将一些数据(运行的最终结果和一些中间数据)输出到磁盘上存放起来,以后需 要时再从磁盘中输入到计算机内存,这就用到磁盘文件。

和其他高级语言一样,MATLAB 把文件看成字符(字节)的序风根据数据的组织形式可分为 ASCII 文件和 二进制文件。ASCII 文件又称文本(text 文件),它的每一个字节放一个 ASCII 代码,代表一个字符。二进制文 件是把内存中的数据按其在内存中的存储形式原样输出到磁盘上存放。用 ASCII 码形式输出与字符一一对应, 一个字节代表一个字符,因而便于对字符进行处理,也便于输出字符。但一般占存储空间较多.而且要花费转 换时间(二进制形式和 ASCII 码问的转换)。用二进制形式输出数值,可以节省外存空间和转换时间,但一个字 节并不对应一个字符,不能直接输出字符形式。一般中间结果数据需要暂时保存在外存上,以后又需要输入到 内存中,常用二进制保存。表 5-2 列出了 MATLAB 中主要的关于文件操作方面的函数。

表 5-2

#### 文件操作函数

函数分类	函数名	说明
打开和关闭文本文件	fopen	打开文件
	fclose	关闭文件
打开和关闭二进制文件	fread	读二进制文件
	fwrite	写二进制文件
字符串转换	sprintf	把格式数据写入字符串
	sscanf	格式读入字符串
文件定位	feof	检验是否为文件结尾
	fseek	设置文件定位器
	ftell	获取文件定位器
	frewind	返回到文件的开头
临时文件操作	tempdir	获取临时文件目录
	tempname	获取临时文件名
格式 I/O	fscanf	从文件中读格式数据
	fprintf	写格式数据
	fget	从文件中读行,不返回行结束符
	fgets	从文件中读行,返回行结束符

### 5.2.1 文件的打开和关闭

和其他高级语言一样,对文件速写之前应该"打开"该文件,在使用结束后应"关闭"该文件。 函数 fopen 用于打开文件,其调用格式为;

fid=fopen('文件名','使用文件方式')

其中, fid 为调用文件时返回的文件句柄, 文件打开后, 可用句柄值来代替该文件。"使用文件方式"项表 示所打开文件的种类及使用文件的权限。使用文件方式如表 5-3 所示。

当不指定文件的类型时,MATLAB 默认为二进制文件,当打开文本文件时,在文件使用方式后加上't',例 如'rt','wt+'等。当打开文件成功后,将会得到一个位为正整数的句柄。有了句柄,在以后对该文件进行操作时, 用该句柄代替指定的文件。文件打开失败时,返回句柄值-1。例如要打开文件 input.dat,并从中读取数据,命令 为:

fid=fopen('input.dat', 'r')

MATLAB 中的许多 I/O 函数都遵循 ANSI C 标准,但并不是所有 MATLAB 的 I/O 函数都和对应的 C 语言函数相同,使用时,最好查找联机帮助。

当文件打开成功后,返回的文件句柄为非负整数,文件句柄可以作为很多 I/O 函数的输入参数。当文件打 开失败时,除了返回句柄值-1 外,还可以在 fopen 语句中增加一个输出参数来获得错误信息,具体调用格式见 下面的循环语句,但并不是所有的系统都提供错误信息。函数 ferror 也可提供一些错误信息。

表 5-3

文件打开方式

使用方式	说明
'r'	为输入打开一个文本文件
'w'	为输出打开(创建)一个文本文件
'a'	向文本文件尾追加数据(创建)
'r+'	为读/写打开一个文本文件(不创建)
'w+'	为读/写建立一个新的文本文件
'a+'	为读/写打开一个文本文件(可创建)
'W'	写但不自动覆盖
'A'	追加数据但不自动覆盖

每次打开文件时检查文件句柄的值,是一个很好的编程习惯,下面是一个打开文件的循环,当以后打开一 个可读的文件后将跳出循环。

fid=0;

while fid<1

```
filename=input('Open file:','r');
```

if fid==-1

disp(message)

end

end

假定文件名 nofile.m 不存在,而文件 goodfile.m 存在。在调用上述循环语句时,第一次输入文件名 nofile.m, 第二次输入文件名 goodfile.m,在命令窗口中将显示下列信息。

open file: nofile.m

Cannot open file.Existence? Permissions? Memory?

open file: goodfile.m

当读写文件完成后,用命令 fclose 关闭文件。函数 fclose 的调用格式为

status=fclose(fid);

status=fclose(all);

其中第一句关闭句柄为 fid 的文件,第二句将关闭所有文件。当文件关闭成功时,上面两句的 status 的值为 0,当关闭失败时,返回值-1。可见,无论打开文件,还是关闭文件,当不成功时,返回值都为-1。当关闭 MATLAB 时,在 MATLAB 中打开的所有文件都将关闭,但文件使用完后用 fclose 关闭,可以增加系统资源。

值得一提的是,当文件关闭后,文件的句柄值仍然存在,但在其他 I/O 函数中,使用该句柄值将不起作用。

在 MATLAB 中,可以建立临时文件。在有些系统中,每次重启计算机时将输出这些临时文件,而在有些系统中,指定为临时文件可以省下文件备份的空间。

函数 tempdir 将返回系统指定的临时目录。函数 tempname 将给出临时文件目录中的一个文件名。可以在该 目录中存储临时数据。当需要把临时数据存入临时文件时,应该用下列目录建立临时文件。

fid=fopen(tempname,'w')

5.2.2 二进制文件

函数 read 用于从指定的二进制文件中读出所有数据,并把所有的数据存入一个矩阵。fread 把二进制文件的 每一个字节当作矩阵的一个元素。例如下面两句命令将把二进制文件 matrix.dat 中的数据读出,并存为矩阵 A。

fid=fopen('matrix.dat','r');

A=fread(fid);

用函数 fread 可把矩阵 A 中的数据的内容显示为 ASCII 码的形式。例如命令 A=fread(fid,100)将从 fid 所代表的文件中读出 100 个数据,也可以把这 100 个数组成 10 × 10 的矩阵。

可在调用函数 fread 时指定第三个输出参数来控制读出数据的类型。这个参数不但指定在所读文件中一个数据所占的位数, MATLAB 还根据该值把所读的数据翻译为相应的类型。

在一般的软件中, 'char'和'uchar'分别代表有符号和无符号字符,一殷采用 8 位存储; 'short'和'long'分别代表 短整数和长整数,一般分别用 16 位和 32 位存储; 'float'和'double'分别代表单精度和双精度浮点数,在多数系统 中分别采用 32 位和 64 位存储。值得注意的是,在不同的系统中,同一数据类型所采用的存储位数可能不一样。 但函数 fread 提供了很丰富的调用格式,如果调用不同的平台产生的二进制文件,采用既指定类型又指定位数的 调用方法,如'int8'、'float32',将不受平台的限制 c

函数 fwrite 将矩阵中的元素按照指定的数值精度存入二进制文件,并返回存入数据的个数。例如下面的语 句将用 64 个字节来存储 4 阶魔方矩阵,每个元素占用四个字节。

file=fopen('magic4,bina', 'w');

count=fwrite(file,magic(4), 'int32');

status=fclose(file);

如果写文件成功, 变量 count 的值为 16。

当用函数 fopen 打开一个文件时, MATLAB 将用一个定位器来指定文件中的某个特定位置。在该文件中定 位器用来指定文件读写的位置。表 5-4 中列出的为 MATIJLB 提供的关于文件定位的函数。

表 5-4

文件定位函数

	命令	说明
_	feof	检验定位器是否在文件结束位置
_	fseek	指定定位器的位置
	ftell	取出定位器的位置
	frewind	重新设置文件定位器到文件开始位置

函数论 fseek 和 ftell 分别用来设置和查询下一步读写的位置。可以用 fseek 函数跳过某些数据或重复读取某些数据。函数 ftell 将给出定位器相对于文件原始位置的偏移量。函数 fseek 的调用格式为: ,

status=fseek(fid, offset, origin)

其中,fid为文件句柄,offset为可正可负的偏移量,用字节数表示。origin为给定的初始位置。origin的值可以为字符串'cof、'bof'、'eof'之一。其中:

- ➢ 'cof'表示文件中的当前位置;
- ➢ 'bof'表示文件的开始位置;
- ➢ 'eof'表示文件的结束位置。

下面为一个函数 fseek 和 ftell 使用的例子。

A=1:5

fid=fopen('five.bin', 'w'); %打开二进制文件 five.binn。 fwrite(fid, A, 'short'); %把数值 1~5 写入文件,每个数占两个字节。

status=fclose(fid);	%关闭文件。
fid=fopen('five.bin','r');	%重新打开文件 five.binn。
status=fseek(fid, 6, 'bof');	%用文件定位器定位到偏离文件开始位置6个字的地方
four=fread(fid, 1, 'short')	%读出第4个值,值为4。
文件中定位器的位置可用表 5-5	表示。

表 5-5

文件定位器的位置

位置	bof	1	2	3	4		5	6	7	8	9	10	eof		
存储内容		0	1	0	2	0	3	0	4	0	5				
定位器位置															

读取元素 4 后,文件定位器的位置向后移了 2 个字节。可用函数 ftell 指定文件中的当前位置。 position=ftell(fid)

position=

8

这时文件定位器的位置如表 5-6 所示。

表 5-6

文件定位器的位置示意

位置	bof	1	2	3	2	1	5	6	7	8	9	10	eof
存储内容		0	1	0	2	0	3	0	4	0	5		
定位器位置													

调用 fseek 文件可以动文件定位器的位置。例如:

status=fseek(fid, -4, 'cof');

这时文件定位器的位置如表 5-7 所示。

表 5-7

文件定位器的位置示意

位置	bof	1	2	3	4		5	6	7	8	9	10	eof		
存储内容		0	1	0	2	0	3	0	4	0	5				
定位器位置															

再用命令 "three=fread(fid,1,'short');"将读出第三个数值,值为3

### 5.2.3 格式文件

在本小节将介绍怎样读写带格式的 ASCII 码文本文件。

1. 函数 fget1 和 fgets

函数 fget1 和 fgets 可从格式文本文件中读出行并存为字符向量。这两个函数的佛功能基本上差不多。下面的函数将打印出所有包含菜字符串的行。

function y = babycount(filename, baby)

% Search for number of string matches per line.

```
fid = fopen(filename, 'rt'); %以只读方式打开文件。
y=0;
while feof(fid) ==0 %文件定位器不在文件结尾
line = fgetl(fid); %读出定位器所在行
matches = findstr(line, baby); %查找行中是否有字符串"baby" o
num= length(matches); % num 为行中字符串 " bBby " 的个数
if num> 0
y = y +num;
fprinff(1,'%d:%s\n',num, line);
end
```

fclose(fid);

假定文件 myfile 的内容如下:

Oranges and les,

Pineapples and tea.

Orangutans and monkeys,

Dragonflys or fles.

在 myfile.m 中查找字符串'es'的结果如下,行前面的数字表示包含字符串的个数。

litcount('badpoem','an')

2:Oranges and les,

1: Pineapples and tea.

1: Dragontiys or fles.

2. 函数 fscanf

MATLAB 中的函数 fscanf 和 C 语言中的同名函数类似,从格式文本文件中读出一行并赋值给一个或多个变量。无论在 MATLAB 还是 C 语言中,fscanf 都使用格式字符串读出数据。格式字符串都以百分符"%"开始。字符串"%s"表示读出一个字符;"%d"表示以 10 为底的整数;"%g"表示双精度浮点数。下面是 fscanf 的使用例子。

假定文件 input.dat 的内容为:

3.6542345333445678

2.7134314231445

3.45677889897897978

下面三句将读出 input.dat 中的所有四个数,而不需采用循环语句。只要文件中数据的格式和指定的数据格式相同,函数 fscanf 将连续不断地读出数据。这一点是 MATLAB 和 C 语言的不同之处。

fid=fopen('input.dat'. 'r');

Mydata=fscanf(fid, '%g');

status=fclose(fid);

函数 fscanf 可以指定读出数据的个数。下面两句分别从 fid 中读出 100 个整数形成列向量和 10 × 10 的矩阵。 A=fscanf(fid, '%5d', 100);

 $A = fscanf(fid, '\%5d', [10\ 10]);$ 

函数 sscanf 和函数 fscanf 类似,不同之处在于函数 sscanf 从字符串中读出数据,而后者从文件中读出数据。 下面两条命令将从字符串向量中读出 5 和 5²,形成一个列向量。

squre=mum2str([5,5^2]);

rootvalues=sscanf(squre, '% f');

3. 函数 fprintf

函数的 fprintf 把数据转换成字符并写入文件或者屏幕上,可以采用格式控制字符串来控制字符串写出的格式。常用的格式控制有:%e表示写成指数形式;%f表示写成一般的浮点数格式;%g表示自动选择指数或浮点数的简短格式,在格式控制中还可以指定显示数据的宽度。其余的格式控制可参考 C 语言的参考书。下面为一个函数 fprintf 的使用例子。

x = 0:0.1:1;

y=[x:exp(x)];

fid=fopen('exptable', 'w');

fprintf(fid, 'Exponential Function\n\n' );

fprintf(fid, '%6.2 f%12.8\n', y) ;

status=fclose(fid);

%输出一段文字 %第1个数占6位,其中小数点后为2位 %每行两个数之间有两个空格

%第2个数占12位,其中小数点后为8位

从上例可知,在 fprintf 中也可不用循环。和函数 sscanf 对应,函数 sprintf 将数据写入字符串,其调用格式和 sscanf 类似。
# 5.3 类和对象

本节讲解如何通过创建类来增加 MATLAB 的数据类型。同时,还讲述类的实例——对象的创建和操作。 通过创建类和对象,可以增加 MATLAB 的数据类型和操作方法。类定义了一种数据结构以及可用于此种数 据的函数和运算符等操作。对象是类的实例。平时常讲的"面向对象编程(object-oriented programming)"就是指 特别注重类和对象使用的一种编程方法。

5.3.1 类和对象概述

MATLAB 本身提供了五种类:double、sparse、char、struct 和 cell,分别表示双精度类、二维双精度复数稀 疏类、字符类、结构类和细胞类。除了这五种基本的类以外,MATLAB 的工具箱还创建了很多类。例如 MATLAB 为积分和微分方程创建的 inline 类、符号计算工具箱中创建的 sym 类、控制系统工具箱中用于分析线性时不变 系统的 Iti 类和三个子类。

通过建立一个具体的数据结构,并创建一个子目录(称为类的子目录),用于存储所有定义该数据结构操作和运算方法的 M 文件,就可创建新的类。类的子目录中包括定义各种数学运算、下标、组合等的 M 文件。定义 MATLAB 提供的操作符称为字符的"超载"。

在 MATLAB 中,使用变量时不用预先定义,例如

A = zeros(10, 10)

s = 'Hello world'

上面的两行命令分别形成一个数值矩阵和一个字符串。这个数值矩阵是双精度类的一个实例,字符串是字 符类的一个实例,称为对象。对于用户定义的类,使用前也不需先定义,类的构造程序将自动生成对象。在本 节中,将用单变量多项式类作为例子。多项式类将生成真正的多项式表达式。单变量多项式类和类的构造程序 的名字都为叫 polynom,当用多项式的系数向量调用函数 polynom 时, polynom 构造程序将自动生成多项式,如

p=polynom([1 0 3 -6])

该命令将生成多项式  $p(x) = x^3 + 3x - 6$ 。

在 MATLAB 里,定义类的方法的所有 M 文件放在一个目录里,目录名为字符"@"加上类名,例如定义 多项式类的所有 M 文件放在目录@polynom 下。

虽然在路径浏览器中没有类的目录,事实上类的目录处于搜索之列。例如@inline 是 toolbox\MATLAB\funfun的子目录,@sym 是 toolbox\symbolic 的子目录,而 toolbox\MATLAB \funfun和 toolbox\symbolic 都是搜索目录, MATLAB 将自动搜索这些目录的子目录。

1. 类的数据结构

选择类的数据结构是定义类的第一步。在 MATLAB 里,对象是用结构数组来存储的。对于结构的属性,以 及对于属性运算的细节,在使用对象时是不可见的,而只能在定义类的子目录中加以修改。

对于多项式类,每个对象(即每个实际的多项式)用降序排列的系数行向量来表示。每个多项式对象都是只有 一个属性的结构,这个属性的值即系数向量。要改变多项式的这个属性,只能修改@polynom 目录下定义方法的 M文件。

MATLAB 采用降序排列的系数行向量来定义多项式并不是表示多项式唯一的方法。事实上,也可采用降序 排列系数行向量,也可用列向量来表示多项式。对于多项式这类简单的类,实现各种表示方法都不很困难,在 这种情况下,表达方式的选择也不很重要。但对于很复杂的对象,数据结构的设计非常关键。例如,对于稀疏 矩阵的数据结构的选择,将大大地影响矩阵的运行速度。

2. 类的构造函数

在定义类方法的子目录中,必须有一个称为构造程序的 M 文件。M 文件和类同名,也就是子目录名去掉字符"@"。构造程序通过初始化类的数据结构和给类分配标号来定义类。

下面是多项式类的构造程序 polynom.m 的内容:

function p=polynom(a)

%POLYNOM Polynomial class constructor .

p = POLYNOM(v) creates a polynomial object from the vector v,

%containing the coefficients of descending powers of x

if nargin==0

p.c=[]; p=class(p, 'polynom'); elseif isa(a, 'polynom') p=a; else

else

p.c=a(:).';

p=calss(p, 'polynom');

end

当采用无输入参数的格式调用函数 polynom 时,构造程序将为该对象生成一个模板,通常情况下属性值为空;当输入参数为多项式时,构造程序将原封不动地返回该多项式(用函数 isa 判断);当输入参数不是多项式时,构造程序将把输入参数变形为行向量并赋给属性 c,最后,用函数 class 加上标号 "polynom",表明该对象为多项式。

一般来说,类的构造函数都采用 polynom 的框架,大致包含以下几个部分:

- 当无输入参数调用时,返回一个模板对象;
- 当输入参数为该类的对象时,无变化返回:
- 把输入参数转换为合适酌格式;
- ➢ 该结构的属性赋值;
- ▶ 用函数 class 加标号。
- 3. 函数 class 和 isa

函数 classs 和 isa 除了用于类的构造函数所在于目录以外,还可在通常的 M 文件和命令行中使用,格式为 isa(a, 'class_name')

上述表达式检查变量 a 是否为 calss_name 指定的对象。下面的表达式都为真

isa(pi, 'double')

isa('hello', 'char')

isa(p, 'polynom')

上述表达式将返回一个字符串,字符串表示该变量所属的类。例如

class(pi)

ans=

'double'

class('hello')

ans=

'char'

class(p)

ans=

'polynom'

4.转换函数

在 MATLAB 中, 类的转换函数用于不同类之间的转换。其调用格式为

b=class_name(a)

其中, a 是某个类的对象。上面的命令将运行类的子目录中名为" class_name"的函数。当输入参数 a 为 " class_name"所指定的类的对象时, 一般情况下, 将原封不动地返回输入参数。

MATLAB 中, double 和 char 是最重要的两个转换函数。double 把其他的类转换为双精度。char 把其他的类转换为字符。

把多项式类转换为双精度的文件为于目录@polynom下的 double.m,其内容为

136

```
function c=double(p)
    %POLYNOM/DOUBLE Convert polynom object to coefficient vector .
    %c=DOUBLE(p) converts a polynomial object to the vector c
    % containing the coefficients of descending power of x.
    c=p.c;
    在该文件中,只有一句可执行语句,用于取出多项式的属性 c。该文件把多项式的系数按降序排列,形成多
项式的系数行向量。
    由于涉及到变量的幂,对于字符的转换是一个比较复杂的方法。文件@opaque/char.m的内容为;
    function chr = char(opaque_array)
    %CHAR Convert an OPAQUE object to CHAR
    %
         Chip Nylander, June 1998
    %
         Copyright 1984-2000 The MathWorks, Inc.
    %
         $Revision: 1.8 $ $Date: 2000/08/01 15:16:55 $
    % For opaque types other than those programmed here, just run the default
    % builtin char function.
    if ~isjava(opaque_array)
        chr = builtin('char', opaque_array);
        return;
    end
    % Convert opaque array to cell array to get the items in it.
    err = 0;
    try
      cel = cell(opaque_array);
    catch
      err = 1:
    end
    if err
      chr = ";
      return;
    end
    % A java.lang.String object becomes a char array.
    if isa(opaque_array,'java.lang.String')
      chr = cel\{1\};
      return:
    end
    % An empty Java array becomes an empty char array.
    sz = builtin('size', cel);
    psz = prod(sz);
    if psz == 0
      try
        chr = reshape(",size(cel));
      catch
        chr = ";
      end
      return;
```

```
end;
```

```
% A java.lang.String array becomes a char array.
    chr = cell(sz);
    for i=1:psz
      chr{i} = ";
    end
    t = opaque_array(1);
    c = class(t);
    while ~isempty(findstr(c,'[]'))
      t = t(1);
      c = class(t);
    end
    if psz == 1 \& ischar(t) \& size(t,1) == 1
      chr = t;
      return;
    end
    if isa(t,'java.lang.String')
      chr = char(cel);
      return;
    end
    % Run toChar on each Java object in the MATLAB array. This will error
    % out if a toChar method is not available for the Java class of the object.
    % A scalar array becomes a single char array.
    if psz == 1
      if ~isjava(opaque_array(1))
        chr = builtin('char',opaque_array(1));
      else
        chr = javaMethod('toChar', opaque_array(1));
      end
    else
      for i = 1:psz;
        if ~isjava(cel{i})
           chr{i} = builtin('char',cel{i});
        else
           chr{i} = javaMethod('toChar', cel{i});
        end;
      end;
    end;
    chr=char(chr);
    5. 输出显示
    当生成某对象的赋值语句的末尾没有分号时,将调用一个称为 display 的方法。对于很多类,函数 display
只能输出变量名,其余部分由转换函数实现。下面为@inline 目录下的 display.m 文件的内容,对于其他的类,函
数 display 的内容差不多。
```

function display(obj)

%DISPLAY Display an INLINE object.

- % Steven L. Eddins, August 1995
- % Copyright 1984-2000 The MathWorks, Inc.

```
$Revision: 1.6 $$Date: 2000/06/02 00:11:15 $
%
isLoose = strcmp(get(0,'FormatSpacing'),'loose');
line1 = sprintf('%s =', inputname(1));
if (obj.isEmpty)
  line2 = 'Inline function (empty)';
else
  line2 = sprintf('Inline function:\n
                                          %s(', inputname(1));
  for k = 1:(obj.numArgs-1)
     line2 = sprintf('%s%s,', line2, deblank(obj.args(k,:)));
  end
  line2 = sprintf('%s%s)', line2, deblank(obj.args(obj.numArgs,:)));
  line2 = sprintf('%s = %s', line2, obj.expr);
end
if (isLoose)
  fprintf('\n');
end
fprintf('%s\n', line1);
if (isLoose)
  fprintf('\n');
end
fprintf('
              %s(n', line2);
if (isLoose)
  fprintf('\n');
end
```

```
5.3.2 运算符超载
```

在很多情况下,希望常用的运算符表示复杂的运算,称为运算符的超载。通过编写相应的 M 文件可以实现 运算符的超载。在超载情况下,运算符可对不同的数据类型进行运算。

1.算术符的超载

MATLAB 的所有指定运算符都有一个相应的文件来实现运算符的功能。用户可以编写自己的文件对运算符 定义算法。例如,当 p和 q 中有一个是多项式时,下面的表达式将调用@polynom 目录下的 plus.m 文件, plus.m 实现加号对于多项式的超载。

```
p+q
```

plus.m 文件的内容为

```
function r= plus(p,q)
```

% POLYNOM/PLUS Implement p + q for polynoms.

p=polynom(p); %把输入参数转换为多项式

q=polynom(q);

k= length(q.c) - length(p.c);

r = polynom([zeros(1,k) p.c] + [zeros(1,-k) q.c];

在函数中先将输入参数转换为多项式,这使得在输入参数中一个为双精度型、一个为多项式的情况下,函 数能正常运行。如

p+1

其次,把两个多项式的系数向量相加,生成两个多项式的和。

下面为两个多项式的积的例子。函数 mtimes 用于计算两个多项式的积 , 如 p*q , 这是对 MATLAB 定义的矩 阵相乘的超载。两个多项式的积的运算转化为多项式的系数向量的卷积。

```
function r = mtimes(p,q)
% POLYNOM/MTIMES Implement p*q for polynoms.
p = polynom(p);
q = polynom(q);
r = polynom(conv(p.c,q.c));
下面为这两个函数的计算实例
p = polynom([10-2-5]);
q=p+1
q=
x^3 - 2*x - 4
r = p*q
r=
```

x^6 - 4*x^4 - 9'x^3 + 4*x^2 + 18*x + 20

表 5-6 列出了定义 MATLAB 的绝大部分运算符的文件名。在创建类时,可参考相应的文件。

表	5-6	

运算符相关文件

±\+	也亡之类的日本书	12 AD
表达式	相应函数调用格式	
a+b	plus(a,b)	两个运算数相加
a-b	minus(a,b)	两个运算数相减
-a	uminus(a)	单运算数减
+a	uplus(a)	单运算数加
a.*b	times(a,b)	对元素乘
a*b	mtimes(a,b)	矩阵乘
a./b	rdivide(a,b)	对元素右除
a.\b	mrdivide(a,b)	对元素右左除
a/b	ldivide(a,b)	矩阵右除
a\b	mldivide(a,b)	矩阵左除
a.^b	power(a,b)	对元素求幂
a^b	mpower(a,b)	矩阵幂
a <b< td=""><td>lt(a,b)</td><td>小于</td></b<>	lt(a,b)	小于
a>b	gt(a,b)	大于
a<=b	le(a,b)	小于等于
a>=b	ge(a,b)	大于等于
a~=b	ne(a,b)	不等于
a==b	eq(a,b)	等于
a&b	and(a,b)	逻辑与
a b	or(a,b)	逻辑或
~a	not(a)	逻辑非
a:d:b	colon(a,d,b)	冒号运算
a:b	colon(a,b)	
a'	ctranspose(a)	复共轭转置
a.'	transpose(a)	普通转置
命令窗口输出	display(a)	显示信息
[a b]	horzcat(a,b)	横向组合
[a;b]	vertcat(a,b)	纵向组合
a(s1,s2,sn)	subsref(a,s)	下标引用
a(s1,s2,sn)=b	subsasgn(a,s,b)	对数组局部赋值
b(a)	subsindex(a,b)	下标引用

### 2. 函数的超载

通过定义和已有函数相同的文件名可以超载函数,该文件需放在类的子目录下。调用某个函数时,最先搜索的目录为输入参数所属类的于目录。比如,现已建立一个新类 myclass,要对函数 plot 超载,只需把编写的 plot.m

文件放入 myclass 类所在的子目录即可。 在 MATLAB 中,有几个对多项式的系数向量运算的函数,在超载这些函数以用于新的多项式类时,可以调 用 MATLAB 定义的初始函数。例如在新建多项式类的子日录中建立下述文件@polynom/roots.m。 function r=roots(p) %POLYNOM/ROOTS.ROOTS(p) is a vector containing the roots of p. r=roots(p.c) 在文件中,调用了 MATLAB 的 roots 函数。当调用该函数时,也将给出多项式的根。 p=polynom([1 0 -2 -5]); roots(p) ans= 2.0946 -1.0473+1.1359i -1.0473-1.1359i 函数 polyval 用于求多项式在给定的多个点上的值。下面为@polynom/polyval 的内容 function y=polyval(p,x) %POLYNOM/POLYVAL POLYVAL(p,) evaluates p at the points x. y=0; for a=p.c y=y.*x+a; end 函数 polyval 和 roots 都用于下面的超载函数 plot 中。该函数将在多项式根的绝对值中的最大值和最小值之 间画多项式曲线。@polynom/plot.m 文件的内容为 function plot(p) %POLYNOM/PLOT PLOT(p) plots the polynom.p. r=max(abs(roots(p))); x=(-1.0:.01:1.0)*r; y=polyval(p,x); plot(x,y); title(char(p)) grid on 下面的程序@polynom/diff.m 将对多项式求导。只要把多项式的系数向量点与逆序下标向量相乘,即可对多 项式求导 function q=diff(p) %POLYNOM/DIFF DIFF(p) is the derivative of the polynom p. c=p.c; d=length(c)-1;%degree q=polynom(p.c(1:d).*(d:-1:1)); 函数 methods 用于显示某个类的所有方法,其调用方法有以下两种 methods('类名') methods 类名 对于 MATLAB 提供的细胞数组,其结果为 methods('cell') 回车显示结果

Methods for class cell:

deblank	ismembe	er setxor	strcat	union
intersect	setdiff	sort	strmatch	unique

#### 5.3.3 对象优先级

默认情况下,MATLAB认为所有的对象都处于向一优先级,运算时采用表达式最左边的对象所属类的方法。 通过定义对象的优先级,可以控制运算符所采用的方法。在设置对象的优先级中,inferiorto和 superiorto是两个 很常用的函数。函数。superiorto把一个对象置于另一个对象更高的优先级。函数 inferiorto把一个对象置于比另 一个对象更低的优先级。假如新建一个类,名为 rational,为了实现 rational 对象和 polyfun 的对象的混合运算, 不需改变 polyfun 类的方法,而只需在 rational 类的构造程序中加入以下语句

superiorto('polyfun')

该语句把 rational 设为比 polyfun 类高一个优先级,这样表达式 p+r, r+p, P*r, r*P

等(其中 p 为多项式, r 为 rational 的对象)将采用 ratioanl 类的方法。

在 MATLAB 中,经常会遇到对象和数组,正确的设置对象和数组的关系操作过程。下面对著名的 Chebyshev 多项式序列提出三种解决方案。

下面三行程序定义了比 Chebyshev 多项式序列:

ro(X) = 1

71(X) = x

7 " * , N = 2xX(1)—yodN , 其中 M , 1。

至少有三种方法可以生成 chebyshev 多项式序列:

把多项式对象的属性设置为数组。前面定义的多项式类有一个属性为数组,数组表示多项式的系数向量。 由于每个多项式的 c 属性为一个向量,当把多项式的属性设置为 cheby6h 则多项式序列时,这时的多项式对象 的属性为一个二维数组。

采用这种设置时,所有定义在多项式类上的方法将对一个二维数组操作,这并不是求 chebyshev 多项式序列的最好方法。

把多项式序列定义为数组。这时,多项式对象必须是一个结构数组。在这种情况下,所有定义在多项式类上的方法将采用 p(k)的方式引用序列中的某个多项式,每个多项式又有存储系数向量的属性,所以,当方法对 某个多项式的系数向量操作时,必须采用 PQ) '的引用格式。由于 chebyshev 多项式序列采用循环的方式。

#### 5.3.4 类和对象的继承

如前所述,MATLAB 中的对象都是结构,这使得对象的继承很容易实现。继承指一种对象可获得其他类的 对象的属性。当一个对象(子对象)继承另一个对象(父对象)时,子对象将继承父对象的所有属性,能够调用父对 象的所有方法。继承是面向对象编程的一个主要特点。通过继承,子对象可以利用父对象的代码。继承可分为 两类:单继承和多继承。在单继承中,每个子对象只能继承一个父类。在多继承中,一个子对象可以继承多个 父类的属性。在这部分里,还要介绍聚积的概念。通过聚积,一个对象可以把另一个对象当作其中的一个属性。

1. 单继承

当—个类只继承另—个类的属性时,采用单继承。单继承中,了类除了继承父类的属性外,还可添加自己的属性。于对象可以调用父类的方法,但父类的对象不能调用子类的方法。

子类的构造函数有两个主要特点

子类的构造函数通过调用父类的构造函数来生成于类的继承属性;

构造函数的语法和父类的构造函数稍有不同,以示区别父类和子类。

控制系统工具箱中用于分析线型时不变系统的 tf, zpk、ss(子类)和 lti(父类)采用单继承。其中 tf, zpk、ss 分别表示系统的转移函数、零极点和状态空间。父类 Iti 只包含一些通用的属性,如系统是离散的还是连续的、 抽样间隔、输入和输出的名称等;tf 类的对象则包含系统转移函数的分子多项式和分母多项式的系数向量;zpk 对象包含系统的极点和零点;ss 对象包含四个描述状态空间的矩阵。

表达式 L=lti(1,1)只能建立一个系统的框架,生成一个连续时间的没有输入和输出的系统。

表达式 T=tf(1, [10-3-5])建立一个转移函数对象,该对象表示的系统为连续系统,系统的转移函数为

 $\frac{1}{s^3 - 3s - 5}$ 

对象 T 有四个属性

➤ T.m 为转移函数的分子,为1;

➤ T.den 为转移函数的分母,为[10-3-5];

➤ T.variable 为转移因数的变量,为"s";

▶ T.lti 为从父类 lti 处继承的属性。

继承的机制是通过于对象的构造函数来实现的。子类 tf 的构造函数中包含下面的语句 L=iti(Ny,Nu,Ts)该语 句将建立一个带某些参数的 lti 对象,然后用下面的语句建立 tf 对象。

sys=class(sys, 'tf', L)

在这个语句中,既指定对象 sys 的类标号,又指明该对象继承前面建立的父对象 L 的属性。事实上,继承可以跨越多代,即一个子对象既可继承父对象的属性,也可继承祖父对象的属性,甚至更前代的属性。子对象 不仅可以调用父对象的方法,还可调用更前几代的方法。

2. 多继承

在多继承中,一个子对象可以继承多个父对象的属性,同时还可建立自己的属性。和单继承类似,多继承 也可跨代继承,即一个子对象可继承多个父对象的属性,而每个父对象又有多个祖父对象,子对象可继承所有 沮父对象的属性。多继承也是通过在类的构造函数中调用 class 函数实现的。其调用格式为

obj=class(structure, 'class_name', parent1, parent2....)

多继承时,有可能某两个父对象的属性同名,这时子对象将选用在上述公式中最先出现的父对象的属性。 除了标准的继承外 MATLAB 还支持对象的嵌套,即把一个对象当作另一个对象的一个属性。

3. 方法调用

当使用对象和方法时,MATLAB采用一套特殊的规则来保证调用指定的函数,而不致产生混乱。对于运算数中有多个对象的表达式,当所有的对象为同一优先级时,将调用所有对象中最左端的对象所属类的相应方法。 当对象不为同一优先级时,再从左到石搜索优先级最高的对象。并采用下面的规则来选择方法

(1)检查在该类中是否有所调用函数的超载函数,如果没有,再检查该类的父类中是否有该函数的超载函数.继续往上检查所有的父类、祖父类、...当找不到超载函数时,进入第二步。

(2)如果调用的函数名和某个类的目录同名,MATLAB 首先检查该函数是否为转换函如果是,将调用该转换函数;如果不是,将调用类的构造函数,进而转入第三步。

(3) 按照先后顺序执行下面四步

▶ 该类中是否有同名的方法,如果有,调用;如果无,进入下一步;

▶ 该类的父类中是否有同名的方法,如果有,调用;如果无,进入下一步;

➢ 在 MATLAB 的搜索路径中是否有同名的函数,如果有,调用;如果无,进入下一步;

▶ 返回出错信息。

4. 私有目录和对象调试

在 MATLAB 中,每个对象允许有自己的私有目录。在私有目录中,可以包含私有函数和私有方法。私有函数只在该类中有效,私有函数并不对对象运算,但能执行一些普通的运算。私有方法只对本类的对象有效。和 其他私有目录的创建相同,只需建立一个名为"Private"的子目录就可以了。

和 M 文件类似,对象和类也可使用 MATLAB 的调试命令。调试时,类名应放在方法名的前面,如 dbstop class/method

# 5.4 程序的优化

这一节将介绍一些提高 M 文件执行速度,优化内存管理的方法,包括循环的向量化和向量的预分配。

5.4.1 循环的向量化

前面提过,MATLAB的一个缺点是当对矩阵的单个元素作循环时运算速度很慢。编程时,把循环向量化, 不但能缩短程序的长度,更能提高程序的执行效率。由于 MATLAB 的基本数据为矩阵和向量,所以编程时,应 尽量对向量和矩阵编程,而不是像在其他应用程序里,对矩阵的元素编程。下面是一个向量化编程的例子。

%一般循环编程

x=1;

- for k=1:1001
- y(k)=log10(x);
- x=x+0.01;

end

%向量化编程。

MATLAB 提供的 repmat 函数是一个极好的向量化编程的例子。该函数把输入矩阵在横向和纵向重复。下面 是一个调用函数 repmat 的例子,对于矩阵

 $\begin{array}{ccc} A = [ \\ 1 & 2 & 3 ; \\ 4 & 5 & 6 ] ; \end{array}$ 

在命令窗口输入

B=repmat(A,2,3)

回车显示结果

B =

2	3	1	2	3	1	2	3
5	6	4	5	6	4	5	6
2	3	1	2	3	1	2	3
5	6	4	5	6	4	5	6
	2 5 2 5	2 3 5 6 2 3 5 6	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$

函数 repmat 的代码为

```
function B = repmat(A,M,N)
```

```
% REPMAT Replicate and tile an array.
% B = repmat(A,M,N) creates a large matrix B consisting of an M-by-N
% tiling of copies of A.
```

```
% B = REPMAT(A,[M N]) accomplishes the same result as repmat(A,M,N).
```

```
% B = REPMAT(A,[M N P ...]) tiles the array A to produce a
```

```
% M-by-N-by-P-by-... block array. A can be N-D.
```

```
%
```

%

% REPMAT(A,M,N) when A is a scalar is commonly used to produce

```
% an M-by-N matrix filled with A's value. This can be much faster
```

% than A*ONES(M,N) when M and/or N are large.

%

% Example:

% %

```
repmat(magic(2),2,3)
repmat(NaN,2,3)
```

%

```
% See also MESHGRID.
```

```
$Revision: 1.15 $ $Date: 2000/06/01 16:46:52 $
%
if nargin < 2
                                       %当输入参数少于两个时,终止程序
   error('Requires at least 2 inputs.')
                                       %当输入参数为两个时,
elseif nargin == 2
                                       %横向重复次数等于纵向重复次数
  if length(M) == 1
     siz = [M M];
  else
     siz = M;
  end
else
                                       %输入参数为三个时, 取两个方向的重复次数
  siz = [M N];
end
if length(A) == 1
  nelems = prod(siz);
  if nelems>0
    % Since B doesn't exist, the first statement creates a B with
    % the right size and type. Then use scalar expansion to
    % fill the array.. Finally reshape to the specified size.
    B(nelems) = A;
    B(:) = A;
    B = reshape(B,siz);
  else
    B = A(ones(siz));
  end
elseif ndims(A)==2 & length(siz)==2
  [m,n] = size(A);
  mind = (1:m)';
                                       %生成列向量[1:2]
                                       %生成列向量[1:2:3]
  nind = (1:n)';
  mind = mind(:,ones(1,siz(1)));
                                       %生成矩阵[11;22]
                                       %生成矩阵[111222333]
  nind = nind(:,ones(1,siz(2)));
  B = A(mind,nind);
  Asiz = size(A);
  Asiz = [Asiz ones(1,length(siz)-length(Asiz))];
  siz = [siz ones(1,length(Asiz)-length(siz))];
  for i=length(Asiz):-1:1
    ind = (1:Asiz(i))';
    subs{i} = ind(:,ones(1,siz(i)));
  end
  B = A(subs\{:\});
end
最后几句的执行过程为:把矩阵 mind 和 nind 按列优先重组为列向量,然后取 mind 的每个元素和 nind 中的
```

所有元素配对形成横、纵下标,取出 A 中相应元素作为 B 的元素。例如在计算的开始,取 mind 的第一个元素 1, 分别和 nind 中的元素形成元素下标(1,1)、(1,2)、(1,3)、(1,1)1(1,2).....(1,3),在 A 中取这九个下标对应的元素, 形成矩阵 B 的第一行,然后再取 mind 的第二个元素和 nind 的所有元素配对,形成矩阵 B 的第 2 行.....。可以 想象,如果对每个元素循环编程,将使用很长的代码。

#### 5.4.2 数组的预定维

数组的预定维可以提高程序的执行效率。由于在 MATLAB 里,变量使用之前不用定义和指定维数,如果未预定义数组,每当新赋值的元素的下标超出向量的维数时,MATLAB 就为该数组扩维一次,这样大大降低了程序的执行效率。另外,数组的预定维可以提高内存的使用效率。不使用预定维时,数组的多次扩维会增加内存的碎片。下面的例子中,没有预定维的程序段每执行一次循环,r向量的大小就增1,在计算机上计算时间将比预定维的程序执行时间长。

%矩阵未予定位编程 for n=1:32 r(n)=rank(magic(n)); end %矩阵预定维编程 r=zeros(32,1) for n=1:32

```
r(n)=rank(magic(n));
```

```
end
```

数值数组的预定维一般采用 zeros 函数;细胞数组预定维采用 cell 函数;结构型数组用 struct 函数预定维。 下面为用这三个函数预定维的例子。

%数值数组预定维。

y=zeros(1,100) for i=1:100 y(i)=det(X^i); end %细胞数组预定维。 B=cell(2,3) B{1,3}=1:3; B{2,2}='string'; %结构数组预定维。 data=struct([1 3], 'x', [1 3], 'y', [5 6]) data(3).x=[9 0 2]; data(3).y=[5 6 7];

#### 5.4.3 内存的管理

MATLAB 提供了五个用于提高内存工作效率的函数

- ➢ clear 命令从内存中删除变量;
- > pack 命令把内存中的变量存入磁盘,再用内存中的连续空间载回这些变量;
- ▶ quit 命令将退出 MATLAB,把 MATLAB 占用的内存全部交还系统;
- ➢ save 命令把指定的变量存入磁盘;
- ▶ load 命令把 save 代命令存入酌变量载入内存。

在循环内部和 M 文件里,最好不要使用 pack 命令,因为这将影响运行速度。值得一提的是,由于命令 save 和命令 load 存取的文件为二进制文件,所以其存取速度比低级的文件输入、输出命令执行效率更高。

在一些系统中,可用命令 whos 来显示剩余内存的大小。但是,如果变量不处于高端内存,删除该变量后, 用命令 whos 显示的剩余内存并不增多,这意味着实际可用内存可能比 whos 显示的多。对于使用虚拟内存(在硬 盘上划定的一部分空间,当内存不足时,内存和该空间可交换数据)的系统,由了硬件对内存限制不大,MATLAB 不受剩余内存多少的限制。

在 MATLAB 启动时,将在内存中为所有在 matlab\toolbox 子目录下的 M 文件和 MEX 文文件建立一个列表。 除非用 path 命令建立一个新列表,这个列表将一直保留在内存中。当某个 M 文件或 MEX 文件被调用时 MATLAB 再将该文件的代码载入内存。在下列情况下,这些文件代码将被从内存中删除。

当该文件被重新调用,在内存中生成了新的代码;

➢ 当所调用的函数被命令 clear 凹或 clear functions 从内存中删除时;

▶ 当 MATLAB 内存不足时。

函数的镶嵌调用并不能减少所需内存空间。例如下面的第一局命令和连续的第二、第三句命令占用的空间 是一样的。

result=function2(function1(input0);

result=function1(input);

result=function2(result);

当被赋值的变量在内存中不存在时,MATLAB 将给该变量分配相应的内存空间;当变量所赋值的元素不存 在时,MATLAB 会自动为该变量扩维,并分配相应内存。

为了节省内存,在编程时,应尽量避免生成大的临时变量,当临时变量不再有用时,尽快删除;当变量不 再使用时,可把变量置为空矩阵或用 clear 彻底删除;尽量重复使用变量名而减少变量的数目。

在 MATLAB 里,定义变量为全局变量并不多占内存,当使用" clear 变量名"命令时,可把该全局变量从 当前的工作空间中删除,但在其他定义该变量的函数中仍可使用,为了彻底删除该变量,可佼用命令" clear global 变量名"。

在 Pc 机上,MATLAB 不提供特殊的函数用于管理系统资源。为了提高 MATLAB 的运行速度,使用时,可 以关掉非活动窗口,井把其他活动窗口缩为图标。另外,在系统中使用永久性的交换文件比临时交换文件更快。 一般情况下,把交换文件的大小设置为内存的两倍足矣。

## 5.5 时间与日期函数

在 MATLAB 编程时,有些情况下需要在生成的文件中加入日期,或需要计算程序的执行时间,这都可以由 MATLAB 提供的日期和时间处理函数来完成。

MATLAB 提供的日期和时间处理函数在 MATLAB 的安装目录下的 toolbox\MATLAB\TIMEFUN 目录中。表 5-8 列出了这些函数及其功能。

表 5-8

#### 日期和时间函数

函数	说明
now	获取当前日期和时间,用连续日期数表示
date	获职当丽日期,用日期子符串表示
clock	获取当前日期和时间,用日期向量表示
datenum	转换为连续日期数
datestr	转换为日期字符串
datevec	转换为日期向量
calendar	日历
weekday	星期几
eomday	指定月的最后一天
datetick	日期格式的刻度标注
cputime	MATLAB 启动以来所使用的 CPU 时间,单位是秒
tic	秒表
toc	
etime	经过的时间

#### 5.5.1 日期的获取和表示

MATLAB 表示日期有三种基本格式:日期字符串(date strings),连续日期数(serial date numbers),日期向量 (date vectors)。

比如,当前日期是1999年8月1日,在 MATLAB 命令窗口分别输入三个获取日期的命令 date、now、clock, 它们返回的结果分别为上述的三种日期格式,用户可以通过返回值比较三种格式的区别 date ans = 23-April-2001 now ans = 7.3103e+005 clock ans = 1.0e+003 *

2.0010 0.0060 0.0230 0.0090 0.0280

函数 date 返回的是日期字符串,使用字符串格式直观的表示的年月日; now 返回的时连续日期数,它是从 公元元年第一天开始到当前日期的累计天数;而 clock 返回的时日期向量,也就是当前的[年 月 日 时 分 秒] 的数值。

0.0129

用日期字符串格式表示日期还包括 19 种不同的格式,各种格式都有相应的序号,序号从 0 到 18,如标 5-9 所示。

表 5-9

日期字符串格式

序号	格式	说明
0	01-Aug-1999 14:45:50	日-月—年时:分:秒
1	01-Aug-1999	日-月—年
2	08 / 01 / 99	月/日/年
3	Aug	月,'个字母
4	А	月,一个字母
5	8	月
6	08/01	月/日
7	1	E
8	Sun	早期门-二个字母
9	S	星期几 , 一个字母
10	1999	年,四位数
11	99	年,两位数
12	Apr99	月年
13	14 : 45 : 50	时:分:秒
14	02:45:50PM	时 :分 :秒 AM(上午)或 PM(下午)
15	14:45	时:分
16	02:45PM	时:分AM(上午)或PM(下午)
17	Q3-99	季度-年
18	Q3	季度

#### 5.5.2 日期的转换

表示日期的各种基本格式可以使用函数 datenum, datestr, datevec 进行相互转换。函数 datestr 还可以在日期字符串的 19 种日期字符串各时间进行转换。

1. datenum 函数

函数 datenum 的调用格式:

(1)n=datenum(s):把字符串 s 转换为连续日期数。其中字符串 s 必须是日期字符串格式列表中第 0、1、2、 6、13、14、15、16 种格式之一。字符串中用两位数表示年时,指的是一当前年为中心的一百年内的年份。

(2) n=datenum(y,m,d):数 y、m、d 分别指年、月、日,可以是标量,也可以是维数相同的数组。

(3) =datenum(y, m, d, h, mi, s): 参数 y、m、d、h、mi、s 分别指年、月、日、时、分、秒, 可以使 标量或维数相同的数组。

(4) n=datenum(s,pivotyear):用两位数表示年时,用参数 pivotyear 可以指定该年份所在的 100 年范围的起 始年。例如 pivotyear=1920 和 pivotyear=1900,都表示起始年是 1900 年。缺省的 pivotyear 值是当前的年份减去 50 年。

148

```
下面的命令列出了函数 datenum 的几种调用方法及其结果:
   datenum('08/01/2000')
   ans =
        730699
   datenum('08/01/2000',1800)
   ans =
        730699
   datenum(1998,9,1)
   ans =
        729999
   datenum(1998,9,1,13,45,10)
   ans =
     7.3000e+005
   2. datevec 函数
   函数 datevec 的调用格式为:
    (1) c=datevec(t):把日期字符串和连续日期数分成日期向量,格式为[年 月 日 时 分秒]。如果 t 是日期
字符串, 它必须是格式列表中的第0、1、2、6、13、14、15、16种格式之一。
    (2) [y, m, d, h, mi, s]=datevec(t):一单独变量的形式返回日期的成分。
    (3) [...]=datevec(t, pivoyear):当使用两位数表示年时,用参数 pivoyear 可以指定该年份所在的 100 年范围
的起始年。缺省的 pivotyear 值是当前的年份减去 50 年。
   下面的命令列出了函数 davevec 的几种调用方法及其结果
   [y, m, d, h, mi, s]=datevec('08/01/99')
   \mathbf{y} =
          1999
   m =
        8
   d =
        1
   h =
        0
   mi =
       0
   s =
       0
   c=datevec('08/01/99',1800)
   c =
          1899
                        8
                                  1
                                             0
                                                        0
                                                                   0
   c=datevec('08/01/01')
                                    1
                                               0
                                                                     0
   c =
            2001
                          8
                                                          0
   3. datestr 函数
   函数 datestr 调用格式为
```

(1) datestr(d, dateform):把连续日期数 d 转换为日期字符串,日期字符串的格式由参数 dateform 的值指定。 缺省情况下,根据 d 包含的内容分别取为 1、16 或 0。

(2) datestr(d, dateform, pivotyear):当使用两位数表示年时,用参数 pivotyear 可以指定该年份所在的 100 年范围的起始年。缺省的值是当前的年份减去 50 年。dateform=-1 时使用缺省值。

下面的命令列出了函数 davestr 的几种调用方法及其结果

datestr(now) ans = 23-Jun-2001 10:06:56 datestr(now,12) ans = Jun01 datestr(now,14) ans = 10:07:11 AM datestr(now,17) ans = Q2-01 datestr('8/1/99',8) ans = Sun datestr('8/1/99',1) ans = 01-Aug-1999 datestr('8/1/99',1,1800) ans = 01-Aug-1899

5.5.3 几个特殊的日期函数

calendar 函数
 函数 calendar 显示日历,下面列出了该函数的两种使用方法
 在命令窗口输入
 calendar, calendar('09/01/2000')
 回车将显示结果

#### Jun 2001

S	Μ	Tu	W	Th	F	S
0	0	0	0	0	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
0	0	0	0	0	0	0

#### Sep 2000

S	М	Tu	W	Th	F	S
0	0	0	0	0	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
0	0	0	0	0	0	0

```
2. weekday 函数
函数 weekday 显示星期,下面给除了该函数的使用方法,在命令窗口输入
[d1,w1]=weekday(now), [d2,w2]=weekday('07/01/1999')
回车将显示结果
d1 =
   7
w1 =
Sat
d2 =
   5
w^2 =
Thu
3. eomday 函数
函数 emoday 显示月的最后一天,下面给除了该函数的使用方法,在命令窗口输入
eomday(2001,2)
回车将显示结果
ans =
   28
```

5.5.4 用日期格式标注坐标轴

函数用来为坐标轴添加日期格式的标注,它的调用格式为 datetick(tickaxis, dateform):参数 tickaxis 必须是 字符串'x'、'y'、或'z'之一,缺省是'x'。参数 dateform 指定标注的日期格式字符串个时。可以是格式编号也可以是 格式字符串。如果没有指定参数 dateform,那么该函数根据绘图的数据进行猜测。

下面的例子是使用日期标注坐标轴的某城市的年气温变化曲线,在命令窗口输入

```
m=(1:1:12);
t=datenum(1998,m,1);
y=[-10, -6, 5, 10, 20, 25, 30, 24, 22, 19, 10, 6];
plot(t,y);
datetick('x',12)
axis tight
在图形窗口的到结果如图 5-1 所示
```



图 5-1 用日期标注的坐标图

#### 5.5.5 计算程序执行时间

程序的执行速度一直是编程人员关心的主要问题之一,MATLAB 提供了一刻计算某段程序执行速度的函数,包括:cputime、tic和 toc、etime。

这几个函数的用途大致相同,但是使用方式略有不同。

- ▶ 函数 cputime 计算从 MATLAB 启动到当前时刻, MATLAB 用去的时间, 单位是秒
- ▶ 函数 tic 和 toc 相当于秒表记时器的启动和停止
- ▶ 函数 etime(t1, t0)计算t1和t0之间经过的时间。这里的参数t1和t0必须是含有[年月日时分秒] 六个元素的向量。

下面分别举例说明如何使用这三个函数计算程序的执行时间。

```
1. cputime
建立如下程序
t1=cputime;
for i=1:0.01:1000
end
t=cputime-t1
```

#### 执行后显示结果

```
t =
```

```
0.1100
```

2. tict和 oc

建立如下程序

```
tic
```

```
for i=1:0.01:1000
```

end toc

执行后显示结果

elapsed_time = 0.1000 3.etime 建立如下程序 t1=clock; for i=1:0.01:1000 end t=etime(clock,t1) 执行后显示结果 t = 0.0600

🔉 计算的时间不完全相同是误差造成的,和各个函数本身没有必然联系

### 5.6 文件调试

MATLAB 调试过程是程序设计中的重要步骤,可以从中寻找错误,设置和清除断点,逐行运行 M 文件, 或在不同的工作区检查变量。MATLAB 的程序编辑器同时又是调试器。利用调试功能,可以比较容易地发现程 序中的错误。

调试的目的是改正程序中的错误。调试对于改正下面两种错误很有帮助。

- 语法错误(synax errors) 例如函数名的错误拼写、括号的省略等,MATLAB 可以检查出大部分 语法错误,并显示错误信息,标出错误在程序中的行号。
- ➢ 运行错误(runtime errors) 这类错误的根源一般为算法,例如,当除数为0时,将出现运行错误。 当运行结果错误时,可以很明显地判断为运行错误。

一般来说,命令窗口中显示的错误信息多为语法错误,比较容易修改。而运行错误出现时,将关闭函数的 工作空间,回到 MATLAB 的主工作空间,这样很多中间数据将会丢失,致使错误很难追踪。这时,可采用下述 几种方法来修正错误。

- > 去掉命令末尾的分号,将显示运行的中间结果。
- 在 M 文件中加入 keyboard 命令。运行 M 文件时,在 keyboard 命令的地方将暂停,同时在命令窗口中显示字符" K>>",这时,可以检查和修改函数工作空间中的变量的值。在命令窗口输入 return,再按回车链即可继续运行程序。
- 注释掉函数的定义行,把函数当作命令文件来运行。这样,所有的中间结果都可在 MATLAB 的 主工作空间中查看和修改。
- 使用 MATLAB 的调试器。由于可以查看和修改函数工作空间中的变量,调试器可以准确地找到运行错误。通过调试器设置断点可使程序运行到某行暂停,这时可以查看和修改各个工作空间中的变量。通过调试,可以一行一行运行程序。

MATLAB 的调试功能既可使用调试器实现,也司采用命令行实现。下面分别就两种方法举例说明。

5.6.1 用调试器调试

首先建立一个名为啊 var.m 的文件,该文件有一个输入参数,返回方差的无偏估计。var.m 调用一个名为 sqs.m 的文件来计算输入向量的中点平方和。

var.m的内容是 function y=var(x) mu=sum(x)/length(x); tot=sqs(x,mu); y=tot/(length(x)-1);

sqs.m 的内容是 function tot=sqs(x,mu) tot=0;for i=1:length(mu) tot=tot+((x(i)-mu).^2); end MATLAB 内部提供了一个函数来计算标准差,对于向量 v = [1 2 3 4 5];可以使用 std 计算其方差,在命令窗口输入  $var1=std(v).^{2}$ 回车显示结果 var1 =2.5000 使用自己定义的程序 var 计算方差 myvar1 = var(v)将见到错误信息,下面使用调试器来查错。 1. 启动调试器 如果采用 MATLAB 的程序编辑器输入文件 sqs.m,则可直接进行调试;如果使用外部文本编辑器编写文件 sqs.m,则在 MATLAB 的程序编辑器中打开该文件。 2. 设置断点

调试过程基本上都是从设置断点开始的。程序运行时,将在断点处暂停,允许用户查看和修改函数的工作 空间中的变量值。断点用行首的大红点来表示。当设置断点的行不是可执行时,断点将被设置在下一个可执行 行的行首。

在调试的开始阶段,并不知道错误在哪一行,甚至不知道错误是在文件 var.m 中,还是在文件 sqs.m 中,现 在把断点设置在 var.m 的第四行,如图 5-2 所示。

光标所在行的行号显示在图 5-2 下边的状态栏的右边。设置断点时,既可单击工具栏的按钮 🗐,也可使用 鼠标右键菜单中选中 "Set/Clear Breakpoints " 项,如图 5-3 所示。

💫 D:\matlabR12\work\var.m	
<u>File Edit View Text Debug Breakpoints Web Window H</u> elp	
🗈 😅 🛃 🎒 🐰 🛍 🋍 🕫 📯 👬 🌮 📲 🎦 🖓 🗐 🛱 🗊 🚛 🏭 Stack 🕬	× ×
1 - 3 - 4 ● 5 ← 5 ←	<ul> <li>Evaluate Selection</li> <li>Open Selection</li> <li>Help on Selection</li> <li>Cut</li> <li>Copy</li> <li>Paste</li> <li>Smart Indent</li> <li>Comment</li> <li>Uncomment</li> </ul>
	Set/Clear Breakpoint
var.m sqs.m	Co Lintil Horo
Ready	Su Ontil Here
	图:2、调试网口的上于去共产

图 5-2 断点设置示意图

图 5-3 调试窗口的上下文菜单

#### 3.检查变量

为了使程序运行到断点所在行,在命令窗口中输入以下命令

var(v)

当执行到断点处时,在断点和文本之间将产生一个黄箭头,表示所指行为将要运行的行,如图 5-4 所示。 这时可以查看和修改已经产生的变量。

🖗 D : \	matla	bR12\w	ork\va	ar.m						
<u>F</u> ile	<u>E</u> dit	View	<u>T</u> ext	Debug	Brea <u>k</u> poi	nts We	<u>b W</u> indo	w <u>H</u> elp		
🗋 🖻	🛛 🖨	み 暗	🔒 🗠 (	🗠 🏙 ታኑ	🗧 🛃	4 <b>1</b>	) 🕼 🎝	Stack: var	-	• • • 💌
1 2 - 3 - 4 • 5	function mu=sum tot=sqs tot=sqs	on y=var(x (x)/length s(x,mu); (length(x)	) .(x); -1);							*
Ready	var.m	sqs.m		and a second		n an	ee Careeraa	n e tra a como de la como		

#### 图 5-4 执行到断点处的程序编辑器

在选中 tot 或 mu 以后,单击鼠标右键,选中" Evaluate "项,或者从" View "菜单中选中" Evaluate Selection " 项,都将在命令窗口中显示

K>>mu

mu =

3

K>>tot

tot=

4

可见问题出在程序 sqs.m 上。

4.转换工作空间

选中调试器右上方的 "Stack " 下拉式菜单选择工作空间。选中 "base",即 MATLAB 的主工作空间,这时, 在命令窗口中输入命令 whos 或者使用变量浏览器可以查看主工作空间中的变量。

5.继续调试

将光标移到文件 var.m 的第四行,清除该断点,单击"Debug"菜单中的"Continue"项,程序将继续执行。 打开文件 sqs.m,并在第四行设置断点。采用前面的向量 v 再一次调用函数 var,这时查看"Stack"菜单中的 var,在文件的第三行前有一个中间有竖线的箭头,如图 5-5 所示。

🐴 D : \	matlal	bR12\w	ork\v	ar.m						
File	<u>E</u> dit	View	<u>T</u> ext	Debug	Brea <u>k</u> poi	nts We <u>b</u>	Windo	w <u>H</u> elp		
🗅 😅	8	X 酯	🖹 💫 (	2 🖗 f)	- 🔒 🗶	48 98 Bİ	13 13	Stack: var	-	
1 2 - 3 - • 4 - 5	function mu=sum () tot=sqs y=tot/ ()	n y=var(x x)/length (x, mu); length(x)	) (x); -1);							*
										▼
Ready	var.m	sqs.m		an the second	an an the test		Satisfaction			<u>.                                    </u>

图 5-5 在被调用函数中设置断点时的主函数窗口

通过"Evaluate"选项获取循环变量 i 的值,命令窗口中将显示

K>>i

i=

1

从菜单"Debug"中选中"Single Step"或单击"Single Step"按钮,将执行文件 sqs.m 的下一行,再查看变量 tot 的值,命令窗口中将显示:

K>>tot

tot =

4

再继续选择" Single Step"项,可见 sqs.m 中的循环只进行一次便返回到函数 var 而不是对 x 的每个元素循环,可以发现是 sqs 中循环语句有误,即下面语句有误:

```
for i=1:length(mu)
```

```
6.停止调试
```

```
选中 "Debug " 菜单中的 "Quit Debugging " 或工具栏中的相应按钮将停止调试。把
```

for i = 1:length(mu)

改为

for i = 1:length(x),并清除所有断点,并在文件 var.m 的第四行设置断点,重新运行下列命令

var(v)

查看 tot 的值,命令窗口将显示

K>>tot

tot=

10

选择"Contune Execution"项继续执行,将得到正确结果。

ans =

2.5000

7.退出调试

选择"File"菜单中的"Exit Editor/Debugger"选项退出调试器。

#### 5.6.2 在命令窗口中调试

除了采用调试器调试程序外, MATLAB 还提供了一些调试命令,用调试命令可在命令窗口中调试程序。所 有的调试命令都是以字母 db 开头,表 5-10 列出了常用断点调试命令的基本调用方法。

表 5-10

断点调试命令

命令	说明
dbstop in fname	在 M 文件 fname 的第一可执行程序上设置断点
dbstop at r in fname	在 M 文件的第 r 行程序上设置断点。如果第 r 行程序是不可执行的 , 则
	程序会在运行 r 行可执行程序后停止
dbstop if v	当遇到条件 v 时,停止运行程序。当发生错误时,条件 v 可以是 error ,
	当发生 NaN 或 inf 时,也可以是 naninf/infnan
dstop if warning	如果有警告,则停止运行程序
dbclear at r in fname	清除文件 fname 的第 r 行处断点
dbclear all in fname	清除文件 fname 中的所有断点
dbclear all	清除所有 M 文件中的所有断点
dbclear in fname	清除文件 fname 第一可执行程序上的所有断点
dbclear if v	清除第 v 行由 dbstop if v 设置的断点
dbstatus fname	在文件 fname 中列出所有的断点
dbstatus	显示存放在 dbstatus 中用分号隔开的行数信息

这些命令与下面表 5-11 中列出的运行控制命令用来跟踪和控制 M 文件运行是非常有用的。

表 5-11

运行控制命令

命令	说明
dbstep	运行 M 文件的下一行程序
dbstep n	执行下 n 行程序,然后停止
dbstep in	在下一个调用函数的第一可执行程序处停止运行
dbcount	执行所有行程序直至遇到下一个断点或到达文件尾
dbmex	调试 MEX 文件的命令
dbquit	退出调试模式

采用上例中的 var.m 和 sps.m 讨论这些命令的用法。

1. 设置断点

函数 dbstop 可在文件的任意行设置断点,当文件执行到断点时将暂停。在文件 var.m 中求 meam 后设置断点,并在求中心平方和语句后设置断点,命令为

dbstop var 3

dbstop var 4

2.运行程序

仍用向量

[1 0 0 4 F

v = [1 2 3 4 5];

作为计算的输入参数。正确的平均值 mean 应为 3,中心平方和应为 10。为了检验是否能得到正确答案,在 命令窗口中输入命令

var(v)

命令窗口中显示将要运行的下一行(同时打开调试器)

usage: [bnds,rowd,sens,rowp,rowg] = mu(matin,blk,opt)

Warning: One or more output arguments not assigned during call to 'mu'.

> In D:\matlabR12\work\var.m at line 2

Warning: Divide by zero.

> In D:\matlabR12\work\var.m at line 2

 $K \!\!>\!\!>$ 

当执行到断点处时,可在命令窗口中输入命令查看和修改变旦的值,输入 dbcount 命令将继续进行调试。 当在第一个断点处暂停时,用命令 whos 查看工作空问中的变量

K>> whos Name Size Bytes Class 1x1 8 double array mu х 1x5 40 double array Grand total is 6 elements using 48 bytes 由于工作空间中无变量 v,该工作空间为函数 var 的工作空间。查看 mu 的值 K>>mu mu= 3 用命令 dbstep 执行第一个断点所在行,查看变量的值 K>>dbstep 4 y=tot/(length(x)-1);K>>tot tot= 4 由于 tot 和中心平方和有较大差距,可以大致判断 sqs.m 有错。 3.转换工作空间 命令 dbup 和 dbdown 用来转换工作空间。用命令从函数的工作空间转到主工作空间 K>> dbup, whos 回车显示结果 In base workspace. Name Size Bytes Class 8 double array ans 1x1 v 1x5 40 double array Grand total is 6 elements using 48 bytes 用命令 dbdown 将返回到函数 var 的工作空间。 4.显示大行号的 M 文件 在命令窗口中用函数 dbtype 来查看函数 sqs.m 的内容 K>> dbtype sqs 回车显示结果 1 function tot=sqs(x,mu) 2 tot=0;3 for i=1:length(x) 4 tot=tot+((x(i)-mu).^2); 5 end 5. 查看未执行完的命令行并继续调试 在 MATLAB 命令窗口中输入下面一行命令将清除文件 var 中的断点: dbclear var 用命令 dbs 毗查看程序调试过程中未运行的程序行: K>> dbstack 回车显示结果 In D:\matlabR12\work\var.m at line 3 6.退出调试 用命令 dbquit 退出调试,返回主工作空间。修改文件 sqs.m,再运行将会得到正确结果。 5.6.3 文件评述

MATLAB 提供的程序评述器可以记录程序中每一步的运行时间。评述(profile)文件是提高文件运行速度的一种方法。

采用估计的方式来猜测文件中什么地方耗时最多,往往会做出错误的判断,而采用程序评述器可以得出准确的结果。提高程序运行效率的方法是尽量用最简单的代码来编写程序,并用程序评述器来查找运行速度的"瓶颈",逐步解决各个瓶颈。下面列出了提高速度的一些常用措施:

去掉不必要的计算,

> 采用省时的算法。

▶ 避免重复计算。

采用程序评述器评述文件的目的是提高程序的运行速度,当运行程序的时间基本上都花在调用 MATLAB 的 内部函数上时,代码优化的目的就达到了。

评述文件是通过命令 profile 来实现的。在 MATLAB 中,一欢只能评述一个文件。当文件运行时,程序评述 器将以 0.01 秒为单位,记录下每行的执行时间。时间的记录采用累计方式。profile 命令的调用方式为:

profile 关键字

其中的"关键字"可为:文件名、on、off、done、reset, report 或 plot。关键字为文件名时,将评述指定的 文件;关键字 on、off、done、reset 用于控制评述的状态;关键字 report 将给出刚才评述的文字报告;关键字 plot 将采用图形的方式绘制出评述结果。

# 5.7 小结

在本章中,主要介绍了 M 文件的编写、优化、调试方法和时间日期函数,本章中还介绍了 MATLAB 中的 类与对象、MATLAB 对文件的管理方法,通过本章的学习,可以熟悉 MATLAB 中 M 文件的编写与调试。

# 第六章 数据可视化

MATLAB 不仅在数值计算上独占整头,而且在数据可视化方面也有上乘表现,本章主要讲述的内容包括

- ➢ MATLAB 中二维图形的绘制
- ➢ MATLAB 中三维图形的绘制
- ▶ 特殊图形
- ▶ 色彩控制
- > 图形对象控制
- ▶ MATLAB 中的图形文件操作
- ▶ 动画制作

人们很难直接从一大堆原始的离散数据中体会到它们的含义,用数据画出图形却能使人们用视觉器官直接 感受到数据的许多内在本质。因此,数据可视化是人们研究科学、认识世界所不可缺少的手段。

作为一个优秀的科技软件,MATLAB 不仅在数值计算上独占整头,而且在数据可视化方面也有上乘表现。 MATLAB 可以给出数据的二维、三维乃至四维的图形表现。通过对图形线型、立面、色彩、渲染、光线、视角 等的控制,可把数据的特征表现得淋漓尽致。MATLAB 提供了两个层次的图形命令:一种是对图形句柄进行的 低级图形命令,另一种是建立在低级图形命令之上的高级图形命令。

MATLAB 提供了很多高级图形命令,这些命令可以绘制一般科技绘图软件所能绘制的几乎所有图形,如曲 线图、极点图、直方图、等高线图、网格图、表面图等。用户还可控制图形的颜色、视角、坐标标注和阴影等 和图形外观有关的因素。高级图形命令可自动控制如坐标刻度等图形特性。

MATLAB 还提供了句柄图形,用户利用句柄图形可以对图形的显示进行精密的控制,也可利用句柄图形的 命令生成用户自己的图形命令。句柄图形是 MATLAB 的面向对象的图形系统。句柄图形中定义了一些图形对象, 如线、面、文本框等,每个图形对象都有很多可以更改的属性。通过更改句柄图形对象的属性值,可以生成理 想的图形。对于已经生成的图形,可在命令窗口中输入相应的句柄图形命令来改变图形的外观。

利用图形界面来输入数据以及在程序运行中和计算机交互非常方便。句柄图形命令可以建立菜单、按钮、 文本框以及其他的图形界面部件。MATLAB 的句柄图形命令,可以用于开发图形界面。

# 6.1 二维图形

MATLAB 提供了很多绘制、标注二维图形的函数。在本节中将介绍这些函数,并对一些常用的函数举例说 明。表 6-1 介绍了绘制线图的函数,这些函数的不同之处在于,不同函数对坐标轴的标注不同。这些函数的输 入参数为向量或矩阵,MATLAB 会自动根据数据的大小调整坐标的刻度范围。

表 6-1

部分绘制二维图形的函数

命令	轴的刻度方式
plot	纵横轴均采用线性坐标
logog	纵横轴均采用对数坐标
semilogx	纵轴采用线性坐标,横轴采用对数坐标
semilogy	纵轴采用对数坐标,横轴采用线性坐标
plotyy	在图的左右两侧分别建立纵坐标轴

在 MATLAB 中建立二维图形的完整步骤可分为七步。但并不是绘制所有的二维图形都需要这七步,例如绘 图只是为了检验数据是否正确,只用第一步和第三步就足够了,其他的步骤只是为了把图形修饰得更漂亮。建 立二维图形的步骤如下(每一步下面的命令为示例)。

(1) 准备数据

x=0:0.2:12;

y1=bessel(1, x); y2=bessel(2, x); (2) 选定绘图窗口或窗口中的区域 figure(1) subplot(2,2,1)(3) 调用基本绘图命令 f=plot(x, y1, x, y2, x, y3); (4)选择线型和数据点图标 set(h, 'LineWidth',2, {'LineStyle'}, {'--'; ':'; '-.'}) set(h, {'Color'}, {'r'; 'g'; 'b'}) (5) 选定轴的显示范围、刻度线和网格线 axis([0 12 -0.5 1]) grid on (6)标注轴、图标和文本 xlabel('Time'); ylabel('Amplitude') legend(h, 'First', 'Second', 'Third') title('Bessel Functions'); [y, ix]=min(y1); text(x(ix), y, 'First Min \rightarrow', ...) (7)打印图形 print -dps2

#### 6.1.1 基本绘图命令

1.线性坐标平面图

plot 是最基本的二维图形命令,它是以 MATLAB 的内部函数(Build-in Function)的形式出现的。MATLAB 其他二维函数中的绝大多数是以 plot 为基础构造的。表 6-2 列出了 plot 命令的常用形式,列表中 A 表示一个 m×n 的矩阵,线条的类型和颜色可以通过 str 字符串来指定。

表 6-2

绘图命令

命令	说明
plot(x, y)	对向量 x 绘制向量 y。以 x 为横坐标 , y 为纵坐标 , 按照坐标(x _i ,y _j )的有序排列绘
	制曲线
plot(y)	以 j 为横坐标,y _j 为纵坐标,绘制(j, y _j )的有序集合的图形
plot(z)	以横轴为实轴,纵轴为虚轴,绘制(real $(z_k)$ , imag $(z_k)$ )的有序集合的图形。这样,复
	数 z _k 就在复平面上
plot(A)	绘制矩阵 A 的列对它下标的图形。对于 m × n 的矩阵 A,有 n 个含有 m 个元素的
	数对,或是 n 条有 m 个点曲线,且这 n 条曲线均采用颜色监视器上不同的颜色绘
	制而成
plot(A, x)	绘制矩阵 A 对向量 x 的图形。对 m × n 的矩阵 A 和长度为 m 的向量 x , 绘制矩
	阵 A 的列对向量 x 的图形。如果 x 的长度为 n , 则绘制矩阵 A 的行对向量 x 的
	图形。向量 x 可以是行向量也可以是列向量
plot(A, B)	对矩阵 A 的行绘制矩阵 B 的列的图形。如果 A 和 B 都是 m × n 的矩阵,将绘制
	n 条由 m 个有序对连成的曲线
plot(,str)	使用字符串 str 指定的颜色和线型进行绘图
plot(x1,y1,str1,x2,	用字符串 str 指定的颜色和线型对 y1 绘制 x1 的图形 , 用字符串 str2 指定的颜色
y2,str2)	和线型对 y2 绘制 x2 的图形
[l,f,p,errorm]=cols	返回 str 中不同部分的值。其中1代表线型,f 代表颜色,p 代表点的类型,errorm
tyle(str)	用来保存系统错误信息

下面介绍函数 plot 的使用方法。函数 plot 的基本调用方法为:

(1) plot(y)

当 y 为向量时,则以 y 的元素值为纵坐标,以相应元素下标为横坐标,绘制连线图。如 y 为实矩阵,则按

列绘制每列元素值相对其下标的连线图,图中曲线数等于矩阵的列数。若 y 为复数矩阵,则分别以每列元素的 实部和虚部为纵横坐标绘制多条连线图。

例如对于向量 y,使用 plot 命令,在命令窗口输入命令行 y=[4 2 9 0 4 6]; plot(y) 回车将在绘图窗口绘出曲线,如图 6-1 所示。



图 6-1 plot(y 函数绘制的曲线)

(2) plot(x,y)

若 y 和 x 为同维向量,则以 x 为横坐标、y 为纵坐标绘制连线图。若 x 是向量, y 是有一维和 x 同维的矩阵,则绘制多条不同色彩的连线图,曲线的根数等于矩阵的另一维, x 被作为这些曲线的共同横坐标。如果 x 为矩阵, y 为向量,情况与上相同,只是都以 y 为共同纵坐标。若 x 和 y 为同维矩阵,则以 x、y 对应列元素为横纵坐标 分别绘制曲线,曲线条数等于短阵的列数。若 x 和 y 为复数矩阵时,MATLAB 将忽略虚数部分。

例如,在命令窗口输入命令

t=0:0.1:4*pi;

y=sin(t);

plot(t,y);

回车将在绘图窗口绘制两个周期的正弦曲线,如图 6-2 所示



图 6-2 plot 函数绘制的正弦曲线

在此格式中,每对 x, y 必须符合 plot(x1,y1, x2,y2,...)中的要求,不同对之间没有影响,命令将对每一对 x、 y 绘制曲线。使用这种方式可以在同一窗口中绘制多条相互没有影响的曲线。

例如,在命令窗口输入

t1=0:0.1:2*pi;

t2=0:0.1:4;

plot(t1,sin(t1),t2,sqrt(t2));

回车将在绘图窗口绘制两条相互没有影响的曲线,如图 6-3 所示。

需要注意的是,在以上三种格式中,输入参数 x, y都可以是表达式,但表达的结果应符合上述格式的要求。 (4)plot(x,y, '选项')

在 MATLAB 中,用户可自己设置图形中颜色、线型和数据点等图形属性。这里"选项"包括了线型、颜色、 数据点标记符号等特性的设置,表 6-3 到 6-5 列出了常用的几种选项。



图 6-3 plot 函数的多向量绘制图

表 6-3

常用线型选项

选项	说明
none	无线
'-'	实线
''	虚线
':'	点线
''	点划线

表 6-4

```
常用点选项
```

选项	说明
	用点号标出数据点
! * !	用星号标出数据点
'0'	用圆圈标出数据点
'x'	用叉号标出数据点
'+'	用加号标出数据点
'S'	用小正方形标出数据点
'd'	用菱形标出数据点
'v'	用下三角标出数据点
' <b>^</b> '	用上三角标出数据点
'<'	用左三角标出数据点
'>'	用右三角标出数据点
'h'	用六角形标出数据点
'p'	用五角形标出数据点

6-5	常用颜色选项
选项	说明
'r'	红色
'g'	绿色
'b'	蓝色
'y'	黄色
'm'	洋红
'c'	青色
'w'	白色
'k'	

利用表中这些选项可以把同一窗口中不同的曲线设置为不同的线型和颜色,可以只画出数据点,也可以在 绘制的曲线上同时标出数据点。这些选项可以组合使用,例如选项"--r"表示绘制红色的虚线,":yx"表示绘 制黄色点线,同时用符号'x'标记数据点。

下面使用不同的线型和标注来绘制两条曲线,在命令窗口输入命令

t1=0:0.1:2*pi;

t2=1:0.1:4;

plot(t1,sin(t1),'ob',t2,cos(t2),'--g');

回车将在图形窗口显示结果,如图 6-4 所示



图 6-4 使用不同线型绘图

2. 对数坐标曲线

函数 semilogx、 semilogy 和 loglog 可以用来绘制二维对数坐标曲线,这几个函数的用法和 plot 函数相同。 例如,在命令窗口输入如下命令

t=0:0.1:2*pi; y=sin(t); semilogx(t,y); grid on

在图形窗口绘制出以对数坐标为横坐标的正弦曲线,并用"grid on"命令为图形窗口添加了网格。如图 6-5 所示。

表



图 6-5 绘制对数坐标曲线

函数 semilogy 绘制的图形纵坐标为对数坐标, loglog 函数绘制的图形横纵坐标均为对数坐标。 3. 双 y 轴图形

使用 plotyy 函数可以绘制左右各有一个 y 轴的图形。该函数有以下几种常用的形式

- 在一个图形窗口同时绘制两条曲线(x1,y1)和(x2,y2),曲线(x1,y1)用左侧的 y  $\blacktriangleright$  plotyy(x1,y1,x2,y2) 轴,曲线(x2,y2)用右侧的 y 轴。
- $\triangleright$  plotyy(x1,y1,x2,y2,FUN) "FUN"是字符串格式,用来指定绘图的函数名,如plot、semilogx 等。
- plotyy(x1,y1,x2,y2,FUN1,FUN2) 和第二种形式类似,只是用"FUNI"和"FUN2  $\geq$

可以指定不同的绘图函数分别绘制这两条曲线。

例如,在命令窗口输入下面的命令

x=1:0.01:5;

y = exp(x);

plotyy(x,y,x,y,'semilogy','plot');

将在图形窗口用不同的函数、不同的 y 坐标轴绘制同一条指数曲线, 如图 6-6 所示。

图中左侧的 y 轴为 4 对数坐标, 图中由 semilogy 函数绘制的直线对应着这个 y 轴(在对数坐标系中, 指数函 数 y=exp(x)为一条直线);右侧的 y 轴为线性坐标,图中由 plot 函数绘制的曲线对应着这个 y 轴。在彩色显示方 式下每条曲线的颜色和相应的坐标轴颜色相同,而两个 y 轴的颜色(两条曲线的颜色)互不相同,以示区别。

🗻 在 plotyy 函数中不能加入设置曲线的线型、颜色以及标出数据点的参数。如果要进行这种设置可 以借助我们在第7章将要讲到的 set 函数

4. 极坐标曲线

在 MATLAB 中可以使用 polar 函数绘制极坐标曲线。该函数的调用格式为

polar(theta,rho, '选项')

其中" theta"和" rho"分别为角度向量和幅值向量,要求向量" theta"和" rho"的长度相同,其"选项" 的内容和用法与 plot 函数基本一致。

例如,可以在命令窗口中输入下面的命令,绘制了如图 6-7 所示的极坐标曲线

x=0:0.1:4*pi;

y = cos(x/4) + 1/3;

polar(x,y);



图 6-6 双 y 轴图型



图 6-7 极坐标曲线

5.复数数据

当绘图数据为复数时,使用 plot 函数绘制图形根据 plot 函数的不同调用方式会有两种不同的情况。

 $\succ$  plot(x,y)

这种方式下不论参数 x 和 y 哪个是复数,都将被忽略掉虚部数据,只绘制该复数的实部数据。其余用法和 绘制一般实数数据相同。

 $\succ$  plot(z)

用这种方式绘制复数 z 的图形时,会以复数的实部为横坐标、虚部为纵坐标绘制实部和虚部的关系曲线, 其作用相当于使用 plot(real(z), imag(z))命令。

下面举例来说明两种方式的区别。例如,在命令窗口输入如下一组命令,定义复数 z

t=0:0.1:2*pi;

x=sin(t);

y=cos(t);

z=x+i*y;

如果在命令窗口中输入命令行

plot(t,z);

命令执行结果将在图形窗口显示如图 6-8 所示的正弦曲线,如果在命令窗口中输入命令行。



图 6-8 plot(t,z)绘制的复数图形

plot(z); 那么将在图形窗口显示如图 6-9 所示的正余弦关系曲线。



图 6-9 plot(z) 绘制的复数图形

6.1.2 基本绘图控制

在使用 MATLAB 绘制图形时,给出 M AT L A B 命令要使用命令窗口,而展示图形的是图形窗口。图形窗 口可以用将要介绍的命令进行控制,这些命令可以用来在各个窗口之间进行切换、清除窗口内容或保持当前图 形。

1. 基本窗口命令

在表 6-6 中列出了常用的窗口命令。

表 6-6		窗口命令
-	命令	说明
_	figure(gcf)	显示当前图形窗口。figure 命令还可以用来在两个图形窗口之间进行切换和创
_		建新的图形窗口
	shg	显示当前图形窗口,等价于 figure(gcf)
	clf	清除当前图形窗口。警告:如果设置 hold on 状态,窗口内容也将被清除
	clg	早期版本中等价于 clf 命令。在 MATLAB 以后的版本中可能会被淘汰
	clc	清除命令窗口
_	home	移动光标到命令窗口的左上角
	hold on	保持当前图形。允许在当前图形状态下,使用同样的缩放比例加入另一个图形
	hold off	释放图形窗口,这样下一个图形将称为当前图形。这是缺省状态
	hold	在 hold on 和 hold off 之间进行切换
	ishold	如果当前图形处于 hold on 状态,则返回 1 ;否则,返回 0

命令 hold 用于向已有的图形窗口中加入图形,当 hold 设为 on 时,绘图命令并不删除当前图形窗口中的线

条,只是把新的数据加入到该图形中,如果新的数据超出坐标的范围,MATLAB 会自动调整坐标轴的显示范围。 例如在命令窗口输入

semilogy(1:100,'+') hold on plot(1:3:300,1:100,'--') hold off 回车在图形窗口显示结果如图 6-10 所示。



图 6-10 用 hold on 命令添加图形

2.子图形窗口

有时需要在一个图形窗口显示几幅图,这可以通过用 subplot 函数把图形窗口分割为几个子窗口来实现。函 数 subplot 本身并不绘制任何图形,但是,它决定了如何分割图形窗口以及下一幅图将被画在哪个子窗口中。

函数 subplot 的调用格式为

subplot(m , n , i)

它的含意是把图形窗口分割为 m 行 n 列子窗口, 然后选定第 i 个窗口为当前窗口。

例如,命令

subplot(2, 2, 3)

指的是把图形窗口分为2行2列共四个子窗口选择第二行第一列(排序是3)的子窗口为当前窗口进行操作。 下面的一个例子中, subplot 函数把两种不同的图形综合在一个图形窗口。在命令窗口输入命令 t=0.1:0.1:4*pi;

z=x+i*y; subplot(2,2,1) plot(z,'s') grid on subplot(2,2,2) polar(t,x) subplot(2,2,3) semilogx(t,x) grid on subplot(2,2,4)	$y=\cos(t);$
subplot(2,2,1) plot(z,'s') grid on subplot(2,2,2) polar(t,x) subplot(2,2,3) semilogx(t,x) grid on subplot(2,2,4)	z=x+i*y;
subplot(2,2,1) plot(z,'s') grid on subplot(2,2,2) polar(t,x) subplot(2,2,3) semilogx(t,x) grid on subplot(2,2,4)	
plot(z,'s') grid on subplot(2,2,2) polar(t,x) subplot(2,2,3) semilogx(t,x) grid on subplot(2,2,4)	subplot(2,2,1)
grid on subplot(2,2,2) polar(t,x) subplot(2,2,3) semilogx(t,x) grid on subplot(2,2,4)	plot(z,'s')
subplot(2,2,2) polar(t,x) subplot(2,2,3) semilogx(t,x) grid on subplot(2,2,4)	grid on
polar(t,x) subplot(2,2,3) semilogx(t,x) grid on subplot(2,2,4)	subplot(2,2,2)
subplot(2,2,3) semilogx(t,x) grid on subplot(2,2,4)	polar(t,x)
subplot(2,2,3) semilogx(t,x) grid on subplot(2,2,4)	
semilogx(t,x) grid on subplot(2,2,4)	subplot(2,2,3)
grid on subplot(2,2,4)	semilogx(t,x)
subplot(2,2,4)	grid on
subplot(2,2,4)	
	subplot(2,2,4)

x = sin(t);

plotyy(t,x,4*x+2*pi,2*y) 这一组命令绘制出图 6-11 中所示的四幅不同的图形。





3. 坐标轴刻度的调整

在前面的例子中已经看到,MATLAB 的绘图函数可以根据要绘制曲线数据的范围自动选择合适的坐标系, 使得曲线尽可能清晰地显示出来,所以一般情况下用户不必自己选择绘图坐标。但对有些图形,如果用户觉得 自动选择的坐标不合适,则可以用手动的方式选择新的坐标系。手动选择坐标系的工作可以由函数 axis 来完成, 该函数的调用格式为:

axis([xmin,xmax,ymin,ymax,zmin,zmax)

↘ 坐标的最小值(xmin, ymin, zmin)必须小于相应的最大值(xmax, ymax, zmax), 否则会出错 如下几种调用格式常用于指定选取坐标刻度的方式

- ▶ axis auto 设为自动选取刻度的方式,缺省值。
- ▶ axis manual 使坐标刻度保持当前状态不变。
- ➤ axis tight 限定坐标刻度等于数据范围。
- ➤ axis fill 使坐标轴充满边框。

在这个函数中可以输入 4 个或 6 个参数 , 分别对应于二维或三维坐标系每一坐标轴的最小值和最大值。参

数可以是数值也可以是某些函数。MATLAB 会按照用户指定的值来选择合适的坐标。

例如,对于绘制了正弦曲线的命令

t=0:0.1:2*pi;

y=sin(t);

plot(t,y)

其坐标为自动坐标,如果想要调整该坐标系使得正弦曲线能充满整个窗口,那么可以由下面的命令来完成 axis([0,max(t),min(y),max(y)])

该命令执行后,显示如图 6-12 所示的坐标轴显示范围。



图 6-12 使用函数调整坐标刻度

坐标系的调整还可以用命令 set 来完成,关于该命令的用法将在后面具体介绍。

通常状况下,坐标轴的纵横比也由 MATLAB 根据图形窗口的大小自动选取,但是,有时需要按照确定的纵 横比例显示图形,在图形窗口发生变化时它的比例也保持不变,甚至有时图形的尺寸都不发生变化。这些特殊 要求可以借助 axis 函数来实现。相关的选项有

- ➢ axis equal 使得在几个方向的坐标轴上对于相等的刻度增量尺寸也相同
- ▶ axis image 在的基础上,还使得坐标轴的边框紧靠数据曲线
- ➤ axis square 使坐标的边框尺寸相等
- ➢ axis normal 恢复缺省状态
- ➤ axis vis3d 锁定纵横比

例如,使用 MATLAB 中的 sphere 函数绘出的球体,在默认情况下,看到的球体是扁的,在调用函数 axis equal 之后,才会变成真正的圆球。

#### 4.坐标轴缩放

在 MATLAB 中可以使用函数 zoom 来指定是否可以对图形进行缩放,这在分析数据量比较大的图形时很常用。zomm 函数有以下这些用法

- ➢ zoom 在 zoomOn 和 zoomoff 状态之间切换
- ▶ zoom(factor) 以系数 factor 为倍数对图形进行缩放的操作,但不改变图形的缩放状态
- ➢ zoom on 允许对图形进行缩放
- ➢ zoom off 不允许对图形进行缩放
- ➢ zoom out 恢复图形的初始大小
- ➢ zoom xon 只允许对图形的 x 轴方向缩放
- ➢ zoom yon 只允许对图形的 y 轴方向缩放

在允许缩放状态下,用户可以用鼠标对图形进行放大或缩小操作:用鼠标左键每单击一下图形要放大的部
位,图形就会放大一倍;放大后的图形用鼠标右键每单击一下要缩小的部位,图形就会缩小一倍;双击鼠标左

# 键,图形会恢复原始大小。

🔉 原始大小的图形只能放大,不能缩小,缩小的操作只是对于被放大的图形有效

## 6.1.3 图形的标注

MATLAB 有若干个命令可以用来在图形窗口中输出文本,所有这些命令都可以用在当前子窗口中,并且通 常应该在窗口中画完图形之后给出。使用这些命令,MATLAB 可以在画出的图形上加各种标注及文字说明,以 丰富图形的表现力。在中文 Windows 环境下,还可以用汉字进行标注。表 6-7 列出了常用的标注命令。

#### 表 6-7

图形窗口文本命令

命令	说明
title(txt)	在图形窗口顶端的中间位置输出字符串 txt 作为标题
xlabel(txt)	在 x 轴下的中间位置输出字符串 txt 作为标注
ylabel(txt)	在 y 轴边上的中间位置输出字符串 txt 作为标注
zlabel(txt)	在 z 轴边上的中间位置输出字符串 txt 作为标注
text(x,y,txt)	在图形窗口的(x,y)处写字符串 txt 。坐标 x 和 y 按照与所绘制图形相同的刻度
	给出。对于向量 x 和 y , 字符串 txt 写在(xi, yi)的位置上。如果 txt 是一个字符
	串向量,即一个字符矩阵,且与 x,y 有相同的行数,则第 i 行的字符串将写在
	图形窗口的(x _i , y _i )的位置上
text(x,y,txt,'sec')	在图形窗口的(x, y)处输出字符串 txt , 给定左下角的坐标为(0.0, 0.0),右上角的
	坐标则为(1.0, 1.0)
gtext(txt)	通过使用鼠标或方向键,移动图形窗口中的十字光标让用户将字符串 txt 放置
	在图形窗口中。当十字光标走到所期望的位置时,用户按下任意键或鼠标上的
	任意按钮,字符串将会写入在窗口中

# 1.为坐标轴加标注,给图形加标题

给坐标轴 x、y、z 加标注只要调用相应的函数 xlabel、ylabel、zlabel。以函数 xlabel 为例,其调用格式为: xlabel('text','propertyl',propertyvaluel,'property2',propertyvalue2....)

其中"text"为要添加的标注文本,"property"指该文本的属性,"propertyvalue"为相应的属性值。该命令把文本"text"按照指定的格式添加到 x 轴下方。

# 给图形加标题的函数为 title,它的调用格式和 xlabel 类似

title('text', 'propertyl', propertyvaluel, 'property2', propertyvalue2,...) 区别只是 title 函数把文本 "text "加到图形上方。 例如,在命令窗口输入如下命令,为图形添加标题和坐标轴标注 t=0:0.1:2*pi; y=cos(t); plot(t,y); xlabel('x 轴 (1-2 pi)') ylabel('y 轴') title('余弦曲线',Fontsize',13,FontWeight','Bold','FontName','宋体') 结果如图 6-13 所示。



图 6-13 图形基本标注

在函数 xlabel、ylabel 和 title 中可以设置的属性有很多,例如上例中用到的"FontWeigh"(字体粗细)、 "FontName"(字体名称)、"FontSize"(字体大小)等,将在后面结合图形属性一并介绍。

另外,可以看到在函数 title 的文本字符串中的" \pi "显示到图形标题上是字符" *p*",这里用特征字符串来 表示 TeX 格式的字符,斜线" \"是引导特征字符串的标志符号。使用特征字符串可以把很多数学公式或工程中 的 TeX 符号标注到图形上,大大增加了标注的灵活性。MATLAB 中的特征字符串及其对应的 TeX 符号见表 6-8。

2. 在图形窗口添加文本字符串

使用函数 text 可以在图形窗口的任何位置加入文本字符串。该函数调用格式为

text(x,y,'string')

这里, x, y用于指定加入字符串的位置, "string"是需要添加的字符串, 该字符串中也可以添加由引导"\"的特征字符串来表示特殊符号。

例如,可以使用如下一组命令对上例中的图做标注

text(2*pi-1,-0.8,'图形标注示例')

 $text(pi/2,0,'\leftarrowcos(pi/2)=0')$ 

结果如图 6-14 所示

#### 表 6-8

#### 特征字符串和 Tex 符号

特征字符串	符号	特征字符串	符号
\alpha	α	\beta	β
\gamma	γ	\delta	δ
\epsilon	ε	\zeta	ζ
\eta	η	\iota	l
\theta	θ	\vartheta	ϑ
\kappa	κ	\lambda	λ
\mu	μ	\nu	ν
\xi	بح	\pi	π
\rho	ρ	\tau	τ
\sigma	σ	\varsigma	ς
\upsilon	υ	\phi	φ
\chi	χ	\Gamma	Г
\omega	ω	\Theta	Θ
\Delta	Δ	\Xi	Ξ
\Lambda	Λ	\Sigma	Σ
\Pi	П	\phi	φ
\Upsilon	Y	\psi	ψ

## MATLAB 6 实用教程

特征字符串	符号	特征字符串	符号
\Psi	Ψ	\Omega	Ω
\forall	A	\exists	Ээ
\ni	Э	\equiv	≡
\cong	≅	\approx	≈
\neq	≠	\propto	×
\leq	≤	\geq	≥
\pm	±	\div	÷
\times	×	\surd	ÿ
\otimes	$\otimes$	\oplus	$\oplus$
\cdot	0	\neg	-
\cap	$\cap$	\cup	U
\subseteq	⊇	\supset	ے ۲
\supseteq	C	\subset	С
\int	ſ	\in	E
\partial	9	\varpi	ω
/o	0	\circ	0
0/	0	\infty	∞
\Im	3	\Re	R
\aleph	х	\wp	<i>{i)</i>
\wedge	^	\vee	V
\ langle	<	\rangle	$\rangle$
\nabla	$\nabla$	\bullet	•
\sim	~	\leftrightarrow	$\leftrightarrow$
\leftarrrow	$\leftarrow$	\rightarrow	$\rightarrow$
\uparrow	1	\downarrow	$\downarrow$
\clubsuit	*	\diamondsuit	•
\heartsuit	•	\spadesuit	٨
\lceil	ð	\rceil	
\floor		\rfloor	
\perp		\mid	
\ldorts		\prime	/
\coopyright	©		



从标注字符串的位置和 x、y 值之间的关系可见, x、y 是插入点在图形的坐标系中的坐标值。我们注意到在标注字符串中使用了特征字符串 " \leftarrow " 和 " \pi ", 分别表示左箭头 " " 和符号 " p "。

标注字符串还具有更加灵活的功能,在字符串中可以使用字符变量或返回值为字符形 式的函数。例如, 下面这组命令:

text(pi,-0.8,['\downarrowcon(\pi)=',num2str(cos(pi))])

text(0,-0.7,['绘图日期:',date])

text(0,-0.85,['MATLAB版本:',version])

在图形窗口的标注结果如图 6-15 所示。

需要注意在几条命令中字符串变量的使用方法

命令 " text(pi,-0.8,['\downarrowcon(\pi)=',num2str(cos(pi))]) " 中 " sin(pi) " 的结果是一个实数,不是字符串,这种情况下不能直接和前面的字符串连接,必须首先用 " num2str " 函数把它转化为字符串格式:

最后两条命令类似, date 函数给出当前日期, version 函数返回使用的 MATLAB 版本, 它们的返回值都是字符串格式,因而不用进行类型转换,可以直接和其它字符串连接。



图 6-15 在标注字符串中使用变量

3.图例

在 MATLAB 中,为了便于对图形的观察和分析,用户可以使用 legend 函数为图形添加图例。legend 函数函数的一般调用格式为:

legend(string1 , string2 , string3 , ...)

只要指定标注字符串,该函数就会按顺序把字符串添加到相应的曲线线型符号之后。

例如,在命令窗口输入如下命令

t=0:0.1:4*pi;

plot(t,sin(t),':',t,cos(t),'*');

legend('Sin Wave','Cos Wave','None')

回车将在图形窗口显示标注结果如图 6-16 所示。



图 6-16 添加图例

从图中可见,图例在缺省情况下自动置于图形的右上角,并自动把线型和标注字符串按顺序排列到一起, 由于图中只有两条曲线,所以,第三个字符串 "None"不对应图中的任何曲线,因而其前面是空的。

MATLAB 还允许很方便地对图例进行调整:用鼠标左键点住图例拖动即可移动图例到需要的位置;用鼠标 左键双击图例中的某个字符串就可以对该字符串进行编辑。

在 legend 函数中还可以加入一个参数也可以对图例的位置作出设置,它的使用格式如下

legend(stringl, string2, string3, ..., Pos)

下面列出了上式中 "Pos"的取值及含义

- > 0 自动把图例置于最佳位置,即和图中曲线重复最少
- ▶ 1 置于图形右上角(缺省值)
- ▶ 2 置于图形左上角
- ▶ 3 置于图形左下角
- ▶ 4 置于图形右下角
- ▶ -1 置于图形右侧(外部)

当 Pos=0 时,每改变一次图形大小,图例都会重新选择一次最佳位置。在命令窗口输入

t=0:0.1:4*pi;

plot(t,sin(t),':',t,cos(t),'*');

legend('Sin Wave','Cos Wave','None', 0)

回车后,图形窗口的显示结果如图 6-17 所示



图 6-17 参数 Pos 为 0 时的图例位置

当 Pos=-1 时,图例会被置于图形外部,这样层次更加分明。在命令窗口输入

t=0:0.1:4*pi;

plot(t,sin(t),':',t,cos(t),'*');

legend('Sin Wave','Cos Wave','None', 0)

回车后,图形窗口的显示结果如图 6-18 所示



图 6-18 参数 Pos 为-1 时的图例位置

用户可以使用 legend off 清除所有图例。

6.1.4 交互式图形

MATLAB 中提供了一些函数,使用户可以和图形方便的进行简单的交互式访问。

1.gtext

MATLAB 提供了一个使用鼠标交互式添加文本的函数 gtext。这个函数的调用格式为 gtext('string', 'PropertyName', Property value,...)

其中的可以是一个字符串,也可以是含有多行字符串的字符串数组。调用这个函数后,图形窗口中的鼠标 指针会成为十字光标,通过移动鼠标来移动十字光标进行定位,光标移动到适当的位置后,单击鼠标或按下键 盘上的任意健,都会在光标所在位置显示指定的文本。

例如,在命令窗口输入命令

gtext({'April','\downarrow','May','\downarrow','June'})

在出现的空白图形窗口中,鼠标箭头会变成十字形的光标,使用鼠标选中适当位置,单击鼠标或按下键盘 上的任意健,都会在光标所在位置显示指定的文本,如图 6-19 所示。

函数 ginput 允许用鼠标获取图形上坐标轴范围内点的坐标。它有如下三种调用方式

- ▶ [x,y]=ginput(n) 从图形中获取 n 个点的坐标值,获得的数据保存在长度为 n 的向量" x "(横坐标) 和 " y "(纵坐标)中。
- ▶ [x, y]=ginput 从图形中获取任意个点的坐标,直到按下回车键为止。
- ▷ [x, y, button]=ginput(n) 返回值增加了 "button"向量,该向量中的元素为整数,反映选取数据 点时按下了哪个鼠标键(左、中、右键分别对应 1、2、3)或返回使用的键盘上的键的 ASCII 码值。

调用了 ginput 函数后,在图形窗口中鼠标箭头会变成十字形的光标。移动鼠标,光标随之移动,在关心的 数据点上单击鼠标左键,该点的坐标就被记录下来,直到点数达到指定的个数或按下回车键终止取值为止。

🤌 Fig	ure No							_ 🗆 🗙
File	<u>E</u> dit	View	<u>I</u> nsert	Tools	Window	Help		
D 🗳		λA	718	େଇ				
	1					1		
	0.9 -							
	~ ~							
	0.8 -			April				
	0.7 -			↓ Mav				
				↓ ↓				
	0.6 -			June				
	0.5 -							
	0.4							
	0.4							
	0.3 -							
	0.2							
	0.2							
	0.1 -							
	0				1			
	Ő		0.2	0	.4	0.6	0.8	1

图 6-19 使用 gtext 函数添加字符串

6.2 三维图形

MATLAB 具有强大的 3 维图形功能,包括 3 维数据显示、空间曲线、曲面、分块及填充,以及曲面光顺着 色、视点变换、旋转、隐藏等功能和操作。本节将分别对上述功能予以详细的介绍。

## 6.2.1 基本绘图命令简介

MATLAB 的基本系统中主要包括表 6-9 种列出几个三维图形函数:

表 6-9	9
-------	---

```
基本三维绘图命令
```

命令	说明
plot3	这是与平面线型绘图函数 plot 对应的 3 维绘图函数
contour, contour3	生成矩阵数据图形或函数曲面的等高线图形
pcolor	根据矩阵的数值对平面的小矩形区域着伪色,可以用来标识矩
	阵元素的大小分布
image	用于在平面坐标系上显示一幅图像,该图像的像素的灰度值或
	颜色素引号由该函数的输入矩阵变量决定,该矩阵称为颜色矩
	阵。一般使用当前的图形窗口色谱对图像像素着色
mesh, meshc, meshz	生成3维网格曲面
surf, surfc, surfz	与上述的网格曲面生成函数相对应.其差别在于这些函数还将

命令	说明
	对网格曲面进行着色处理.得到视觉效果更好的彩色曲面
fill3	生成3维的多边形区域并按一定的方式填充区域内部
view	设置3维坐标系的视点

6.2.2 三维图形的建立

1. 三维线型图形

函数 plot3 是函数 plot 在三维时的调用方式,用来生成 3 维空间的折线或曲线, plot3 和二维绘图函数 plot 相比区别只在于 plot3 多了第三维数据,调用格式都相同。它主要有以下几种调用方式。

(1) 向量式 plot3(x,y,z)

这是最基本的调用形式,其中 x、y 和 z 分别为第一到三维数据,都是长度相同的向量,它的功能是在 3 维 坐标系中,生成顺序连接顶点((x(i),y(i),z(i)))的折(曲)线。s 是设置线型、颜色、数据点标记的字符串。

例如,在命令窗口输入命令

t=0:0.1:8*pi;

plot3(sin(t),cos(t),t)

xlabel('sin(t)','FontSize',12)

ylabel('cos(t)','FontSize',12)

zlabel('t','FontSize',12)

回车将在图形窗口生成 3 维坐标系中的螺旋线,并为其坐标轴加上标注,如图 6-20 所示。想象如果图 6-20 中的 z 轴(t)去掉,也就是相当于从上往下看这幅图,那么它将是一个圆,这就和前面曾讲过的 plot(sin(t), cos(t)) 绘制的曲线相同,由此可见, plot3 实际上就是二维函数 plot 在三维空间上的扩展。



图 6-20 三维螺旋线

(2) 矩阵式 plot3(X,Y,Z)

在这种调用方式中,X、Y、Z都是维数相同的矩阵,它按3个短阵的列向量对应他生成多条折(曲)线。每条折(曲)线的颜色由 MATLAB 自动设置,与 plot 函数一样。另外,变量X、Y、Z中可以有一个或两个是向量。 这时,向量的长度必须等于矩阵变量的行数或列数。MATLAB 按长度对应的方式选择矩阵的行或列向量来生成 折(曲)线。

例如,在命令窗口输入命令行
[X,Y]=meshgrid(-pi:0.1:pi);
Z=sin(X)-cos(Y);
plot3(X,Y,Z)
这里的函数 meshgrid 用于从向量生成方阵,生成的矩阵 X 的行和 Y 的列相同,矩阵 X 的每一行都是向量

"-pi:0.1:pi",行数和列数相等。绘制的三维曲线如图 6-21 所示。在彩色显示方式下,这些曲线是用多种颜色绘制的,相邻曲线的颜色都不相同,便于区分。



图 6-21 参数为矩阵的三维图形

(3)标准式 plot3(x,y,z,s)

与 plot 相类似,用户可以设置绘制折(曲)线的线型、顶点标记和颜色。plot3(x,y,z,s)中的 s 就是上节中介绍的线型与颜色或顶点标记与颜色的字符串变量,这是比较一般的调用方式。更为一般的调用形式即所谓的 4 元 组形式:

plot3(X1,Y1,Z1,S1,X2,Y2,Z2,S2....)

其中,Xi,Yi、Zi可以是矩阵或向量,而Si是类型字符串变量。每一组(Xi,Yi,Zi)必须满足维数的要求,而不同的组之可矩阵的维数可以不同。在不至混淆的情况下,Si可以被省略,这时、线型类型由 MATLAB 按缺省的方式自动设定。

2. 三维曲面

(1) 平面网格点的生成

所谓网格图,是指把相邻的数据点连接起来形成网状曲面。

在数学上,函数 z = f(x,y)的图形是 3 维空间的曲面,MATLAB 是如何实现 3 维函数曲面的绘制的呢?在 MATLAB 中,总是假设函数 s = f(x, y)是定义在一个矩形的区域  $D = [x0,xm] \times [y0,yn]$ 上的。为了绘制在区域 D 上的 3 维曲面,MATLAB 的方法是首先将[ $x0 \cdot xm$ ]在 x 方向分成 m 份,将[ $y0 \cdot yn$ ]在 y 方向分成 n 份,由各分 划点分别作平行于坐标轴的直线,将区域 D 分成 m×n 个小矩形块,计算出在网格点的函数值。对于每个小矩 形,在空间中决定出 4 个顶点(xi,yj,f(xi,yj)),连接 4 个顶点得到一个空间中的四边形片。所有这些四边形片 一起构成函数 z = f(x, y)定义在区域 D 上的空间网格曲面。

为方便起见, MATLAB 用函数 meshgrid 来生成 x-y 平面上的小矩形顶点坐标值的矩阵。函数 meshgrid 的调用形式是

[X,Y]=meshgrid(x);

或者

[X,Y]=meshgrid(x,y)

第一种形式等价于 meshgrid(x,x)。这里 x 是区间[x0,xm]上分划点组成的向量,而 y 是区间[y0,yn]上分划 点组成的向量。输出变量 X 和 Y 都是矩阵,矩阵 X 的行向量都是向量 x,Y 的列向量都是向量 y。于是,X 和 Y 的元素组(X(i,j),Y(i,j))恰好是区域 D 的第(i,j)网格顶点。例如,(X(1,1).Y(1,1))对应于(x0,y0),而[X(m+1, n+1)、Y(m+1,n+1))对应于(xm,yn)。换句话说函数 meshgrid 将由两个向量决定的区域转换成对应的网格点矩 阵。

下一步的工作就是计算出所有网格点处的函数值。由于矩阵 X 和 Y 的对应元素恰好组成某个网格点,因此, 利用 MATLAB 的矩阵运算能力,可以很容易地求出所有网格点上的函数值组成的矩阵。下面通过例子来说明其 具体做法。 考虑数学函数  $z = \sin(\sqrt{x^2 + y^2})/\sqrt{x^2 + y^2}$ ,函数定义在区域[-8,8]×[-8,8]上。在生成网格点后,为了计算网

格点上的函数值,需按下列方法进行

[X,Y]=meshgrid(-8:0.5:8);

R=sqrt(X^.2+Y^.2)+eps;

Z=sin(R)./R

由于在邻近原点处, R 的某些元素可能会很小. 加入 eps 是避免出现零除数。

(2) 网格曲面

在得到网格点上的函数值矩阵后,即可以用 MATLAB 函数 mesh 来生成函数的网格曲面,即各问格线段组成的曲面。例如对上述的例子,在命令窗口输入

## mesh(X,Y,Z)

回车在图形窗口显示如图 6-22 所示的图形。



图 6-22 网格曲面

函数 mesh 有多种调用形式,在固 2-12 中, x 轴和 y 轴是按矩阵 Z 的行列序号定义的。为了使其与该函数的 定义域一致,需要使用其他的调用方式。下面分别给予介绍。

- mesh(X、Y,Z,C) 这是最一般的调用形式,X、Y、Z、C是同维数的矩阵,C称为颜色矩阵。 网格曲面的顶点对应于空间的顶点(X(i,j),Y(i,j),Z(i,j),而网络曲面的网格线的颜色由C值根据 当前的色谱来着色。这种调用形式还可以用来生 成参数曲面片;
- ▶ mesh(X,Y,Z) 这是在上一种调用形式中,取C=Z的简单调用形式;
- mesh(x, y, Z, C) 其中, x和 y 是向量, Z和 C 是同维数的矩阵, 且向量 x 的长度等于矩阵 Z 的列数, 而向量 y 的长度等于短阵 Z 的行数。在这种情况下、网格曲 面的网格顶点是(x(j), y(i); Z(i,j)), 网格线的颜色由矩阵 C 决定;
- ▶ mesh(x, y, Z) 这是在上一种调用形式中, 取C=Z的简单调用形式;
- mesh(Z, C) Z和C都是m×n的矩阵、该形式与mesh(x, y, Z, C)等价,其中向量 x=1:n,
   向量 y = 1:m;
- ▶ mesh(Z) 这是在上一种调用形式中, 取 C = Z 的简单调用形式。

与 mesh 相关的另外两个函数是 meshc 和 meshz,它们的调用形式与 mesh 相同,meshc 除了生成网格曲面外,还在"x-y平面上生成曲面的等高线图形,如图 6-23 所示。



图 6-23 meshc 绘制的三维网格曲面

而函数 meshz 的作用除了生成与 mesh 相同的网格曲面之外,还在曲面下面加上一个长方体的台柱,使图形更加美观,如图 6-24 所示。

在图 6-22 中,网格线之间的区域是不透明的,因而显示的网格只是前面的部分,被遮住的部分没有显示出来。MATLAB 用 hidden 函数控制网格图的这个特性

hidden on 表示隐藏被遮住的部分 hidden off 表示显示遮住的部分。例如,在绘制图 6-22 命令之后加上命令 hidden off 结果将如图 6-25 所示。



图 6-24 meshz 绘制的三维网格曲面



图 6-25 显示被遮住的部分

MATLAB 总是在平面的矩形区域上表达曲面,要绘制其他区域上的曲面可以使用 MATLAB 的曲面裁剪功 能来实现。在 MATLAB 中有两种实曲面的裁剪技巧

- 一种是利用着色原理将要裁剪的部分着当前的图形窗口的背景色,使其在坐标系中不可见。这种 方法的技巧性较高,而且用这种方法裁剪的部分在图形打印时是可见的;
- 另一种方法是使用 MATLAB 定义的特殊数 NaN,这个特殊数的一个重要性质是,它可以用于表示那些不可使用的数据。可利用这种待性进行曲面的裁剪。这种方法的原理是,将要裁剪的曲面片部分对应的矩阵(或函数值)元素设置为 NaN,然后,或者特该矩阵作为图形函数的矩阵变量,或者作为图形函数的颜色短阵进行作因,即可以达到曲面裁剪的目的。

例如,读者可以看到下列语句生成的图形,在命令窗口输入

p=peaks;

p(30:40,20:30)=NaN+p(30:40,20:30); mesh(p) 结果如图 6-26 所示:



图 6-26 曲面裁减图

对于函数  $z=x^2+y^2$  在区域  $x^2+y^2<9$  上的曲面片,可以在命令窗口输入 [x,y]=meshgrid(-3:0.1:3); z=x.^2+y.^2; [I,J]=find(x.^2+y.^2>9); for ii=1:length(I) z(I(ii),J(ii))=NaN; end surf(x,y,z) 显示结果如图 6-27 所示。 3. 实曲面的绘制 实曲面就是对网格曲面的网格块(四边形片)区域进行着色的结果。

(1) surf

MATLAB 函数 surf 可提供这种功能,它的调用方式与 mesh 完全一样。 surf 的曲面生成过程与 mesh 是类 似的,所不同的是 mesh 仅对网格线进行着色,surf 是对网格片进行着色,而网格线用黑色标出(可以修改)。 通常,surf 用缺省的着色方式对曲面片着色。

例如,在命令窗口输入如下一组命令

[X,Y]=meshgrid(-8:0.5:8);

R=sqrt(X.^2+Y.^2)+eps;

Z=sin(R)./R;

surf(X,Y,Z)

将建立填充了网格的草帽图,如图 6-28 所示



图 6-27 区域剪裁曲面



图 6-28 surf 绘制的实曲面图

可以看到曲面图的线条色为黑色,补片是彩色的,这也是它和网格图的区别所在。

(2) shading

有时想要得到曲面图的整体效果,那么其中的线条应该去掉,并可以对颜色作平滑和插值处理,这需要借助函数 shading。shading 函数有三种用法

- ▶ shading flat 去掉各片连接处的线条,平滑当前图形的颜色。
- ➢ shading interp 去掉连接线条,在各片之间使用颜色插值,使得片与片之间以及片内部的颜色 过渡都很平滑。
- ➢ shading faceted 缺省值,带有连接线条的曲面。

如果在建立图 6-28 的命令后加上命令行

shading flat

那么将产生如图 6-29 所示的效果,网格去掉了,但片与片之间的过渡还比较明显。



图 6-29 平滑效果图

若加上如下命令 shading interp

axis off

那么将产生一个非常平滑、完整的效果,如果 6-30 所示。



图 6-30 颜色插值后的平滑效果

(3) surfc和 surfl

函数 surf 也有两个类似的函数:surfc 和 surfl。下面的例子说明了这两个函数的功能。

仍使用前面的例子,在命令窗口中输入如下命令

[X,Y]=meshgrid(-8:0.5:8);

R=sqrt(X^.2+Y^.2)+eps;

Z=sin(R)./R

surfc(X,Y,Z)

图形窗口将绘制出具有基本等高线的表面图,如图 6-31 所示。





将上述命令行中的 surfc 换成 surfl 将绘制出具有光照效果的表面图, 如图 6-32 所示。



图 6-32 surfl 产生的曲面图

(4) surface

MATLAB 还提供了一个建立表面对象的低级函数:surface。该函数用于把表面对象添加到当前坐标系中。 例如,在命令窗口输入

[X,Y]=meshgrid(-8:0.5:8);

R=sqrt(X.^2+Y.^2)+eps;

Z=sin(R)./R;

mesh(X,Y,Z)

surface(X,Y,Z)

命令的执行结果和调用 surf 函数是一样的,也生成如图 6-26 所示的彩色表面图。

但如果在调用 surface 函数之前没有用 mesh 命令生成网格图 ,那么直接运行 surface 的结果将如图 6-33 所示 , 这实际上是图 6-26 中的表面图向 x-y 平面的投影。



图 6-33 直接执行 surface 函数的结果

#### 6.2.3 调整视角

所谓视角简单讲就是观察(显示)图形的方向。调整视角可以使一幅图显示出来自不同方向的观察效果。

设置视角的基本函数是 view,它有如下几种不同的调用方式:

- ➢ view(az,e1)或 view([az, e1]) 设置观察图形的视角
- ▶ view(2) 设置缺省的二维视角
- ▶ view(3) 设置缺省的三维视角
- ➤ [az, e1]=view

返回当前的视角。

下面我们就来介绍如何用第一种调用方式来设置图形的视角。式中的"az"是方位角(azimuth),指在 x-y 平面内从 y 轴负方向绕 z 轴旋转的角度,逆时针方向为正;"el"是仰角(elevation),指从 x-y 平面沿 z 轴方向仰起的角度,向 z 轴正方向的仰角为正。"az"和"el"的单位都是"度"。

二维图形视角的缺省值是: az=0 度, e1=90 度。

三维图形视角的缺省值是: az=37.5 度, e1=30 度。

例如,首先建立表面图,在命令窗口输入

[X,Y,Z]=peaks(30);

1. 用函数 view 调整视角

surf(X,Y,Z);

xlabel('x 轴');

ylabel('y 轴');

zlabel('z 轴');

如图 6-34 所示,该图的视角为缺省值:az=37=5,el=30。



图 6-34 缺省视角

下面使用命令 view 来调整视角,在命令窗口输入

view(90,0)

该命令把方位角在 x-y 平面内从 y 轴负方向逆时针旋转 90 度,转到了 x 轴的正方向,而仰角为 0 度,也就 是说视线从 x 轴的正方向水平看过去的效果,如图 6-35 所示。

在命令窗口输入命令行

view(-37.5,80)

把方位角恢复为缺省值,仰角设为80度,也就是从斜上方往下看,这时的效果如图6-36所示。

在命令窗口输入 view(0,90)

设置观察方向为从正上方向下看,如图 6-37 所示。

用 view 函数靠调整方位角和仰角来调整观察点是有一定限制的,它不能指定观察点的距离,只能指定方向, 而且 z 轴总是指向上的。

要实现对观察点的高级控制,还需要借助于下面讲到的 MATLAB 照相制图技术(camera graphics)。



图 6-35 方位角为 90 度, 仰角为 0

图 6-36 方位角-37.5 度, 仰角 80 度



图 6-37 方位角为 0 度, 仰角为 90 度

## 2.照相制图技术

为了实现对观察点的控制, MATLAB 提供了类似照相机变焦透镜的高级技术, 这种技术由表 6-10 中的函数 实现。

#### 表 6-10

#### 照相制图函数

命令	说明
camdolly	移动照相机的位置和被照目标
camlookat	移动照相机和目标以观察指定对象
camorbit	在被照目标周围旋转照相机
campan	在照相机的位置周围旋转被照目标
campos	设置或获取照相机的位置
camproj	设置或获取投影类型(正交或远景)
camroll	绕着观察坐标轴旋转照相机
camtarget	设置或获取被照目标的位置
camup	设置或获取照相机向上向量的值
camva	设置或获取照相机视角的值
camzoom	推近(放大)或拉远(缩小)照相机

例如,对于上例中建立一个缺省情况下的三维高斯分布曲线

	surf(X,Y,Z);			
	xlabel('x 轴');			
	ylabel('y 轴');			
	zlabel('z 轴');			
	参见图 6-32,	下面首先获取	Q当前照相机的位置和视角,在命	i令窗口输入命令
,	campos, canmv	a		
	回车显示结果			
,	ans=			
	-36.5257	-47.6012	86.6025	

ans=

10.3396

命令 campos 返回的值是照相机的位置坐标[x,y,z1, camva 返回的是照相机观察目标的视线夹角大小。

下面首先把照相机的位置移到和当前位置关于坐标原点对称的地方,在命令窗口输入 campos=[36.5257, 47.6012, -86.6025D

结果如图 6-38 所示。



图 6-38 移动照相机的位置

在上一步基础上,把观察的角度减小,从而相当于放大了目标,在命令窗口输入命令 camva(5) 结果如图 6-39 所示。



图 6-39 减小视角

下面再把放大了的目标绕观察轴旋转 135 度,命令如下

camroll(135) 结果如图 6-40 所示。



图 6-40 沿观察轴旋转

从上例可以看出,照相制图技术功能要强大很多,如果能够灵活运用,会得到逼真的效果。

#### 6.2.4 光照控制

为图形加上光源能够得到更加逼真的效果。MATLAB 提供了放置光源和调整光照目标特性的一些函数,如表 6-11 所示。

表 6-11

光源设置函数

命令	说明
camlight	根据照相机的位置建立或移动光源
lightangle	在球形坐标系中建立或移动光源
light	建立光源对象
lighting	选择光照方法
material	设置被照目标的反射特性

## 合理地运用这些函数会得到很真实的视觉效果。

1.建立光源

 $\triangleright$ 

在设置光照效果之前要首先建立光源,可以用 light、camlight 或 lightangle 函数来完成。light 函数一般有三种调用方式:

在建立光源的同时设置参数

- ▶ light 使用缺省值建立光源。
- light(paraml,valuel...,paramn,valuen)

H=light(...)

返回建立光源对象的句柄

camlight 函数用于建立类似照相时的光源,其调用方式有

- ➢ camlight headlight 在当前坐标系中照相机的位置建立光源
- ➤ camlight right 在照相机的右上方建立光源
- ▶ camlight left 在照相机的左上方建立光源
- ▶ camlight 缺省情况,在照相机的右上方建立光源
- ▶ camlight(az,e1) 在相对照相机方位角为 "az", 仰角为 "el"的位置建立光源
- ➤ camlight(...,style) 设置光源的类型,可以是"local"(缺省值)或"infinite"
- ▶ camlight(h,...) 把指定的光源放在指定的位置
- ▶ h=camlight(...) 返回光源对象的句柄。

lightangle 函数是在球形坐标系中建立光源,用法如下

- ➢ lightangle(az,e1)
  在指定位置建立光源
- ▶ h=lightangle(a2,e1) 建立光源并返回其句柄
- ➢ lightangle(h,az,e1) 设置指定光源的位置
- ➢ [az el]=lightangle(h) 获取指定光源的位置

例如,首先建立一个三维高斯分布表面图,然后用 light 函数在指定位置添加光源,效果见图 6-41。

[x,y,z]=peaks(30);

surf(x,y,z);

axis tight

light('Position',[-4 -6 10]);



图 6-41 用 light 函数加光源

下面的命令首先改变图形的视角,然后用 lightangle 函数指定光源位置,效果见图 6-42。

[x,y,z]=peaks(30); surf(x,y,z); axis tight view(90,0); lightangle(45,30);



图 6-42 用 lightangle 函数加光源

2. 设置照明方式

在为图形添加了光源后,还要考虑选取合适的照明方式,不同的照明方式效果也不一样。设置照明方式由 lighting 函数完成,MATLAB 提供了三种照明方式,分别由 lighting 函数的不同调用方式来设置

- lighting fiat
  - 均衡图形中每个小表面的交叉颜色,这是缺省方式
- ➢ lighting gouraud 对小表面的交叉颜色做颜色插值
- ➢ lighting phong 对小表面的交叉颜色做颜色插值,并计算每个像素的反射比
- ▶ lighting none 关闭照明。

下面的例子中可以看出几种方式的区别,首先绘制一个球体,去掉网格,然后用 camlight 命令在照相机的 左上方和右上方各设置一个光源,最后用 lighting 命令指定照明的不同方式。为便于比较,把四种情况用 subplot 函数合到一幅图中,如图 6-43 所示。在命令窗口输入

subplot(2,2,1); sphere shading flat camlight left camlight right lighting flat subplot(2,2,2); sphere shading flat camlight left camlight right lighting gouraud subplot(2,2,3); sphere shading flat

camlight right

camlight right lighting phong

subplot(2,2,4); sphere shading flat camlight left camlight right lighting none

subplot(2,2,1) title('Flat Mode'); subplot(2,2,2) title('Gouraud Mode');; subplot(2,2,3) title('Phong Mode');; subplot(2,2,4) title('None Mode');



图 6-43 三种照明方式的比较

6.3 特殊图形

在很多研究领域中,常常为满足一些特殊的要而需要采用特殊的图形,MATLAB为此设计了几种专门用于 绘制特殊图形的函数,使得这项工作变得非常简单。

# 6.3.1 条形图

条形图常用于统计数据的作图。绘制条形图的函数如表 6-12 所示。

表 6-12

冬形图函数	

命令	说明
bar	数值条形图
barh	水平条形图
bar3	三维数值条形图
bar3h	三维水平条形图

这几个函数的调用方法类似,以函数 bar 为例,它有以下几种用法

- ▷ bar(X,Y) "X"是横坐标向量。"Y"可以是向量或矩阵,"Y"是向量时,每一个元素对应 一个竖条;"Y"是 m行 n 列的矩阵时,将画出 m 组竖条,每组包括 n 个条。
- ▶ bar(Y) 横坐标使用缺省值 1:1:m。
- ▷ bar(X,Y,width)或 bar(Y,width) 用 "width "指定竖条的宽度;缺省宽度是 0.8。如果宽度大于 1, 那么条与条之间将重合。
- ▹ bar(...,'grouped') 产生组合的条形图。
- ➢ bar(...,'stacked') 产生堆叠的条形图。
- ▹ bar(...,linespec) 指定条的颜色。
- ▶ h=bar(...) 返回补片对象的句柄向量。

## 例如,对于向量

y=5 2 1;8 7 3;9 5 6;5 1 5;4 3 2;

# 在命令窗口输入

bar(y)

显示结果如图 6-44 所示。



图 6-44 二维条形图

入在命令窗口输入 bar3(y) 显示结果如图 6-45 所示。



图 6-45 三维条形图

如果选择 "group "、" stack " 选项,分别显示结果如图 6-46、图 6-47。



图 6-46 "group"选项的效果





## 6.3.2 直方图

直方图和条形图的形状类似,但作用不同,它主要用于显示数据的分布规律。用于建立直方图的函数有 hist 和 rose,其中函数 hist 在直角坐标系中建立直方图,函 数 rose 在极坐标系中建立直方图。

函数 hist 有如下几种主要用法

- ▷ N=hist(Y) 把"Y"按其中数据的大小分为 10 个长度相等的段,统计每段中的元素个数并返回 给"N"。如果"Y"是矩阵,那么按列分段。
- ▶ N=hist(Y,m) "m"是标量,用来设置分段的个数。
- ▶ N=hist(Y,X) "X"是向量,用于指定所分的每个数据段的中间值。
- ▶ hist(...) 不带输出参数时,直接绘制直方图。

例如,考虑随机数的分布规律,函数 rand 用于产生具有均匀分布规律的随机数,函数 randn 用于产生具有 正态分布的随机数,使用下列命令产生不同的随机数并绘制直方图,来看一下二者的区别。在命令窗口输入

y1=rand(10000,1); hist(y1,20)

y2=randn(10000,1);

# hist(y2,30)

图 3-48 显示的是均匀分布随机序列的分布情况,图 3-49 是正态分布的随机序列。



图 6-48 均匀分布随机序列





如果"Y"是矩阵,我们来看结果如何:

y=randn(10000,2);

hist(y)

上述命令产生了一个 10000 行 2 列的正态分布随机数矩阵, 它的分布规律的表示方式见图 6-50 所示。



图 6-50 两列随机数的分布直方图

在极坐标中建立直方图的函数是 rose。rose 用法和 hist 类似,只是把数据作为弧度值处理,在极坐标系中建 立直方图。例如,在命令窗口输入

a=randn(1000,1); a=pi*a/max(a); rose(a) 式中的 a 示归一到区间[-p, p]上的随机弧度值,随即数正态分布,这时的结果如图 6-51 所示。 1 - 🗆 × File Edit View Insert Tools Window Help |口 译日番| トムオノタのつ 90 150 120 60 100 150 180 210 230 240 300 270

图 6-51 极坐标中直方图

6.3.3 面积图

在 MATLAB 中,函数 area 用于绘制面积图,其输入参数为向量或矩阵。函数 area 画出向量或矩阵的列向 量中的数据,连成线,并填充 x 轴和曲线之间的面积。面积图特别适合查看一个数在该列所有数总和中占的比 例。

```
例如,在命令窗口输入
y=[542;867;998;585;453];
area(y)
结果显示如图 6-52。
```





从下例中可以看出面积图在比较数据方面的作用。某公司 1994 年至 1998 年的销售额如向量 sales 所示,成本如 expense 向量所示。该程序将成本和销售额画在同一坐标轴内,并指定了"FaceColor"、"EdgeColor"、 "LineWidth"三种属性,绘制结果如图 6-53 所示。

sale=[51.6 82.4 90.8 59.1 47.0]; x=94:98; expense=[19.3 34.2 61.4 50.5 29.4]; area(x,sale,'FaceColor',[0.5 0.9 0.6],'EdgeColor','b','LineWidth',2) hold on area(x,expense,'FaceColor',[0.9 0.85 0.7],'EdgeColor','y','LineWidth',2) hold off set(gca,'XTick',[94:98]) set(gca,'Layer','top') gtext('\leftarrow 销售额') xlabel('年份')

ylabel('销售额 in 1,000"s')



图 6-53 面积图的标注

6.3.4 饼图

命令 pie 和 pie3 用于绘制饼图。饼图便于比较一个整体中各个元素所占比例。命令 pie 和 pie3 的输入参数 均为向量或矩阵,如果输入参数为向量,则绘制各个元素在向量所有元素之和中所占比例;当输入参数为矩阵 时,绘制出各个元素在矩阵所有元素之和中所占比例。命令 pie 用于绘制平面饼图,命令 pie3 用于绘制三维饼 图。

饼图命令中参数 "explode "用于指定讲图中的某些片是否和整个饼图脱开。参数 "explode "和函数的输入数据同维,当 "explode "中的元素值非0时,该元素对应的片和整个图脱开。下面为一个饼图的例子。

某学生的四门课的成绩如 x 变量所示,先后顺序为数学、语文、英文和物理。在饼图中,由于 MATLAB 在 绘制的时候在饼图的每一条的旁边标注了百分比,标注图形时只需修改这些文本即可。

本例介绍的标注方法虽然比较烦琐,但熟悉这些命令对于标注复杂图形是有好处的。本例在标注时,先查 找出图形上的文本对象中的字符串和字符串的长度,修改字符串这个属性。由于字符串的变化,也应修改字符 串的位置。图形绘制结果如图 6-54 所示。

```
x=[78 89 61 98];
explode=zeros(size(x));
[c,offset]=min(x);
h=pie(x,explode);
textObjs=findobj(h,'Type','text')
oldStr=get(textObjs,{'String'});
Names={'数学 78';'语文 89';'英文 61';'物理 98'};
set(textObjs,{'String'},Names);
```



图 6-54 饼图的绘制和标注

还可以使用函数 pie3 绘制三维饼图,例如,在命令窗口输入 sale=[100 150 400 250]; pie3(sale,[0 0 1 0],{'Spring','Summer','Autumn','Winter'});

显示结果如图 6-55 所示。



图 6-55 三维饼图

如果需要把图中的一部分区域去掉,首先需要把绘图数据归一化,每个数据用其所占的百分比表示,然后 把不显示的区域数据去掉。例如,在数据窗口输入

sale=[0.11 0.17 0.44 0.28]; pie(sale(2:4)); 显示结果如图 6-56 所示。



图 6-56 去掉一块的饼图

## 6.3.5 火柴杆图

函数 stem 用于绘制火柴杆图。火柴杆图把每个数据点画成一根直线,在数据值的地方,用黑点表示,看起 来很像火柴杆。在二维火柴杆图中,火柴杆从 x 轴开始,在三维火柴杆图中,火柴杆从 x-y 平面开始。

下面的例子中,在横坐标 0~2 p 之间均匀形成 40 个数据点,用不同的线型和颜色绘出图形。在程序中,命 令 legend 用于设置图标。函数 stem 绘图时产生两个句柄,第一个是竖直杆上数据点图标的句柄,第二个是竖直 杆的句柄,取出第一个句柄生成图标。整个程序段的绘制结果如图 6-57 所示。

x=linspace(0,2*pi,40); a=sin(2*x); b=cos(x); stem_handles=stem(x,a+b); hold on plot_handles=plot(x,a,'-r',x,b,'-g'); hold off legend_handles=[stem_handles(1);plot_handles]; legend(legend_handles,'a+b','a=sin(2x)','b=cos(x)'); xlabel('时间(单位:秒)'); ylabel('喃值'); title('两个函数相加');



图 6-57 火柴杆图

函数 stem3 用于绘制三维空间中的火柴杆图。当数据较少时,这种图的效果很好。下面是一个螺旋线的例子,绘制出的螺旋线如图 6-58 所示。

t=(0:127)/128*2*pi; x=t.*cos(t); y=t.*sin(t); stem3(x,y,t,'fill'); xlabel('x'); ylabel('y'); zlabel('z'); title('火柴杆图示例'); hold on plot3(x,y,t,'k'); hold off



图 6-58 空间火柴杆图

## 6.3.6 阶梯图

在中,可以使用函数 stairs 绘制阶梯图。画曲线图时,相邻数据点之间用直线连接;在阶梯图中,相邻两点 之间的区间的值全为两点中起点的值,两点间用阶梯线相连。阶梯图对于绘制数字系统的时程曲线比较有用。 下面为一个阶梯图的例子。例中,同时绘出了曲线图和阶梯图,以供比较,如图 6-59 所示。

```
alpha=0.01;
beta=0.5;
t=0:10;
f=exp(-alpha*t).*sin(2*beta*t);
stairs(t,f);
hold on
plot(t,f,'-*');
hold off
label='函数阶梯图'
xlabel('t=0:10','FontSize',14);
text(0.5,0.2,label);
```



图 6-59 阶梯图

6.3.7 矢量图

矢量图在工程中的应用比较广泛。在 MATLAB 中,函数 compass、feather、quiver 和 quiver3 用于绘制矢量 图。使用这些函数时,如果只输入一个参数,则把参数的实部当作矢量的 x 分量,把虚部当作矢量的 y 分量; 当输入参数为两个时,则把第一个参数当作 x 分量,第二个参数当作 y 分量。

函数 compass 把所有指定的矢量,从同一始点绘制。图形中显示环向的栅格,函数名的英文字面意思为罗 盘。图形中每个矢量相当于一根指针。函数 feather 指定矢量的起点为水平轴上等间距的点。调用函数 quiver 和 quiver3 时,要指定每个矢量的起始点的坐标。函数 quiver3 用于绘制三维空间中的矢量图。

下面为这四个函数的使用示例。例中,函数 compass 绘制由风向和风的强度所确定的矢量,函数 feather 把极坐标中的数据转换成直角坐标中的数据。

## 在命令窗口输入

direction=[45 90 90 45 360 335 360 270 335 270 335 335];

strength=[10 6 8 6 3 9 10 8 9 10 14 12];

rdir=direction*pi/180;

[x,y]=pol2cart(rdir,strength);

compass(x,y);

结果如图 6-60 所示。



图 6-60 compass 函数绘图示例

#### 在命令窗口输入

t=90:-2:0; r=ones(size(t)); [u,v]=pol2cart(t*pi/180,5*sin(t)); feather(u,v); axis equal 结果如图 6-61 所示。

函数 quiver 用于绘制二维矢量,函数 quiver3 用于绘制三维矢量。在下面的例子中先绘制出等高线图,然后 在图上绘出各点的梯度矢量,在命令窗口输入

[X,Y,Z]=peaks; contour(X,Y,Z,15); [U,V]=gradient(Z,0.4); hold on quiver(X,Y,U,V);





图 6-61 函数 feather 绘图示例



图 6-62 函数 quiver 绘图示例

假定飞机飞行速度为:vx=2t , vy=2t² , vz=6t³-t² , 其中 t 为时间 , 绘制飞机飞行轨迹上各点的加速度矢量。 在命令窗口中输入

t=0:0.1:1; vx=2*t; x=vx.*t/2; vy=2*t.^2; y=vy.*t/3; vz=6*t.^3-vy/2; z=6*t.^3.*t/4-vy/2.*t/3; u=gradient(x); v=gradient(y); w=gradient(z); scale=0; quiver3(x,y,z,u,v,w,scale); axis square 所绘图形如图 6-63 所示。



图 6-63 函数 quiver3 的绘图示例

#### 6.3.8 等高线图

在 MATLAB 中,函数 clabel、contour、contour3、contourf 用于绘制等高线图。函数 meshc 和 surfc 也可在 绘制网线图和表面团的同时绘制等高线图。

函数 contour 和 contour3 分别用于绘制二维和三维的等高线图。这两个函数自动根据 Z 轴的最大值和最小值 确定等高线的条数,也可以指定等高线的条数。例如,指定等高线的条数为 30,在命令窗口输入

[X,Y,Z]=peaks;
contour(X,Y,Z,30);
绘制结果如图 6-64 所示。
如果使用三维等高线命令,
[X,Y,Z]=peaks;
contour3(X,Y,Z,30);
绘制结果如图 6-65 所示。



图 6-64 二维等高线图



图 6-65 三维等高线图

顾名思义,每条等高线上点的高度相等,在调用函数 contour 和 contour3 时,可以返回每条等高线的高度值 和句柄。函数 clabel 利用返回的高度值和句柄给等高线标注(只标注那些有足够空间标注的等高线)。函数 calbel 可返回所标注文本的句柄。下面语句标注函数 peaks 产生的 5 条等高线

z=peaks; [c,h]=contour(z,5); >> clabel(c,h); title('用函数 clabel 标注'); 结果如图 6-66 所示。



图 6-66 标注等高线图

也可使用命令 clabel 的"manual"选项,来指定需要标注的等高线。运行时,光标将在等高线图上变成十字,将光标移到某条等高线内单击,则标注该条等高线,其命令格式为

clabel(c,h,'manual') 函数 contourf 绘制二维等高线,并在等高线之间填充不同的颜色。下面的例子将绘出函数 peaks 生成的等高线,函数 caxis 用于匹配填充色,在命令窗口输入

z=peaks; [c,h]=contourf(z,15); caxis([-25 25]); title('绘制带填充色的等高线图');
图形如图 6-67 所示。



#### 图 6-67 着色等高线图

在绘制等高线图的命令中,可以指定等高线的数目,还可以指定等高线的值,即在那些高度值处绘制等高 线。下面以命令 contour 为例来说明高度值的指定。指定等高线数目的调用方法为:contour(z,n),其中 n 为标量。 指定等高线值的调用方法为:contour(z,v),其中 v 为向量,函数将在向量 v 中元素指定的高度处画等高线,等 高线条数为向量 v 的元素个数。当只指定一条等高线的高度时,向量 v 应为元素值相等的二元向量,元素值即 为指定高度,而不能用标量表示。

[X,Y]=meshgrid(-3:0.125:3); z=peaks(X,Y); contour3(X,Y,z,[3.5 3.5]);; hold on contour3(X,Y,z,[4 6]); hold off

在上面的绘制等高线图的所有函数中,都使用函数 contourc 来计算等高线。函数 contourc 是一个底层函数, 在命令窗口中不能调用。函数 contourc 计算时,先确定需要绘制等高线的高度值,当用户不指定时,则自动指 定 20 条以内的等高线;函数 contourc 把输入的数据点看成网格,把 x-y 平面的区域分成小的长方形。当指定的 某一高度值比某个小长方形区域的某些端点的值大,而比某些端点的值小时,采用线性插值计算长方形区域的 边界上的点。最后将所有的高度值相等的点连成线,即为等高线。

当采用函数 surfc 和 meshc 绘制等高线时,将在一个坐标轴里的上部绘制表面图或网格图,在图的下部绘制 等高线图。等高线图画在 z 轴所能显示的最小值处的平面上。可以通过等高线图的"Zdata"属性来增强图形效 果,其步骤为:首先取出等高线图的句柄值。命令是 h=meshc(peaks(20))。句柄 h 中的第一个分量为网格图的句 柄,第二个分量为等高线图的句柄,然后修改属性"Zdata"的值。

除了在直角坐标中,也可以在极坐标中绘制等高线图。例如绘制 z=((x+yi)⁶-1)^{0.25} 的等高线。例子中将把数 值从极坐标转换到直角坐标,并分别在直角坐标中和极坐标中绘制函数的等高线图,在命令窗口输入。

```
[t,r]=meshgrid((0:5:360)*pi/180,0:0.05:1);
[x,y]=pol2cart(t,r);
z=x+i*y;
f=(z.^6-1).^(1/4);
surf(x,y,abs(f));
hold on
surf(x,y,zeros(size(x)));
hold off
xlabel('实部')
```

ylabel('虚部'); zlabel('abs(f)'); 结果如图 6-68 所示。



图 6-68 带等高线的三维图

# 在命令窗口中输入

contour(x,y,abs(f),30); axis equal; xlabel('实部'); ylabel('虚部') 将在直角坐标中绘制曲线的等高线,如图 6-69 所示。



图 6-69 直角坐标中的等高线图

在命令窗口输入下列命令,将在图形窗口绘制如图 6-70 所示的极坐标中等高线图。 h=polar([0 2*pi],[0,1]); delete(h) hold on contour(x,y,abs(f),30);



图 6-70 极坐标中的等高线图

## 6.3.9 瀑布图

在 MATLAB 中,绘制瀑布图的函数是 waterfall,它和 mesh 函数基本相同,只是它并不绘出纵向的线条,因而产生类是"瀑布"的效果。

例如,在命令窗口输入

waterfall(peaks);

axis tight

在图形窗口将绘出如图 6-71 所示的三维高斯分布的瀑布图。



图 6-71 瀑布图

6.4 色彩控制

在 MATLAB 里,可以通过控制图形的着色,来增加图形的表现力。绘制图形时,既可以选择自己喜欢的色彩,也可以用 MATLAB 提供的色图。MATLAB 中,图形着色可分为两类,一类称为对应着色,一类称为真彩着色。对应着色将固中的每一个数据点和色固中的一种颜色对应,具体的着色方法视图形的渲染方式而定。真彩着色采用用户指定的色值(如 RGB 二元组)来着色图形,具体的着色方法也和图形的渲染方式有关。要得到真

正的真彩着色图,显示器应采用真彩显示(24 位)。在其他条件下,MATLAB 也可模仿真彩着色。绘图时,如果用户不指定色值,MAnAB 将采用自带的色图对图形进行对应着色:真彩着色时,用户需为指定数据点指定元素值(RGB)。

6.4.1 MATLAB 颜色表示法

在 MATLAB 中,每种颜色都用一个长度为3的实数向量表示,向量的元索取值范围是[0,1],这3个数值分别表示计算机屏幕显像管红、绿、蓝3种颜色的光强度值,称为 RGB3 元数组值。表 6-13 中列出了几种常见颜色的 RGB 值。

表 6-13

常见颜色 RGB 对照表

红	绿	蓝	颜色
0	0	0	黑色
1	1	1	白色
1	0	0	红色
0	1	0	绿色
0	0	1	蓝色
1	1	0	黄色
1	0	1	洋红色
0	1	1	淡青色
0.5	0.5	0.5	灰色
0.5	0	0	暗红色
1	0.62	0.40	黄铜色

在 MATLAB 中,每个图形窗口都配备着一个色谱。色谱定义为由一组颜色的 R、G、B 值组成的列数为 3 的数值矩阵,称为色谱矩阵。色谱矩阵的第一列为各颜色的 R 值、第二列为 G 值,第三列为 B 值。色语矩阵的行数称为色谱矩阵的长度。在弄清楚着色原理之前,有必要理解下列几个概念的区别:

- 颜色索引表 这是计算机的硬件术语
- ▷ 色谱 是由 RGB 值组成的 m×3 结数值矩阵,每个 MATLAB 图形窗口关联一个色谱 矩阵,即该图形窗口中图形对象可使用的颜色
- ▶ 伪色谱 相对于图像来说的非图像真实颜色的任何一个色谱
- ▶ 调色板 图像的特定色谱,在该色谱下,图像的颜色被真实地表示出来

MATLAB 的函数 colormap 可以用来取得当前的图形窗口的色谱,其调用命令为

M=colormap;

MATLAB 在创建图形窗口时,自动地为图形窗口设置一个色谱.随后用户可以通过函数 colormap 为图形窗口设定特定色谱。设 M 是一个色谱矩阵,那么语句

colormap(M)

的作用就是将当前图形窗口的色谱设置为 M,同时该语旬使得该图形窗口中已存在的所有图形对象按新色 谱改变颜色;随后创建新的图形对象时,也将使用这个新的色谱进行着色。

由于色谱矩阵是一个数值矩阵,因此,它可以被看作是一个数据表,可以由 MATLAB 的数组或矩阵运算来 定义。MATLAB 提供几个函数定义几种系统色谱,表 6-14 中列出了这些函数。

表 6-1	14	色彩寒暑表
•	命令	说明
-	hsv	色度饱和度色图
-	hot	黑红黄白色图
-	gray	线性灰度色图
	bone	蓝色色调色图
-	copper	线性纯铜色色图
	pink	淡阴影粉红色色图
-	white	全白色图
-	flag	交互的红白蓝和黑色图
-	lines	线性色图
-	colorcube	增强的立方色图
-	jet	HSV 的变异图

命令	说明
prism	棱镜色图
cool	青色和洋红色阴影色图
autumn	红色黄色阴影色图
spring	洋红黄色阴影色图
winter	蓝色绿色阴影色图
summer	绿色黄色阴影色图

每个函数可以有一个可选择的输入参数 m,表示生成长度为 m 的色谱矩阵。例如,函数 hot 的命令形式是 M=hot(m)

其他的函数调用方式相同。M 是一个 m × 3 的色谱短阵 , hot 生成的色谱矩阵表示的颜色是从黑色、暗红色、 橙色、黄色到白色的由暗到明变化的各种颜色。

值得一提的是,在不指定参数 m 时,上述的色谱矩阵生成函数都是失取得当前图形窗口的色谱长度,再生 成同样长度的色谱矩阵。通常 MATLAB 自动生成图形窗口时,设置的色谱矩阵是由 hsv 生成的色谱矩阵,长度 为 64。这种做法的好处在于,可以同时有 3 个或 4 个图形窗口的色谱的颜色互不相同(即不超过 256 种颜色)。 对于颜色资源有限的计算机系统.如果在不同的图形窗口中使用较长的色谱矩阵,那么在不同的图形窗口之司 进行切换时,wtndows 操作系统可能会改变非活跃图形窗口中图形对象的颜色。

使用色谱着色的图形生成函数有 mesh、surf、pcolor 和 image,所有以这些函数为基础的图形生成函数都将 使用色谱矩阵进行着色,但是线型图形生成函数.如 plot、plot3、contour 和 contour3 等则使用系统预定义的 7 种颜色着色,与色谱无关。

#### 6.4.2 着色原理

MATLAB 通过线性变技的方式将领色矩阵的数值元素映射到色谱矩阵的行索引号,以该行的 RGB 颜色值 所决定的颜色对颜色矩阵天素对应的图形位置进行着色。例如,下述命令

surf(z,c)

中,c是颜色矩阵,MATLAB将c中的最小元素映射到色谱矩阵的第一行 RGB值,最大元素映射到色谱矩阵的最后一行 RGB值,其他的元素接线性的方式映射到色谱短阵的某一行 RGB值。例如,如果颜色矩阵的元素 c(1,1)被映射到色谱矩阵的第一行 RGB值,即 c(1,1)是 c 的最小元素,那么 c(1,1)对应的顶点或网格片就按色谱矩阵第一行 RGB值对应的颜色进行着色处理,具体的着色方法由着色模式决定。

为了有效地使用和控制色谱矩阵的颜色 MATLAB 提供了坐标系颜色范围的概念。坐标系颜色范围是一个二 元的向量,它描述了在进行图形元着色时,最小和最大的可见颜色的颜色值的范围。例如,前面的缺省着色方 式的颜色范围是[cmin, cmax],亦即颜色矩阵元素的最小值和最大值。这种做法有一个缺点,如果只有极少数 元家数值远大于或远小于其他的颜色矩阵元素。那么,按缺省方式着色后,就很难通过图形的颜色来判定数值 矩阵元素之间的大小关系。为此,MATLAB 通过设置[minc, maxc]的办法剔除一些不很明显的部分。设置当前坐 标系的颜色范围可以通过下面的语句实现

caxis([cmin,cmax])

利用 caxis,可以达到以下的效果

- 如果将[cmin, cmax]设置得比颜色矩阵 c 的最小与最大值范围还要大, 那么, 真实的数据着色只是 用到色谱中的部分颜色;
- 如果将[mmc cmax]设置在颜色矩阵 c 的最小值与最大值的范围内,那么那些小于 cmim 和大于 cmax 的 c 的元素对应的顶点或区域将不被着色,或者说着色值为 NaN。

如果计算机系统支持 24 位的全真颜色 ,即同时支持 2²⁴种不同的颜色 ,那么 MATLAB 提供了一种着色方案。 下面以函数 surf 为例来说明这种方法。对于命令

surf(x,y,z)

变量 c 是一个与 z 同维数的矩阵。对于曲面片上的每小块,先将颜色矩阵 c 相应的元素映射到色谱矩阵的 行索引号个、再由该行提供的 RGB 值来决定用什么样的颜色着色。在 MATLAB 中,变量 c 称为颜色数据,它 既可以是一个与 z 同维数的矩阵,也可以是一个 m×n×3 的 3 维数组,其中 m 是 z 的行数,n 是 z 的列数。此 时{c(i,j,1),c(i,j,2),c(i,j,3)}组成 RGB 值,用于对 z(i,j)对应的曲面小块着色。也就是说,在这种情况下.不必从色 谱中挑选颜色。 例如,在命令窗口中输入

z=peaks(25);

c(:,:,1)=rand(25,25); c(:,:,2)=rand(25,25);

c(:,:,3)=rand(25,25);

surf(z,c)

可以对 peaks 曲面随机地进行着色

6.4.3 色谱矩阵分析

一幅图像具有或明或暗的效果,这种效果可以从图像的调色板色谱上反映出来。

设 M 是 m × 3 的色谱矩阵,命令语句

plot(M)

将生成红、绿、蓝 3 种颜色光强度的曲线图形。在本书中用实线、虚线和点线分别表示红、绿、蓝 3 种颜 色光强度的曲线。在计算机屏幕上,可以用语句

rgbplot(M)

将 M 用红、绿、蓝的线型表示出来。

MATLAB 的灰色色谱用函数 gray 生成 , 该色谱矩阵每行的 3 个元素的数值相同 , 即 3 种颜色的强度比例是 一样的 , 于是 , 利用语句

plot(gray)

生成的图形如图 6-72 所示, 3 条曲线是重合的。



图 6-72 灰色色谱的 RGB 曲线

色谱 hsv 也是 MATLAB 系统预定义的色谱,它常用来显示极坐标系中的周期函数的图形。色谱 hot 对应的 颜色是由暗到明地经过黑、暗红、橙、黄到白各种颜色。数学上,红、绿、蓝3种颜色的强度值是按斜梯方式 增加的。在命令窗口输入

plot(hot)

显示的图形可以证实这一点,如图 6-73 所示。

利用 pcolor 函数可以将色谱在平面上展示出来。例如,在命令窗口输入

colormap(gray);

pcolor([1:9;1:9])

将看到 8 个灰度的灰色色谱, 如图 6-74 所示。

由于色谱是一个数值矩阵,故对色谱矩阵进行的数学运算可以获得其他性质的色谱。MATLAB 函数 brighten 可以用来增加或减弱色谱暗色部分的强度值,从而改变色谱的视觉效果。例如,在命令窗口输入

# plot(brighten(gray,0.5))

- 🗆 × 1 File Edit View Insert Tools Window Help 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0 60 10 40 20 30 50 70

结果显示如图 6-75,将灰色色谱的强度按 $\sqrt{x}$  增强,产生了新的色谱强度曲线。





图 6-74 8 个灰度的灰色色谱

如果在命令窗口输入

plot(brighten(hot , -0.5))

其的作用是按 x²的方式减弱色谱的强度,如图 6-76 所示。

也可以用数学的方式对色谱矩阵进行处理,得到新的色谱矩阵。当然,新的色谱矩阵表示的颜色效果有时 是很难知道的。



图 6-75 灰色的增强色谱 RGB 曲线



图 6-76 hot 的减弱色谱 RGB 曲线

6.4.4 图像处理

在 MATLAB 的基本系统中,有几个函数用于图像处理、进行图像的显示、矩阵伪色图像、动画演示等操作。 这对于设计教学演示程序是很有用的。除此之外,MATLAB 的图像处理工具箱则提供了更多的图像操作功能, 对图像处理感兴趣的读者可以进一步参考图像处理工具箱的使用说明。

1. 伪色图像

在命令窗口中输入

z=peaks;

pcolor(z);

colormap(hot);

这将在 MATLAB 系统中根据色谱 hot 在平面坐标系中生成矩阵 z 的伪色图。函数名 pcolor 是英文 pseudocolor(伪色)的缩写,这个函数的作用是将矩阵 z 作为颜色矩阵,根据上一节提供的着色方法,在平面网格 点(i,j)的右上角小区域内用 z(i,j)对应的色谱矩阵的颜色着色。同时,网格线用黑色着色。如图 6-77 所示。



图 6-77 伪色图

由于 contour 和 pcolor 都以图示的方法表示矩阵的数值大小关系,前者用等高线分出数值大小不同的区域, 后者通过不同的颜色分出数值大小不同的区域。将二者结合在一起,可以更清楚地表达矩阵数值的大小关系。 例如,在命令窗口输入

colormap(hot) pcolor(peaks) shading flat hold on contour(peaks,20,'k') hold off 将生成如图 6-78 所示的图形。

这里,pcolor 用 flat 的的着色模式,以消去黑色的网格线,而将所有的等高线着黑色,这样会使得效果更加明显。



图 6-78 伪色图与等高线

2. 图像显示技术

我们知道,数字图像是指将一幅2维的图像表示成一个数值矩阵,矩阵的元素被解释为像素的颜色值(或灰

度值),或被解释为调色板颜色的索引号。为了显示由矩阵表示的数字图像,MATLAB 最一般的做法是将矩阵的 每个元素对应到当前色谱的某个行标号,并取该行的颜色值作为图像相应点的颜色。一般说来,每幅图的色调 不同。因此,作为图像,必须有自己特殊的色谱,这样才能真实地显示图像。这个持殊的色谱就称为该图像的 调色板。

MATLAB 用函数 image 显示图像,命令形式为

image(x)

为了正确地显示图像,还必须将该图像的调色板装入到图形窗口的色谱中。例如.在 MATLAB 中、有一幅 图像的数值 MAT 文件是 gatlin.mat,它包括了图像矩阵 x 和调色板矩阵 map,于是,下列语句就可以真实地再 现这幅图像

load gatlin

image(x);

colormap(map);

axis equal

axis off

在一般情况下,要保证正确的图像纵、横方向的比例。上述例子中的语句 axis equal 就是为了达到这样的目的。

在没有装入图橡的调色板之前,image(x)用的是当前的色谱,它的效果与 pcolor 的效果相类似。但是它们之间有明显的区别。首先,lmage 是用来显示真正的图像的,包含装入调色板;而 pcolor 则是用图形的方式表达抽象的数学对象的数量关系。除此之外,它们还在以下几个方面存在着差别。

设矩阵 A 是 m×n 的数值矩阵

- ▶ image(A)对 m×n 片小矩形进行着色,横轴和纵轴网格的分点在 k+1/2 点处。而 pcolor(A)在(m-1)
   × (n-1)片小矩形上着色,且对网格线着黑色(缺省色);
- ➢ image(A)对每小片矩形区域按 flat 模式(分片常值)着色,而 pcolor(x)可以有多种着色模式,用 shading 函数来设置;
- ▶ image (A)采用门矩阵坐标系,而 pcolor(A)的缺省方式是 x-y 笛卡尔坐标系;
- ▶ image(A)总是生成等分的网格,而 pcolor(X,Y,A)可以生成参数型的网格;
- ➢ image(A)生成的 2 维图像,只能从标准的 2 维视点(0,90)观看,即不能改变视点。而 pcolor(A)生成 的图形可以从任何观点观看;
- image(A)直接用 X 的值作为索引在色谱中对应颜色值,即 A 的元素被截成整数,其范围是 0 到 length(colormap),而 pcolor 是根据坐标系的颜色区间(可以由 caxis 设置)按线性变换方式计算索引 值来对应色谱矩阵的颜色值。image(A)的图像不受 caxis 函数的影响;

▶ image(A)能固定与图像同样大小的坐标系.使图像充满坐标系,而 pcolor 不具备这种功能。

另一个与 image 函数类似的函数是 imagesc, 它的命令形式与 image 的命令形式是一样的。在讨论 image 与 pcolor 函数的区别时,已经看到 image(x)是将数据炬阵 x 的值直接作为索引号在色谱矩阵中提取 RGB 颜色值进 行着色的。事实上.对于任何矩阵 x, image(x)可以生成一幅图像,如果 x 的元素的数值大小十分接近,或超出 色谙矩阵的长度,那么 image(x)就不能有效地用图像表达矩阵 x, 而函数 imagesc 就可以做到这一点。ima8esc 在功能上与 image 是一样的,只是按线性变换的方式计算索引号,即与 pcolor 使用的方式相同。于是, image(x) 生成的图像将受 caxis 函数的影响。

# 6.5 图形对象控制

到目前为止,本书所介绍的 MATLAB 图形的生成、管理和控制,都是 MATLAB 图形系统的高层操作。事实上,MATLAB 的图形系统还具有低层的图形操作功能,这就是 MATLAB 的图形句柄管理模式。

## 6.5.1 MATLAB 图形对象结构

MATLAB 系统内部使用对象语言描述各种图形单元,并将这些图形单元按照树型结构组织起来进行管理和 实施各种操作。在 MATLAB 中,图形对象被分成:计算机屏幕(root)、图形窗口(figure)、坐标系(axes)、线段(line)、 区域片(Patch)、曲面(surface)、图像(image)、文字(text)、光源(light)、用户界面控制元(uicontrol)和用户界面菜单 (uimenu)等图形对象类型。后两种对象统称为用户界面对象。

在图形对象的树型结构中,计算机屏幕作为该结构的根,它的一级树节点是图形窗口对象;二级节点即图 形窗口的子对象可以是坐标系对象、两种用户界面对象等;第三级树节点是坐标系对象的子对象,它们是线段 对象、区域片对象、曲面对象、图像对象、文字对象、光源对象等,有时称为叶节点。

各种图形对象类型及其功能分述如下

- root MATLAB 图形系统中的 root 对象是计算机屏幕,也是唯一的 root 对象,其他任何图形对 象都是它的子对象
- figure 图形窗口对象是根对象的子对象。根对象可以有多个图形窗口对象,所有其他的图形对象都是图形窗口对象的子对象。在图形窗口对象末被创建时,所有的对象生成函数或高层的图形生成函数自动地创建一个图形窗口对象。也可以用函数 figure 直接创建一个图形窗口对象
- axes 称为坐标系对象。坐标系对象是图形窗口对象的子对象,其表现形式是图形窗口对象中的一块矩形区域。它的图形子对象都将在这块区域中显示。坐标系对象又是线段对象、曲面对象、文字对象、区域片对象等的父对象。在坐标系对象末被创建之前,所有的子对象生成函数和所有的高层图形生成函数都将创建一个坐标系对象,也可以用函数 axes 直接创建坐标系对象
- line 线段对象是基本的图形对象之一,可以用来构成大部分的2维相3维图形,它们是坐标系 对象的子对象,它的位置由它的坐标系对象决定。线段对象主要由高层函数如 plot、p1ot3、contour 和 contour3等创建、也可以由低层生成函数 1ine 创建
- patch 区域片对象是带边界的填充的多边形区域。它们是坐标系对象的子对象。其位置由坐标 系对象决定,有常值型和插值型的填充模式。高层绘图画数 fill 和 fill3 生成区域片对象,也可以 用低层函数 patch 创建
- surface 曲面对象主要用来表达数据矩阵的 3 维图形,由数据矩阵定义的众多小四边形片构成。 曲面对象可以按常值型或插值型的方式进行着色,曲面对象是坐标系对象的子对象,其位置由坐 标系对象决定。高层函数 pcolor、surface 和 mesh 及其变形的函数都可以创建曲面对象,也可以 由低层函数 surface 直接创建
- image 图像对象是数据矩阵的映射图,数据矩阵被当作颜色矩阵来对平面的网格区域进行着 色。图像对象通常有自己的调色板,即色谱,图像对象是2维的图形对象,不能改变图像对象坐 标系的观点。图像对象是坐标系对象的子对象,它由函数 image 创建
- text 文字对象是坐标系中的字符串。文字对象也是坐标系对象的于对象,其位置由坐标系决定。它们可以用函数 text 和 gtext 等创建
- light 光源对象在一个坐标系内定义一个光源。坐标系中所有的图形对象都受它的影响。光源 对象是不可见的,但是可以根据需要设置它的届性值,以控制光源的类型、颜色和位置,也可以 设置其他一些图形对象的公共属性,它们可以闻函数 light 创建
- uicontrol 用户界面控制元对象是交互程序设计中引入的图形对象,利用它们可以启动某个函数或子程序的执行。它们是图形窗口对象的子对象,与坐标系无关
- uimenu 用户界面菜单对象是图形窗口对象的子对象,它是图形窗口菜单条中的菜单项,也与 坐标系无关

#### 6.5.2 通用函数

MATLAB 系统管理着众多的图形对象,在 MATLAB 中,每个具体的图形对象都有唯一的标识值,称为句 柄。MATLAB 系统正是通过图形对象的句柄来标识、管理和操作每个图形对象的。句柄值是在图形对象创建时 由系统自动设定的。下面对图形对象句柄的特点作简单介绍。

根对象(即计算机屏幕)的句柄值被赋予特别的标识值0,且保持不变。图形窗口对象的句柄值是正整数,根

据图形窗口对象创建的顺序给定其值。在缺省状态下,这个正整数被显示在图形窗口的标题栏上。其他的图形 对象的句柄值是一个双精度的实数,句柄值关联着图形对象的系统管理信息。当用户使用句柄值来标识某个图 形对象时.一定要保证该值的正确性。在不同的数据输出模式下,命令窗口回显的数据不一定是精确的句柄值, 所以最好的办法是将要使用的句柄值传递给一个变量,由这个变量将句柄值传送到所需要的地方。

所有的图形生成函数(无论低层还是高层函数)都返回生成对象的句柄值。在命令窗口输入

h=plot([1 2 3 4 5 6])

就得到该线段对象的句柄值 h。有些函数如 contour 等生成多条线段对象,因此返回一个句柄值的向量。 1.当前窗口函数

MATLAB 定义了 3 个特别的函数来访问一些当前的图形对象,它们是

➢ gcf 取得当前的图形窗口的句柄值

- ▶ gca 取得当前的坐标系的句柄值
- ▶ gco 取得当前的(第三级)图形对象的句柄值

可以将这些函数取得的句柄值作为输入变量送到需要这些句柄值的函数中。

2. 拷贝与删除函数

(1) copyobj 函数

MATLAB 提供了一个函数 copyobj,利用这个函数可以将图形对象从一个父对象拷贝到另一个父对象中, 从而创建一个新的图形对象。新的图形对象与旧的图形对象只有句柄值和 Parent 属性值是不同的。既可以将多 个图形对象拷贝到一个父对象中,也可将一个图形对象拷贝到多个父对象中,只要有正确的 Parent/Child 关系即 可。

copyobj 的命令形式是

copyobj(h1, h2)

其中,hl是被拷贝的图形对象的句柄值,而h2是希望拷到的父对象的句柄值。

(2) delete 函数

MATLAB 的函数 delete 可以用来删除句柄值标识的任何图形对象。例如,要删去当前坐标系,可在命令窗 口输入

delete(gca)

可以使用这种方法来关闭一个图形窗口。而在命令窗口输入

delete(2)

将关闭2号图形窗口。

3. 属性设置函数

在 MATLAB 中,每种图形对象都具有一定的属性,MATLAB 正是通过对属性的操作来控制和改变图形对 象的。这些届性包括对象类型、对象的父对象和子对象句柄、对象是否可见等对象的公共属性,还包括不同类 型对象的特殊属性,如坐标系对象的 x 轴性质等等。

在生成一个新的对象时,必须给对象的各种属性赋予必要的属性值。当用户不提供某种属性值时原统将自 动使用缺省的属性值。用户可以向系统询问某个对象的当前属性值,也可以设置新的属性值。

为了方便起见,MATLAB 给每种对象的每种属性规定了一个名字,称为属性名,而属性名的取值称为属性 值。例如,LineStyle 是线段对象的一个属性名字,它的值决定着该线段的显示类型,比如是实型线段、虚型线 段,还是点型线段等等。函数 plot 调用中的字符串变量的值就被传送到这个属性上。在属性名的写法中,MATLAB 不区分大小写字母。

在程序设计中,经常使用的访问属性的方法是用 MATLAB 函数 set 和 get,使用这两个函数之前,对象应该已存在。

(1) set 函数

set 函数可以让用户设置图形对象的任何属性(除了受 MATLAB 保护的私有属性外)的属性值,它的第一个输入参数是图形对象的句柄值,其他的参数为"属性/属性值"对。

例如,在命令窗口输入

f=figure('Color','white');

a=axes('View',[-37.5 30], 'XColor', 'k','YColor','k','ZColor','k'); surfh=surface(peaks,'LineStyle','.'); grid on

第一条语句创建一个图形窗口对象,用于画图,将图形窗口的背景色设置为白色(缺省色为黑色);第二条语 句在图形窗口中创建一个坐标系,坐标系的视点为[一 37.5,30](缺省的坐标系为标准的 2 维坐标系,即视点为[0, 90]),3条坐标轴都设置为黑色(缺省为白色);第三条语句生成 peaks 曲面,网格线段的线型为实点标记,结果 如图 6-79 所示。



图 6-79 peak 函数图

如果要将视点移到[-45,45]方向,在命令窗口输入 set(a, 'View',[-45,45]) 结果显示如图 6-80 所示。 如果要将网格线的线型改为实线线型,只需执行语句 set(surfh, 'LineStyle', '--'); 显示结果如图 6-81 所示。



图 6-80 用 set 设置视点



图 6-81 用 set 设置线型

set 函数的第一个变量可以是句柄值向量,此时它的作用是将指定的所有对象的指定属性设置为持定的属性值,当然每个句柄值对应的图形对象必须具有指定的属性。

若要得出某个类型的对象的所有可设置的属性,调用 set 函数即可以达到目的。例如,若要得出某个类型的 对象的所有可设置的属性,可在命令窗口输入

set(surface) 回车显示结果 AlphaData AlphaDataMapping: [ none | direct | {scaled} ] **C**Data CDataMapping: [direct | {scaled} ] EdgeAlpha: [ flat | interp ] -or- {an Alpha}. EdgeColor: [ none | flat | interp ] -or- {a ColorSpec}. EraseMode: [ {normal} | background | xor | none ] FaceAlpha: [flat | interp | texturemap ] -or- {an Alpha}. FaceColor: [ none | {flat} | interp | texturemap ] -or- a ColorSpec. LineStyle: [ {-} | -- | : | -. | none ] LineWidth Marker:  $[+ | o | * | . | x | square | diamond | v | ^ | > | < | pentagram | hexagram | {none} ]$ MarkerEdgeColor: [ none | {auto} | flat ] -or- a ColorSpec. MarkerFaceColor: [ {none} | auto | flat ] -or- a ColorSpec. MarkerSize MeshStyle: [ {both} | row | column ] **XD**ata YData ZData FaceLighting: [ none | {flat} | gouraud | phong ] EdgeLighting: [ {none} | flat | gouraud | phong ] BackFaceLighting: [unlit | lit | {reverselit} ] AmbientStrength DiffuseStrength SpecularStrength

SpecularExponent SpecularColorReflectance VertexNormals NormalMode: [ {auto} | manual ] ButtonDownFcn: string -or- function handle -or- cell array Children Clipping: [ {on} | off ] CreateFcn: string -or- function handle -or- cell array DeleteFcn: string -or- function handle -or- cell array BusyAction: [ {queue} | cancel ] HandleVisibility: [ {on} | callback | off ] HitTest: [ {on} | off ] Interruptible: [ {on} | off ] Parent Selected: [ on | off ] SelectionHighlight: [ {on} | off ] Tag **UIContextMenu** UserData Visible: [{on} | off] 也可以向 MATLAB 系统询问对单独一个属性可以设置哪些合法的属性值,例如在命令窗口输入 h=plot([1 2 3 4]); set(h, 'LineStyle') 回车将显示结果 [ {-} | -- | : | -. | none ] (2) get 函数 与 set 相对应, get 函数可以用来取得图形对象的各项属性的当前值。想要得到某个图形对象的所有当前值 时,可以用 get(h),其中 h 为该图形对象的句柄值。例如,在命令窗口输入 h=surface(peaks,'LineStyle','.'); get(h) 回车显示结果 AlphaData = [1]AlphaDataMapping = scaled CData = [(49 by 49) double array]CDataMapping = scaled EdgeAlpha = [1]EdgeColor = [0 0 0]EraseMode = normal FaceAlpha = [1]FaceColor = flat LineStyle = none LineWidth = [0.5]Marker = . MarkerEdgeColor = auto

```
MarkerFaceColor = none
    MarkerSize = [6]
    MeshStyle = both
    XData = [(1 by 49) double array]
    YData = [(49 by 1) double array]
    ZData = [(49 by 49) double array]
    FaceLighting = flat
    EdgeLighting = none
    BackFaceLighting = reverselit
    AmbientStrength = [0.3]
    DiffuseStrength = [0.6]
    SpecularStrength = [0.9]
    SpecularExponent = [10]
    SpecularColorReflectance = [1]
    VertexNormals = [ (49 by 49 by 3) double array]
    NormalMode = auto
    BeingDeleted = off
    ButtonDownFcn =
    Children = []
    Clipping = on
    CreateFcn =
    DeleteFcn =
    BusyAction = queue
    HandleVisibility = on
    HitTest = on
    Interruptible = on
    Parent = [100.001]
    Selected = off
    SelectionHighlight = on
    Tag =
    Type = surface
    UIContextMenu = []
    UserData = []
    Visible = on
    为了取得某个特殊属性值,除了给出图形对象的句柄外、只要在函数 get 的调用中给出属性名即可。例如,
要求得当前坐标系对象的所有子对象的句柄,可在命令窗口输入
    h=get(gca, 'children')
```

如果将 get 的输出值赋给一个变量,那么,MATLAB 将创建一个结构数组,该结构的各个域名是图形对象的属性名,而每个域值是当前的属性值。例如,设 x 和 y 是两个等长的向量,下面的命令将创建一个 line 对象:

h=plot(x,y); a=get(h);

而第二条语句将该对象的所有属性值按结构的方式赋给变量 a,例如,a.color 的值就是该线段对象的颜色值。 通常,h是一个列向量,其元素的顺序由相应的对象的显示顺序决定,第一个元素是最上面的图形对象的句 柄,如果还需要知道每个子对象的类型,只要执行语句

get(h(1), 'Type')

get(h(2), 'Type')

即可。有些高层的图形生成函数生成多个图形对象,这些函数返回所生成的图形对象句柄值的列向量,例 如,在命令窗口中输入

비, 비미 오이 마귀, 비

h=surfc(peaks) 回车显示结果

h =

100.0026 101.0012 102.0007 103.0007 104.0004 105.0002 106.0002 107.0002 108.0002 109.0002 110.0002

111.0002

112.0002

由于该函数可生成曲面和等高线,因此,通过这些句柄值,可以访问它的属性 Type,来判别每个句柄对应的图形对象的类型。例如:

get(h(1), 'Type') ans = surface

get(h(2), 'Type')

ans =

patch

使用函数 get 和 set 访问对象同性的好处在于可以改变单个的图形对象的特性,而不会破坏其他图形对象的 属性。

(4) findobi 函数

函数 surfc 生成的曲面用图形窗口的色谱着色,而平面上的等高线用 MATLAB 预定义的 7 种颜色着色,如 果希望将等高线的颜色改变成曲面上对应的颜色,只要重新设置线段对象的 color 属性值即可达到目的。为了简 化这个过程,可以使用 findobj 函数找出全部线段对象,该函数接受句柄值向量,而输出具有指定局性和属性值 酌对象的勾柄值组成的向量。

6.5.3 共有属性

MATLAB 为每种图形对象定义了许多不同的属性,但是有些属性是所有的图形对象共同具备的。如 Children、Clipping、Parent、Select、SelectedHighlight、Tag、Type、UserData、Visible 等等。这里将对这些公共 属性及其作用进行详细介绍。

1 . Children

Children 属性的取值是对象句柄值组成的向量。这个向量由该图形对象所有的子对象的句柄值组成。例如, 对图形窗口对象来说,其 Children 属性值就是所有的 Axes 句柄值, Uimenu 句柄值和 Uicontrol 句柄值。用户可 以改变该向量元素的顺序,那么,所对应的子对象的显示顺序也就随之改变了。

坐标系的图形子对象可以是图像、光源、线段、区域片、曲面、文字等对象。虽然坐标轴的说明文字对象 也是坐标系的子对象,但是它们的句柄值不会出现在 children 属性向量中,因为这些文字对象的 HandleVisibility 属性取值为 "callback"。如果需要临时从 Children 属性中查看这些子对象的句柄值,可以将根对象的 ShowHiddenHandles 属性设置为 "on"。例如

h1=plot([1, 2, 3]);

h2=xlabel('This is a test')

h=get(gca, 'Children')

这时 h2 的值不会出现在的分量中,但是

set(0, 'ShowHiddenHandles', 'on')

h=get(gca, 'Children')

可以使得 h2 出现在向量 h 约分量中。

对于所有第三级图形对象, Children 属性的取值为空矩阵。例如, 因为线段对象是第三级子对象, 故它自己 没有子对象。

2. Clipping

Clipping 属性的取值是 on(缺省值)或 off。它决定图形对象被显示时,超出图形窗口或坐标系范围的部分是 否需要剪裁。

图形窗口对象本身不用这个属性,它对坐标系对象也没有任何影响。对坐标系对象,Clipping 属性值为 on 时,超出当前坐标系范围之外的曲面、区域、文字部分不会显示出来。通常,MATLAB 总是将图像故在坐标系 的整个区域上。但是,如果该属性值为 off 图像可以在坐标系区域外显示出来。

3. Parent

Parent 属性的取值是某个图形对象的句柄值。对图形窗口对象而言,它的 Parent 属性取值总是 0,这是因为 图形窗口对象的父对象总是根对象,即计算机屏幕,其句柄值为 0。

坐标系对象的 PaMn?属性的取值是某个图形窗口的句柄值,这是因为坐标系的父对象总是图形窗口对象。 函数 gcf 返回当前坐标系所在的图形窗口对象的句柄值。

线段对象、曲面对象、区域片对象、图像对象、文字对象、光源对象等的 Parent 属性的取值是某个坐标系 对象的句柄值,因为这些对象总是坐标系对象的子对象,该句柄值标明了这些对象所在的坐标系。如果用 s 碘 函数将 Parent 属性设置为另一个坐标系的句柄值、相当于将这些对象移到另一个坐标系中。例如,

hold on

load clown

h_image2=image(x)

h_axes=subplot(212)

handles=get(h_axes, 'Children')

set(h_axes, 'Children', [handles, h,.. image2])

4 . Tag

Tag 属性的取值是一个字符串。在 MATLAB 系统内,系统是用句柄值来标识图形对象的。为了在程序与被调用的 M 函数之间共享该句柄值,需要将保存句柄值的变量定义为 global 变量,或作为实参数传递到被调用的 M 函数,这显然不够方便。

例如,如果希望将图形都输出到某个特定的图形窗口,可以用下列语句

figure('Tag', 'My plotting figure')

达到这个目的。即定义一个 Tag 属性值为 "My plotting figure?" 的图形窗口,然后在每次要输出图形之前用 findobj 函数找到这个图形窗口,并使用命令:

figure(findobj('Tag', 'My ploting figure'))

使它成为当前的图形窗口。例如,如果在某段 M 文件子程序中,希望将所有的图形输出到同一个坐标系中, 那么只要定义一个具有特定 Tag 属性的坐标系即可

axes('Tag', 'My special axes')

然后,在每条图形输出语句前加入下列语句

axes(findobj('Tag', 'My special axes'))

即可达到目的。当然,此时其他的固形对象 Tag 属性值不能为 My special axes。

句柄值是 MATLAB 系统管理图形对象时所用的标识值 ,而 Tag 属性值是用户管理图形对象时所用的标识值。 Tag 属性的取值是字符串,它结该图形对象定义了一个标识值,当定义了 Tag 属性后,在任何程序中都可以 通过这个标识值找出该图形对象。在整个程序中,每个图形对象的 Tag 属性值必须是唯一的。

5. Type

Type 属性的取值是字符串,这个属性说明该图形对象的类型,它的值是不可改变的。若 Type 属性的取值是 axes,说明该图形对象是坐标系对象;取值是 line,说明图形对象是线段对象;若取值是 surface,说明图形对象 是曲面对象;取值是 patch,说明图形对象是区域片对象;若取值是 image,说明图形对象是图像对象;取值是 text,说明图形对象是文字对象;若取值是 light。说明图形对象是光源对象,等等。

6. UserData

UserData 属性的取值是一个矩阵。对所有的图形对象,这个属性的意义是一样的。在程序设计中,可以将 一个与图形对象有关的比较重要的数据存储在这个属性中。图形对象本身并不一定位用这个矩阵,但是,用户 可以用函数 set 和 get 访问这个属性。

利用这种方法,可以将与坐标系相关的重要信息保存在这个属性中,借此可以达到传递数据或信息的目的。 它的缺省值为空矩阵。

7. Visible

Visible 属性的取值是 on(缺省值)或 off,它是一个开关,决定着图形对象是否在屏幕上显示出来。当它的值为 off 时,可以用来隐藏该图形对象。

通常,坐标系是可见的。如果坐标系这个属性值被设置为 off,那么坐标系的坐标轴、刻度标记、坐标轴说 明文字等都被隐藏起来,成为不可见的,但这些对象仍存在。该属性的取值对其他的坐标系子对象没有影响。 例如,在显示一幅图像时,通常将坐标系 visible 属性置为 off。

由于光源对象本身是不可见的,它的存在是通过它对区域片对象和曲面对象的作用效果反映出来的。如果 该属性值为 off,那么光源对象就不起任何作用,相当于没有该光源对象。但是,如果要保证区域片对象和曲面 对象的某些控制光照效应的属性,如 AmbientStrength 等,包括坐标系对象的 AmbientLightColor 属性起作用, 那么在坐标系中至少有一个光源对象的 visible 属性值为 on。

对象存在与否与对象不可见是两码事,对象可以存在,同时又是不可见的。

8. Selected

Selected 属性的取值是 on 或 off,这是所有图形对象的公共属性,这个属性表明图形对象是否被选中。

当该属性取值为 on ,且相应地图形对象的 SelectionHighlighht 属性的取值也是 on 时,MATLAB 就显示图形 对象被鼠标选中的状态。

如果将 ButtonDownFcn 对应的操作功能,即 callback 定义为将 Selected 属性值设置为" on",那么当选中这个图形对象时,图形对象就会有一个被选中的方框(图形窗口对象和坐标系对象除外)。

例如,在执行下列几条语句后,再在图像上点按鼠标,即可看出效果

load gatlin

h=image(x);

colormap(map);

axis off

set(h, 'Selected', 'on', 'SelectionHinghlight', on)

这种技巧可以用在 GUI 程序设计中,表明图形对象被选中时的状态。光源对象不使用这个属性。

9. SelectionHighlight

SelectionHighlight 属性的取值是 on(缺省值)或 off,这是所有图形对象的公共属性。如果该属性值是 on,则 MATLAB 画出一个矩形,指明图形对象被鼠标选中的状态;如果该属性值为 off,则没有标记来指明图形子对象 被选中。这个属性必须与 selected 属性配合使用,共同控制图形对象被选中时的状态。

图形窗口对象、坐标系对象和光源对象不使用这个属性。

另外还有一些用于控制对用户的动作作出反应的某些高级属性、它们是:ButtonDownFcn, ChangeFcn, CreateFcn, DeleteFcn, BusyAction, HandleVisibility, Interruptible 等。在介绍这些属性之前,先来简要地说明 MATLAB 的一个很重要的概念 callback。在 MATLAB 中,一个 callback 通常是指那种打断程序运行的某种操作,

即中断。它们可以是由一小段 MATLAB 语句或 M 文件定义的。

10 . HandleVisibility

HandleVisibility 属性的取值为 on(缺省值),但有些对象,例如坐标轴说明文字的该缺省值为 callback 或 off, 这个属性定义图形对象句柄的可访问权限,决定其句柄值是否在父对象的 Children 周性中出现。如果 HandleVisibility 的属性值是 on,那么图形对象的句柄值总是可见的;例如,该图形对象的句柄值可以出现在定 义对象的 Children 属性中;如果 HandleVibility 的属性值为 callback,那么该图形对象句柄值只在图形对象的 Callback 调用以及 Callback 调用函数内是可见的,但是命令行内调用的函数则不能访问这样的图形对象句柄, 也就是说,这样的句柄对于该图形对象自己的 Callback 调用是私有的。这对于保护交互式的程序是十分有用的; 如果 HandleVisibility 属性值被设置为 off,那么图形对象的句柄在任何时刻都是不可见的。如果句柄是不可见的, 则任何查询句柄的函数都不能返回它的值。这些函数包括个 get、findobj、gcf、gca、gco、newplot、cla、clf、 close 等。

当 HandleVisibility 属性值是 callback 或 off 时,这样的图形窗口对象不会出现在根对象的 Children 属性和 CurrentFigure 属性中,而且它的任何子对象也不会出现在根对象的 CallbaekObject 属性和 CurrentObject 属性中。 但是,可以将根对象的 ShowHidenHandles 属性设置为 on,临时地使得所有句柄都是可见的,而不管其 HsndleVisibility 属性值是什么。

被隐藏的句柄仍是有效的,如果知道这样的句柄值,则一切关于句柄的操作都是合法的。

11 . BusyAction

BusyAction 属性的取值是 cancel 或 queue(缺省值),该属性决定 MATLAB 采取的控制中断执行对象的 Callback 调用的方式。在 MATLAB 环境中,如果某个图形对象的 Callback 调用的方式。在 MATLAB 环境中,如果某个图形对象的 Callback 调用的方式。在 MATLAB 环境中,如果某个图形对象的 Callback 调用的方式。在 MATLAB 环境中,如果某个图形对象的 Callback 调用(例如用户按下某个控制按钮,希望运行另一个子任务)总是试图打断正在运行的 Callback。此时,正在运行的 Callback 是否可以被中断,要由该图形对象的 Interruptible 属性和 BusyAction 属性 共同决定。如果此时 Interruptible 属性值为 yes 那么 MATLAB 在处理事件队列时,就会中断正在运行的 Callback,转而执行用户所希望的动作;如果 Interruptible 属性值为 no,那么就由 BusyAction 属性来决定 MATLAB 对事件 处理的方式。如果 BusyAction 属性值是 Cancel,就从事件队列中取消企图运行第二个 Callback 调用的事件(例如,按下鼠标的事件);如果 BusyAction 的属性值是 queue,就将企图运行第二个 Callback 调用的事件加入到事件队列,直到当前的 Callback 调用完成为止,再处理该事件对应的 callback 调用。

12 . ButtonDownFcn

ButtonDownFcn 属性的取值是字符串,一般是某个 M 文件名或一小段 MATLAB 语句。当用户在图形对象 上按下鼠标时,程序执行该属性定义的 Callback 调用。图形对象决定了一个作用区域,称为热区。鼠标键在该 区域内按下时,就执行 ButtonDownFcn 定义的 Callback。

13 . CreateFcn

CreateFcn 属性的取值是字符串,一般是某个 M 文件名或一小段 MAILAB 语句,即当 MATLAB 创建该图形 对象时,MATLAB 事先应该执行的程序段。

这个属性只能按设置缺省值的方式定义、这是为了保证它能事先运行。对于已经存在的线段对象,改变该 后性之值,对该线段无任何影响。

MATLAB 在创建图形对象时,总是在完成所有缺省属性值的设置后,再执行 CreateFcn 属性定义的操作。 事实上 CreateFcn 属性是用户自定义的缺省值、当然在图形对象生成之前执行。

14 . DeleteFcn

DeleteFcn 属性的取值是字符串、一般是某个 M 文件名或一小段 MATLAB 语句。当删除图形对象时, MATLAB 在清除该对象属性之前,执行由 DeleteFcn 属性定义的 Callback。

当用 delete 命令删除图形窗口对象时,一般不执行 DeleteFcn 属性定义的 Callback。

如果一个图形对象 DeleteFcn 定义的 Callback 正在运行,那么该图形对象的句柄值不可访问,必须用函数 gcbo 获取。

15 . Interruptible

Interruptible 属性的取值是 on 或 yes(缺省值)与 off 或 no,这个属性决定图形对象的 Callback 是否可以被随 后的 Callback 调用中断。只有由 ButtonDownFcn 属性定义的 Callback 受 Interruptible 属性值的影响。

6.5.4 根对象

在 MATLAB 图形对象的树型结构中,根对象是根节点,也就是计算机屏幕。它是所有其他对象的父对象。 MATLAB 中可以用 set 和 get 设置和获取它的一些属性。

根对象的句柄是 0 ,可以使用下面的方法来调用 scrsize=get(0, 'ScreenSize') 命令返回得到屏幕大小的像素值 scrsize =

1 1 1024 768

6.5.5 图形窗口对象

图形窗口对象是 MATLAB 系统管理的一类很重要的图形对象。MATLAB 的一切图形图像的输出都是在图 形窗口对象中完成的。同时,在交互式程序设计中,图形窗口对象的设计是任务的重点。因此,掌握好图形窗 口对象对于充分发挥 MATLAB 系统的优秀图形功能和设计高质量的交互界面是十分必要的。

1.图像窗口创建函数

在 MATLAB 图形对象的树型结构中,图形窗口对象是第一级树节点,亦即根节点〔计算机屏幕)的子对象。 创建图形窗口对象有几种常用方法,一是按照 MATLAB 的规则,当不存在任何图形窗口对象时,所有的图形窗 口子对象创建函数将自动地创建一个图形窗口对象作为该子对象的图形输出窗口;二是利用高层图形窗口创建 函数 figure,它具有3种基本命令形式:

figure

h = figure

figure(h)

第一、二种形式用于创建一个新的图形窗口对象,且第二种形式返回该对象的句柄值;第三种形式有两种 用处,如果以 h 为句柄值的图形窗口存在,那么这条语句的作用是使这个图形窗口成为当前的图形窗口;如果 句柄值为 h 的图形宙口对象不存在,那么这条语句的作用就是创建一个新的图形窗口对象,并且以给定的 h 为 其句柄值。

创建固形窗口对象的更一般方法是使用低层创建函数,该函数名为 figure, 它的命令形式是

h=figure('属性名',属性值...)

这种方法主要用在需要特定局性的图形窗口对象的情况下。通常都是使用高层创建函数按 MATLAB 设定的 缺省届性值方式创建图形窗口对象的。

2.图形窗口对象的属性

MATLAB 为每个图形窗口对象提供了 60 多个属性,这些属性及其取值控制着图形窗口对象。下满分述一些常用的图形窗口对象属性。

(1) 外观特征属性

- MenuBar 属性的取值可以是的 Mre(缺省值)或 none。它是一个开关,用来控制图形窗口是否应该 具有菜单条。如果它的属性值为 none,则表示该图形窗口没有菜单条。
- > Name 属性的取值可以是任何字符串,它的缺省值为空。这个字符串作为图形窗口的标题。
- NumberTide 属性的取值是 on (缺省值)或 off。它是一个开关,决定着在图形窗口的标题中是否以"Figure No. N:"为标题前缀,这里 N 是图形窗口的序号,即句柄值。
- Resize 属性取值是 on(缺省值)或 off。它是一个开关,决定着在图形窗口创建后可否用鼠标按 windows 系统的方式改变该宙口的大小。在交互式程序设计中,Resize 属性值为 off 的图形窗口一 般用来作说明页,其窗口大小不可用鼠标改变。WindowStyle 属性的取值是 normal(缺省值)或 modal, modal 类图形窗口捕捉所有 MATLAB 窗口上的键盘和鼠标数据,只要这些窗口是可见的(对 于图形窗口,它的 Visible 属性值为 on)。modal 图形窗口总是位于窗口栈的员上层,如果有多个 modal 图形窗口存在,最后一个创建的 modal 图形窗口位于所有窗口的上端,直到它不可见,或 WindowStyle 属性转为 normal,或被删除时,第二个 modal 图形窗口便成为最上层的 modal 图形 窗口。如果一个 modal 图形宙口的 Visible 属性值为 off,那么这个图形窗口便不再有 modal 图形 窗口的功能,除非它成为可见的窗口。因此,将一个 modal 图形窗口隐藏起来比删除该窗口更有

效些。modal 图形容口不显示菜单栏,但是可以在一个图形窗口中创建菜单对象。这些对象是存在的,当 WindowStyle 属性被设置为 normal 时,这些菜单项就会显示出来。使用命令 " control C "将所有的 modal 图形窗口转为 normal 图形窗口。

- Position 属性的取值是一个 4 元素的向量,其形式为[left, bottom, width, height]。这个向量定义 了图形窗口对象在计算机屏幕上的位置,其中,left 为图形窗口的内左边界到计算机屏幕左边界 的距离;bottom 为图形窗口内底边界到计算机屏幕底边界的距离;width 为图形窗口内区域的宽 度;height 为图形窗口内区域的高度,它们的单位由 Units 属性决定。Position 定义的区域不包括 图形窗口的边界和窗口标题栏及菜单栏。
- Units 属性的取值可以是下列字符串中的任何一种:pixel(像素,为缺省值),normal(相对单位), inches(英寸),centimeters(厘米),points(磅)。这个属性定义图形窗口使用的长度单位,由此决定 图形窗口的大小与位置。除了 normal 以外,其他单位都是绝对度量单位。相对单位 normal 将计 算机屏幕左下角对应为(0,0),而右上角对应为(1.0, 1.0)。该属性对 CurrentPoint 属性和 Position 属性的取值有影响,如果在程序中改变过 Units 属性值,在完成相应的计算后,最好将 Units 届性 值设置为缺省值,以防止影响其他函数计算。在创建图形宙口对象时,如果不使用 Units 属性缺 省值,那么必须在给定依赖 Units 属性的属性值之前先定义 Units 属性值。
- Color 属性的取值是一个颜色值,既可以是 MATLAB 预定义的几种颜色(用字符表示,如'r'代表红色),也可以是任何一个三元素的 RGB 向量。它的取值决定了图形窗口的背景颜色。缺省值为'k',即黑色。
- (2) 控制色谱的属性
  - ColorMap 属性的取值是一个 m×3 阶的 RGB 颜色矩阵,它的缺省值是包含 MATLAB 预定义的 64 种颜色的色谱。色谱矩阵的每一行 RGB 值决定一种颜色。在这个图形窗口中,MATLAB 通过 行索引号决定某种图形对象应着的颜色。色谱矩阵的长度 m 可以是任意的、但在 Pc 机上一般限 制为 256。只有 surface、image、patch 等图形对象的着色受 Colormap 属性值的影响。
  - DitherMap 属性的取值是一个 m×3 阶的 RGB 颜色矩阵,它的缺省值所定义的颜色均匀地覆盖了 整个色谱。这个色谱矩阵用在低颜色分辨窍的计算机系统上,模拟全真颜色(Truecolor)的色谱。 为了模拟出全真颜色的图形和图像,此时就要使用这个色谱矩阵。用户可以改变这个色谱矩阵,以使模拟的图形图像更逼真。
  - DitherMapMode 属性的取值是 auto 或 manual(缺省值),定义 MATLAB 系统使用与生成 Dithermap 色谱矩阵的模式。在 manual 模式下, MATLAB 用 DitherMap 属性定义的色谱矩阵模拟全真颜色 的图形和图像;而在 auto 模式下, MATLAB 根据图形窗口当前使用的颜色生成一个 DitherMap 色谱矩阵,这对于显示出较满意的图形图像是必要的。但是重新生成 DitherMap 色谱矩阵是一件 费时的工作,而且当向该图形窗口增加新图形对象或改变图形窗口大小时,生成 DitherMap 色谱 矩阵的过程会重新进行。一般情况下,可以将生成的 DitherMap 色谱矩阵保存下来,并将 DitherMapMod 属性值设置为 mannual。
  - FixedColors 属性的取值是一个 m×3 阶的 RGB 颜色矩阵,它的缺省值为[0,0,0]。这是一个只读属性,用户不能用 set 函数对其进行更改。它记录着图形窗口中使用的那些不是由 ColorMap 定义的所有颜色的 RGB 值,这些颜色包括图形窗口的背景色、坐标轴颜色、Line、Text、Uicontrol 和 Uimenu 等子对象的颜色,以及用户直接定义的颜色等等。由 FixedColors 属性定义的颜色与colormap 色谱矩阵定义的颜色在计算机系统的色谱表中占据着不同的位置,因此,如果 FixedColors 属性定义的颜色太多,对于颜色资源有限的计算机系统来说,可同时显示的颜色数会受到影响。
  - MinColorMap 属性的取值是一个整数值,它的缺省值为 64。它定义了 MATLAB 向操作系统申请 的系统颜色表项的最小数目。每个表项对应 ColorMap 中的一种颜色。通常情况下,需要增大这 个属性值,以保证 MATLAB 的图形窗口真正使用 ColorMap 定义的全部颜色。但是这样做的结果 是可能导致其他非活跃窗口的颜色失真。
- (3) 控制窗口刷新方式的属性
  - ▶ BackingStore 属性的取值是 on(缺省值)或 off。如果 BackingStore 属性值为 on ,则 MATLAB 在内

存缓冲区中保存该图形窗口的拷贝。当图形窗口被挡住的部分需要重新显示时,MATLAB就直接从缓冲区中拷贝出窗口的内容,而不必重新生成那些图形子对象。这样可以加快屏幕的刷新速度。如果计算机系统内存资源有限,而不在乎屏幕刷新速度时,就可以将 BackingStore 届性值设置为 off 以便节省内存开销。

- Renderer 属性的取值是 painters(缺省值)或 zbuffer,这个属性决定了图形窗口中图形的着色手段。 painters 模式是 MATLAB 定义的基本方式,当图形窗口中只有简单或较小的图形子对象时,这种 模式提供的显示速度较快;在 zbuffer 模式下,MATLAB 能更快地以逐个像素的方式绘制图形子 对象,因为 MATLAB 只需要对在屏幕中可见的像素点进行着色。但是,这种模式要消耗大量的 内存资源。Renderer 属性的取值由 RendererMode 属性决定,Rendcrer 属性的缺省值是 auto,意味 着 Renderer 属性的取值由 MATLAB 自己根据情况决定。如果它的取值设置为 manual,则表明 Renderer 的属性值可以由用户用函数 set 来设置。
- (4) 当前子对象属性
  - CurrentAxes 属性的取值是图形窗口中当前坐标系子对象的句柄值。一个图形窗口对象可以有多个 坐标系子对象,但是在任一时刻只有一个坐标系子对象是当前的坐标系,它的句柄值记录在 CurrentAxes 属性值中。
  - > CurrentCharacter 属性的取值是一个字符,记录着用户在该图形窗口中最后一次键入的字符。
  - CurrentObject 属性的取值为图形窗口的某个子对象的句柄值,这个图形子对象位于图形窗口当前 点(Current Point)下,且图形窗口的子对象栈顺序中位于最上层。所以,可以通过这个属性知道在 图形窗口中用户用鼠标最后选择的子对象。一般用函数 get 取得这个句柄值。
  - CurrentPoint 属性的取值是两元素的向量。这个向量表示在图形窃口中用户最后一次 按鼠标的位置,其值是相对于图形窗口的左下角点为原点的坐标值,单位由 Units 属性确定。在图形窗口中, 每按一次鼠标,MATLAB 就会更新这个属性值。
  - SelectionType 属性的取值可以是 normal(缺省值)、extended、alt、open 中的任何一个,用户不能 修改这个属性值。SelectionType 记录着在图形窗口中最后一次按鼠标的方式。在 Windows 环境中, normal 表明单击鼠标左键; extended 表明按 Shift 键+单击鼠标左键或同时单击鼠标左右键; alt 表 明按 Ctrl 键+单击鼠标左键; open 表明双击鼠标左键。
- (5) Callback 调用函数

除了图形对象的公共属性外,图形窗口对象还有以下几个定义 Callback 调用的属性。

- CloseRequestFcn 属性的取值是字符串,一般是某个 M 文件名或一小段 MATLAB 语句。它定义该 图形窗口在关闭时,若用户发出 close 命令,应执行的 Callback 调用。但是,delete 命令会无条件 地关闭图形窗口,此时这个属性定义的 Callback 调用不会被执行。DeleteFcn 属性与 CloseRequesFcn 属性的作用类似,所不同的是,无论用户采用什么样的方式关闭图形窗口,DeleteFcn 属性定义 的 Callback 都要被执行。
- KeyPressFcn 属性的取值是字符串,一般是某个 M 文件名或一小段 MATLAB 语句。它定义了用 户在图形宙口中按下某键时所要执行的 callback 调用。例如,在这个属性的 Callback 调用中可以 定义一条语句访问 CMrrentCharackr 届性,然后判别 CurrentCharacter 属性值是否为某个特别的字 符,再决定是否执行某些操作。这种方法可以让用户在图形窗口中按下某个字符键,完成一定的 操作任务的字符时执行。另一方面,在 KeyPMssFcn 届性定义的 Callback 中,可以访问根对象的 PointerWindow 属性,以确定是在哪个图形窗口中按下该键的,因为鼠标标记落在某个图形窗口 上时,键入一个字符,并不能使这个图形窗口成为当前的图形窗口。KeyPressFcn 属性定义的 Callback 受 Interruptible 属性的影响。
- ResizeFcn 属性的取值是字符串,一般是某个 M 文件名或一小段 MATLAB 语句。它定义了用户在 改变图形窗口大小时所要执行的 Callback 调用。此时,可以访问图形窗口的 Position 属性,以了 解图形窗口的大小与位置。如果正在用鼠标改变图形窗口的大小,那么正在运行的程序中就不能 直接访问读取这个窗口的句柄,只能从根对象的 CallbackObject 属性中访问该图形窗口。
- ▶ WindowButtonUpFcn 属性的取值是字符串,一般是某个 M 文件名或一小段 MATLAB 语句。它定

义用户在图形窗口中释放鼠标按键时程序应执行的 Callback 调用。WindowButtonUPFcn 属性定义的 Callback 受 Interruptible 属性的影响。

WindowButtonDownFcn 属性的取值是字符串,一般是某个 M 文件或一小段 MATLAB 语句。它定 义用户在图形窗口中按下鼠标键时程序应执行的 Callback 调用。WindowButtonDownFcn 属性定义 的 Callback 受 Interruptible 属性的影响。

WindowButtonMotionFcn 属性的取值是字符串,一般是某个 M 文件或一小段 MATLAB 语句。它定义用户在 图形窗口中移动鼠标时程序应执行的 callback 调用。WindowButtonMotionFcn 属性定义的 Callback 受 Interruptible 属性的影响。值得注意的是,ButtonDownFcn 属性和 WindowButtonDownFcn 属性都是定义用户在图形窗口中按 下鼠标铝时所要执行的 Callback。如果某个图形窗口的这两个属性都有定义,那么当在该图形窗口中按下鼠标键 时,先执行 WindowButtonDownFcn 定义的 Callback,执行完后,再执行 ButtonDownFcn 属性定义的 Callback。

- (6) 鼠标标记属性
  - Pointer 属性的取值是 arrow(缺省值)、crosshair、watch、topl、topr、botl、botr、circle、cross、fleur、 left、right、top、bottom、fullcrosshair、ibeam、custom 等,用户可以根据工作任务的类型,设定 不同的鼠标标记。其中,custom表示由用户自定义鼠标标记。
  - PointerShapeCData 属性的取值是 16×16 阶矩阵。当 Pointer 属性值为 custom 时, MATLAB 以 这个矩阵定义的 16×16 像素位因为用户自定义的鼠标标记。该矩阵各元素之值为 1、2 或 NaN。 1 表示像素标记黑色,2 表示像素标记白色,而 NaN 表示像寒是透明的。 PointerShapeCData 矩 阵的(1,1)元素对应它定义的鼠标标记的左上角点确定。
- (7)图形窗口管理属性
  - IntegerHandle 属性的取值是 on(缺省值)或 off,这个属性值决定了图形窗口对象句柄值的数值类型。 一般来讲,图形窗口句柄值总是正整数。当 MATLAB 创建一个新的图形窗口时,总是将最小的 可用的正整数作为新图形窗口的句柄值。如果用户删除了某个图形窗口,那么再创建图形窗口时, 就可以使用删除的图形窗口句柄值。如果将 ItegerHandle 属性设置为 off,则 MATLAB 在创建新 图形窗口对象时,就用一个实数型数值表示图形窗口的句柄值。
  - NextPlot 属性的取值是 add(缺省值)、replsce 或 replacechildren,这个属性决定了 MATLAB 用于图 形输出的图形窗口。如果当前的图形窗口的 NextPlot 属性值是 add,则表明用当前的图形窗口作 图形输出;如果它的取值是 replace,那么除了它的 Position 属性外,所有的属性都设置为缺省值, 并且在输出图形之前,删除已存在的所有图形于对象;如果它的取值是 replacechildren,那么只是 在新图形输出之前,删除已存在的所有图形于对象,而不改变其他的属性值。
- (8)管理图形打印机属性
  - InvertHardcopy 属性的取值是 on(缺省值)或 off,这个属性只影响打印输出。当它的属性值是 on 时,图形窗口的背景色等转换成白色,而坐标抽、线段等转换为黑色;若其属性值为 offl 则打印 图形窗口时不会进行任何颜色的转换。
  - PaperOrientation 属性的取值是 portrait(缺省值)或 1andscape,这个属性决定着图形窗口打印输出时的方向。portrait 是标准的打印方式,而设置 landscape 会使打印的结果按逆时针方向旋转 90 度。
  - PaperPosition 属性的取值是一个4元素向量:[left, bottom, width, height],此向量定义了打印纸 上的一个长方形区域,其左下角相对于纸张的左下角的位置,由(left, bottom)决定,它的长度单位 由 PaperUnits 属性定义。图形窗口的内容按一定的比例缩放打印在这个区域上,所以用户可以设 置这个届性值来得到所需要大小的图形输出,但要注意打印的结果还受下面其它属性值的影响。
  - PaperPositionMode 属性的取值是 manual(缺省值)或 auto。在 manMal 模式下,输出的图形在打印 纸上的位置由 PaperPosition 属性决定;而在 auto 模式下,MATLAB 将按照屏幕上图形窗口的尺 寸把图形打印在打印纸的中央。
  - PaperSize 属性的取值是二元素向量[width, heighy],它定义打印纸的大小。这个属性只可读,用 户不能更改。它的具体值由 PaperType 属性决定,其单位由 PaperUnits 属性定义。
  - PaperType 属性的取值是 uslener(缺省值)、uslegal、a4letter、a5、b4 和 tabloid。它定义了 MATLAB 打印支持的几种标准纸张。如果用户使用 A4 纸,最好将这个属性设置为 a4letter。

PaperUnits 属性的取值是 normalized(相对单位)、inches(英寸:缺省值)、centimeters(厘米)和 points(磅)。这个属性定义打印纸的度量单位。

6.5.6 轴对象

坐标系对象是 MATLAB 系统管理的另一类很重要的图形对象。坐标系对象是图形窗口对象的子对象,每个 图形窗口对象中可以定义多个坐标系对象,但是只有一个坐标系是当前坐标系,在没有指明坐标系时,所有的 图形生成函数都是在当前坐标系中输出图形或图像。必须弄清楚一个慨念,所谓在某个图形窗口中输出图形图 像,是指在该图形窗口的当前坐标系中输出图形图像。

1. 坐标系对象生成函数

通常,在图形窗口中没有定义坐标系之前,MATLAB的每个图形生成函数会自动地创建一个坐标系作为它 输出的图形图像的坐标系。用户可以根据自己需要利用坐标系创建函数创建特殊的坐标系。坐标系创建函数是 axes,它有如下多种命令形式:

axes

h=axes

axes('坐标系属性名',坐标系属性值...)

h=axes('坐标系属性名',坐标系属性值...)

axes(h)

坐标系生成函数 axes 是系统低层生成函数,上述第一和第二条命令用系统坐标系缺省属性值在当前图形窗 口中生成一个坐标系,并将该坐标系句柄值存在变量 h 中;第三和第四条命令根据指定的"属性名/属性值" 对在当前图形窗口中创建坐标系,并且将该坐标系的句柄值存在变量 h 中;第五条命令使句柄值为 h 的坐标系 成为当前的坐标系,且该坐标系所在的图形窗口自动成为当前的图形窗口。在这种命令形式中,h必须是一个合 法坐标系句柄,即这个坐标系是存在的,这与图形窗口创建函数的 Mre(h)的调用是不同的。如果 h 对应的图形 窗口不存在,那么蛤 axes(h)就创建一个新的图形窗口对象,并且以 h 为其句柄值。

可以用函数 gca,获得当前图形窗口的当前坐标系的句柄值,即当前图形窗口对象 CurrentAxes 属性值。如 果当前图形窗口中没有坐标系子对象,或没有图形窗口对象存在,那么 gca 将在当前图形窗口中创建一个坐标 系,或同时创建一个图形宙口对象,这等价子函数 axes 调用。

2.坐标系对象属性

除共有属性外,MATLAB 为每个坐标系对象定义了多种属性,它们共同控制着坐标系的外形、方位和大小 等。

(1) 外观特征属性

- Box 属性的取值是 on 或 off(缺省值),它决定坐标系是否带有边框。在2维情形,坐标系绘制成矩形框,图形在框内输出;在3维情形.坐标系绘制成长方体,图形在体内输出。大部分的2维图形函数,如 plot 上 ontour 等在绘图时,自动为图形加上边框。
- ➢ GridLineStyle 属性的取值是-、--、:(缺省值)、-..或 none, 该属性定义坐标系网格线的线型。" "表示实线,"--"表示虚线,":"表示点线,"-."表示点虚线。
- Layer 属性的取值是 bottom(缺省值)或 top,对于 2 维坐标系,或视点[0 90],当该属性值为 top 时, 坐标轴刻度标记、网格线等画在坐标系图形子对象之上(即可见的);而当该属性值为 bottom 时, 坐标轴刻度、网格线等在图形子对象之下。
- LineStyleOrder 属性的取值为线段类型值,其缺省值为-(实线),表示对每条线段型图形使用实线作 图。该属性决定在生成多条线段型图形时使用的线段类型和顶点标记及其顺序。MATLAB 自定义 了4种线段类型,在作图时、依次使用这4种类型的线段作图,而每种线段类型重复地使用 ColorOrder 属性定义的不同颜色,直到所有颜色都使用过一次后,再换下一种线段类型。
- LineWidth 属性的取值是以磅为单位的宽度值,缺省值为 0.5 磅,它定义坐标轴线段的宽度。
- TickDir 属性的取值是 in 或 out,定义坐标轴到度标记的方向。在缺省模式下,对2维坐标系,坐标轴刻度标记向坐标系内;对于3维坐标系,坐标轴刻度标记向坐标系外。
- ▶ TickDirMode 属性的取值是 auto(缺省值)或 manual, 定义 MATLAB 选择坐标轴刻度方向的方式。

在 auto 模式下,MATLAB 总是将 2 维坐标系的坐标轴刻度方向置为 in,而把 3 维坐标系置为 out。 但是,如果用户设置了 TickDir 的值,MATLAB 会自动地将 TickDirMode 属性设置为 manual。在 manunl 模式下,MATLAB 不改变特定的坐标轴刻度方向

- TickLength 属性的取值是二元素向量。第一个元素定义2维视点时坐标系刻度线段的长度,第二 个元素定义3绍视点时坐标系刻度线段的长度,其值是相对于最长坐标轴的相对单位值,缺省值 为[0.010 0.025]。
- CurrentPoint 属性的取值为 2 × 3 的数值矩阵,它定义最后一次按下鼠标键的位置,由坐标系数据 单位决定其数值单位,这个矩阵记录着鼠标决定的两个顶点的 3 维坐标值,这两个顶点位于垂直 于屏幕并通过鼠标顶点的直线上、是该直线与 3 维坐标系的 x 轴、y 轴、z 轴的 Limits 属性决定 的矩形体边界面的交点。其中一个交点位于前端,另一个交点位于后端。数值矩阵的第一行为后 端交点的坐标系值,第二行为前端交点曲坐标系值。对于 2 维坐标系,前端交点的 z 坐标值为 1, 后端交点的 z 坐标值为 0,而它们的 x 和 y 坐标值是相同的。当在图形宙口中点按鼠标键时, MATLAB 就更新该图形宙口令所有坐标系的 CurrentPoint 属性值。
- Position 属性的取值为 4 元素的数值向量[left, bottom, width, height]。这个向量在图形窗口中决定一个矩形区域,坐标系就位于这区域中,或者说这个向量确定坐标系在图形窗口中的位置。该矩形的左下角相对于图形窗口左下角的坐标为(left, bottom),矩形的宽度为 width,高度为 height, 其数值单位由坐标系 units 属性值定义。在通常情况下,坐标轴标记、坐标轴说明文字等部落在由Position 定义的矩形区域之外,即坐标系撑满模式。但是如果坐标系不取撑满模式,那么坐标系及坐标抽标记、坐标轴说明文字等都落在 Position 决定的区域内。
- Units 属性的取值是 inches(英寸)、centimeters(厘米)、normalized(相对单位:缺省值)或 points(磅)。 这个属性值决定 Position 属性的度量单位。所有的度量都是从图形窗口左下角(0,0)开始,而相对 单位 normalized 将图形窗口左下角对应为(0,0),右上角对应为(1,1),其他都是绝对单位。
- (2) 辅助属性
  - ➢ FontAngle 属性的取值是 normal(缺省值)、italic 或 oblique,它定义坐标系的坐标轴刻度值、坐标 轴说明文字等所用的字体。normal 表示使用正体,italic 和 oblique 都表示使用斜体。
  - FontName 属性的取值为字符串,它定义坐标系的坐标轴刻度值、坐标轴说明文字等所用的字体。 它的合法值与计算机系统上使用的字体资源有关,通常的缺省字体是方头字体。在改变这个属性时,坐标系到度值字体立即更新,但是坐标系说明文字的字体必须通过重新设置 Xlabel、Ylabel和 ZLabel 属性或重新用 xlabel、ylabel和 z1abel 命令来更新。
  - FontSize 属性的取值是一个数值,它定义坐标系坐标轴刻度数值、坐标轴说明文字以及坐标系标题(Title)文字的大小尺寸,其尺寸单位由坐标系 FontUnits 属性决定,缺省值为 12 磅。当改变 FontSize 属性值后,坐标轴刻度数值字体大小立刻改变,但是必须重新分别设置 Xlabel、Ylabel、 Zlabel 属性或调用命令 xlabel、ylabel、zlsbel 才能改变坐标轴说明文字的字体大小。
  - FontUnits 属性的取值可以是 Points(磅 缺省值)、normalized(相对单位)、inches(英寸)、centimeters(厘米)或 pixels(像素)。FontUnits 属性值 normalized 是相对于坐标系高度的相对单位。
  - FontWeight 属性的取值是 normal(缺省值)、bo1d、light 或 demi。该属性决定坐标系使用文字字体的粗细类型。通常,bold 表示某种字体的黑体类型,light 表示相应的细体类型。与 FontsSize 属性一样,更新该属性值时,坐标轴刻度数值说明文字立即改变,但是必须重新分别设置 XLabel、YLabel、ZLabel 属性或调用命令 xlabel、ylabel、zlabel 才能改变坐标轴说明文字的字体形态。
  - Tttle 属性的取值为坐标系标题文字对象的句柄值,可以通过该属性对坐标系标题文字对象进行操作。
  - XTickLabels、YTickLabels、ZTickLabels 3 种属性的取值都是字符串矩阵或字符串块数组,它定义 了各坐标轴刻度上的标记,矩阵的每行文字对应一个刻度。如果矩阵行不足以对应所有坐标轴刻 度,矩阵行就被循环使用。3 种属性接受多种形式的输入,在用函数 set 设置这些属性时,其值可 以是字符串矩阵、字符串块数组、由竖线分隔的字符串或数值向量(其中数值被转化为数字字符 串)。

XTickLabelMod、YTickLabelMode、ZTickLabelMode 3 种属性的取值都是 auto(缺省值)或 manual, 它们分别记录着各坐标轴刻度标记的生成方式。在 auto 模式下,MATLAB 用数值作刻度标记, 而 manual 模式是指用户自定义的刻度标记,如文字标记等等。当用户用 set 函数分别设置 XTickLabels、YTickLabels、ZTickLabels 属性时,MATLAB 分别自动地将 XTickLabelMode、 YTickLabelMode、ZTickLabelMode 属性置为 manual。

# (3) 控制坐标系的属性

- XDir、YDir、ZDir 3 种属性的取值都是 normal(缺省值)或 reverse,这些属性决定着各坐标轴的方向。在缺省设定下,取右手坐标系,x 轴从左到有为正向,xDir 属性值 reverse 则将这个方向反过来;y 轴从下到上(2 维视点)或由前向后(3 维视点)为正向,YDir 属性值 reverse 将这个方向反过来;z 轴从屏幕向外(2 维视点)或从下到上(3 维视点)为正向,ZDir 属性值 reverse 将这个方向反过来。
- XGrid、YGrid、ZGrid 3 种属性的取值都是 on 或 off(缺省值),当属性值分别为 on 时,在坐标系 上相应的坐标轴刻度处作垂直于该坐标轴的网络线,函数 grid on 或 grid off 同时对 3 条坐标轴起 作用。
- ➢ xLim、YLim、ZLim 3 种属性的取信都是 2 元素的数值向量,它们分别定义各坐标轴的可见范围, 缺省值为[0,1]。改变该向量,相当于改变坐标轴的比例。
- XLimMode、YLimMode、ZLimMode 3 种属性的取值都是 auto(缺省值)或 manual。这些属性记录 着 MATLAB 设定坐标轴范围向量的方式。在 auto 模式下, MATLAB 根据坐标系中图形的数据自 动计算范围向量,保证所有图形于对象都落在坐标系中。用户可以用 set 函数设置 xLim、YLim、 Zlim 属性值,此时 MATLAB 将 XLimMode、YLimMode、ZLimMode 属性置为 manual。
- ➢ XScale、YScale、ZScsle 3 种属性的取值都是 1inear(缺省值)或 log,这些属性定义各坐标轴的比例 尺度。MATLAB 定义了两种比例尺度,一种为 1inear(线性)坐标尺度,另一种为 log(对数)坐标尺 度。
- XTick、YTick、ZTlck 3 种属性的取值都是数值向量,这些向量记录着各坐标轴刻度位置处的坐标 值。每个向量的元素必须是单调增加的、用户可以设置这些向量,可以将坐标轴刻度标记在所需 的地方。如果不希望坐标轴上有任何刻度,则将这些向量设置为空向量即可。
- XTickMod e、YTzckMode、ZTickMode 3 种属性的取值都是 auto(缺省值)或 manual。记录坐标轴 刻度的生成方式。在 auto 模式下,MATLAB 根据各坐标轴的数据范围计算标记刻度的坐标值; 如果用户自己定义 XTick、YTick、ZTick 属性,那么 MATLAB 自动地将 XTickMode、YTzckMode、 ZTickMode 属性置为 manual。
- XAxisLocation 属性的取值是 top 或 bottom(缺省值),该属性决定 x 轴刻度和说明文字的位置。当 它的取值是 bottom 时,就是通常的坐标系;如果它的取值是 top,那么 x 轴就被移到坐标系区域 的上方。
- YAxisLocation 属性的取值是 right 或 left(缺省值),该属性决定 y 轴刻度和说明文字的位置。当它的取值是 left 时,就是通常的坐标系,如果它的取值是 right,那么 y 轴就被移到坐标系区域的右边。

(4) 视点属性

在 MATLAB 中, view 属性可以控制坐标系的视点,但是这个属性只定义了视点的方位,用户不能控制视点 与目标点的距离。但实际上,即使方位相同,如果视点离坐标系的距离不同,看到的坐标系的效果也应该是不 一样的。为了更好地控制视点、根据照相机原理,MATLAB 提供了几种其他属性来控制坐标系的视点位置。

- CameraPosition 属性的取值是 3 元素数值向量[x, y, z], 该属性定义视点位置的坐标值,可以将 其比作照相机从该处取景。假设照相机的视角(由 CameraTarget 定义)保持不变,那么改变 CamemPosition 就可以变化取景范围,当视点向目标点(由 CameraTarget 定义)靠近时,目标被放大; 远离目标点时,目标将缩小。
- CameraPositionMode 属性的取值是 auto(缺省值)或 manual。当它的取值是 auto 时, MATLAB 根据 View 属性定义的视点方向,自动计算 CameraPosition 属性值,使得从 CameTaPosition 到 CameraTarget 保持固定的距离。如果由用户设置 CameraPosition 属性值,则 MATLAB 将

CameraPositionMode 置为 manual。

- CameraTarget 属性的取值是 3 元素数值向量[x,y,z],该属性定义目标点位置的坐标值,或说是 照相机的对焦点。显然,视点方向在 CameraPosition 和 CameraTarget 两点决定的直线上。
- CameraTargetMode 属性的取值是 auto(缺省值)或 manual。当它的取值是 auto 时, MATLAB 自动 地将 CameraTarget 点固定在坐标系绘图区域的重心点处。如果用户自定义 CameraTarget 属性值, 那么 MATLAB 就置 CameraTargetMode 为 manual。
- CameraUpVector 属性的取值为 3 元素数值向量[x, y, z], 该属性定义坐标系中的自由向量,决定 照像机取景时的上方方向。设置不同的 CameraUpVector 向量,可以让照相机绕由 CameraTarget 与 cameraPosition 决定的直线旋转。它的缺省值为[0 0 1],即 z 轴正向为视点上方方向。
- CameraUpVectorMode 属性的取值是 auto(缺省值)或 manual。当它的取值是 auto 时, MATLAB 用 [0 0 1]作为 3 维视点的上方方向,而用[0 1 0]作为 2 维视点的上方方向。当用户自定义 CameraUpVector 属性值时, MATLAB 将 CameraUpVectorMode 置为 manual。
- CameraViewAngle 属性的取值是范围为 0-180 度,该属性定义视点或照相机的视角。它的取值影 响坐标系中图形对象的大小、值越大,取景范围越大,那么坐标系中的同一图形就越小。
- CameraVewAngleMode 属性的取值是 auto(缺省值)或 manual。当它的取值是 auto 时, MATLAB 将 CameraViewAng1e 设置为能使照相机取到坐标系全景的最小视角。当用户自定义 CameraViewAngle 属性值时, MATLAB 将 CameraViewAngeMode 置为 manual。
- ➢ View 属性的取值是 2 元素数值向量 [x ,y] ,它定义视点方向 ,即视点方向的径度和纬度。在 MATLAB 中 ,可以用上述的照相机属性代替 View 的作用。
- (5)比例属性
  - ➢ DataAspectRatio 属性的取值是 3 元素数值向量[dx, dy, dz], 该向量控制 x、y、z 轴单位长度的相对比例。例如,如果该属性值为[1 2 3],则表示坐标系 x 轴上的数据单位长度等于 2 倍的 y 轴数据单位长度,等于 3 倍的 z 轴数据单位长度。
  - DalaAspectRatioMode 属性的取值是 auto(缺省值)或 manual。当它的取值是 auto 时, MATLAB 自动设置 DataAspectRatio 属性值;如果用户自定义 DataAspectRatio 属性值,则 MATLAB 置 DataAspectRatioMode 属性值为 manual。
  - ➢ PlotBoxAspectRatio 属性的取值是 3 元素数值向量[px, py, pz], 该向量控制坐标系长方体区域在 x、y、z 方向的相对比例。例如,当它取值为[1 1 1]时,坐标系呈正方体形式。
  - PlotBoxAspectRatioMode 属性的取值是 auto(缺省值)或 manual。当它的取值是 auto 时, MATLAB 自动设置 PlotBoxAspectRatio 属性值:如果用户自定义 PlotBoxAspectRatio 属性值,则 MATLAB 置 PlotBoxAspectRatioMode 属性值为 manual。
  - Projection 属性的取值为 orthographic(缺省值)或 perspective,它决定坐标系在平面(如计算机屏幕 平面)上的投影方式。对观察者来说,在 orthographic 模式下,是保持图形的相对宽度,例如数据 决定的平行线在屏幕上画为平行线;在 perspective 模式下,3 维图形投影为2 维图形时,则保持 景深效应,即图形离观察者越远,看上去越小。
- (6)图形刷新模式
  - DrawMode 属性的取值是 normal(缺省值)或 fast。当坐标系所在的图形窗口 Renderer 属性值为 painters 时,该属性控制 MATLAB 对坐标系中图形进行刷新的方式。在 normal 模式下,MATLAB 根据当前的视点方向,按从后向前的顺序刷新或输出图形,体现出图形在3 维空间中的前后层次 关系;而在 fast 模式下,MATLAB 按绘图命令的顺序刷新或输出图形,而不考虑图形对象在3 维 空间中的顺序关系。
  - NextPlot 属性的取值是 add、replace 〔缺省值)或 replacechildren。这个属性决定了 MATLAB 的高 层图形函数图形输出的方式。如果当前坐标系的 NextPlot 属性值是 add,则表明作图形输出时, 保持原有的子图像不变,加入新的图形;如果它的取值是 replace,则除了坐标系的 Position 属性 外,所有的属性都设置为缺省值,并且在输出图形之前,删除已存在的所有图形子对象;如果它 的取值是 replacechildren,那么只是在新图形输出之前,删除坐标系中己存在的所有图形子对象,

而不改变坐标系其他的属性值。

# (7) 颜色属性

- AmbientLightColor 属性的取值为 MATLAB 系统定义额色的字符或 RGB 颜色值,该属性定义坐标 系中环境照明光的颜色。环境照明光是无方向的光源,它均匀地散射在坐标系中的图形上。但是 如果在坐标系中没有定义光源对象,则该属性不起作用。如果有光源对象存在,那么 AmbientLightColor 的作用就加到光照效应中去。
- CLim 属性的取值是 2 元素数值向量[cmin, cmax],它决定着 MATLAB 将曲面对象和区域片对象的 CData 数据映射到其图形窗口 Colorrmap 色谱矩阵行索引号的方式。cmin 值映射到色谱矩阵的第一行, cmax 值映射到色谱矩阵的最后一行、介于 cmin 与 cmax 之间的数值按线性方式映射到色谱矩阵的各行。在 CData 中超出[rmin, cmax]范围的数值,则不对应任何颜色。
- CLimMode 属性的取值是 auto(缺省值)或 manual。MATLAB 取 CData 饱数据中的最小值为 cmin, 最大值为 cmax。当用户自己设置 CLim 属性值时, MATLAB 将 CLimMode 属性置为 manual。在 manual 模式下, CData 数据发生变化,对 CLim 属性没有影响。
- Color 属性的取值为 none(缺省值),或由 MATLAB 系统定义颜色的字符或 RGB 颜色值,它定义 坐标系区域背景颜色。当它的值为 none 时,坐标系区域是透明的,即坐标系区域的颜色为图形窗 口的背景色。
- ColorOrder 属性的取值是 m×3 的 RGB 颜色值矩阵,其缺省值是系统预定义的 7 种颜色。它定义 在坐标系中用 plot、plot3 等函数同时绘制多条曲线时,曲线着色的颜色顺序。当坐标系 NextPlot 属性值为 replace(缺省值),函数 plot 在决定要使用的颜色时,将 ColorOrder 设置为系统的缺省值。 如果希望使用自定义的颜色顺序,则要将坐标系的 NextPlot 属性设置为 replacedata。
- XColor、YColor、ZColor 3 种属性的取值都是 MATLAB 系统定义颜色的字符或 RGB 颜色值,它 们的缺省值都是 w(白色)。它们分别定义坐标系的坐标轴、坐标轴刻度、坐标轴刻度标记以及坐 标系网格线的颜色。

6.5.7 线段对象

线段对象(Line)是最简单的图形对象,它是坐标系对象的子对象。在 MATLAB 中,曲线是用线段对象表示的。高层图形函数,如 plot、plot3、contour、contour3 等等都可以创建线段对象。线段对象既可以定义在2 维坐标系中,也可以定义在3 维坐标系中。

1. 线段对象创建函数

线段对象的低层创建函数是 line, 它的命令形式有下列几种

line(x, y)

```
line(x, y, z)
```

line(x, y, z, '属性名', 属性值...)

line('属性名',属性值...)

h=line( ... )

每种命令形式都能在当前坐标系中创建一个或多个线段对象、最后一种形式还返回创建线段对象的句柄值。 如果不给出"属性名/属性值"对,MATLAB就用线段对象的缺省值创建该线段对象,在这些命令形式中,对 变量 x、y、z 的解释与高层线段函数 plot 和 plot3 是一样的,当 x、y、z 等都是矩阵时,按列的对应方式创建多 个线段对象。在这种多个线段对象的创建中、线段对象、线段类型与着色分别由坐标系 LineStyleOrder 属性和 ColorOrder 属性共同决定,即对 LineStyleOrder 定义的每种线段类型,依次用 ColorOrder 定义的颜色作线段,如 此循环。

line 创建函数与 plot 等高层创建函数的重要区别在于, line 函数不检查当前坐标系的 NextPlot 属性值,它总是直接向当前坐标系中添加新的线段对象,并自动更新坐标系的 XLim、YLim、ZLim 等属性,使得新加入的线段对象完全落在坐标系中,而高层函数 plot 等调用另一个函数 newplot 来决定当前坐标系 Nextplot 属性的状态,除非 NextPlot 的属性值为 add。plot 等在输出图形之前,会先删除当前坐标系中已存在的图形对象。

## 2. 线段对象属性

与其它的图形对象一样,MATLAB 为线段对象定义了各种属性,并通过这些属性来管理线段对象。

- (1)外形属性
  - Color 属性的取值是某些颜色的预定义字符或颜色 RGB 数值,它决定线段对象输出时的颜色。可以用 set 函数设置该属性值,将线段对象按指定颜色着色。
  - ▶ LineStyle 属性的取值是 "-"(实线:缺省值)、"--"(虚线)、":"(点线)、"-."(点虚线)或 none(无线), 这是 MATLAB 预定义的线段类型。该属性决定线段对象输出(显示)时所用的线段类型。
  - ➢ LineWidth 属性的取值是以磅为单位的数值,缺省值为 0.5 磅。它定义线段对象输出(显示)时线段的宽度。增大 LineWidth 属性值,可以得到较粗的线段(曲线)。
  - Marker 属性的取值可以是 "+"(加号顶点标记)、"。"(圆圈顶点标记)、"*"(星号顶点标记)、
     "."(实点顶点标记)、"×"(乘号顶点标记)、square(方块顶点标记)、diamond(钻石形顶点标记)、

"^"(三角形顶点标记)、"v"(三角形顶点标记∇)、">"(三角形标记AA)、"<"(三角标标

记 ■)、pentagram(五角星标记)、hexagram(六角星标记)、none(空顶点标记,缺省值)。在 MATLAB 中,曲线是通过一些顶点的折线表示的,该属性定义这段折线顶点处的标记,则线段看上去会更 美观。缺省属性值是 none,表示无顶点标记。Marker 属性与 LineStyle 属性无关,但可以通过这 两者的不同组合得到不同式样的线段。

- MarkerEdgeColor 属性的取值是 auto(缺省值)、none 或某些颜色的预定义字符,或颜色 RGB 数值。 该属性的取值如果是颜色值,那么它定义顶点标记的颜色,或某些有边界的顶点标记(圆圈标记、 方块标记、钻石形标记、五角形标记、六角形标记,以及4种三角形标记)的边界颜色。如果取值 为 none,则对有边界的顶点标记来说,边界没有颜色,边界也不可见,而对其他的顶点标记,其 效果是顶点标记不可见。
- MarkerFaceColor 属性的取值是 auto、none(缺省值),或某些颜色的预定义字符或颜色 RGB 数值。 该属性的取值如果是颜色值,那么它定义某些有边界的顶点标记(圆圈标记、方块标记、钻石形标 记、五角形标记、六角形标记、以及四种三角形标记)的内部区域填充颜色。如果取值为 none, 则有边界的顶点标记内部无颜色,并且内部区域是透明的。如果 MarkerFaceColor 取值为 auto, 那么内部区域用坐标系 Color 属性定义的颜色填充。如果坐标系 Color 属性值为 none,则使用图 形窗口 color 属性定义的颜色填充。对无边界的顶点标记,该属性不起作用。
- > MarkerSize 属性的取值是以磅为单位的数值,缺省值为6磅。该属性决定顶点标记的大小。
- XData、YData、ZData 3 种属性的取值都是数值向量,这 3 个向量的对应元素构成线段对象各顶 点在坐标系令的 x、y、z 坐标值,线段对象就是通过这些顶点定义的。线段对象创建函数的最低 层形式为

line('XData', x, 'YData', y, 'ZData', z, '属性名', 属性值....)

- (2) 辅助属性
  - EraseMode 属性的取值是 normal(缺省值)、none、xor 或 background。该属性定义 MATLAB 系统 使用何种方式在屏幕上面线段对象和擦去线段对象,这些技术在生成动画序列时特别有用。在 normal 模式下,当画线段或者擦去线段时,重新刷新整个图形,相当于重新生成该对象。在有遮 挡的情况下,还要进行3维分析,保证原有的因形对象恢复到正确状况。虽然在这种模式下,重 新刷新的图形对象保持原有的着色,但是这种模式速度太慢,并且要消耗大量内存。在 none 模式 下,当线段对象被删除时,该线段对象原来的图形仍保留在屏幕上。在 xor 模式下,当在屏幕上 画线段对象或者擦去线段对象时,用线段的颜色与线段下屏幕上应有的颜色值作"异或"运算, 所以最后输出的线段图形的颜色依赖于屏幕上原有的颜色。在 background 模式下,为了擦去线段, 用坐标系的背景颜色画线段,这样做的结果会破坏线段下原有的图形对象的着色。

# 6.5.8 曲面对象

曲面对象(surface)是图形对象的一种,它是坐标系对象的子对象。在 MATLAB 中,数学曲面是用曲面对象

表示的。高层图形函数,如 mesh、surf、 surfc、suffl 等等都可以创建曲面对象。曲面对象定义在3维坐标系中, 而坐标系可以在任何视点模式下。

1. 曲面对象创建函数

曲面对象的低层创建函数是 surface, 它的命令形式有下列几种

surface(z)

surface(z, c)

surface(x, y, z)

- surface(x, y, z, c)
- surface(..., '属性名', '属性值', ...)
- h = surface(...)

每种命令形式都在当前坐标系中创建一个曲面对象,最后一种形式还返回创建曲面对象的句柄值。如果不 给出"属性名/属性值"对,MATLAB就用曲面对象的缺省值创建该曲面对象。在这些命令形式中,对变量 x、 y、z、c 的解释与高层曲面函数 mesh 和 surf 等的解释是一样的,其中 x 和 y 可以是向量,通常,变量 c 称为颜 色数据矩阵。

surface 创建函数与 mesh 等高层创建函数的重要区别在于, surface 函数不检查当前坐标的 NextPlot 属性值, 它总是直接向当前坐标系中添加新的曲面对象,并不更新坐标系的 XLim、YLim、ZLim 等属性,使新加入的曲 面对象完全落在坐标系中。而高层函数 mesh 等要调用另一个函数 newplot 指令决定当前坐标系 NextPlot 属性的 状态,除非 NextPlot 属性值为 add。mesh 等在输出图形之前,会先删除当前坐标系中已存在的图形对象。

通常,如果变量 x、y、z都是矩阵,那么它们是同维数的矩阵,但是颜色数据矩阵 c 可以与 z 是不同维的, 这时 FaceColor 属性值应该是 texturemap。

1. 曲面对象属性

与对其他的图形对象一样,MATLAB为曲面对象定义了各种属性,并通过这些属性来

管理曲面对象。

(1)外形属性

- CData 属性的取值是数值矩阵或全真颜色 RGB 3 维数组。在第一种情况下,称 CData 为索引式颜 色数据矩阵;在第二种情况下,称 CData 为 RGB 颜色数组,统称为颜色数据矩阵。它是曲面数 据 ZData 对应的曲面各顶点处的颜色数据。通常,CData 矩阵的维数与 ZData 矩阵线数相同,但 是如果 FaceColor 属性值为 texturemap,那么这两个短阵的维数可以不必相同。在这种情况下, MATLAB 将 CData 矩阵线性地映射为与 ZData 同维数的矩阵。如果 CData 为索引式颜色数据矩阵, 那么根据 CDataMapping 属性的取值情况,或者将 CData 的数据按通常的线性映射方式映射到图 形窗口 Colormap 色谱矩阵的索引号中,或者将 CData 的数据直接解释为 colormap 色谱矩阵的索 引号。如果 CData 为 RGB 颜色数组,此时如果坐标数据,如 XData 的维数是 m×n,那么 CData 的维数必须是 m×n×3。如果计算机系统不支持全真颜色,MATLAB 就用 C01。rmap 和 Dithermap 对全真颜色进行模拟。
- CDataMapping 属性的取值是 scaled(缺省值)或 direct,该属性定义对索引式颜色数据矩阵的解释方式。如果 CData 定义的是全真颜色数据,这个属性无任何作用。当该属性值为 scaled 时,MATLAB 根据坐标系 CLim 属性值,按线性方式将 CDta 数据映射到 ColorMap 索引号中;如果它的属性值为 direct,那么 MATLAB 将 CData 数据直接解释为 ColorMap 索引号。因此,在这种情况下,CData 数据通常为整数。如果数值小于1,就解释为1,数值大于 length(colormap),就解释为 length(colormap)。对于非整数数值,按其整数部分截断。
- EdgeColor 属性的取值是 MATLAB 合法的颜色值以及 none、flat 或 interp。取缺省值时为黑色。 该属性定义曲面网格线的颜色或着色方式。在 MATLAB 中,曲面都是用网格曲面表示的,对网格线划分的小曲面片进行着色后就可得到一张视觉良好的实曲面。如果属性值是 RGB 颜色值或 MATLAB 表示的预定义颜色的字符,那么曲面网格线就用这种颜色着色,取缺省值为 k,即网格 线为黑色。但是网格线上顶点处的标记不受该值影响。如果属性值为 none,则在曲面上不画网格 线;属性值为 flat,则用 CData 第(i,j)项数值定义的颜色分别对 x 方向网格线段(x(i,j),y(i,j),z(i,j)),

((x(i+1,j), y(i+1,j), z(i+1,j))和 y 方向的网格线段((x(i,j), y(i,j), z(i,j)), (x(i,j+1), y(i,j+1), z(i,j+1))) 涂色,这里,x、y、z 分别是曲面顶点的坐标值;如果属性值为 interp,则根据网络线段两顶点的 颜色值按线性插值方式对两顶点之间的线段着色。

- FaceColor 属性的取值是 MATLAB 合法的颜色值以及 none、flat(缺省值)或 interp,该属性定义曲面上网格片的着色方式。如果属性是 RGB 颜色值或 MATLAB 表示的预定义颜色的字符,则用该颜色对整个曲面着色,曲面网格线除外;如果属性值为 none,那么不对曲面片进行着色,此时曲面是由网格线构成的透明曲面,而 mesh函数生成的网格曲面的曲面片用黑色着色。虽然在屏幕上看上去 mesh函数生成的曲面与 FaceColor 属性值为 none 时定义的曲面是一样的,但它们之间是有区别的。在透明的情况下,可以看到曲面背后的图形对象。如果属性值为 flat(缺省值),那么用 CData 第(i,j)项数值定义的颜色对由四顶点(x(i,j),y(i,j),z(i,j)),(x(i+1,j),y(i+1,j),z(i+1,j)),(x(i+1,j+1),y(i+1,j+1)),z(i+1,j+1)),(x(i,j+1),y(i,j+1),z(i,j+1))决定的网格片进行着色。这里,x、y、z 分别是曲面顶点的坐标值。如果属性值为 interp,则用每个曲面片 4 顶点对应的颜色值按双线性插值的方式定义曲面片的颜色;如果属性值为舵 texturemap,则 MATLAB 将 CData 数据线性地映射到整个曲面上,通常 CData 的维数远大于坐标数据矩阵上。
- LineStyle 属性的取值是 "-"(实线:缺省值)、"--"(虚线)、":"(点线)、"-."(点虚线)或 none(无线), 这是 MATLAB 预定义的线段类型,该属性决定曲面对象上网格线所使用的线段类型。
- LineWidth 属性的取值是以磅为单位的数值,缺省值为 0.5 磅,它定义曲面对象上网格线的宽度。 增大 LineWidth 属性值,可以得到较粗的网格线。
- Marker 属性,可以参照线段对象 Marker 属性,两者意义相同。该属性定义曲面顶点上的标记, 缺省后性值是 none,表示无顶点标记。Marker 属性与 LineStyle 属性无关,可以通过这二者的不同组合得到不同式样的曲面网格线。
- MarkerEdgeColor 属性与线段对象的 MarkerEdgeColor 属性的取值相同。如果 MarkerEdgeColor 取值为 auto,则使用曲面对象 EdgeColor 属性定义的颜色对标记或标记边界着色。
- MarkerFaceColor 属性与线段对象 MarkerFaceColor 属性的取值相同。如果取值为 none,则有边界 的顶点标记内部无颜色,并且内部区域是透明的;如果 MarkerFaceColor 取值为 auto,则那么内 部区域用该顶点的 cData 数据定义的颜色填充。
- > MarkerSize 属性的取值是以磅为单位的数值,缺省值为6磅,该属性决定顶点标记的大小。
- MeshStyle 属性的取值是 both(缺省值)、row 或 column,该属性定义画曲面网络线的方式。如果取值为 both,表示画出 x 和 y 方向的所有网格线;如果其取值为 row,只画 x 方向的网格线;如果取值为 column,只画 y 方向的网格线。
- XData、YData、ZData 3 种属性的取值都是数值向量或矩阵,它们的对应元素构成曲面对象各顶 点在坐标系中的 x、y、z 坐标值,曲面对象就是这些顶点定义的。下面是常用的创建曲面对象的 语句 sur[ace('xData', x, 'YData', y, 'zDala', z, '属性名',属性值...')如果输入的数据 x 和 y 分别 是向量,那么 MATLAB 将这些向量扩展为矩阵时、x 作为行向量扩展为 XData,且行数与 zDa 船 的行数相同 v 作为列向量扩展为 YDa?a,且列数与 zDa 船的列数相同。
- (2)光照效应属性
  - AmbientStrength 属性的取值是介于 0 到 1 之间的数值,它定义曲面上均匀分布的照明光强度。只有当坐标系中存在有可见的光源对象时,该属性定义的照明光才会增加曲面上的光照效应。而照明光的颜色由坐标系 AmbientColor 属性确定。
  - BackFaccLighting 属性的取值是 unlit、lit 或 reverselit(缺省值)。MATLAB 为曲面上的顶点定义了 法向方向,该属性定义对那些顶点法向背离观点的曲面片(称为背向曲面片)的明暗着色方式。如 果此属性取值为 unlit,那么背向曲面片上无光照效应;如果取值为 reverselit,则背向曲面片上也 有光照效应;如果取值为 lit,其效果基本上与 reverselit 模式类似,只是对闭曲面(例如球面)其效 果有点区别。对于罚曲面,在 revers?1it 模式下,边界点上也有光照效应,这不符合实际情况,而 lit 模式即可避免这个问题。该属性对于在视觉上区分曲面的内侧与外侧是有用的。
  - ▶ DiffuseStrength 属性的取值是介于 0 到 1 之间的数值,它定义光照落在曲面上的散射成分的强度,

散射光来自光源对象。

- EdgeLighting 属性的取值是 none(缺省值)、flat、gouraud 或 phong,该属性定义计算光源对象对曲面网格线作用效果的算法。在 none 模式下,光源对象对曲面网格线无影响;在 flat 模式下,光源对象在网格线线段上的光照效果是均匀一致的;在 gouraud 模式下,根据网格线线段两端点的光照效应作线性插值决定线段上的光照效应;在 phong 模式下,先根据线段两端点的法向沿线段作线性插值,再计算线段上每点(像素点)的反射系数,据此决定线段上的光照效应。这样计算的结果在视觉上效果最好,但花费时间较长。
- FaceLighting 属性的取值是 none(缺省值)、flat、gouraud 或 phong,该属性定义计算光源对象对曲面片作用效果的算法。在 none 模式下,光源对象对曲面片无影响;在 flat 模式下,光源对象在曲面片上的光照效果是均匀一致的;在 gouraud 模式下,根据计算曲面片顶点的光照效应,再作线性插值决定曲面片上的光照效应;在 phong 模式下,先根据曲面片顶点的法向在曲面片上作线性插值,再计算曲面片上每点(像素点)的反射系数,据此决定曲面片上的光照效应。这样计算的结果在视觉上效果最好,但花费时间较长。
- NormalMode 属性的取值是 auto(缺省值)或 manual,该属性定义曲面上顶点法向向量的生成方式。 如果它的取值为 auto,则 MATLAB 会自动根据曲面坐标数据计算法向向量;如果用户自定义法 向向量,那么 MATLAB 将该属性值置为 manual。
- SpecularColorReflectance 属性的取值是介于 0 到 1 之间的数值,该属性定义镜面反射光颜色。如 果它的值是 0 那么镜面反射光颜色就同时依赖于曲面的颜色和光源对象的颜色,如果它的值是 1, 镜面反射光颜色就仅与光源 Color 属性有关;在取其他值的情况下,MATLAB 就在曲面团色与光 源对象颜色之间接线性比例进行调节。
- > SpecularExponent 属性的取值是大于或等于1的数值,该属性定义镜面反射点大小。
- > SpecularStrength 属性的取值是介于 0 到 1 之间的数值,该属性定义光源在曲面上的镜面反射强度。
- VertexNormals 属性的取值是数值向量或矩阵,该属性记录着曲面顶点的法向数据。一般情况下, MATLAB 会自动生成 VertexNormals 属性值,用户也可以自定义法向数据。MATLAB 生成曲面光 照效应时会使用 VertexNormals 数据。这样。就可以达到各种光照效应下的曲面。

## (3)辅助属性

▶ EraseMode 属性与线段对象 EraseMode 属性的取值及意义相同。

## 6.5.9 区域片对象

区域片对象(Patch)是图形对象的一种,它是坐标系对象的子对象。区域片对象的图形是由平面或空间中某 些顶点构成的 3 维立体,例如长方体就可以看成是具有六个面的区域片对象。每个面是由一些顶点组成的多边 形,多边形可能是凹的或自相交的。可以将曲面对象看成为一个特殊的区域片对象,因为曲面对象的每个面就 是由 4 个顶点构成的多边形区域。MATLAB 对区域性的概念略有更新,区域先被认为是由面构成的几何体。

1.区域片对象创建函数

区域片对象的低层创建函数是 patch, 它的命令形式有下列几种

- patch(x, y, c)
- patch(x, y, z, c)

patch(..., '属性名', 属性值, ...)

patch('属性名',属性值...)

 $h = patch( \dots )$ 

每种命令形式都可在当前坐标系中创建一个区域片对象,最后一种形式还返回创建区域片对象的句柄值。 如果不给出"属性名 / 属性值"对,MATLAB 就用区域片对象的缺省值创建该区域片对象。

在这些命令形式中,变量 x、y、z 为向量或矩阵,它们的维数相同,对应的元素定义顶点的坐标值。若它 们都是矩阵,则 MATLAB 按列定义区域片对象的面,变量 c 称为颜色数据,它的解释较复杂,与区域片的其他 属性值有关,共同定义区域片上的着色方式。假设 x、y、z 的维数都是 m×n ,那么当 c 的取值是数值(作为 Colorrmap 的索引值)或全真颜色 RGB 单色值时,MATLAB 对所有的区域都着这种颜色;当 c 的取值是 1×n 向量(作为 Colormap 的索引值)或 n×3(或 1×n×3)RGB 全真颜色值时,MATLAB 对每个不同面将着不同的颜色;当 c 的 取值是 m×n 矩阵(作为 Colormap 索引值)或 m×n×3RGB 全真颜色值时,MATLAB 将 c 定义的颜色对应到每个 顶点,而面可以用 flat 或 interp 模式着色。另外,在某些情况下,着色还受 FaceVertexCData 属性值的影响。

🔉 这里说的着色是对面的内部的着色,区域片边界的颜色由 EdgeColor 等属性定义

第一种调用形式在平面上定义区域片对象,第二种形式在空间中定义区域片对象。patch 创建函数与 fill 等高层创建函数的重要区别在于,patch 函数不检查当前坐标 NextPlot 属性值,它总是直接向当前坐标系中添加新的区域片对象,而高层区域片生成函数 fill 等需调用另一个函数 newplot 来决定当前坐标系 NextPlot 属性的状态,除非 NextPlot 属性值为 add。fill 等在输出图形之前,会先删除当前坐标系中已存在的图形对象。

当没有坐标系存在时, patch 函数创建的坐标系的视点为[0 90], 即 2 维坐标, 如有必要, 要由用户用 view 等函数将其转换到 3 维坐标系。

有两种方式可以定义区域片对象,第一种是直接定义顶点坐标和颜色值,例如

x=[0 1;1 1;0 0];

y=[2 2;2 1;1 1];

z=[1 1;1 1;1 1];

id_color=[24 37];

patch(x, y, z, id_color)

另一种更方便的方式是先定义所有的顶点,然后定义顶点之间的联系,即每个面由哪些顶点构成,最后定 义颜色数据。例如,下面语句组的作用与上述例子相同

vert=[0 1 1;0 2 1;1 2 1;1 1 11;

fac=[1 2 3;1 3 4];

id_color=[24 37];

patch('faces', fac, 'vertices', vent, 'FaceVertexCDatar', id_color)

vert 矩阵中的每一行数值为顶点 x、y、z 坐标值,而 fac 短阵的元素值是 vert 中顶点的序号,它的每行定义 区域片对象的一个面,当然每个面的顶点个数不必相同,顶点数较少时,可以用 NaN 代替 vert 使得 fac 每行元 素个数相同。

2.区域片对象属性

区域片对象属性大体上与曲面对象属性相同,并通过这些属性来管理区域片对象。

(1) 外形属性

- CData 属性的取值可以是数值或 MATLAB 预定义的颜色字符值,也可以是向量,还可以是数值矩 阵或全真颜色 RGB 3 维数组,该属性定义区域片颜色及着色方式。MATLAB 针对不同的 CData 值,来解释如何对区域片对象进行着色。当 CData 属性值的数据是数值时,MATLAB 或者将数据 按线性方式映射到整个 Colormap 色谱上,或者将数据直接作为 Colormap 色谱的索引值,例如对 整数值就是如此,当然也依赖于 CDataMapping 属性。如果 CData 数据是 RGB 3 维数组,MATLAB 就直接引用 RGB 值定义的颜色,而不必在 Colormap 色谱中作索引,但是当计算机系统不支持全 真颜色时,MATLAB 就用 Dithermap 色谱作全真颜色模拟。属性值矩阵维数与坐标值 XData、 YData、zData 的维数之间有一定的匹配关系。如果 CData 属性值是长度为 n 的数值向量,或者是 1×n×3RGB 全真颜色值,那么区域片的 n 个面分别用这 n 种颜色着色;如果属性值是与坐标值 同维数的矩阵,或 m×n×双 GB 全真颜色值,那么区域片对象的每个顶点对应一种颜色,而每个 区域片上的着色方式分为 f1at 和 interp 两种方式。
- CDataMapping 属性的取值是 scaled(缺省值)或 direct,该属性定义对索引式颜色数据矩阵的解程方式。如果 CData 定义的是全真颜色数据 则这个属性无任何作用。当该属性值为 scaled 时,MATLAB 会根据坐标系 CLim 属性值,按线性方式将 CData 数据映射到 Colorrmap 索引号中;如果属性值为 direct,则 MATLAB 将 CData 数据直接解释为 Colormap 索引号。因此,在这种情况下,CData 数据通常为整数。如果数值小于1,就解释为1,数值大于 length(colormap),就解释为 length(colormap)。对于非整数数值,则按其整数部分截断。
- ▶ EdgeColor 属性的取值是 MATLAB 合法的颜色值,以及 none、f1at 或 interp, 取缺省值时为黑色。

该属性定义单个区域片对象边界线颜色或着色方式。如果属性值是 RGB 颜色值或 MATLAB 预定 义的颜色的字符,那么边界线就用这种颜色着色。取缺省值为 b 时,边界线为黑色,但是边界线 上顶点处的标记不受该值。属性值为 none,则不画边界线。属性值为 flat,则边界线上两顶点间 线段的颜色与起点的颜色相同,边界方向是由给定坐标值的顺序确定的;如果属性值为 interp, 则根据两顶点的颜色值按线性括值方式对两顶点之间的线段进行着色。

- FaceColor 属性的取值是 MATLAB 合法的颜色值, none、flat 或 interp, 缺省值为w, 该属性定义 区域片酌颜色与着色方式。如果属性值是 RGB 颜色值或 MATLAB 表示预定义颜色的字符,则用 该颜色对整个区域片着色,区域边界线除外;属性值为 none,则不对区域片进行着色,此时区域 片是由边界线构成的透明片;属性值为 flat(缺省值),那么区域片中每个面的颜色就由其第一个顶 点的颜色决定;属性值为 interp,则用每个面片上顶点对应的颜色值按多线性插值的方式定义面 上的颜色。
- Faces 属性的取值是 m×n数值矩阵,称为连接矩阵。它定义了用 Vertices 属性定义顶点的连接关系。连接矩阵的元素都是 Vertices 中顶点的序号,或 NaN,每一行的顶点构成区域片对象的面。 Faces 属性和 Vertices 属性一起提供了一种更经济地定义区域片对象的方法。
- ➢ FaceVertexCData 属性的取值为矩阵,它决定了由 Faces 属性和 Vertices 属性定义的区域片及顶点的颜色与着色方式。FaceVertexCData 属性值的解释依赖于它的维数,对于索引式颜色数据,属性可以取单个颜色值,它定义整个区域片的颜色;可以是 n×1 向量,这里 n 是 Faces 属性值的行数,它定义区域片每个面的颜色;也可以是 n×1 向量,这里 n 是 Vertices 属性值的行数,它定义每个顶点的颜色。对于全真颜色 RGB 值,FaceVertexCData 属性的值可以是 1×3 矩阵,它定义整个区域片的颜色;可以是 n×3 矩阵,其中 n 是 Faces 属性值的行数,它定义区域片每个面的颜色;可以是 n×3 矩阵,其中 n 是 Vertex 属性值的行数,它定义每个顶点的颜色。
- LineStyle 属性的取值是 "-"(实线:缺省值)、"--"(虚线)、":"(点线)、"-."(点虚线)或 none(无线), 这是 MATLAB 预定义的线段类型。该属性决定区域片边界线所用的线段类型。
- ➢ LineWidth 属性的取值是以磅为单位的数值,缺省值为 0.5 磅,定义区域片边界线的宽度。增大 LineWidth 属性值,可以得到较粗的边界线。
- Marker 属性的取值与线段对象的 Marker 属性的取值相同。该属性定义区域片顶点上的标记,缺 省属性值是 none,表示无顶点标记。Marker 属性与 LineStyle 属性无关,可以通过这二者的不同 组合得到不同式样的区域片边界线。
- MarkerEdgeColor 属性的取值及意义与线段对象 MarkeEdgeColor 属性相同。如果属性取值为 auto,则用区域片对象 EdgeColor 属性定义的颜色对标记或标记边界进行着色。
- MarkerFaceColor 属性的取值及意义与线段对象 MarkerFaceColor 属性相同。MarkerFaceColor 取值 为 auto,则内部区域用坐标系背景色或图形窗口背景色填充。
- > MarkerSize 属性的取值是以磅为单位的数值,缺省值为6磅,该属性决定顶点标记的大小。
- Vertices 属性的取值是 n×3 矩阵,它定义区域片各顶点的坐标值。每一行对应一个顶点,其值为 x、y、z 坐标值。
- XData、YData、ZData 3 种属性的取值都是数值向量或矩阵,它们的对应元素构成区域片对象各 顶点的 x、y、z 坐标值,区域片对象就是这些顶点定义的。

(2) 光照效应属性

控制光照效应的属性的属性名、属性取值、属性的含义与曲面对象的光照效应属性完全相同,在此不再重

- 复。
- (3) 辅助属性
  - EraseMode 属性的取值是 normal(缺省值)、none、xor 或 background,该属性定义 MATLAB 系统 使用何种方式在屏幕上画区域片对象或撩去区域片对象,这些技术在生成动画序列时待别有用。 在 normal 模式下,当画区域片或者擦去区域片时,会重新刷新有变化的图形部分,必要时还要进行3 维分析,保证所有的图形对象被正确着色。虽然在这种模式下,重新刷新的图形对象还保持 原有的着色和层次,但是这种模式速度太慢,并且要消耗大量内存;在 none 模式下,当区域片对

象被删除时、该区域片对象原来的图形仍保留在屏幕上;在 xor 模式下,当在屏幕上面区域片或 者控去区域片对象时,用区域片每个像素的颜色与区域片下屏幕上像素应有的颜色值作"异或" 运算,所以最后输出的图形的颜色依赖于屏幕上原有的颜色;在 background 模式下,为了擦去区 域片,要用坐标系的背景颜色画区域片,这样做的结果会破坏区域片下原有的图形对象的着色。

6.5.10 图像对象

MATLAB 通过创建图像对象的方法显示一幅图像,图像对象也是图形对象的一种,它是坐标系对象的子对象,即第三级子对象。图像对象本身没有子对象。

1.图像对象创建函数

图像对象的创建函数是 image, 它既是高层创建函数, 也是低层创建函数, 视其调用形而定。调用 image 函数的形式有如下几种:

image(c)

image(x, y, c)

image(...,'属性名',属性值,...)

ma8e('属性名',属性值,...)

h=image( ... )

前 3 种调用形式是其高层调用,第四种形式是其低层调用,而第五种调用形式则返回创建图像对象的句柄 值。低层调用直接在当前坐标系中显示图像对象,而高层形式调用 newplot 函数决定坐标系并检查坐标系 NextPlot 属性值,再决定加入图像对象方式。这些语句都是用于创建图像对象,将图像数据 c 按其数据形式或者解释为 图形窗口 Colormap 的索引值,或者直接解释为 RGB 全真颜色值。

图像数据 c 可以是索引式颜色值,也可以是 RGB 全真颜色数组值。取索引式颜色值时, c 数据作为图形窗 口 Colormap 色谱的索引来引用颜色对因像对象的相应点进行着色;取 RGB 全真颜色数组值时, MATLAB 直接 用 RGB 定义的颜色对图像对象的相应点进行着色。在不支持全真颜色的计算机系统上,用 Dithhermap 色谱中 定义的颜色作模拟图像颜色,图像数据保存在图像对象 CData 属性中,所以它或者是 m×n 数值矩阵,或者是 m ×n×33 维 RGB 颜色数组。

在 MATLAB 中,图像数据有两种数据格式,即 8 位数据格式和 64 位数据格式(双精度)。通常,MATLAB 在内存中对任何数据都使用双精度格式(不管它的显示格式是什么),由于图像数据的数量很大,为了节省内存,MATLAB 提供了 8 位数据格式。在显示图像时,MATLAB 对这两种数据格式的数据使用不同的解释。对于索引式颜色数据矩阵 c,如果它是 64 位数据格式的,那么其元素值通常在范围[1,length(colormap)]内。当元素值为 1 时,就映射到 Colormap 色谱的第一行,如此等等。如果它是 8 位数据格式,那么其元素值在范围[0 255]中,那 么作索引映射时必须偏移一位.即数值 0 映射到 Colormap 色谱第一行,如此等等。对于 3 维 RGB 颜色数组 c,在 64 位数据格式下,各元素之值落在[0 1]范围内;而在 8 位数据格式下,各元素之值落在[0 255]范围内。

2.图像对象属性

(1)外形属性

- CData 属性的取值可以是数值矩阵或全真颜色 RGB 3 维数组,该属性定义图像对象的颜色数据。 函数 image(c)将变量 c 的值传送给 CData 属性,MATLAB 针对不同的数据 CData,按不同的方式 进行解释:如果 CData 是数据矩阵,则或者将其值直接作为 Colormap 色谱的索引值,或者按线 性方式映射到整个 Colormap 色谱上,这取决于 CDataMapping 色谱的索引值,或者按线性方式映 射到整个 Colormap 色谱上,这取决于 CDataMapping 属性值;如果 CData 数据是 RGB 3 维数组, MATLAB 就直接引用 RGB 值定义的颜色,而不必在 Colormap 色谱中作索引,但是当计算机系统 不支持全真颜色时,MATLAB 就用 Dithermap 色谱作全真颜色模拟。
- CDataMapping 属性的取值是 scaled 或 direct(缺省值),该属性定义对索引式颜色数据矩阵的解释 方式。如果 CData 定义的是全真颜色数据,这个属性无任何作用。当该属性值为 scaled 时,MATLAB 根据坐标系 CLim 的属性值,按线性方式将 CData 数据映射到 Colormap 索引号中;如果它的属性 值为 direct,那么 MATLAB 将 CData 数据直接解释为 Colormap 索引号。因此,在这种情况下 CData 数据通常为整数。如果数值小于1,就解释为1;数值大于1ength(Colormap),就解释为 1ength(Colormap)。对于非整数数值,则按其整数部分裁断。
- ➢ XData、YData 两种属性都是 2 元素向量。通常, XData 的取值是[1, size(c, 2)], YData 的取值是 [1, size(c, 1)], 它们分别定义了图像占用的 x、y 坐标范围。
- (2) 辅助属性
  - EraseMode 属性的取值是 normal(缺省值)、none、xor 或 background,该属性定义 MATLAB 系统 使用何种方式在屏幕上显示和接去图像对象,这些技术在生成动画序列时特别有用。在 normaL(缺 省)模式下,当画或者擦去图像时,重新刷新有变化的图形部分,必要时还要进行3维分析,保证 所有的图形对象被正确着色。虽然在这种模式下,重新刷新的图形对象保持原有的着色和层次, 但是这种模式速度太慢,并且要消耗大量内存。在 none 模式下,当图像对象被删除时,该图像对 象的图形仍保留在屏幕上;在 xor 模式下,当在屏幕上画或者擦去图像对象时,MATLAB 用图像 每个像素的颜色值与图像下屏幕上像素现有的颜色值作"异或"运算,所以最后输出的图形的颜 色依赖于屏幕上原有的额色;在的颜色值作"异或"运算,所以最后输出的图形的颜色依赖于屏 幕上原有的额色;在 back8round 模式下,为了擦去图像,MATLAB 用坐标系的背景颜色重新显 示图像。这样他的结果会破坏图像下原有的图形对象的颜色。

6.5.11 文字对象

文字对象主要用于在坐标系中写说明文字。在文字对象中还可以使用希腊字母、数学符号及公式等,这就 是 MATLAB 处理文字对象时对 LATEX 文本(LATEX 是一种用于数学文章的排版语言及排版系统。)的解释功能。 这样就可以在坐标系中添加更形象的说明文字。

1. 文字对象创建函数

文字对象的低层创建函数是 text, 交互式的高层文字对象创建函数是 gtext。可以有下列几种命令形式:

text(x, y, '字符串')

text(x, y, z, '字符串')

text(..., '属性名', 属性值, ...)

h = text(...)

第一种命令形式是在 2 维坐标系,或者以 z 坐标值为 0,在 3 维坐标系的指定位置上写出给定的字符串(文 字);第二种命令形式是在 3 维坐标系的指定位置上写出给定的字符串(文字);第三种命令形式是根据指定位置 和属性值写出说明文字;第四种命令形式返回创建的文字对象的句柄值。由于 text 是低层函数,所以它只是向 当前坐标系添加文字对象,而不改变坐标系的属性。如果给出的坐标值超出了当前坐标系的范围,则文字对象 是不可见的。坦所创建的文字对象是存在的。

变量 x、y、z 是同长度向量,而字符串变量可以是字符矩阵.其行数与坐标变虽向量长度相同,也可以是 字符串块数组或用竖线 "|"隔开的字符串,其块个数与坐标变量向量长度相同。此时,MATLAB 将创建多个文 字对象。

坐标值定义文字对象字符串定位的相对点,文字对象 VerticalAltgnment 属性和 HorizontalAlignment 属性决定文字对象的最后位置。坐标值按照当前坐标系的数值单位解释。

2. 文字对象属性

(1) 基本属性

Color 属性的取值是 RGB 颜色值或 MATLAB 预定义的颜色名称,它定义文字对象的显示颜色,其缺省值为w(白色)。

Interpreter 属性的取值是 latex(缺省值)或 none,该属性控制 MATLAB 对 String 属性中字符及控制字符串的 解释方式。在 latex 模式下,MATLAB 按 LATEX 意义解释 String 属性中的控制符。LATEX 是一种十分流行的 数学公式排版工具,在 LATEX 意义解释下,MATLAB 可以在说明文字加入数学公式以及希腊字母等;在 none 模式下,则按 ASCII 码解释控制字符串。

String 属性的取值是字符串、字符串矩阵或字符串块数组。它记录着文字对象的说明文字内容。在 MATLAB 中,文字对象既可以是单行说明文字,也可以是多行说明文字,后者由字符串矩阵或字符串块数组定义。在字 符串中,除了标准的 ASCII 字符外、还可以有以 LATEX 命令形式出现的控制字符串。在 latex 模式下,MATLAB 将控制字符串解释为数学符号或希腊字母等。为了嵌入特殊符号和字母(如俄文字母等),可以使用其他系统字 库。控制字符串"\fontname(字库名)",表示使用给定的字库。利用 LATEX 的变换字体的命令,还可以改变说 明文字中部分字符的字体,这些命令包括" \bf "(黑体)、" \it "(斜体)、" \sl "(斜体的一种,很少使用)、" \rm "(正体)。对于在这些命令的作用范围之外的字符,仍使用字体属性定义的字体。

Units 属性的取值是 pixels(像素)、normalized(相对单位)、inches(英寸)、centimeters(厘米)、points(磅)、data(坐标数据单位:缺省值),该属性定义 Extent 属性和 Position 属性的度量单位。坐标数据单位是指按坐标系各坐标单位进行解释。其他单位都是从坐标系矩形的左下角开始度量的。

(2)字体属性

FontAngle 属性的取值是 normal(缺省值)、italic 或 oblique ,该属性定义文字对象使用的某种系统字库 ,normal 表示使用系统的正体字库 , italic 和 oblique 在通常的计算机系统上都表示使用系统的斜体字库。

FontName 属性的取值是字符串,它给出文字对象使用的字库类名,缺省值是 Helvetica。

FontSize 属性的取值是整数值,它定义文字对象使用的字体大小,其单位由 FontUnits 属性定义。缺省值是 10,以磅为单位。

FontWeight 属性的取值是 light、normal(缺省值)、demi 或 bold。该属性决定文字对象字体的租细,MATLAB 据此从系统字库中挑选一种适当的字库。通常取 demi 和 bold,都表示黑体。

FontUnits 属性的取值是 points(磅 缺省值)、normalized(相对单位)、inches(英寸)、centimeters(厘米)或 pixels(像素),该属性定义 FontSize 属性的度量单位。相对单位将 FontSize 属性值解释为坐标系高度的比例。

(3) 位置属性

Extent 属性的取值为 4 元素数值向量[1eft, bottom.width, height], 该属性只可读,用户不能改写。它定 义了文字对象所占据的矩形区域的大小,(1eft, bottom)是该区域左下角相对坐标系矩形左下角距离的值,而 width 和 height 是区域宽度和高度值,其单位由 Units 属性定义。

HorizontalAlignment 属性的取值是 1eft(缺省值)、center 或 right,它定义文字对象在水平方向相对于 Position 点的对齐方式。

Position 属性的取值是 2 元素或 3 元素数值向量[x, y]或[x, y, z], 如果略去 z, 则在系统内以 0 代替 z, 缺省 值为[0, 0,0]。它定义坐标系中的一个参考点,文字对象以该点定位,并根据 HorizontalAlignment 和 VerticalAlignment 属性值决定文字对象的最后位置。

Rotation 属性的取值是数量值,缺省值为0,它定义文字对象的旋转角度,取正值时表示逆时针方向旋转, 取负值时表示顺时针方向旋转。

VerticalAlignment 属性的取值是 top、cap、middle(缺省值)、baseline 或 bottom,该属性定义文字对象的纵向 定位方式。top 表示将文字对象的顶端在纵向上与 Position 参考点对齐; cap 表示将文字对象中的大写字母的顶 端与 Position 参考点对齐,这种方式大体上与 top 方式是一样的;middle(缺省方式)表示将文字对象的纵向中滨 与 Position 参考点对齐; baseline 表示将文字对象字符串的基线点与 Position 参考点对齐; bottom 表示将文字对 象的底端与 Position 参考点对齐。

(4)辅助属性

EraseMode 属性的取值是 normal(缺省值)、none、xor 或 background,该属性定义了 MATLAB 系统使用何种 方式在屏幕上写文字对象或擦去文字对象。在 normal(缺省)模式下,当写或者擦去说明文字时,MATLAB 重新 刷新有变化的图形部分,必要时还要进行3维分析,保证所有的图形对象被正确着色;在 none 模式下,当文字 对象被删除时,该文字对象的图形仍保留在屏幕上;在 xor 模式下,当在屏幕上写或者镑去文字对象时,用文 字上每个像素颜色值与文字下屏幕上像素原有的颜色值作"异或"运算,所以最后输出的图形的颜色依赖于屏 幕上原有的颜色;在 background 模式下,为了擦去说明文字,MATLAE 用坐标系的背景颜色重写说明文字,这 样做的结果会破坏文字下原有的图形对象的着色。

6.5.12 光源对象

光源对象仅对区域片对象和曲面对象产生光照效应,利用光源对象可以生成更逼真的3维实体模型和曲面。 用户不仅可以控制光源的方向,还可以控制光源的位置、颜色、强度等。光源对象在坐标系中是不可见的,它 的存在是通过它对区域片对象和曲面对象的作用效果来反映的。

1. 光源对象创建函数 创建光源对象的函数是 light,它具有下列两种命令形式 light('属性名',属性值...) h=light('属性名',属性值...)

它们的作用都是在当前坐标系中创建光源对象,除非对 Parent 属性给定特定的坐标系句柄值。第二种形式 还返回所创建的光源对象的句柄值。

光源对象对区域片对象和曲面对象产生光照效应,其效果是由区域片对象和曲面对象的 AmbientStrength、 DiffuseStrength、SpecularStrength、SpecularExponent、SpecularColorReflec tance、VertexNormals 等属性共同作用 的结果。

2. 光源对象属性

由于光源对象对用户来说是不可见的,因此,许多图形对象的公共属性对光源对象就毫无意义。事实上, 光源对象只使用如下一些基本属性。

- Color 属性的取值是 RGB 颜色值或 MATLAB 预定义的颜色名称,它定义光源对象发出的光的颜色,缺省颜色为白色。
- Style 属性的取值是 infinite(缺省值)或 local,它定义光源对象的类型。在 infinite 模式下,MATLAB 将光源点置于无穷远处、相当于是平行光束;在 local 模式下,光源点被置于坐标系的特定位置上, 此时光线是从该点发出的散射光束。该属性值决定了 Position 属性的解释方式。
- Position 属性的取值是 3 元素数值向量[x, y, z],它定义了光源对象的位置。如果 Style 属性值为 local,那么光源对象置于由该向量定义的坐标位置上,它发出的光线是散射光束;如果属性值为 infinite,那么该向量将决定无穷远光源对象发出的平行光束的方向。

### 6.5.13 缺省属性及其设置

为了应用上的方便,MATLAB 对所有图形对象的属性都规定有缺省的属性值,又称为 factory 定义值。 MATLAB 也允许用户定义自己的缺省属性值。从定义处开始,用户定义的缺省属性值将覆盖 factory 定义值,随 后的图形对象生成函数控用户定义的缺省值来设置末指定的属性值。

任何一个图形创建函数启动时,对于没有提供属性值的那些属性,MATLAB都将使用属性缺省值。缺省属 性值的寻找过程是这样的:首先从当前的图形对象开始逐步向上检查它的高层图形对象类,直到找到用户定义 的缺省值或 MATLAB预定义的缺省值,即 factory 属性值。用户定义的缺省值优先级最高。在使用缺省值时, 以当前被创建图形对象的最邻近高层图形对象类中用户定义的缺省值优先,而 factory 定义值的优先级最低,也 就是 factory 定义值的定义级别最高,但它的使用级别最低。当用户定义了新的缺省属性值后,MATLAB将保留 着 factory 定义值的备份,必要时可以恢复 factory 定义值。

新的缺省属性值一经定义后,会在创建该类图形对象时随即发挥作用,除非在低层创建函数中用"属性名/属性值"指定了特殊值。但是,新的缺省值对已存在的该类图形对象没有影响。

1. 定义缺省属性值

除了最低级别的图形对象外,MATLAB中的每种类型的图形对象都为其各于级图形对象的属性定义了缺省 属性名,借此来设置子级图形对象的缺省属性值。缺省属性名由3部分组成,第一部分是 default,第二部分是 图形对象类型名,第三部分是该类型图形对象的属性名。例如,DefaultLineLineWidth,这个缺省属性名的值就 是线段对象的线段宽度的缺省值。

MATLAB 图形对象缺省值的定义方法是:在其高层图形对象上用 set 函数将相应的缺省属性名定义为所期望的缺省值,其一般命令形式为

set(handle, '缺省属性名',属性值)

其中,handlc 是高层图形对象的句柄值。那么在这个高层图形对象的范围内。所有相应的子图形对象就使用所定义的缺省值,定义缺省值的高层图形对象级别越高,相应的缺省属性值的作用范围就越广。

例如,为了在当前的窗口内用白色画线段对象,只要将线段对象的 Color 属性的缺省值设置为白色,即 set(gcf, 'DefaultLineColor', 'w')

随后,无论在这个图形窗口的哪个坐标系中画线段对象,都用白色对线段对象进行着色。由于线段对象是 坐标系对象的子对象,因而也可以在坐标系设置线段对象属性的缺省值。只有在该坐标系中输出线段对象时, 才使用用户定义的缺省值。例如:

set(handle_axes, 'DefaultLineColor', 'w')

由于线段对象是坐标系对象的子对象,又是图形窗口对象的子对象的子对象,因此,DefaultLineColor 是线

245

段对象的高层对象,即轴对象、图形窗口对象和根对象的缺省属性,而不是线段对象自身的属性。线段对象是 最低层的图形对象,MATLAB 对最低层的图形对象没有定义缺省属性名。又如,如果要将图形窗口对象的缺省 背景色黑色改为蓝色,那么必须在图形窗口对象的父对象,即根对象中定义相应的缺省属性设置

set(0, 'DefaultFigureColor', 'w')

此后,再创建的图形窗口的背景颜色就是蓝色。

如果用户对某些属性总是有自己的特殊要求,那么可以在根对象层上设置所期望的缺省属性值,并把这些 设置语句写入到 startup.m 文件中,这样 MATLAB 系统启动后就在根对象层上定义了用户的缺省属性值。

2. 缺省属性值的使用

在创建图形对象时,没有被指定的属性,将使用用户定义的缺省属性值或 factory 定义值。MATLAB 还提供 了一种方法,可以将已存在的图形对象的任何属性值设定成用户定义的缺省值或 factory 定义值。在 MATLAB 中,任何属性都接受 default 和 factory 值输入。属性值" default"含义是:将在根据缺省值搜索过程用首先遇到 的缺省值作为相应的属性值。例如,下面的语句设置曲面的 EdgeColor 属性为绿色

h = surfc(peaks);

set(0, 'DefaultSurfaceEdgeColor' , 'g') ;

set(h, 'EdgeColor', 'Default')

第一条语句对创建的曲面网格线着黑色,这是因为系统定义的曲面对象 EdglColor 的缺省属性值为 k(黑色), 即 factory 定义值为 k;第二条语句是在根对象层上定义曲面对象 EdgeColor 的缺省属性值为 g(绿色);而第三条 语句的作用是将已存在的曲面对象 EdgeColor 属性值设置为用户定义的缺省值,即"g"。虽然 EdgeColor 的缺省 属性值是在根对象层上定义的,但是它的定义级别低于 factory 定义值的定义级别,所以执行第三条语句后, MATLAB 就使用根对象层上定义的缺省属性值。

如果 EdgeColor 属性的缺省值已经在坐标系对象或图形窗口对象中存在,则它们将首先被使用,代替在根 对象层中设置的缺省值。

属性值 factory 的含义是将相应的属性值设置成 factory 定义值。例如下面的语句将曲面 h 的 EdgeColor 属性 重新设置成黑色

h = surf(peaks);

set(0, 'DcfaultSurfaceEdgeColor', 'g')

set(h, 'EdgeColor', 'Default')

pause

swt(h, 'EdgeColor', 'factory')

🔉 如果对于某个属性没有用户定义的缺省属性值存在,那么 default 和 factory 的作用是相同的

3. 用户定义缺省值的删除

一旦用户定义了自己的缺省属性值,MATLAB 总是试图使用用户定义的缺省属性值。在其后的过程中,有可能不再希望使用己定义的缺省属性值,这时就需要删除用户定义的缺省属性值。为此,MATLAB 对所有的缺 省属性都定义了属性值 remove,其含义是删除用户定义的缺省值,例如,语句 set(0, 'DefaultSurfaceEdgeColor', 'remove')的作用就是删除在根对象层中定义的曲面 EdgeColor 属性的缺省值。

## 6.6 图像操作

MATLAB 在图像处理方面也提供了强大的功能,本节将为大家介绍其中的基本内容,包括图像的读写和显示、修改。

6.6.1 图像的读写

MATLAB 中的不同类型图像有自己固定的数据格式,它们都是以数组的形式存储的,灰度图像存储在二维数组中,真彩色(RGB)图像存储在三维数组中。

要在 MATLAB 下使用在其他软件中生成的图像,需要用 imread 函数读取该图像,这实际上也是一个数据转换的过程,即把该图像的数据转换为 MATLAB 图像的数据格式。

函数 imread 的调用格式为:

- A=imread(filename, fmt) 式中字符串"filename"是图像的文件名,字符串"fmt"指定图像的 类型。数组"A"是由图像文件中读出来并转化为 MATLAB 可以识别的图像格式的数据,如果该 图像是灰度图像,那么数组"A"是二维的,如果图像是真彩色的,那么数组"A"是三维数组。
- ▶ [x, map]=imread(filename, fmt) 把经转化的索引图像数据保存到数组"x"中,同时把相关的颜 色映象读到数组"map"中。
- ▶ […]=imread(filename) 通过文件的内容自动判断其格式。
- 由"filename''指定的文件必须位于当前路径下,或在 MATLAB 指定的路径范围内

MATLAB 支持的图像类型(字符串"fmt"的可能取值)为: JPEG('jpg'或'jpeg'), TIFF('tif'或'tiff'), BMP('bmp'), PNG('png'), HDF('hdf), PCX('pcx'), XWD('xwd')。其中括号内的字符串为"fmt''的可能取值。

比如,准备读取当前目录下的一个真彩色图像文件"12.jpg",我们可以用如下三种方式中的任意一种

```
A=imread('12.jpg', 'jpg');
```

```
A=imread('12', 'jpg');
```

A=imread('12.jpg');

```
结果读进来的数组"A"为三维数组,大小为
```

3

size(A)

ans =

X

571 800

可以看到三维数组 " A " 有三个面,他们依次对应 R、G、B 三种颜色,而面中的数据则分别是这三种颜色的强度值,面中的元素对应于图像中的象素点,因而面中的行数和列数与图像中象素的行数和列数是一致的。

MATLAB 中的函数 inswrite 用于把图像输出到文件,函数调用格式如下

- ➢ imwrite(A, filename, fmt) 把图像的数据"A"输出到文件"filename",图像的类型为"fmt"。
- ➢ imwrite(...,filename) MATLAB 自动从文件的扩展名推断图像类型。
- imwrite(...,param1,vall,param2,val2,...) 设置输出图像文件的参数,可以设置的参数及取值请参见关于图像处理的书籍。

例如,如下命令把前面读进来的图像另存为 BMP 类型的文件

imwrite(A,'13.bmp', 'bmp') ;

保存后的文件"13.bmp"占了1.3M字节硬盘空间,而原文件"12.jpg"只有24k字节,这是由于 BMP 文件 未经过压缩,另外,文件的大小还和参数的设置有关。

### 6.6.2 图像显示

读进来的图像用函数 image 可以在 MATLAB 环境下显示。例如,前面读进来的图像数据可以用如下命令显示

image(A)

axisimage

axisoff

显示的结果如图 6-82 所示。



图 6-82 读取并显示的图像文件

其中,命令 axis image 设置坐标轴的纵横比保持不变,从而避免图像发生变形。

#### 6.6.3 图形窗口的编辑

MATLAB 提供了图形窗口的交互方式,按下按钮 < 后就允许对窗口中的图形对象进行交互编辑。例如,

### 在命令窗口中输入

t=0:0.1:2*pi; y1=sin(t); plot(t,y1,':'); hold on y2=cos(t); plot(t,y2,'*'); xlabel('Time(0-2\pi)','FontWeight','bold'); text(pi,0,'\leftarrowsin wave'); text(pi/2,0,'\leftarrowcos wave'); hold off 回车将在图形窗口显示如图 6-83 的图形。



图 6-83 包含坐标轴、线条和文本对象的图形窗口

按下按钮 🕨 后就允许对窗口中的坐标轴、线条和文本对象进行交互式编辑。

#### (1) 编辑坐标轴属性

在坐标轴范围内的空白区域或坐标轴的边框处单击鼠标右键,坐标轴在四个角和四条边的中央会各出现一个黑色的小方块,表示坐标轴已被选中,同时弹出一个菜单(这种菜单就是上下文菜单(Context menu),此时的状态如图 6-84 所示。



图 6-84 成为编辑状态的坐标轴及弹出菜单

弹出菜单中的 Show Legend 选项用于显示缺省的图例。Unlock Axes Position 用于解锁坐标轴的位置,选中这一项后,可以用鼠标拖动坐标轴来改变其位置,缺省情况下坐标轴的位置是锁定的,不能直接移动。

选择菜单中的 Properies 选项,将弹出如图 6-85 所示的编辑坐标轴属性对话框。这个对话框也可以通过用鼠标左键双击坐标轴范围内的空白区域或坐标轴的边框得到。还可以先选中要编辑的坐标轴,然后选择菜单 Edit下的 Axis Properties 选项。

🖗 Property Editor - Axes				_ 🗆 🗙
Edit Properties for: axes :				
Scale Style Labels Aspect	Lights V	iewpoint 📔 Info	1	
×		Y		z
Limits 🔽 Auto 0.00 7.00	Auto -1.00	1.00	Auto -1.00	1.00
Ticks 🗹 Auto [ 0.0 1.0 2.0 3.0 4.0 5.0	Auto [-1.0 -	0.8 -0.6 -0.3999!	Auto [-1.0 0	.0 1.0 ]
Labels 🔽 Auto 01234567	Auto 1	<u> </u>	Auto Auto	<u>^</u>
	-0.8 -0.6	<b>v</b>		<b>v</b>
Scale 🖲 Linear 💽 Normal	Linear	Normal	Linear	Normal
C Log C Reverse	🔿 Log	C Reverse	🔿 Log	C Reverse
Grid 🗖 Show	☐ Show		☐ Show	
Set axes auto shane Set tight limits				
OK Cancel Apply	I Immed	iate apply		Help
Ready Apply		and alters		

图 6-85 坐标轴属性编辑对话框

在这个对话框中的各个选项卡分别可以编辑坐标轴的如下几个属性

- Scale 坐标类型(线性、对数)和坐标轴的方向、刻度范围、坐标标注、刻度间隔、网格线等
- ▶ Style 坐标轴的色彩、线宽、显示位置、标注字体及是否隐藏
- ▶ Label 标题及坐标轴的标注、
- ➢ Light 光源设置

➢ Viewpoint 视点设置

➢ Aspect 外观设置

➢ info 相关信息

在这里设置坐标轴属性和使用相关命令进行设置的效果是相同的。

### (2) 编辑线条属性

在需要编辑的线条上单击鼠标右键,线条的所有数据点都会成为一个黑色的小方块,表示线条已被选中, 同时弹出一个菜单,此时的状态如图 6-86 所示。下一步选择菜单中的 Properties 选项,将弹出如图 6-87 所示的 编辑线条属性对话框。这个对话框也可以通过用鼠标左键双击该线条得到。还可以先选中要编辑的线条,然后 选择菜单 Edit 下的 Current Object Properties 选项。





Property Editor - Line				
Data Style Info				
Line Properties	Marker Properties			
Line style Dotted line (:)	Style No marker (none)	•		
Line width 0.5	Size 6.0	•		
Color Blue	Edge color Inherited (auto)	•		
	Face color No color (none)	•		
Example				
OK Cancel Apply 🔽 In	nmediate apply	Help		
Ready				

图 6-87 线条编辑对话框

从图 8-87 中可以看到在此对话框的选项卡中能够编辑的线条属性有如下几种:

- ➢ Style 线条宽度、线型、颜色、标记的大小、
- ➢ Data 数据类型
- ➢ info 相关信息

在 Style 中 Color 的下拉选单中,提供了多种供选择的颜色,选择 "Custom Color"会弹出一个通用的颜色 设置对话框,如图 6-88 所示,可以自己定义颜色。

 Zdit
 Color
 X

 Red
 0
 25
 50
 75
 100
 0
 %

 Green
 0
 25
 50
 75
 100
 0
 %
 New

 Blue
 0
 25
 60
 75
 100
 100
 %
 Original

 OK
 Cancel
 0
 0
 100
 %
 0
 0

图 6-88 颜色设置对话框

### (3) 编辑文本属性

在需要编辑的文本上单击鼠标右键,文本框的四个角和四条边中央会各出现一个黑色的小方块,表示文本已被选中,同时弹出一个菜单,此时的状态如图 6-89 所示。下一步选择菜单中的 Properties 选项,将弹出如图 6-90 所示的文本的编辑字体属性对话框。这等效于先选中要编辑的文本,然后选择菜单 Edit 下的 Current Object Properties 选项。



图 6-89 文本对象选中状态及弹出菜单

如果用鼠标左键双击文本对象,或者从图 8-89 所示的弹出式菜单中选择 String,那么文本字符串将还原为 原始字符串的内容,并能够对其中的字符进行编辑。

用鼠标左键选中文本并拖动鼠标可以任意改变文本的位置。

🗖 Property Editor - Text		_ 🗆 🗵
Edit Properties for: text :		- 📢 📎
Text Alignment Info		
Font	Text	
Font name Helvetica	Veftarrowcos wave	*
Font weight Normal		
Font angle Normal		
Font size 10.0		
Font units Points		-
Color Black		
	Use LaTeX interpreter	
OK Cancel Apply 🔽	Immediate apply	Help
Ready		

图 6-90 文本属性编辑对话框

在图 6-90 所示的编辑文本属性对话框选项卡中能够编辑的文本属性有如下几种 → Text 文本字符串的内容、样式、大小、颜色、角度等属性 ➢ info 相关信息

#### 6.6.4 图像打印

在 MATLAB 中,有多种方法可以输出绘制的图形,其中包括

- 从菜单或命令行打印;
- ▶ 用 MATLAB 指定的打印引擎或系统的打印服务项打印
- 制作成特定的图形文件插入到其他文档中打印:
- ▶ 生成 M 文件,以便再次绘制图形。

下面分别介绍各种打印方法。

1. 从菜单中打印

在很多情况下,从图形窗口的"file"菜单中打印是最简单的方法。在不同的机型上打印情况不尽相同,下 面简单介绍在 PC 机上打印的方法。

选中 "File " 菜单下的 "Print " 项将打印图形。

选中 "Filc"菜单下的 "Page Setup"项,将弹出如图 6-91 所示的对话柜。可以设置图形的打印属性。

age Setup - Figur	e 1	X
age Setup - Figur Size and Position Mode C Use screen size O Use manual size Manual size and po Top: 2.50	• 1 Paper Lines and Text e, centered on page re and position sition Use defaults	Xes and Figure
Left: 0.25 Width: 3.00 Height: 6.00 Units: inches	Fill page Fix aspect ratio	
Help		OK Cancel

图 6-91 打印机设置

2. 从命令行打印

从命令行打印的命令为 print。当选择从命令行打印时,图形的输出由 print 命令的选项和句柄图形的属性值 决定。下面简单介绍 print 命令的使用。

函数 print 的基本调用方法为

print - 打印设备 - 打印选项

当不指定打印设备时,则以系统的默认打印机打印。下面是一个打印 cos 曲线的例子

x=-pi:0.1:pi;

plot(x, cos(x));

print

把图形打印为文件的命令格式为

print -打印设备 -打印选项 文件名

print 命令有很多选项。调用 print 命令时,若不指定这些选项, print 命令将调用命令 printopt 来获得这些选项的默认值。用户可以通过修改目录 tootbox/local 下的 printopt.m 文件来重设这些默认值。

6.6.5 在不同应用程序间拷贝图形

除了通过打印直接生成图形的硬拷贝外,MATLAB 还能把生成的图形转换成多种图形格式,以便其他府用 程序使用。

在 MATLAB 中,有两种方法用于生成图形文件。一种利用函数 print 生成图形文件,一种利用抓图函数 capture 生成位图的图形数据,再用函数 imwrite 把位图的数据写成其他格式的图形文件。下面简单介绍函数 capture 和 imwrite 的用法。

capture 函数抓取所绘制的图形 (包括图形中的用户控制界面和用户菜单界面),并转换为位图。其常用调用

### 格式为

capture(图形句柄)

这种调用格式将抓取该图形并在一个新窗口中绘制抓取到的图形。

capture 函数的另一种调用格式为

[x,map]=capture(图形句柄)

这种调用格式将返回一个矩阵 x 和色图矩阵 map,但不显示抓到的图形。下面两行命令把句柄值为 3 的图 形写入 JPEG 格式的图形文件 fig3.jpeg

[x, map]=capture(3);

imwrite(x, map, 'fig3,jpeg')

函数 imwrite 支持的图形文件的格式有:BMP、HDF、JPEG、PCX、TIFF 和 XWD。值得一提的是,函数 capture 抓取的图形为屏幕上显示的图形,所生成图形的分辨率不及原图,所以,一般在使用 capture 命令后,还 是用 print 命令生成原图的硬拷贝。

在 Pc 机和苹果机上,还可用剪贴板来复制图形,复制的格式可为位图或者图元格式。把图形拷贝到剪贴板 上的方法有以下两种

▶ 单击图形窗口的 "Edit"菜单中的 "Copy Figure"选项;

使用 print 命令,在命令中选中选顶-dbitmap 或-dmeta 命令中不提供文件名,将以位图格式或图元格式拷贝到剪贴板。

在其它的应用程序中,使用粘贴命令即可。

位图格式存储图形中的每一个点的颜色值,为标量图形;图元格式是矢量图的一种。一般情况下,位图格 式的分辨率比图元格式低。

## 6.7 动画制作

MATLAB 提供了一种能力,它可以存储一系列各种类型的二维或三维图,然后象放电影一样把它们按次序 重放出来。在某种意义上,动画提供的运动为图形增加另一个维数。通常图形的次序不必以任意的方式关联起 来。一种明显的动画类型是取出三维图形然后缓慢地将它旋转,这样我们就可以从不同角度来观察它。另一种 类型是当一个参数变化时,依次显示某些问题解的图形。

MATLAB 中的函数 moviein, getframe 和 movie 提供了捕捉和播放动画的所需工具。函数 moviein 可以产生 一个帧矩阵来存放动画中的帧;函数 getframe 对当前的图象进行快照;而函数 movie 按顺序回放各帧。照这样, 捕捉和回放动画的方法是

创建帧矩阵;

- 对动画中的每一帧生成图形,并把它捕捉到到帧矩阵里;
- 从帧矩阵里回放动画。

下面是一个绘制了函数 peaks 并且将它绕 z 轴旋转的例子

% movie making example: rotate a 3-D surface plot

[X,Y,Z]=peaks(30); surfl(X,Y,Z) axis([-3 3 -3 3 -10 10]) axis off

shading interp

colormap(hot)

m=moviein(15);

for I=1:15

```
view(-37.5+24*(i-1),30)
m(:,i)=getframe;
```

movie(m)

end

命令运行最终结果如图 6-92 所示。

注意到动画中的每一帧在帧矩阵中占据一个不同的列。帧矩阵的大小随着动画中的帧数和图形窗口的大小 而增加,而与所绘图形的复杂性无关,这是因为函数 getframe 仅仅是捕捉位图。按缺省,函数 movie 只放一遍 动画。通过加入其它输入参量,它可以向前放、向后倒放、放指定次数或按特定的桢速率播放。关于这些特征 的详细信息,参阅 MATLAB 参考指南或使用在线帮助。



图 6-92 动画最终结果

由于上述的动画制作策略很有用,它已体现在精通 MATLAB 工具箱的函数 mmspin3d 中。

function M=mmspin3d(n)

- % MMSPIN3D Make Movie by 3D Azimuth Rotation of Current Figure.
- % MMSPIN3D(N) captures and plays N frames of the current figure
- % through one rotation about the Z-axis at the current elevation.
- % M=MMSPIN3D(N) returns the movie in M for later playing with movie.
- % If not given,N=18 is used.
- % MMSPIN3D fixes the axis limits and issues axis off.
- % Copyright (c) 1996 by Prentice-Hall,Inc.

```
if nargin<1,n=18;end
n=max(abs(round(n)),2);</pre>
```

```
axis(axis);
axis off
incaz=round(360/n);
[az,el]=view;
```

```
m=movie(n);
for i=1:n
view(az+incaz*(i-1),el)
m(:,i)=getframe;
```

end if nargout, M=m; else, movie(m); end 使用 mmspin3d,可以将上述脚本简化如下:

% movie-making example:rotate a 3-D surface plot

[X,Y,Z]=peaks(30);	%	create data
surfl(X,Y,Z)	%	plot surface with lighting
shading interp	%	make it pretty with interpolated shading
colormap(hot)	%	choose a good colormap for lighting
mmspin3d(15)		

# 6.8 小结

在本章中,从基本的二维图形到动画制作,从简单的高级命令的使用到句柄图形的操作,详细地介绍了 MATLAB 图形功能的各个方面。图形功能是 MATLAB 的一大特色。由于 MATLAB 的函数众多,在本章中只能 介绍—部分函数,其余函数请参考联机手册。

# 第七章 符号运算功能

MATLAB 的符号计算是除数值计算和图形功能外,另一个重要的功能。在本章中,介绍了 MATLAB 符号 计算的各个主要方面,其中包括

- ▶ 符号计算的基础
- ▶ 符号矩阵运算
- ▶ 符号微积分
- ➢ 符号代数方程求解
- > 符号函数的二维图
- ▶ MATLAB 中的特殊函数及 Maple 的访问

MATLAB 自产生之日起就在数值计算功能上独占鳌头,广受各专业计算人员的欢迎。但是由于在数学、物理、力学等各种科研、工程应用中还经常遇到符号运算的问题,因此一部分 MATLAB 用户还不得不同时掌握一种符号计算语言,如 Maple、Mathematic、Mathcad 等,这就带来了一些不便。为了解决这个问题,MathWorks 公司于 1993 年从加拿大滑铁卢大学购买了 Maple 的使用权,并以此为基础利用 Maple 的函数库开发了 MATLAB 语言的重要工具箱——符号计算工具箱(Symbolic Toolbox)。从此 MATLAB 便集数值计算、符号计算和图形可视 化三大基本功能于一体,成为在科学计算各种语言中功能最强、操作最简单、最受用户喜爱的语言。

基本符号计算工具箱中包含 100 多个 MATLM 函数。通过这 100 多个函数,可以采用 MATLAB 的语法和风格来使用 Maple 的内核。通过这些函数还可使用 Maple 的线性代数包 (linear algebra package)。通过扩展符号计算工具箱可以便用 Maple 的所有非图形工具包。通过这两个工具箱,可以编写自己的 M 文件来调用 Maple 的函数和访问 Maple 的工作空间。

MAILAB 的这两个符号计算工具箱提供的主要数学运算见表 7-1 所示。

表	7-	1
---	----	---

符号计算工具提供的数学计算分类

数学分类	主要内容
微积分	微分、积分、台劳级数、求极限、级数求和
线性代数	求逆、特征值、奇异值分解、行列式
化简	代数方程的解的化简
方程求解	代数方程和微分方程的数值解和符号解
指定精度求解	数学表达式的指定精度求值
积分变换	傅里叶变换、拉氏变换、z 变换
特殊数学函数	数学领域的比较特殊的经典函数

在 MATLAB 中实现符号计算功能主要有以下 2 种途径。

- 通过调用 MATLAB 的各种功能函数进行常用符号运算。这些功能主要包括字符串操作、符号表达式与符号矩阵的基本操作、符号矩阵的运算、符号微积分运算、符号线性方程求解、符号微分方程求解、特殊数学符号函数、符号函数图形等。
- MATLAB 语言中的符号计算功能已经很强大了,但为了给一些特殊专业的人员提供方便, MATLAB 中还保留着 maple.m、mpa.m 与 Maple 接口,实现更多功能。应用这 2 个函数时,需对 Maple 的一些操作语法加以了解。

## 7.1 符号计算入门

在 MATLAB 的演示程序中,提供了符号计算工具箱的介绍。通过这些演示程序可对 MATLAB 的符号计算

工具箱有一个快速的了解。在命令窗口中键入 demos 命令,将弹出如图 6-1 所示的演示程序对话框,选中图 6-1 中左边的"Toolbox"下的"Symbolic Math"项,再选中右边的某个子项,便可以浏览符号计算的各个方面的简介和示例。



图 6-1 演示程序对话框

在符号计算工具箱中,创建了一个新的数据类型,称 sym 类,即符号类。符号类的实例即符号对象。在符 号计算工具箱中,用符号对象来表示符号变量、符号表达式和符号矩阵。在本节中,将介绍如何创建和使用符 号对象。

7.1.1 符号变量和符号表达式的生成

符号表达式包括符号函数和符号方程,两者的区别是:前者不包括等号,而后者必须 带有等号。但这 两者的创建方式是相同的。而他们的最简单易用的生成方法和 MATLAB 中的字符串变量的生成方法相同。

▶ 创建符号函数

 $f{=}log(x)$  ,

f =

```
log(x)
```

▶ 创建符号方程

```
diffeq='Dy-y=x';
```

由这种方法创建的符号表达式对空格是很敏感的。因此不要在字符间乱加空格符,否则在调用此 表达式时会出错

在 MATLAB 中,另外一种生成符号变量和符号表达式函数的方法使用函数 sym 和 syms。例如下面的两句 命令将分别创建变量 x 和变量 y

x = sym('x')

y=sym('y')

上面四句命令可以用下面的命令代替

syms x y

一段来说,由于输入字符较少,5yms 命令更好用。

7.1.2 符号和数值间的转换

有时符号运算的目的是得到精确的数值解,这样就需要对得到的解析解进行数值转 换。在 MATLAB 中这种转换主要由 2 个函数实现,即 digits 和 vpa。而这 2 个函数在实际使用中经常同变量替换函数 subs 配合使用。 另外在 MATLAB 旧版本中的数值转换函数 numeric 仍然有效。

➤ digits

R=vpa(S) 函数返回符号表达式在 digits 函数设置下的精度的数值解。 vpa(S, D)函数返回符号表达式的 digits(D)精度下的数值解。 > subs subs(S,OLD,NEW) 即将符号表达式中的 OLD 变量替换为 NEW 变量。 ➤ numeric n=numeric(S) 将不含自由变量的符号表达式转换为数值形式,其效果同于 n= double(sym(S)) 函数 sym 有四个选项用来控制符号变量的数值表示格式。对于变量 rho=(1+sqrt(5)) /2, 下面分别给出不同选 项的表示格式。 ▶ 浮点格式 sym(rho, 'f') ans ='1.9e3779b97f4a8'*2^(0) ▶ 有理格式 sym(rho, 'r') ans = 7286977268806824*2^(-52) 选项 e 将给出变量的有理格式表示再加上有理格式的理论值和机器的浮点值之间的误差。 sym(rho, 'e') ans =7286977268806824*2^(-52) 选项 d 将给出变量的十进制展开 sym(rho, 'd') ans =

函数设置返回有效数字个数为 D 的近似解精度。

1.6180339887498949025257388711907

有理格式是 MATLAB 的默认格式。变量的十进制展开格式的默认位数为 32 位,可以使用命令 digits 指定 有效位数。

函数 sym 的另一个用途是把数值矩阵转换为任意精度的符号形式,3×3 的希尔伯特数值矩阵,再用函数转换为任意精度的符号矩阵。例如

A=hilb(3)

```
A = 1.0000 \quad 0.5000 \quad 0.3333 \\ 0.5000 \quad 0.3333 \quad 0.2500 \\ 0.3333 \quad 0.2500 \quad 0.2000 \\ A = sym(A) \\ A = 1 \\ [ 1, 1/2, 1/3] \\ [ 1/2, 1/3, 1/4] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [ 1/3, 1/4, 1/5] \\ [
```

## 7.1.3 突变量、复变量和抽象函数

在函数 sym 中,通过"rea1"选项可把变量设置为实数,下面的命令将把变量 x 和 y 都设置为实数变量 x=sym('x', 'real'); y=sym('y','real'); 和上面两句命令等价的简洁格式为

digits(D)

➤ vpa

syms x y real 把变量 x 和 y 部设置为实数后, 变量 f = x^2+y^2 的值非负。表达式 z=x+i*y 是复数的标准表示形式, 下面 为复数 z 的运算实例 z=x+i*y; conj(z)ans = x-i*y expand(z*conj(z))ans =  $x^{2+v^{2}}$ 在 MATLAB 中常用下面的表达式来查询变量 x 是否为实数, coni(x) = x当表达式是1时,x为实数。为了去掉变量x的实数属性,可选用下面两句命令中的任一句。 syms x unreal x=sym('x', 'unreal') 函数 sym 还可以用来表示确定的函数,例如建立函数 f(x),并对 f(x)求一阶导数: f = sy "(f(x)') %生成函数。 syms x unKd X : Xym(, x', 'un "eal')函数 sym 还可以用来表示确定的函数,例如建立函数 f(J), f=sym(f(x));df=(subs(f, 'x', 'x+h')-f)/'h'上面的求导命令也可用下面的命令代替 syms x h df=(subs(f, x,x+h)-f)/h 求导的结果为 df = (f(x+h)-f(x))/h: 函数 sym 的这种用法对于傅里叶变换、拉氏变换和 z 变换很有用处。

#### 7.1.4 默认符号变量

在日常的数学表达式中,一般都可通过上下文清楚地辨别出其中的自变量。表 7-2 中列出了两个表达式的 数学表示和在 MATLAB 中的表示。

表	7	-2

数学表示和命令表示

数学表示	命令表示
$f = x^n$	$f = x^{\wedge} n$
$g = \sin(at + b)$	$g = \sin(a * t + b)$

在不指定自变量的情况下,按照数学常规,表中的表达使得子变量分别是 x, t,

表达式的导数分别为

 $f'=nx^{n-1}$ 

 $g' = a\cos(at+b)$ 

表达式中的其他符号 n, a, b, 通常被认为是常数或者参数。如果要把表达式 f = x' 中的 n 看作自变量,

求导时,采用下面的表达式区分

 $dx^n$ 

dn

上式将得到结果 xⁿ1nx。

按照数学的习惯,自变量一般为拉丁宁母表中较靠后的字母,如 x,y,z等。在 MATLAB中,函数且 findsym 采用这种数学习惯来确定表达式中的自变量。在微积分、化简、解方程中,确定自变量是必不可少的。

7.1.5 替代

在符号计算工具箱中有两个函数由表达式中的符号替代:subexpr 和 subs。函数 subexpr 把表达式中重复出现的字符串用变量代替。例如,下面的命令将求解三次方程 x³+a*x¹=0 的根。

syms a x

 $s=solve(x^3+a*x+1)$ 

s =

 $[1/6*(-108+12*(12*a^{3}+81)^{(1/2)})^{(1/3)}-2*a/(-108+12*(12*a^{3}+81)^{(1/2)})^{(1/3)}]$ 

 $\begin{bmatrix} -1/12^{*}(-108+12^{*}(12^{*}a^{*}3+81)^{*}(1/2))^{*}(1/3) + a/(-108+12^{*}(12^{*}a^{*}3+81)^{*}(1/2))^{*}(1/3) + 1/2^{*}i^{*}3^{*}(1/2)^{*}(1/6^{*}(-108+12^{*}(12^{*}a^{*}3+81)^{*}(1/2))^{*}(1/3)) + 1/2^{*}i^{*}3^{*}(1/2)^{*}(1/6^{*}(-108+12^{*}(12^{*}a^{*}3+81)^{*}(1/2))) + 1/2^{*}i^{*}3^{*}(1/2)^{*}(1/6^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}(-108+12^{*}$ 

 $[-1/12^{(-108+12^{(12*a^3+81)^{(1/2)})^{(1/3)+a/(-108+12^{(12*a^3+81)^{(1/2)})^{(1/3)-1/2*i^{3}^{(1/2)^{(1/6^{(-108+12^{(12*a^3+81)^{(1/2)})^{(1/3)+2^{(1/2)}}}}]$ 

函数 subexpr 把重复使用的字表达式用的分量 sigma 表示,变量 sigma 存在 MATLAB 工作空间中,例如 r=subexpr(s)

sigma = -108+12*(12*a^3+81)^(1/2)

r =

[ 1/6*sigma^(1/3)-2*a/sigma^(1/3)]

 $[-1/12*sigma^{(1/3)}+a/sigma^{(1/3)}+1/2*i*3^{(1/2)}*(1/6*sigma^{(1/3)}+2*a/sigma^{(1/3)})]$ 

 $[-1/12*sigma^{(1/3)}+a/sigma^{(1/3)}-1/2*i*3^{(1/2)}*(1/6*sigma^{(1/3)}+2*a/sigma^{(1/3)})]$ 

#### 函数 subs 的调用格式有两种

subs(S, NEW) 用新变量 NEW 替代 S 中的默认变量 用新变量 NEW 替代中 S 的变量 OLD subs(S, NEW, OLD) 例如 t=sym('[exp(a*x),sqrt(x);a*sin(b*x),a+b]');A=subs(t, 'pi'),... B=subs(A'2', a')A =[ exp(a*(pi)), sqrt((pi))] [ a*sin(b*(pi)), a+b] B =[ exp((2)*(pi)),sqrt((pi))]  $[(2)*\sin(b*(pi)),$ (2)+b]

#### 7.1.6 指定精度计算

在符号计算工具箱中存在三种算术运算

➢ 数值运算(MATLAB 的浮点运算);

▶ 符号运算;

▶ 任意精度运算。

下面为三种计算的示例。

format long

1/2+2/3

ans =

1.16666666666667

sym(1/2+2/3) ans =

7/6

digits(25)

vpa(1/2+2/3)

ans =

上面的命令分别采用了数值计算、符号计算、任意精度计算。在这三种运算中,浮点运算最快,且占用内存最少,但计算精度有限。在计算机中,双精度的浮点数用 8 个字节存储。在上面的浮点运算中,有三次舍入 误差

▶ 2 被 3 除将产生舍入误差;

把 1/2 加到 2/3 上也会产生舍入误差;

> 把二进制数转换为十进制数显示出来时的舍入误差。

符号计算占用时间和内容都较其他两种多,但它的计算结果是最精确的。

任意精度计算无论在计算精度和所花时间上都介于其他两种运算之间。函数 digits 用于指定计算结果的有效 位数,随着有效位数的增加,计算时间和内存需要也将增加。digits 的默认值为 32 位,大致和浮点计算相当。

值得注意的是,在 Maple 和 MATLAB 中的运算类型的说法稍有不同。在 Maple 中,用"硬件浮点(hardware floating-point)"来代替 MATLAB 中的浮点计算或者数值计算,用"浮点运算(floating-point arithmetric)来代替 MATLAB 中的任意精度计算。

默认情况下,在符号计算工具箱中采用符号计算。当用函数 sym 生成符号变量后,MATLAB 将对这些变量进行符号计算。

函数 sym 可把双精度矩阵转换为符号矩阵,例如

A =

1.100000000000000	1.200000000000000	1.3000000000000000
2.10000000000000	2.200000000000000	2.300000000000000
3.100000000000000	3.200000000000000	3.300000000000000

```
s=sym(A)
```

```
s =
```

[11/10, 6/5, 13/10]

[21/10, 11/5, 23/10]

[31/10, 16/5, 33/10]

在上面的矩阵 s 中 , 数值用小的整数来表示 , 当数值不能用小的整数的比值来表示时 , sym 将生成新的符点 表示 , 例如

```
E=[exp(1),sqrt(2);log(3),rand]
E =
2.71828182845905 1.41421356237310
1.09861228866811 0.95012928514718
```

sym(E)

ans = [ 6121026514868074*2^(-51), sqrt(2)] [ 4947709893870346*2^(-52), 8558003789085230*2^(-53)]

任意精度数的表示方法和符号表示的区别在于采用十进制标注小数点,函数 vpa 用于生成任意精度的表示。 以上面的矩阵 s 和 E 为例,

digits(4) vpa(s) ans = [ 1.100, 1.200, 1.300] [ 2.100, 2.200, 2.300] [ 3.100, 3.200, 3.300]

digits(25)

f=vpa(E)

f =

[2.718281828459045534884808, 1.414213562373095048801689]

[1.098612288668109560063613, .9501292851471754286052374]

函数 double 可把符号矩阵和任意精度表示的矩阵转换为双精度矩阵。对于上面生成的矩阵 E,命令 double(sym(E))和 double(vpa(E)))都会生成矩阵 E 本身。

## 7.2 符号矩阵运算

7.2.1 符号矩阵的创立

在 MATLAB 中创建符号矩阵的方法和创建数值矩阵的形式很相似,只不过要用到符号定义函数 sym。下面介绍使用此函数创建符号函数的几种形式。

1. 使用 sym 函数直接创建符号矩阵

此方法和直接创建数值矩阵的方法几乎完全相同。矩阵元素可以是任何不带等号的符号表达式,各符号表 达式的长度可以不同;矩阵元素之间可用空格或逗号分隔。

例如

 $a=sym('[1/s+x, sin(x), cos(x)^2/(b+x);9, exp(x^2+y^2), log(tanh(y))]')$ a =  $[1/s+x, sin(x), cos(x)^2/(b+x)]$  $[9, exp(x^2+y^2), log(tanh(y))]$ 

#### 2. 用创建子阵的方法创建符号矩阵

此方法是仿照 MATLAB 的字符串矩阵的直接输入法而设计的。这种方法不需要调用 sym 命令,但要保证 同一列的各元素字符串具有相同的长度。为此,在较短字符串的前后可用空格符补充。

例如

```
ms=['[1/s,sin(x)]';'[1 ,exp(x)]']
ms =
[1/s,sin(x)]
[1 ,exp(x)]
```

```
b=[a;'[exp(-i),3,x^3+y^9]']
   b =
           1/s+x,
                      sin(x), cos(x)^2/(b+x)]
   [
   [
              9, \exp(x^2+y^2), \log(\tanh(y))]
   ſ
          exp(-i),
                          3.
                                 x^3+y^9]
   3. 将数值矩阵转化为符号矩阵
   在 MATLAB 中,数值矩阵不能直接参与符号运算,必须先转化为符号矩阵。
   🗻 不论数值矩阵的元素原先是用分数还是用浮点数表示的,转化后的符号矩阵将都以最接近的精确
      有理形式给出
   例如:
   a=[2/3, sqrt(2),0.222;1.4,1/0.23,log(3)],...
   b=sym(a)
   a =
      0.6667 1.4142
                     0.2220
      1.4000 4.3478 1.0986
   b =
                     2/3,
   [
                                                         111/500]
                                      sqrt(2),
                     7/5.
                                      100/23, 4947709893870346*2^(-52)]
   ſ
   4. 符号矩阵的索引和修改
   MATLAB 的矩阵索引和修改同数值矩阵的索引和修改完全相同,即用矩阵的坐标括号表达式实现。
   例如,对上例中的矩阵 b,可以使用下面的命令对其进行索引和修改
   b(2,3)
   ans =
   4947709893870346*2^(-52)
   b(2,3)='log(9)'
   b =
   ſ
       2/3, sqrt(2), 111/500]
       7/5, 100/23, log(9)]
   [
7.2.2 符号矩阵的基本运算
   1. 符号矩阵的四则运算
   MATLAB 抛弃了在早期版本中为符号矩阵设计的复杂的函数形式,把符号矩阵的四则运算简化为同数值矩
阵完全相同的运算方式,这大大方便了用户。
   (1)矩阵的加(+)、减(-)法
   例如
   a=sym([1/x 1/(x+1); 1/(x+2) 1/(x+3)]);
   b=sym('[x,1;x+2,0]');
   b-a
   ans =
   ſ
        x-1/x, 1-1/(x+1)]
   [x+2-1/(x+2),
               -1/(x+3)]
   还可以使用函数 symadd、symsub 来计算符号矩阵的和、差,例如对上面例中的符号矩阵 a、b
   g=symadd(a,b),...
```

f=symsub(a,b)

```
g =
       1/x+x, 1/(x+1)+1]
ſ
[1/(x+2)+x+2,
               1/(x+3)]
f =
[
        1/x-x, 1/(x+1)-1]
[1/(x+2)-x-2,
               1/(x+3)]
(2)矩阵的乘(*)、除(/、\)法
例如
a=sym('[1/x 1/(x+1); 1/(x+2) 1/(x+3)]');
b=sym([x,1;x+2,0]);
a∖b
ans =
      -6*x-2*x^3-7*x^2,
                         3/2*x^2+x+1/2*x^3
ſ
\begin{bmatrix} 6+2*x^3+10*x^2+14*x, -1/2*x^3-2*x^2-3/2*x \end{bmatrix}
2. 符号矩阵的其他一些基本运算
```

符号矩阵的其他一些基本运算也与数值矩阵的运算格式相同。这些运算包括矩阵的转置(')、行列式(det)、逆 (inv)、秩(rank)、幂(^)和指数(exp和 expm)等运算。例如

H=hilb(3); H=sym(H); a=inv(H),... b=det(H) a = [ 9, -36, 30] [ -36, 192, -180] [ 30, -180, 180] b = 1/2160

7.2.3 符号矩阵分解

绝大多数符号矩阵分解函数的使用方法是与数值矩阵分解函数相同的。例如符号矩阵的特征值分解函数 eig、奇异值分解函数 Svd、三角抽取函数 diag、tril 和 trin,另外还有一个求解符号矩阵的约当标准型函数 jordan。

1.特征值分解

符号矩阵的特征值求解函数和数值矩阵求解函数的名称相同,其符号计算和指定精度计算的调用格式如下 E=eig(A)

[V, E]=eig(A)

E=eig(vpa(A))

[V, E]=eig(vpa(A))

矩阵的特征值为行列式多项式的根,生成行列式多项式的函数名仍为 poly。对于奇异矩阵,由于奇异性, 肯定有一个特征值为 0。例如有奇异矩阵

h = [ 8/9, 1/2, 1/3] [ 1/2, 1/3, 1/4]

[ 1/3, 1/4, 1/5]

[t,e]=eig(h)

t =

[1,	28/153+2/153*12589^(1/2),	28/153-2/153*12589^(1/2	!)]
[-4,	1,		1]
[10/3,	292/255-1/255*12589^(1/2),	292/255+1/255*12589^(1/2	2)]
e =			
[0,	0,	0]	
[0,	32/45+1/180*12589	9^(1/2), 0]	
[0,	0, 32	2/45-1/180*12589^(1/2)]	
- 11-11	1-14-11		

2.约当标准型

对矩阵进行相似变换时,会产生约当标准形(Jordan Canonical Form)。对于矩阵A,求其若当标准形,也就 是找一个非奇异矩阵V,使 J = V\A*V 最接近对角矩阵。矩阵V称为转换矩阵。

当矩阵对称、所用特征值互异时,其约当标准形为特征值织成的对角矩阵,转换矩阵以特征向量为列形成。 对于非对称或者有重特征值的矩阵,约当标准形的对角元素为特征值,但对角线上面有非 0 元素。函数 jordan 用于计算约当标准形。其调用格式为

➤ J=jordan(A)

▶ [V, J]=jordan(A) 同时给出相似变换矩阵

约当标椎形的计算对于原矩阵中的元素值变换特别敏感,这使得采用数值计算特别困难,在有些情况下, 用指定精度计算也会产生很大的误差,下面举例说明。

A=sym([12,32,66,116;-25,-76,-164,-294;21,66,143,256;-6,-19,-41,-73]) A =[ 12, 32, 66, 116] [-25, -76, -164, -294] [ 21, 66, 143, 256] [ -6, -19, -41, -73] [v, j]=jordan(A)  $\mathbf{v} =$ [4, -2, 4, 3][-6, 8, -11, -8] [ 4, -7, 10, 7] [-1, 2, -3, -2] i = [1, 1, 0, 0][0, 1, 0, 0][0, 0, 2, 1][0, 0, 0, 2]从约当标准形 i 可知,矩阵的特征值1和2都是二重根。 3. 奇异值分解 由于采用符号计算的奇异值结果很繁琐,在符号计算工具箱中,只能对符号矩阵进行指定精度的奇异值分 解。对于符号矩阵 A , 下面的命令将给出 A 的奇异值 , 奇异值的精度由函数 digits 指定。 S=svd(A) 下面的调用格式还将给出两个正交矩阵,满足关系式 A=U*S*V [U, S, V] = svd(A)下面生成 5×5 的矩阵,矩阵的每个元素的值为 1/(i-j+1/2),其中 i 和 j 分别为元素的行号和列号。例如 n=5; for i=1:n for j=1:n a(i,j)=sym(1/(i-j+1/2));

end a a =  $\begin{bmatrix} 2, -2, -2/3, -2/5, -2/7 \end{bmatrix}$  $\begin{bmatrix} 2/3, 2, -2, -2/3, -2/5 \end{bmatrix}$  $\begin{bmatrix} 2/5, 2/3, 2, -2, -2/3 \end{bmatrix}$  $\begin{bmatrix} 2/7, 2/5, 2/3, 2, -2 \end{bmatrix}$  $\begin{bmatrix} 2/9, 2/7, 2/5, 2/3, 2 \end{bmatrix}$ 

#### 看下面的例子。

```
a \equiv sym(, [1 \ 1 \ 2 \ ; \ 0 \ 1 \ 3 \ ; \ 00 \ 2] /)
                          . .
а 二
[1, 1, 2]
[0, 1, 3]
[0, 0, 2]
              %其中 x、y 的含义约当标准型, x 为相应的线性转换阵。
[)c , y]\equiv jordan(a)
ΧΞ
[5, -5]
                -5]
ſ
 3,
              -51
        0,
[ 1, 0,
               0]
  ,
у
[ 2, 0, 0]
[ 0, 1, 1]
[0, 0, 1]
```

7.2.4 符号矩阵的简化

符号工具箱中还提供了符号矩阵因式分解、展开、合并、简化及通分等符号操作函数。 下面将一一进行 介绍。

1.因式分解

函数把多项式分解因式,并且分解后的多项式的所有系数都为有理数。其调用格式为

➢ factor(S) 输入变量 S 为一符号矩阵,此函数将因式分解此矩阵的各个元素。如果 S 包含的所有元素为整数,则最佳因式分解式将被计算。为了分解大于 2^52 的整数 N,可使用 factor(sym('N'))。

```
例如
```

```
syms x
factor(x^9-1)
ans =
(x-1)*(x^2+x+1)*(x^6+x^3+1)
函数也能对含有符号整数的多项式进行分解,例如
one='1'
for n=1:11
N(n,:)=sym(one(1,ones(1,n)));
end
[N factor(N)]
ans =
```

end

[	1,	1]
[	11,	(11)]
[	111,	(3)*(37)]
[	1111,	(11)*(101)]
[	11111,	(41)*(271)]
[	111111, (	3)*(7)*(11)*(13)*(37)]
[	1111111,	(239)*(4649)]
[	11111111,	(11)*(73)*(101)*(137)]
[	111111111,	(3)^2*(37)*(333667)]
[	111111111, (	11)*(41)*(271)*(9091)]
[	11111111111,	(513239)*(21649)]

#### 2. 符号矩阵的展开

```
函数 expand 可以把表达式展开,其调用格式为
```

```
expand(S) 对符号矩阵的各元素的符号表达式进行展开。此函数经常用在多项式的表示中,也常用于三角函数、指数函数和对数函数的展开中。
```

### 例如

```
syms x y
expand((x+1)^3) , ...
expand(sin(x+y))
ans =
x^3+3*x^2+3*x+1
```

```
ans =
```

```
sin(x)*cos(y)+cos(x)*sin(y)
```

3. 同类式合并

```
函数将表达式中相同次幂的项合并,其调用格式为
```

```
▶ collect(S, v) 将符号矩阵 S中的各元素的 v 的同幂项系数合并。
```

```
➢ collect(S) 对由 findsym 函数返回的默认变量进行同类项合并。
```

例如

```
x=syms x
f=(x-1)*(x-2)*(x-3);
collect(f)
```

ans =

 $x^3-6*x^2+11*x-6$ 

```
4.符号简化
```

在 MATLAB 中进行符号简化可由 simple 和 simplify 2 个函数实现。

函数 simplify 利用函数规则对表达式进行化简。该函数运行时,采用了很多代数方法如求和、求平方根等,还采用了很多函数如指数函数、对数函数、Bessel 等。其调用格式为

➢ simplify(S) 简化符号矩阵的每一个元素。

### 例如

```
syms c a b
```

```
simplify(exp(c*log(sqrt(a+b))))
```

```
ans =
```

```
(a+b)^{(1/2*c)}
```

函数 simple 的目标是使表达式用最少的字符表示。虽然并非表达式中的字符越少,表达式就越简洁,但采用这个标准往往能够得到满意的结果。为了达到最少字的标难,simple 函数综合使用了以下函数:

```
用 simpley 对表达式进行化简。
   用函数 radsimp 对包含根式的表达式化简。
   用函数 combine 把表达式中以求和形式、乘积形式
   用函数 collect 合并同幂项系数。
   用函数 factor 实现因式分解。
   用函数 convert 把一种格式转换为一种格式。
   simple 函数比较这些函数的结果,始终把所用字符最少作为标准。函数 simple 有几种不同的调用格式,不
同的调用格式输出的内容不同。
      ▶ simple 寻找符号矩阵或符号表达式的最简型。
      ▶ simple(S) 对表达式 S 尝试多种不同的算法简化,以显示 S 表达式的长度最短的简化形式。若 S
          为一矩阵,则结果是全矩阵的最短形,而可能非每个元素的最短形。
      ▶ [R, HOW]=simple(S) 返回的 R 为简化型, HOW 为简化过程中使用的主要方法。
   例如,在命令窗口输入
   simple(cos(x)^{2}+sin(x)^{2})
   回车显示结果
   simplify:
   1
   radsimp:
   \cos(x)^2 + \sin(x)^2
   combine(trig):
      1
   factor:
   \cos(x)^{2}+\sin(x)^{2}
   expand:
   \cos(x)^2 + \sin(x)^2
   combine:
   1
   convert(exp):
   (1/2*\exp(i*x)+1/2/\exp(i*x))^2-1/4*(\exp(i*x)-1/\exp(i*x))^2
   convert(sincos):
   \cos(x)^2 + \sin(x)^2
   convert(tan):
   (1-\tan(1/2*x)^2)^2/(1+\tan(1/2*x)^2)^2+4*\tan(1/2*x)^2/(1+\tan(1/2*x)^2)^2
   collect(x):
   \cos(x)^{2}+\sin(x)^{2}
```

ans =

```
1
5. 分式通分
```

函数 numden 用来求解符号表达式的分子和分母,其调用格式为

▶ [N, D]=numden(A) 把 A 的各元素转换为分子和分母都是整系数的最佳分式。

6. 符号表达式的"秦九昭型"重写

函数 homer 用来把表达式重写成"秦九昭型"多项式,其调用格式为

▶ homer(P) 将符号多项式转换成嵌套形式表达式。

例如

```
horner(x^3-6*x^2+11*x-6)
ans =
-6+(11+(-6+x)*x)*x
```

#### 7. 嵌套格式

函数 horner 把表达式转换为嵌套格式,其调用格式为

```
\blacktriangleright horner(f)
```

例如

```
syms x
f=1.1+2.2*x+3.3*x^2;
horner(f)
ans =
11/10+(11/5+33/10*x)*x
```

## 7.3 符号微积分

MATLAB 的符号计算工具箱提供了微积分个的常用函数,大致可分为微分、求极限、积分和级数求和四个 方面。下面将分别介绍这些函数。

7.3.1 符号极限

极限是微积分学的基础和出发点,因此在介绍微积分之前先介绍极限的求解方法是必要的。"无穷逼近"是 微积分的——个基本思想,微分就是用极限来定义的。在 MATLAB 中,极限的求解可由 limit 函数来实现。其 调用形式为

- ➢ limit(F, x, a) 计算符号表达式 F 在 x→>a 条件下的极限值;。
- ▶ limit(F, a) 计算符号表达式中由 findsym(F)返回的独立变量趋向于 0 的极限值;
- ➢ limit(F) 计算 a=0 时的极限;
- ▶ limit(F, x, a, 'right') 'righ 或者 t"left'用来指定取极限的方向, 求极限可以分为从两边趋近、从左边 趋近和从右边趋近。各种情况下求得的值可能是不相同的

例如

```
syms x a t h
limit(sin(x)/x)
 ans =
1
limit((1+2*t/x)^{(3*x)},x,inf)
 ans =
```

```
exp(6*t)
```

limit(1/x	x,x,0,'right')			
ans =				
inf				
limit(1/x	x,x,0,'left')			
ans =				
-inf				
表 7-3 列出了函数 limit 的常用调用格式。				
表 7-3	函数 limit 的常用调用格式			
数	牧学表达式	命令格式		
$l_x$	$\lim_{x\to 0} f(x)$	limit(f)		
	$\lim_{x \to 0} f(x)$	limit(f, x, a)或者limit(f, a)		
	$\lim_{x\to 0} f(x)$	limit(f, x, a, 'left')		
	$\lim_{x\to 0} f(x)$	limit(f, x, a, 'right')		

### 7.3.2 符号积分

在数学上,积分和微分是互逆的两个过程。在 MATLAB 中,用函数 int 来求符号表达式的积分。其调用格 式为

- ➢ int(S) 符号表达式对 findsym 返回的符号自变量的 S 的不定积分。S 为符号矩阵或符号数量。如果 S 为常数,则积分针对'x'。
- ▶ int(S, v) 符号表达式 S 对符号自变量 v 的不定积分, v 是一数量符号量。
- ▶ int(S, a, b) 符号表达式 S 对默认符号变量在 a 到 b 的定积分, a 和 b 为双精度或符号数量。
- ➢ int(S, v, a, b) 符号表达式对变量 v 从 a 到 b 的定积分。

## 例如

```
syms x x1 u t

A=[\cos(x^*t),\sin(x^*t);-\sin(x^*t),\cos(x^*t)];

int(A,t), ...

int(x1*log(1+x1),0,1)

ans =

[ 1/x*sin(x*t), -cos(x*t)/x]

[ \cos(x^*t)/x, 1/x^*sin(x^*t)]

ans =

1/4
```

当函数的积分不存在时,MATLAB 将原封不动的返回积分表达式。在 MATLAB 中,常数是符号积分中比

较敏锐的一个问题。例如对于表达式  $e^{-(kx)^2}$ , 当 k 为实数时, 该表达式的值水远为正, 并且当 x 趋于正负无穷时, 表达式的值趋于 0。单对于 Maple 的内核, 并不把  $k^2$ 或者  $x^2$ 当作正数。相反, Maple 假定它们为符号, 其值不定, 即它们是纯粹的符号, 没有丝毫的数学属性。用下面的程序运行该积分, 将给出错误信息。

#### 7.3.3 级数求和

- 函数 symsum 用于对符号表达式求和,其调用格式为
  - ➢ symsum(S) 符号表达式对由 findsym 函数返回的符号变量的不定和。
  - ➢ symsum(S, v) 符号表达式 S 对变量 v 的不定和。
  - ▶ symsum(S, a, b)和 symsum(S, v, a, b) 从 a 到 b 的有限和。

```
syms k
s1=symsum(1/k^2,1, inf)
s1 =
1/6*pi^2
```

#### 7.3.4 符号微分和差分

### 1.微分和差分

- ▶ diff(S,'v')或 diff(S,sym('v')) 变量 v 微分式 S。
- ▶ diff(S,n) 整数 n,函数微分 S 表达式 n 次。
- ➢ diff(S, 'v', n)和 diff(S, n, 'v') 以被识别。

#### 例如

```
syms a x
```

```
f=sin(a*x);
```

df=diff(f)

df =

cos(a*x)*a

```
dfa=diff(f, a)
```

dfa =

```
cos(a*x)*x
```

#### 可以使用 diff 函数对于表达式求二阶导数,例如

d2=diff(f,2)

d2 =

 $-\sin(a^*x)^*a^2$ 

#### 函数的输入参数也可以是符号矩阵,这时,将对矩阵中的每个元素求导,例如

syms a x

```
A=[-\sin(a^*x),\sin(a^*x);\cos(a^*x),\cos(a^*x)]
```

A =

```
[-\sin(a^*x), \sin(a^*x)]
```

```
[\cos(a^*x), \cos(a^*x)]
```

```
dy=diff(A)
```

```
dy =
```

```
[-\cos(a^*x)^*a, \cos(a^*x)^*a]
```

```
[-\sin(a^*x)^*a, -\sin(a^*x)^*a]
```

```
2. 梯度函数
```

- MATLAB 中可以使用 gradient 函数求近似梯度,其调用函数为
  - [FX, FY]=gradient(F) 矩阵 F 的数值梯度, FX 相当于 dF/dx,为 X 方向的差分值。FY 相当于 dF/dy, 为 Y 方向的差分值。各个方向的点间隔设为 1。当 F 为向量时, gradient(F)为一维梯度。
  - ➢ [FX, FY]=gradient(F, H) 为数量时,使用H为各方向点间隔。
  - ▷ [FX, FY]=gradient(F, HX, HY) 为二维时,使用 HX 和 HY 指定点间距。HX 和 HY 可为数量和向量,如果 HX 和 HY 为向量,则它们的长度必须和 F 的长度匹配。
  - ▶ [FX, FY, FZ]=gradient(F) 三维的梯度。
  - ▶ [FX, FY, FZ]=gradient(F, HX, HY, HZ) HX、HY、HZ 指定间距。

例如 [x,y]=meshgrid(-2:0.2:2,-2:.2:2); z=x.*exp(-x.^2-y.^2); [px,py]=gradient(z,.2,.2); contour(z),... hold on quiver(px,py) hold off 得到的图形见图 7-2 所示



图 7-2 梯度函数绘制矢量图

3. 多元函数的导数

在多元函数中,仿照单元函数的"极限"、"可微"的概念引入了 Frechet 导数。多元函数的 Frechet 导数在 非线性方程的求解和变分原理中有极其重要的应用,在 MATLAB中,此问题由 jacobian 函数实现。

▶ jacobian(f, v) 数量或向量 f 对向量 v 的 jacobi 矩阵。注意当 f 为数量时,函数返回 f 的梯度。

```
例如
```

```
x=sym(['x']);
y=sym(['y']);
z=sym(['z']);
jacobian([x^2+y^2;x^2-y^2],[x y])
ans =
[ 2*x, 2*y]
[ 2*x, -2*y]
```

## 7.4 符号代数方程求解

在本节中,将分别介绍代数方程和微分方程的求解。

7.4.1 solve 函数

函数 solve 用于求解代数方程的根。假定 S 为符号表达式,命令 solve(S)求解表达式等于 0 时的根。默认情况下,采用函数 findsym 指定的未知数。例如

272

```
syms a b c x
s=a*x^2+b*x+c;
solve(s)
ans =
[1/2/a^{(-b+(b^2-4^*a^*c)^{(1/2)})]
[1/2/a^{*}(-b-(b^{2}-4^{*}a^{*}c)^{(1/2)})]
在函数中,可以用参数指定未知数,例如
b=solve(s,b)
b =
-(a*x^2+c)/x
函数的输入参数也可以是方程,例如
s=solve(cos(2*x)+sin(x)=1')
s =
ſ
      0]
     pi]
ſ
[ 1/6*pi]
[ 5/6*pi]
下面为求解单变量的代数方
```

[5 / 6 " PU

### 7.4.2 线性方程组的解析解法

在 Symbolic Toolbox 中还提供了线性方程的符号求解函数,如 linsolve。此方法可得到方程组的精确解。所得的解析解可由 vpa 函数转换成浮点近似数值。

例如

```
a=sym('[10,-1,0;-1,10,-2;0,-2,10]');
b=sym('[9;7;6]');
linsolve(a,b)
ans=
[ 473/475]
[ 91/95]
[ 376/475]
vpa(ans)
ans=
[ .99578947368421052631578947368421]
[ .95789473684210526315789473684211]
```

[.79157894736842105263157894736842]

### 7.4.3 非线性方程的解析解法

非线性方程的符号求解由 fsolve 函数实现,其调用格式为

- X=fsolve('FUN', X0) 所要求解的函数名,通常以 M 文件的形式给出,用 以返回函数值
   F=FUN(X)。X0 为求解方程的初始向量或矩阵。
- X=fsolve('FUN', X0, OPTIONS)
   ONS 为选择参数输入向量,如 OPTIONS(2)
   表示求解的精度要求; OPTIONS(3)表示在解处的函数值精度要求。
- ➢ X=fsolve('FUN', X0, OPTIONS, 'GRADFUN') DFUN'为输入函数在 X 处的偏导数。
- X=fsolve('FUN', X0, OPTIONS, 'GRADFUN', P1, P2, ...) 数 P1, P2, ...为问题 定性参数,

274

```
直接赋给函数 FUN 和 GRADFUN,即 FUN(X,n,四,...)和 GRADFUN(X,P1,P2,...)。此时若 OPTIONS 和 'GRADFUN'使用默认值则要输人空矩阵。
```

 [X, OPTIONS]=fsolve('FUN', X0, ...) 使用的优化方法的参数。例如 options(10)表示函数 估值的次数。默认算法为二次三次混合搜索的 Gauss-Newton 方法,若使用 Levenberg-Marquardt 算法,要设置 OPTIONS(5)=1。

例如,要求解非线性方程

x1-0.7sinx1-0.2cosx2=0

x2-0.7cosx1+0.2sinx2=0

首先编制函数文件 fc.m 如下

fc.m

function y=fc(x)

y(1)=x(1)-0.7*sin(x(1))-0.2*cos(x(2));

y(2)=x(2)-0.7*cos(x(1))+0.2*sin(x(2));

y=[y(1),y(2)];

在 MATLAB 命令窗口窗口中输入:

x0=[0.5,0.5];

fsolve('fc',x0)

```
回车显示结果
```

```
ans =
```

0.5265 0.5079

## 7.5 符号微分方程求解

函数 dsolve 用于常微分方程组的符号求解。dsolve 函数的调用格式为 dsolve(lsop)' _____)

dsolve('eqnl', 'eqn2', ....)

以代表微分方程及初始条件的符号方程为输人参数,多个方程或初始条件可在一个输入变量内联立输入, 且以逗号分隔。默认的独立变量为t,也可把t变为其他的符号变量。字符 D 代表对独立变量的—次微分,通常 指 d/dt。紧跟一数字的 D 代表高阶微分,D2,D3 分别表示二阶、三阶微分,符号 D2y 相当于 d²y/dt²。紧跟此 微分操作符的任何符号都可为被微变量,如 D3y 代表对 y(t)的三阶微分。注意,用户所定义的符号变量不能再 包括字符 D。初始条件可以由方程如'y(a)=b'或'Dy(a)=b'的形式给出,这里 y 为被微函数而 a 和 b 为常数。 如果初始条件的数目少于被微变量的数目,则结果中要包含不定常数 C1、C2 等。

此函数有3种可能的输出类型

- > 一个方程和一个输出,则结果返回符号向量中非线性方程的联立解。
- > 对多个方程和多个输出,则结果以字母顺序排序且赋给输出量。
- 对多个方程和单输出,则结果返回解的结构。

```
例如
```

```
dsolve('Dx=-a*x')

ans =

C1*exp(-a*t)

x=dsolve('Dx=-a*x','x(0)=1','s')

x =

exp(-a*s)
```

s=dsolve('Du=v, Dv=w, Dw=-u','u(0)=0,v(0)=0,w(0)=1')

s =

```
u: [1x1 \text{ sym}]

v: [1x1 \text{ sym}]

w: [1x1 \text{ sym}]

s.u

ans =

1/3*3^{(1/2)*\exp(1/2*t)*\sin(1/2*t*3^{(1/2)})-1/3*\exp(1/2*t)*\cos(1/2*t*3^{(1/2)})+1/3*\exp(-t)}

s.v

ans =

1/3*\exp(1/2*t)*\cos(1/2*t*3^{(1/2)})+1/3*3^{(1/2)}*\exp(1/2*t)*\sin(1/2*t*3^{(1/2)})-1/3*\exp(-t)}

s.w

ans =

1/3*\exp(-t)+2/3*\exp(1/2*t)*\cos(1/2*t*3^{(1/2)})
```

## 7.6 符号函数的二维图

## 7.6.1 符号函数的简易绘图函数

ezplot(f)绘制 f(x)的函数图,这里 f 为代表数学表达式的包含单个符号变量 x 的字符串或符号表达式。x 轴的 近似范围为[-2*pi, 2*pi]。

- ➢ ezplot(f, xmin, xmax)或 ezplot(f, [xmin, xmax]) 输入参数来代替默认横坐标范围[-2*pi, 2*pi]。
- ▶ ezplot(f, [xmin xmax], fig) 绘图的图窗号以代替当前图窗。
- 例如,在命令窗口输入
- ezplot('erf(x)')

回车将在图形窗口显示误差函数图,如图7-3所示。



图 7-3 误差函数图

## 7.6.2 绘制函数图函数

fplot(fun,lims)绘制由字符串 fun 指定函数名的函数在 x 轴区间为 lims=[xmin xmax]的函数图。若 lims=[xmin xmax ymin ymax]则 y 轴也被输入限制。fun 必须为 M 文件的函数名或对变量 x 的可执行字符串,此字符串被送入函数 eval 后被执行。函数 fun(x)必须要返回针对向量 x 的每一元素结果的行向量。

- ▶ fplot(fun, lims, to1) 其中 tol<1 用来指定相对误差精度,默认值为 0.001。
- ➢ fplot(fun, lims, n) 其中 n>1,指定以最少 n+1 个点来绘制函数图。默认 n=1。最大步长被约束 在不小于(1/n)*(xmax-xmin)。
- ▶ fplot(fun, lims, 'LineSpec') 以指定线型绘制图形。
- ▶ [x,y]=fplot(fun,lims,...) 只返回应用来绘图的点的向量值,而不绘出图形。用户可自己用 plot(x, y)来输出图形。

## 例如

subplot(2,2,1),fplot('humps',[0 1]) subplot(2,2,2),fplot('abs(exp(-j*x*(0:9))*ones(10,1))',[0 2*pi]) subplot(2,2,3),fplot('[tan(x),sin(x),cos(x)]',2*pi*[-1 1 -1 1]) subplot(2,2,4),fplot('sin(1./x)',[0.01 0.1],1e-3) 得到的图形如图 7-4 所示。



图 7-4 fplot 四种绘图方式

7.7 访问 Maple 函数

为了能够在 MATLAB 环境下,充分地利用 MAPLE 中符号计算能力,Symbolic Math Toolbox 提供了两个 函数 sym 和 maple 实现对 MAPLE 绝大多数符号计算指令的调用。本节 介绍这两个指令的使用。

## 7.7.1 利用 sym 访问 Maple 函数

可以使用 sym 访问 Maple 的阶乘函数 k!,其调用格式为 kfac=sym('k!') kfac= k! 相当于定义了一个符号函数可以用整数的阶乘。 例如 kfac=sym('k!');

```
subs(kfac, k,6),subs(kfac,k,n)
ans =
720
```

ans = n!

mhelp gcd

7.7.2 利用 maple 访问 Maple 函数

采用 maple 函数可以直接访问 MAPLE 中的函数。这个函数采用符号对象、字符串、双精度数作为输入。 返回和输入相对应的符号对象、字符串和双精度数。也可以使用 maple 函数调试用户编写的符号数学程序。

假定我们要写一个 M 一文件,用来求两个多项式或两个整数的最大公约数。例如,14 和 21 的最大公约数 是 7, x^2-y^2 和 x^3-y^3 的最大公约数是 x-y。

首先,需要知道怎样调用 MAPLE 中求最大公约数的函数。可以使用"mhelp"函数取出关于最大公约数(gcd)的 Maple 联机帮助。

将给出关于 gcd 用法的信息。 知道了 Maple 调用 gcd 的语法,就可写一个简单的 M 文件计算最大公约数。 首先,下创建 M 文件 gcd,包含下面的命令 function g=gcd(a,b) g=maple('gcd',a,b) 运行这个文件 syms x y z=gcd(x^2-y^2,x^3-y^3) g = -y+x z = -y+x 扩展这个函数,使其能够按元素求连个矩阵的最大公约数 function g=gcd(a,b) if any(size(a)~=size(b)) error('Input must have the same size.') end for k=1:prod(size(a)) g(k)=maple('gcd',a(k),b(k)); end g=reshape(g,size(a)); 用一些检验数据运行这个函数 a=sym([2 4 6;3 5 6;3 6 4]); b=sym([40 30 8;17 60 20;6 3 20]); gcd(a,b) ans = [2, 2, 2][1, 5, 2] [3, 3, 4]

## 7.8 特殊数学函数

在基本符号计算工具箱中,有 50 点多个特殊函数。下面的命令列出了这些函数的帮助信息。在帮助信息的 结尾,列出了扩展符号计算工具箱中的多项式函数。至于更为详细的帮助信息,可采用命令"mhelp"函数名来 查询。下面列出了 MATLAB 提供的这些特殊函数,一共参考。

help mfunlist

MFUNLIST Special functions for MFUN

%n 表示参数为整数

%x 表示参数为实数

%z 表示参数为复数

bernoulli	n	Bernoulli Numbers
bernoulli	n,z	Bernoulli Polynomials
BesselI	x1,x	Bessel Function of the First Kind
BesselJ	x1,x	Bessel Function of the First Kind
BesselK	x1,x	Bessel Function of the Second Kind
BesselY	x1,x	Bessel Function of the Second Kind
Beta	z1,z2	Beta Function
Binominal	x1,x2	Binominal Coefficients
LegendreKc	Х	Complete Elliptic Integral of First Kind
LegendreEc	Х	Complete Elliptic Integral of Second Kind
LegendrePic	x1,x	Complete Elliptic Integral of Third Kind
LegendreKc1	Х	LegendreKc using Complementary Modulus
LegendrePic1	x1,x	LegendrePic using Complementary Modulus
Erfc	Z	Complementary Error Function
Erfc	n,z	Complementary Error Function's Iterated Integrals
Ci	Z	Cosine Integral
Dawson	Х	Downson's Integral
Psi	Z	Digamma Function
Dilog	Х	Dilogarithm Integral
Erf	Z	Error Function
Euler	n	Euler Numbers
Euler	n,z	Euler Polynomial
Ei	Х	Exponential Integral
Ei	n,z	Exponential Integral
FresnelIC	х	Fresnel Cosine Integral
FresnelIS	Х	Fresnel Sine Integral
GAMMA	Z	GAMMA Function
Harmonic	n	Harmonic Function
Chi	Z	Hyperbolic Cosine Integral
Shi	Z	Hyperbolic Sine Integral
Hypergeom	x1,x2	(Generalized) Hypergeometric Function
LegendreF	x,x1	Incomplete Elliptic Integral of First Kind
LegendreE	x,x1	Incomplete Elliptic Integral of Second Kind
LegendrePi	x,x2,x1	Incomplete Elliptic Integral of Third Kind
GAMMA	z1,z2	Incomplete Gamma Function
W	Z	Lambert's W Function
### MATLAB 6 实用教程

W	n,z	Lambert's W Function
InGAMMA	Z	Logarithmic of the Gamma function
Li	Х	Logarithmic Integral
Psi	n,z	Polygamma Function
Ssi	Z	Shifted Sine Integral
Si	Z	Sine Integral
Zeta	Z	(Riemann) Zeta Function
Zeta	n,z	(Riemann) Zeta Function
Zeta	n,z,x	(Riemann) Zeta Function
Orthogonal Poly	nomials (Extended S	ymbolic Math Toolbox only)
Т	n,x	Chebyshev of the First Kind
U	n,x	Chebyshev of the Second Kind
G	n,x1,x	Gegenbauer
Н	n,x	Hermite
Р	n,x1,x2,x	Jacobi
L	n,x	Laguerre
L	n,x1,x	Generalized Laguerre
Р	n,x	Legendre

7.9 小结

本章介绍了 MATLAB 的符号计算。在本章中,介绍了 MATLAB 符号计算的各个主要方面,其中包括:符 号变量的微积分、表达式的化简、指定精度计算、基本的线性代数计算、符号方程的求解、特殊函数的使用方 法和 Maple 函数的使用方法。本章只介绍了 MATLAB 符号计算功能中最常用的部分,对于有较高数学基础要求 的部分,例如傅里叶变换、拉氏变换和 z 变换,感兴趣的读者可以参考联机帮助。

# 第八章 创建用户界面

对于使用程序的用户而言,界面就是应用程序,他们感觉不到正在幕后运行的代码。无论花多少时间和精 力来编制和优化代码,应用程序的可用性仍然依赖于界面。新的 MATLAB 6,不仅在自身的用户界面有了长足 的长进,其提供的制作界面的能力也有了阶段性的进步。在本章中,将介绍 MATLAB 的界面编辑上具和用户界 面的设计过程,主要内容包括:

- ▶ 界面设计原则
- ➢ MATLAB 的图形设计工具
- ▶ 编程设计用户界面

# 8.1 界面设计原则

虽然人们从事图形界面(GUI)的开发并不长,但对于程序用户界面好坏的评判标准很早就确立了。图形界面 正是由于人们对程序用户界面要求的提高而产生的。不同用户的理想程序界面是不相同的,但总体来说,简单 性(Simplicity)、一致性(Consistency)和熟悉性(Familiarity)是界面的三大标推。

8.1.1 简单性

简单是界面设计的最主要的标难。简单的用户界面给人以简洁明快且统一的感觉。给界面添加一些复杂的 功能并不难,但是,如果这些功能没有什么实用价值,最好删除这些功能,因为这些花哨的东西会把界面搞得 很混乱。在界面上,只提供那些引导用户完成实际工作必需的功能。

对于图形界面,应该强调形状,而不是数值来说明问题。由于图形的人生特性就是直观,在界面上应该尽 量少标数值,以使用户把精力集中在图形上。在图 7—6 中,左图的网格线对于图形并没有多少实际意义,相反, 由于有了这些网格线和数字,降低了用户对图形形状的注意力。

如果图形上的某一部分不是绝对必需的,则应该删除。在编写界面时,常常会发现,即使一个很小的部分 被删除,和这一部分相关的很多东两也都没有存在的必要了,删除这些多余的累赘后,整个界面也变得简洁很 多。

在设计界面时,不应该设计较多的窗口。例如在表示输入输出的关系时,最好把它们都放在—个窗口中。 如果一个窗口用了输入命令,一个窗口用了显示命令所绘制的图形。在操作时,必须在两个窗口中来回切换, 很不方便。而如果能将命令输入和图形绘制集中在一个窗口中,并采用好几个按钮宋代表一些命令输入,使用 户能够集中在图形上,这样的界面使用起来方使得多。

8.1.2 一致性

一般用户在使用程序时,都按照自己使用其他软件的经验来熟悉程序。如果一个程序的界面和其他用户程 序的界面相差很大,或者程序的多个功能类似的窗口中,—些图形窗口的布置和另一些差别很大(比如,在一个 窗口中把图形区域放在左边,而在另一个窗口中把图形区域放在右边;在一个窗口中把"确定"按钮做成长方形, 而在另—个窗口中把"确定"按钮做成六边形)。这样的设计往往会把用户搞得晕头转向。

图 8-1 和图 8-2 都由 MATLAB 中的演示程序产生,但这两个程序的内容完全不同。可以看出,两个图的设 计风格是相向的。如果熟悉了一个演示程序的界面,使用另个程序就根方便了。



图 8-2 一致性示例

在知识"爆炸"的今天,各种软件、软件的各种版本,都以惊人的速度推陈出新。人们在学习新的软件时, 不光按照自己的经验熟悉各种软件的界面,对于软件的具体内容和使用方法,很多人都是根据其他相关软件的 知识来举一反三的。在这种大环境下,程序界面的一致性更显得特别重要。

8.1.3 熟悉性

在日常生活中,经常使用图形来提醒人们的注意。如果在编写图形界面的时候,加入一些用户熟悉的图形, 往往可以更加方便用户的学习和使用。有时候虽然不知道该怎么操作,但根据图形的形状等可以做出有效的猜 测,这就是界面的熟悉性。采用人们熟悉的图标来表示操作,在游戏软件中很常用。在设计其他软件的界面时, 这一点也是应该注意的。

8.1.4 动态交互

前面讲的三条原则都是针对静态图形的,但在使用程序时,会在不同的窗口中切换,进行操作后等待新的 结果等等,这就有一个动态交互的问题。在这一方面,要求界面能够迅速、连续地对操作做出反应,在很多情 况下,还要求可以撤销操作。

对于界面的迅速和连续性,计算时间是最主要的因素。如果计算时间较长,用按钮来调用相应的运算是比 较合适的。在界面设计中,要完成操作的可撤销往住比较困难,但是,具有可撤销的操作往往能比较大程度的 增加程序的魅力。

# 8.2 图形编辑工具

MATLAB 的主要功能是科学计算,在 MATLAB 中,提供了几个用于编写应用程序界面的工具。在 MATLAB 中编写程序,很类似了在纸上写数学公式,同样,通过 MATLAB 提供的界面编辑工具编辑界面,就像在纸上绘制界面一样。由于有了这些界面编辑工具,只用几分钟就可以设计出界面的大体方案,通过事件过程编辑器可以编写各个对象的事件代码。

MATLAB 提供的用户界面编辑工具包括下面五个:

- ➢ 布局设计编辑器;
- ▶ 属性编辑器;
- ➢ 对象浏览器;
- ➢ 对齐工具;
- ▶ 菜单编辑器。

### 8.2.1 布局设计编辑器

布局设计编辑器是 MATLAB 设计用户界面的集成环境,用户可以从布局设计编辑器的菜单中直接找到其他 编辑工具的接口,同时,用户还能够直观的从布局编辑器的面板上选择适当的 GUI 部件并且在图形设计窗口中 安排他们。用户可以使用两种方法启动布局设计编辑器:

- ➤ 在命令窗口输入命令 guide;
- ▶ 从 File 菜单中下来菜单中,选择 New 下的 GUI。

在命令窗口输入:

### guide

回车将显示如图 8-3 所示的布局设计编辑器。



图 8-3 布局设计编辑器

在编辑器的左部的是可以使用的控件列表,空白处为版面设计区。在编辑器的工具栏中,除了常用的工具 外,还有如图 8-4 所示五个新的图标,他们的功能非别是:

₽		<b>₿</b> ≜ Ç	<u> </u>

### 图 8-4 五个新图标工具

₽图标	启动对齐工具
醫图标	启动菜单编辑器
聲图标	启动属性编辑器
♥图标	启动对象浏览器

### ■图像 激活图形

使用 MATLAB 6 的布局设计编辑器设计版面示非常方便的。首先在编辑器面板上选择项使用的组件类型, 在该组件上单击鼠标右键,光标将改变成一个十字,这是用户可以在编辑区使用鼠标绘出想要的控件。用户也 可以直接将选择的组件类型拖到布局设计区,释放鼠标件后,将在版面区出现该控件。如图 8-5 所示。

一旦用户已经建立了 GUI 图形,就可以单击工具栏中的 □图标,或者选择 Tools 菜单下的 Activate Figure,激活产生的 GUI 图形。

当用户选择了激活一个 GUI 图形时,将发生下列事件

- GUIDE 首先将保存该图形的 M 文件和 FIG 文件。如果还没有保存局设计, GUIDE 将打开一个另存为对话框,使用户能够指定 GUIDE 将产生的 M 文件的一个名字。然后 GUIDE 将以相同的文件名保存扩展名为.m 的 M 文件和扩展名为.fig 的 FIG 文件。
- > 如果有同名字的 M 文件存在, GUIDE 将提示用户,并由用户选择覆盖或在 M 文件中添加代码。
- ➢ MATLAB 执行 M 文件显示 GUI 窗口。





🔉 激活 GUI 时, MATLAB的 GUIDE 将自动地保存应用程序 M 文件和 FIG 文件

在使用布局设计编辑器工作时,使用鼠标左键选择一个对象然后单击鼠标右键,将在编辑窗口显示如图 8-6 所示的上下文菜单。

Cut
Сору
Delete
Duplicate
Bring to Front
Send to Back
Inspect Properties
Edit Callback
Edit ButtondownFcn
Edit CreateFcn
Edit DeleteFcn

使用上下文菜单,能够使用户方便的为应用程序 M 中任何对象添加回调函数。

图像显示出的上下文菜单是与特定控件相关联的,使用上下文菜单生成的回调函数都是相应于所 选控件的

8.2.2 对齐工具

在布局设计区域内,用户可以随意拖动任何已经选择的控件和控件组。另外,MATLAB 的 GUIDE 还提供 了更为灵活方便的部件对齐工具。

1. 对齐工具

对齐工具是 MATLAB 的 GUIDE 提供的,在设计区中排布控件的工具。单击工具栏中的 图标,或者选择 Layout 菜单下的 Align Objects,将启动如图 8-7 所示的对齐工具窗口。

📣 Align Objects 📃 🗖 🗙
_Vertical
Align OFF 🕕 🗛 🔂
Distribute 📑 📑 📑
E Set spacing 20 pixels
Horizontal
Align OFF 🖶 🚆 🔒
Distribute
E Set spacing 20 pixels
Apply

图 8-7 对齐工具窗口

对齐工具窗口示用户可以调整选中的组件间的相互位置、队齐方式和间距等。按下 Apply 按钮, 被选择的 组件将按照指定的排列方式分布。

对齐工具提供了两种的类型的派不操作:

▶ 对齐 所有的选择的部件排列到一条直线上

前布 将所选部件彼此不规则的散布。

这两种排布方式在垂直和水平两个方向上都可以应用。在多数情况下,将垂直和水平两个方向的对齐分成 两步,相对独立的进行排布,更为方便合理。

(1) 对齐

对齐选项包括垂直和水平方向两种排列选项。每个排列选项都是以一条参照线排布所选部件。这条参照线 由被选择对象的选择框决定。

例如,图 8-8 中显示了选择了三个按钮的布局区域,虚线代表了封装的选择框。所有排列选项(垂直方向的顶端,中心,底部和水平方向性的左边,中心,右端)都将根据选择框的相应边排布。



图 8-8 部件选择框

### (2) 散布

散布选项将在被选择所有的部件、部件组之间增加相等的空间。散布选项有两种不同的操作模式 在封装框内以同等地间隔散布所选组件(默认模式)

以指定象素间距散布所选组件(选择 Set spacing 并设置象素)。

2. 网格和标尺

在布局设计编辑齐上显示网格和标尺,将很大的方便用 择 Layout 菜单下的 Grid and Rolers 将启动如图 8-9 所示的网 缺省的网格线间隔是 50 象素,用户还可以从 10 到 200 用户可以选择性地启用 snap-to-grid 选项,它的作用是,

放到一根网格线附近的 9 个象素以内时,该对象都将跳到那 3.准线

布局设计编辑器中包含了垂直和水平两种准线,当你移 准线的9个象素以内时,对象将自动跳到该准线上。

当你想要在布局编辑器在一个任意的地点为部件排成直

# 图 8-9 网格与标尺对话框

•

Cancel

- 🗆 ×

📣 Grid and Rulers

Show rulers

Show quides

Grid Size: 50

🔲 Snap to grid

OK

🔽 Show grid

线建立一条参照线

动或缩放对象到在

户的设计过程。选

格与标尺对话框。

象素件进行选择。

根网格线上。

当对象被移动或缩

### 时,准线将是很有用的。

是用鼠标按住编辑区顶或便显得标尺线,将其拖到编辑区内,即可创建准线。如图 8-10 所示。

File Edit Layou	t <u>T</u> ools	Help	1	a	_							
L 🖙 🖬   Å		🖻 🛃	<b>1</b>		1				. 1			
▶ Select			100	150	200	250	- ²⁰⁰	350	400	450	500	550
📧 Push Button												
Toggle Button	33											
Radio Button												
🗹 Checkbox	320				C Radio	Button	_					
🖬 Edit Text		Edit	Text			•	•					
THE Static Text	5.70	• • •										
🚥 Slider												
🔲 Frame	220											
E Listbox												
📼 Popup Menu	120											
Axes												- 11
	120											
	2-		_									
	8-											
		L							l	l		
	4											Þ

图 8-10 创建准线

### 8.2.3 属性编辑器

使用属性编辑器可以使用户非常方便的设置布局中各种组件的属性。单击工具栏中的 经图标或者选择 Tools 菜单下的 Property Inspector、Edit 菜单下的 Inspect Properties、数标右键的上下文菜单中的 Inspect Properties,或者双击布局设计窗口中的选定项目,都将启动如图 8-11 所示的属性编辑器窗口。

285

📑 Property Inspector	>	4
📺 figure (Untitled)		
Alphamap		•
— BackingStore	💌 on	
- BusyAction	🔽 queue	
- ButtonDownFcn		
— Clipping	🔽 on –	_
- CloseRequestFcn	closereq	
🕀 – Color		
— Colormap		
— CreateFcn		
— CurrentCharacter		
⊕– CurrentPoint	[-0.2 -0.077]	
- DeleteFcn		
— Dithermap		
— DithermapMode	🔽 manual	
– DoubleBuffer	▼ off	-1

属性编辑器中列出了所有可以设置的对象属性,并显示了这些对象的当前属性值。表中列出的每个属性都 与一个能够提供该属性可接受属性值的编辑器相联系。双击指定属性,将打开相应的编辑器。

# 8.2.4 对象浏览器

使用对象浏览器,用户可以方便的图像中察看显示对象的层次。单击工具栏上的¹⁹⁸图标,或者选择 Tools 菜单下的 Object Browser,都将启动如图 8-12 所示的对象浏览器窗口。

🏕 Object Browser	_ 🗆 🗵
⊡— 🛅 figure (Untitled)	
🖵 🖲 uicontrol (Radio Button)	

图 8-12 对象浏览器窗口

### 8.2.5 菜单编辑器

菜单编辑器用于编辑图形窗口的菜单项目,单击菜单栏中的 Second ,或者选择 Layout 菜单下的 Menu Editor,都将启动如图 8-13 所示的菜单编辑器。

图 8-11 属性编辑器

Image: Separator above this item     Image: Imag	🣣 Menu Editor	
UlMenu Properties Label: Untitled 1 Tag: Separator above this item Item is checked Callback:		
Monu Bor Context Menus	Image: Second system         Image: Second system         Image: Second system         Manu Bar	UlMenu Properties Label: Untitled 1 Tag: Separator above this item Item is checked Callback:

图 8-13 菜单编辑器

使用 MATLAB 提供的菜单编辑,用户能够创造2种菜单

▶ 菜单条对象 在图形菜单栏上显示的菜单

▶ 上下文菜单 当用户在图形对象上单击鼠标右键时,弹出的菜单条上下文菜单菜单,

用具可以使用菜单编辑器工具栏上的图标方便快捷的创建所需要的菜单项:

🗈 图标 创建新菜单

🖻 图标 创建新的菜单项

🗐 图标 创建新的上下文菜单

Î 图标 删除选定的菜单项

当用户创造菜单时,MATLAB将把定义的菜单加到图形菜单条中。然后用户可以为每个菜单创造不同的菜 单项目。每个菜单项目都可以拥有自己的子菜单,而子菜单项也可以拥有自己的子菜单项。

1.建立菜单条对象

下面的例子将使用菜单工具创建一个新的菜单。

▶ 创建菜单

第一步是使用菜单工具创造新菜单。如图 8-14 所示。

▶ 设置菜单属性。

当用户单击菜单时,属性区显示文本编辑区,允许用户指定菜单标签、隔板、菜单性质检查以及回调函数 等。如图 8-15 所示。



图 8-14 创建菜单

🥠 Menu Editor	<u>-0×</u>
Ē ≓ 'E   Ū	
	UlMenu Properties Label: File Tag: menu_1 Separator above this item Item is checked Callback:

图 8-15 菜单属性设置

到菜单的增加的项目使用新菜单条款工具定义在菜单下面被显示的菜单项目。

▶ 添加菜单项

用户可以使用菜单编辑器为选定的菜单添加菜单项或子菜单。例如,如果用户想为 File 菜单添加项目,可以先选中 File,然后单击添加新菜单项目的工具图标。如图 8-16 所示。

图 8-17 所示为图形菜单栏设计的三个菜单。

▶ 击活图形窗口

击活图形窗口,将显示如图 8-18 所示的菜单栏。





🥠 Menu Editor	
File Close Save Edit Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy Copy	UlMenu Properties Label: Open Tag: menu_item1  Separator above this item Item is checked Caliback:
Menu Bar Context Menus	

图 8-17 包含子菜单的三个菜单

Untitled		
ile Edit View	· · · · · · · · · · · · · · · · · · ·	
Chi		
U K		
Copy	to file	
Past	to clipboard	
Select All		
· · · ·		

图 8-18 新的菜单栏

2. 菜单的回调函数

当用户选择菜单项目时,MATLAB 将执行菜单的回调函数。用户可以直接在菜单编辑器中菜单回调函数编 辑框中输入 MATLAB 代码。在菜单回调函数是一个很简单的命令时,这条途径是可行的。然而,把一个子函数 加到应用程序 M 文件,通常会更好一些的。

🔉 GUIDE并不会不自动地为菜单增加一个到应用程序 M 文件的回调函数子函数。用户必须手工地增

加菜单回调函数

3. 上下文菜单

当用户在定义了菜单的物体上单击鼠标右键时,将显示上下文菜单。使用菜单编辑器,用户可以很方便的 为指定对象建立上下文菜单。

▶ 创造母菜单

上下文菜单中所有的菜单项目是上下文菜单的子项目,它们不会在图形菜单栏上显示。要定义母菜单,从 菜单编辑的工具条上的选择的 New Context Menu 菜单。如图 8-19 所示。

🥠 Menu Editor	
New Context Menu	Properties Nothing selected.
Menu Bar Context Menus	

图 8-19 建立上下文菜单

➢ 在定义上下文菜单之前,必须先选择菜单编辑器窗口的 Context Menus 标签➢ 添加上下文菜单项目

在菜单编辑的工具条上使用 New Menu Item 在上下文菜单上创造项目。如图 8-20 所示。

📣 Menu Editor	
i	
<mark>New Menu Item</mark> ¬Ţ <u>∎</u> ] axes_context_menu	UlContextMenu Properties Tag: [axes_context_menu Callback:
Menu Bar Context Menus	

图 8-20 在上下文菜单上创建菜单项目

▶ 与对象建立联系

在布局编辑器中选择要与上下文菜单建立联系的对象,启动该对象的属性编辑器,在属性编辑器中, UIContextMenu 项目后的下拉选单中选择合适的上下文菜单。如图 8-21 所示。

🚅 Property Inspector	<u> ×</u>
uicontrol (Radio Button)	
- Interruptible	💌 on 👘 🛌
— ListboxTop	0.0
— Max	1.0
— Min	0.0
⊕– Position	[20.200 20.615 23.400 5.2]
- SelectionHighlight	on
⊕– SliderStep	[0.01 0.1]
- String	🗾 Radio Button
- Style	🔽 radiobutton
— Tag	radiobutton2
- TooltipString	
— UIContextMenu	<none></none>
- Units	
— UserData	axes_context_menu
— Value	

图 8-21 上下文菜单与 GUI 对象建立联系

8.2.6 图形中的控件

在 MATLAB 的 GUI 布局设计编辑器的左部,以按钮的形式排列着可供选 择的控件,如图 8-22 所示。

1. 各种控件介绍

这些图标对应控件的中英文名称和含义分别为:

- 是小的矩形面,常常在上面标有文本。 ➢ 按钮 (Push Buttons) 将鼠标指针移动至按钮,单击鼠标,按钮被按下随即自动弹起, 并执行回调程序。按钮的"Style"属性值是 pushbutton;在属性 "String"中定义它上面标注的文本。
- ➢ 开关按钮(Toggle Buttons) 和一般按钮形状相同,区别在于它 有两种状态——"开"(按钮按下)和"关"(按钮弹起),用鼠标单 击按钮,它会从一种状态变成另一种状态,并执行相应的回调程 序(两种状态各对应不同的回调程序)。开关按钮"开"时, Value 属性的值为在 Max 属性中指定 的值;"关"时,Value 属性的值为在 Min 属性中指定的值。按钮的"Style "属性值是" togglebutton "; 在属性"String"中定义它上面标注的文本。
- ➤ 编辑框(Edit Text) 允许用户动态地编辑文本字符串或数字,就象使用文本编辑器或文字处理器 一样。编辑框一般用于让用户输入或修改文本字符串和数字。编辑框的"Style"属性值是"edit", 在"String"属性中设置开始显示的字符串。
- 向用户提出互斥的一系列选项清单,用户可以选择其中的某一项。 ➢ 弹出式菜单(Popup Menu) 弹出式菜单不同于前面讲过的菜单(下拉式菜单),它不受菜单条的限制,可以位于图形窗口内的 任何位置。通常状态下的弹出式菜单以矩形的形式出现,矩形中含有当前选择的选项,在选项右 侧有一个向下的箭头来表明该对象是一个弹出式菜单。当指针处在弹出式菜单的箭头之上并按下 鼠标时,出现所有选项。移动指针到不同的选项,单击鼠标左键就选中了该选项,同时关闭弹出 式菜单,显示新的选项。当选择—个选项时,弹出式菜单的"Value"属性值为该选项的序号。弹 出式菜单的"Style"属性值是"popupmenu"。在"String"属性中设置弹出式菜单的选项字符串, 不同的选项之间用"|"分隔,类似于换行。
- 又称无线按钮,它由一个标注字符串(在"String"属性中设置)和字 ▶ 单选按钮(Radio Buttons) 符串左侧的一个小圆圈组成。当选择时,圆圈被填充一个黑点,且属性"Value"的值为1;若未

k	Select
OK	Push Button
TGL	Toggle Button
۲	Radio Button
Ø	Checkbox
EDĬŢ	Edit Text
TRE	Static Text
80-18	Slider
	Frame
F	Listbox
-	Popup Menu
¥	Axes

图 8-22 GUIDE 的可用控件

被选择,圆圈为空,属性"Value"的值为0。单选按钮一般用于在一组互斥的选项中选择一项。 为了确保互斥性,各单选按钮的回调程序需要将其它各项的"Value"值设为0。单选按钮"style" 的属性值是"radiobutton"。

- 图文框(Frames) 图文框是填充的矩形区域。一般用来把其它控件放入图文框中,组成一组。 图文框本身没有回调程序。注意只有用户界面控件可以在图文框中显示。由于图文框是不透明的, 因而定义图文框的顺序就很重要,必须先定义图文框,然后定义放到图文框中的控件。因为先定 义的对象先画,后定义的对象后画,后画的对象覆盖到先画的对象之上。
- 静态文本框(Static Text) 静态文本框用来显示文本字符串,该字符串是由属性"string"所确定的。静态文本框之所以称之为"静态",是因为文本不能被动态地修改,而只能通过改变"String"属性来更改。静态文本框一般用于显示标记、提示信息及当前值。静态文本框的"Style"属性值是"text"。
- 列表框(List Boxes) 列表框列出一些选项的清单,并允许用户选择其中的一个或多个选项,一 个或多个的模式由 Min 和 Max 属性控制。Value 属性的值为被选中选项的序号,同时也指示了选 中选项的个数。当单击鼠标按钮选中选项后,Value 属性的值被改变,释放鼠标按钮的时候 MATLAB 执行列表框的回调程序。 列表框的 "Style"属性值是"listbox"。
- ▶ 复选框 (Checkboxes) 又称检查框,它由—个标注字符串(在"String"属性中设置)和字符串 左侧的一个小方框所组成。选中时在方框内添加"."符号,"Value"属性值设为1;未选中时方 框变空,"Value"属性值设为0。复选框一般用于表明选项的状态或属性。
- 滑动条(Sliders) 又称滚动条,包括三个部分,分别是滑动槽,表示取值范围;滑动槽内的滑块, 代表滑动条的当前值;以及在滑动条两端的箭头,用于改变滑动条的值。滑动条一般用于从一定 的范围中取值。改变滑动条的值有三种方式,一种是用鼠标指针拖动滑块,当滑块位于期望位置 后松开鼠标;另一种是当指针处于滑动槽中但不在滑块上时,单击鼠标按钮,滑块沿该方向移动 一定的距离,距离的大小在属性"SliderStep"中设置,为该属性值的第二个元素,缺省情况下等 于整个范围的10%:第三种方式是在滑动条的某一端用鼠标单击箭头,滑块沿着箭头的方向移动 一定的距离,距离的大小在属性"SliderStep"中设置,为该属性值的第一个元素,缺省情况下为 整个范围的1%。 滑动条的"Style"属性值是"slider"。

2. 控件的几个重要属性

和其它各种图形对象一样, GUI 控件也有自己的很多可以设置的属性, 这里我们对其中的几个重要属性给 以比较详细的介绍和总结。

(1) Value 属性

控件的当前值,格式为标量或向量。该属性对不同的控件有不同的取值方式,分别为:

- ▶ 复选框 被选中时 Value 的值为属性 Max 中设置的值:未选中时 Value 的值为属性 Min 中设置的值。
- 列表框 被选中选项的序号,当有多个选项被选中时,Value 属性的值为向量。序号指的是选项的排列次序,最上面的选项序号为1,第二个选项序号为2.....。
- 弹出式菜单 和列表框类似,也是被选中选项的序号,只是弹出式菜单只能有一个选项被选中, 因而 Value 属性的值为标量。
- 单选按钮 被选中时 Value 的值为属枇 Max 中设置的值:未选中时 Value 的值为属性 Min 中设置的值。
- ▶ 滑动条 Value 的值等于滑块指定的值。
- ▶ 开关按钮 "开"时 Value 的值为属性 Max 中设置的值;"关"时 Value 的值为属性 Min 中设置的值。
- 按钮、编辑框、图文框、静态文本不设置这个属性值。

(2) Max 属性

- 编辑框 如果 Max 的值减去 Min 的值大于 1,那么编辑框可以接受多行输入文本;如果 Max 的值减去 Min 的值小于或等于 1,那么编辑框只能接受一行输入文本。
- ➢ 只能输入一行文本时用回车键结束输入:可以输入多行文本时用 Ctrl+回车键结束输入
  - 列表框 如果 Max 的值减去 Min 的值大于 1,那么允许选取多个选项;如果 Max 的值减去 Min 的值小于或等于 1,那么只能选取一个选项。
  - ▶ 单选按钮 当单选按钮被选中时 Value 属性的取值。
  - ▶ 滑动条 滑动条的最大值,缺省值是1。
  - ▶ 开关按钮 当开关按钮 "开"(被选中)时 Value 属性的取值。缺省值是 1。
  - ▶ 文本框、弹出式菜单、按钮和静态文本框不使用 Max 属性。
- (3) Min 属性

指定 Value 属性中可以设置的最小值,格式为标量。该属性对不同的控件有不同的含义,分别如下所述:

- ▶ 复选框 复选框未被选中时 Value 属性的取值。
- ▶ 编辑框 如果 Max 的值减去 Min 的值大于 1,那么编辑框可以接受多行输入文本;如果 Max 的值减去 Min 的值小于或等于 1,那么编辑框只能接受一行输入文本。
- 列表框 如果 Max 的值减去 Min 的值大于 1,那么允许选取多个选项;如果 Max 的值减去 Min 的值小于或等于 1,那么只能选取一个选项。
- ▶ 单选按钮 单选按钮未被选中时 Value 属性的取值。
- ▶ 滑动条 滑动条的最小值,缺省值是0。
- ▶ 开关按钮 当开关按钮 "关"(未被选中)时 Value 属性的取值。缺省值是 0。
- > 文本框、弹出式菜单、按钮和静态文本框不使用 Min 属性。

# 8.3 编程创建用户界面

在上一节讲述了使用图形用户界面的图形编辑工具创建用户界面,这一节将讲述编写 M 文件,使用 MATLAB 函数建立用户界面。

### 8.3.1 编程建立菜单

在 MATLAB 中,用于建立菜单的函数是 unimenu。

1. 函数 uimenu 的调用格式

函数 uimenu 用法和其它建立图形对象的函数类似,一般它有如下两种调用格式

- h=uimenu('PropertyNamel', valuel, 'PropertyName2', value2,...)
   在当前图形窗口顶部的菜单条上
   建立菜单,同时返回该菜单的句柄。参数 "PropertyNamel "和 "valuel "分别是菜单对象的属性
   和设置值。
- h=uimenu(Parent...) 用参数 "Parent"指定菜单的上一级菜单的句柄,或是图形窗口的句柄。
   如果 "Parent"是一个图形窗口的句柄,那么 uimenu 在对应图形窗口顶部的菜单条上建立菜单;
   如果 "Parent"是菜单的句柄,那么 uimenu 建立该菜单的子菜单。

在这里强调几种菜单对象的属性设置。

(1) Label 和 Callback

这两个属性是菜单的基本属性,编写一个具有基本功能的菜单必须要设置 Label 和 Callback 属性。

Label 是在菜单项上显示的标注文本,在文本中同时可以设置该菜单项的快捷键:把符号 " & " 放在标注字 符串中用于快捷键的字符前,执行该菜单项时就可以用 " Alt+该字符 " 来完成。

例如,如下命令在当前图形窗口的菜单条上建立一个菜单并设置其快捷键:

h1=uimenu('Label' ' ' &Graph');

建立的菜单项显示将是"File",快捷键定义为"Alt+g"。

属性 Callback 用来设置菜单项的回调程序。如下命令在菜单"Graph"下建立子菜单,并给出回调程序:

h2 1=uimenu(h1, 'Label', '&Grido&n', 'Callback', 'gridon'); 此命令建立的菜单项是"Grid on",快捷键定义为"Alt+n",回调程序"grid on"由于给图形添加网格。 (2) Position 属性 Position 用来调整菜单项的相对位置。 例如, 在菜单"Graph"下增加一个子菜单"Gridoff", 比较一下它和子菜单"Gridon"的属性 Position 的值, 命令如下 h2_2=uimenu(h1, 'Label', 'Grido&ff', 'Callback', 'gridoff'); get(h2_1, 'position') ans= 1 get(h2_2, 'position') ans= 2 可见属性 Position 的值等于菜单项的排列顺序,排在最上面的子菜单"Gridon"的 Position 值为 1,排在第二 位的子菜单 "Grid off"的 Position 值为 2。如果需要调整相对位置可以用 set 函数来完成 set(h2 1, 'position', 2) 该命令把子菜单 "Gridon "的 Position 值设为 2,从而把它移到了第二位,子菜单 "Gridoff"自动移到第一 位。 同样,横向排列在图形窗口顶部菜单条上的菜单也可以用 Position 属性来设置其相对位置,方法和纵向排列 的菜单相同,只是这时的 Position 值表示菜单横向排列的相对位置。 例如,在菜单条上的"Graph"菜单位于图形窗口的五个基本菜单之后,那么它的 Position 属性值为 get(h1, 'position') ans= 6 (3) Checked 和 Separator 属性 Checked 设置是否在菜单项前添加选中标记,设为"on"表示添加,设为"off"表示不添加。 比如,如下命令给子菜单"Gridon"添加选中标记 set(h2 1, 'Checked', 'On') 需要注意,有些菜单的选中标记是相斥的," Gridon " 和 " Gridoff " 就属于这种类型,两个菜单项不能同时 标记为选中或不选中,这就要求给一个菜单项添加选中标记的同时去掉另一个菜单项的标记。 属性 Separator 用于在菜单项前添加分隔符, 是菜单的层次更加清晰。 (4) BackgroundColor 和 ForegroundColor 背景色和前景色这两个属性值均为 RGB 向量, BackgroundColor 是菜单本身的颜色, ForegroundColor 是菜 单上标注字符串的颜色。 3. 使用函数 uimenu 设计菜单 本小结将举例介绍用 uimenu 函数设计菜单的具体过程。 首先建立一个图形窗口,去掉窗口本身的菜单条和工具条,并命名为"My GUI",程序如下: h0 = figure('MenuBar', 'none', ... 'ToolBar', 'none',... 'Name', 'My GUI');

下面从左到右依次建立各级菜单。

把第一个菜单命名为"Mesh",下面有三个选项"Membrane"、"Peaks"、"Sinc",分别用来绘制隔膜的特征 函数、三维高斯分布、Sinc 函数(草帽图)的网格图。程序如下

h1=uimenu(h0, 'Label', '&Mesh'); h21=uimenu(h1, 'Label', '&Membrane',... 'callback','mesh(membrane); axis tight'); h21=uimenu(h1, 'Label', '&Peaks',... 'callback','mesh(peaks);axis tight'); h21=uimenu(h1, 'Label', '&Sinc',... 'callback',... ['[X,Y] = meshgrid(-8:0.5:8);',... 'R = sqrt(X.^2+Y.^2)+eps;',... 'Z=sin(R)./R;',... 'mesh(X,Y,Z)']);

➢ "Sinc"菜单的回调程序书写格式,"callback"属性值是一个 MATLAB 字符串, MATLAB 将它传 给函数 eval 并在命令窗口工作空间执行

字符串可以串联起来生成一个合法的 MATLAB 字符串,只是如果该字符串需要分行,那么每一行的语句必须都加上单引号,而且要把整个字符串的几行括在方括号中。

同时注意,除了最后一句命令,其他引号内的各语句必须用分号或逗号结尾,同时在引号后要用空格或逗 号结尾。实际上"Sinc"菜单"callback"属性中的几条语句后可以不要逗号,下一行的空格可以起到相同的作 用。

下面将建立第二个菜单"",该菜单下有五个菜单项目,分别用来选择不同的颜色映象。当某个选项被选中时,为这个选项添加选中标记,同时去掉其他选项上的选中标记。每个选项还都设置了相应的快捷键。程序如下

```
h1= uimenu(h0, 'Label', '&Colormap');
h22(1) = uimenu(h1, 'Label', '&Default',...
                    'Accelerator', 'd',...
                      'Checked'.'on'....
                     'Callback',...
['set(h22,"Checked","off"); '...
'set(h22(1),"Checked","on");'...
         'colormap("default")']);
h22(2) = uimenu(h1, 'Label', '&Spring',...
      'Accelerator', 's',...
       'Callback'....
      ['set(h22,"Checked","off"); '...
         'set(h22(2),"Checked","on");'...
         'colormap(spring)']);
h22(3) = uimenu(h1, 'Label', 'S&ummer',...
       'Accelerator', 'u',...
       'Callback',...
       ['set(h22,"Checked","off"); '...
        'set(h22(3),"Checked","on");'...
        'colormap(summer)']);
h22(4)= uimenu(h1, 'Label', '&Autumn',...
       'Accelerator', 'a',...
       'Callback'....
```

['set(h22,"Checked","off"); '... 'set(h22(4),' 'Checked",' 'on' '); '... 'colormap(autumn)']);

h22(5) = uimenu(h1, 'Label', '&Winter',...

'Accelerator', 'w',...

'Callback',...

['set(h22,"Checked","off"); '...

'set(h22(5),"Checked","on");'...

'colormap(winter)']);

# 程序中各菜单项的回调程序的字符串,在单引号内的字符串用两个单引号(不等于双引号)表示单引 号

在设置选中标记时,首先用命令"set(h22,"Checked","off");"把向量"h22"中的五个句柄对应的菜单项都 设为未选中状态,然后,把选择的菜单项设为选中状态,这样保证了几个选项之间的互斥性。

最后建立一个控制坐标轴显示的菜单 "Axis",用于选择是否显示坐标轴。和菜单 "Colormap" 类似,它的 两个选项也添加了选中标记,并且指定了快捷键。程序如下

h1 = uimenu(h0, 'Label', '&Axis'); h23(1) = uimenu(h1, 'Label', 'Axis o&ff',... 'Accelerator', 'f',... 'Callback',... ['axis off;'... 'set(h23, "checked", "off");'... 'set(h23(1), "checked", "on");']); h23(2)=uimenu(h1, 'Label', 'Axis o&n', ... 'Accelerator', 'n', ... 'Callback', ... ['axis on;', ... 'set(h23, "checked", "off");'... 'set(h23(2), "checked", "on");']);



把这些程序组合起来,就形成了一个功能比较完整的用户界面菜单,如图 8-23 所示。

#### 图 8-23 编程生成的图形菜单

### 8.3.2 编程建立控件

在 MATLAB 中,用来建立控件的函数是 uicontrol。前面已经对各种控件一一作了介绍,虽然不同类型的控件属性各不相同,但它们都是 uicontrol 对象,都由函数 uicontrol 来建立,在 uicontrol 的属性 "Style"中指定不同的控件类型就可以建立不同的控件。

函数 uicontrol 的用法和函数 uimenu 相同,这里不再重复。

各种控件的 "Style " 属性值在前面已经讲过,下面举例说明如何用 uicontrol 函数编写建立控件的程序。

1.建立图形窗口和坐标轴

首先建立—个图形窗口,指定窗口的大小,以便后面设置坐标轴和控件的位置

h0=figure('MenuBar', 'none',...

'ToolBar', 'none',...

'Position',[198 56 408 468],...

'Name', 'My GUI');

指定一个坐标轴对象的位置,这样后面的绘图将在这个坐标轴范围内,而不会自动充 满整个图形窗口, 而且为后面建立控件预留位置:

h1=axes ('Parent',h0,... 'Position',[0.15 0.45 0.7 0.5],... 'Visible','off');

### 2. 建立编辑框

在坐标轴区域下建立一个编辑框,用户可以在里面输入图形的标题,编辑框前放置文 本框起到提示作用, 程序如下

htext1=uicontrol('Parent',h0,... 'Units','points',... 'Position',[20 110 45 15],... 'String','Input Title',... 'Style','text'); hedit=uicontrol('Parent',h0,... 'Units','points',... 'Position',[65 110 70 15],... 'callback','title(get(hedit,"String"))',... 'Style','edit');

### 3. 建立按钮

在编辑框下方建立三个按钮对象,分别用于建立三种不同的网格图:隔膜的特征函数、 三维高斯分布、 Sinc 函数(草帽图),程序如下

hpushl=uicontrol('Parent',h0, ... 'Units','points',... 'String','Membrn',... 'Position',[15 65 50 18], ... 'callback','mesh(membrane);axis tight'); hpush2=uicontrol('Parent',h0,... 'Units','points',... 'String','Peaks',... 'Position',[75 65 50 18],... 'callback','mesh(peaks);axis tight'); hpush3=uicontrol('Parent',h0, ... 'Units','points', ... 'String','Sinc', ... 'String','Sinc', ... 'Position',[135 65 50 18], ... 'Position',[135 65 50 18], ... 'callback',... ['[X,Y] = meshgrid(-8:0.5:8);'... 'R = sqrt(X.^2+Y.^2)+eps;'... 'Z=sin(R)./R;'... 'mesh(X,Y,Z)']);

4. 建立滑动条

按钮下面安排一个滑动条,调节图形颜色的亮度,滑块从中间向右移动图形变亮,向左移动图形变暗。同时,在滑块左侧建立了一个静态文本,用以提示用户滑动条的功能。程序如下

htext2=uicontrol('Parent',h0,... 'Units','points', ... 'Position',[20 30 45 15], ... 'String','Brightness',... 'Style','text'); hslider=uicontrol('Parent',h0, ... 'Units','points',... 'Units','points',... 'Position',[65 30 115 15], ... 'Min',-1,... 'Max',1,... 'Style','slider',... 'callback',... 'brighten(get(hslider,"Value"))');

上面的程序中使用 brighten 函数改变颜色的亮度,该函数的调用方式为

brighten(beta)

参数 beta 指示调整的幅度,取值范围在[-11]之间,其中[-10]标是把当前颜色变暗,[01]表示把当前颜色变亮。

5.建立单选按钮

现在在坐标轴下方的右侧建立五个单选按钮,用于选取五种颜色映象中的一种,每个单选按钮对应一种颜 色映象,按钮之间是相斥的关系,不能同时选取多个按钮。在五个按钮上方安排一个静态文本框,提示用户在 这里选择颜色映象。把缺省颜色映象('Default')按钮的初始状态设为选中状态。程序如下

htext3=uicontrol('Parent',h0,... 'Units','points', ... 'Position',[200 110 80 15], ... 'String','Select Color :',... 'Style', 'text'); hradiol(1)=uicontrol('Parent',h0, ...

'Units', 'points', ... 'Position',[200 95 80 15], ... 'Style', 'radiobutton', ... 'String','Default',... 'Value',1,... 'Callback',... ['set(hradiol,"Value",0); '... 'set(hradiol(1),"Value",1);'... 'colormap("default")']); hradiol(2) = uicontrol('Parent',h0, ... 'Units', 'points', ... 'Position',[200 80 80 15], ... 'Style', 'radiobutton', ... 'String', 'Spring',... 'Callback',... ['set(hradiol,"Value",0); '... 'set(hradiol(2),"Value",1);'... 'colormap(spring)']); hradiol(3) = uicontrol('Parent',h0, ... 'Units', 'points', ... 'Position',]200 65 80 15], ... 'Style', 'radiobutton', ... 'String', 'Summer',... 'Callback',... ['set(hradiol,"Value",0); '... 'set(hradiol(3),"Value",1);'... 'colormap(summer)']); hradiol(4) = uicontrol('Parent',h0, ... 'Units', 'points',... 'Position',[200 50 80 15], ... 'Style', 'radiobutton', ... 'String', 'Autumn',... 'Callback',... ['set(hradiol,"Value",0); '... 'set(hradiol(4),"Value",1);'... 'colormap(autumn)']); hradiol(5) = uicontrol('Parent',h0, ... 'Units', 'points', ... 'Position',[200 35 80 15], ... 'Style', 'radiobutton', ... 'String', 'Winter',... 'Callback',... ['set(hradiol,"Value",0); '... 'set(hradiol(5),"Value",1);'... 'colormap(winter)']);

到此为止,就建立了一个用户界面,把上述程序保存为一个 M 文件,如 figure1.m,在命令窗口执行该文件, 将出现如图 8-24 的用户界面。

📣 Figure No. 1: My GUI	
Input Title	Select Color :
	Derault     C Spring
Membrn Peaks Sinc	C Summer
	C Autumn
Brightness	C Winter

图 8-24 控件界面

7. 建立对话框

几乎所有的 Windows 程序都要借助于对话框来和用户打交道,所谓对话框实际上就是 一种带有各种控件 的用户界面,一般是弹出显示的单独窗口,用来要求或提供信息。通过使 用对话框,用户可以通知计算机一 些自己的选择,也可以将一些参数赋给计算机,而计算机 也可以通过对话框将一些信息反馈给用户。

理论上,每个对话框都可以通过编程从最基本的图形窗口开始逐步建立起来,但这样 的工作量很大,而 且有很多有专门作用的对话框格式都很固定,因而 MATLAB 提供了多种建立专用对话框的函数,这些函数给用 户提供了很大的方便,同时也保证了界面的—致。

下面分别介绍这些函数的用法。

(1) errordlg 函数

该函数的调用格式为

- errordlg 建立一个错误提示对话框,如果已有错误对话框存在,则把该对话框弹处到前面来显示。
- ➢ errodlg('errorstring') 建立一个包含字符串 "errorstring" 错误对话框。
- ➢ errordlg('errorstring', 'dlgname') 建立名为" dlgname",包含字符串" errorstring "的错误对话框。
- errordlg('errorstring','dlgname','on') 指定是否代替已经存在的同名对话框," on "表示把已有的 对话框弹出道前面来显示,不建立新的对话框;如果是" off "则新建立一个对话框。
- ▶ h=errordlg(...) 返回对话框的句柄。

例如,可以使用如下命令建立一个如图 8-25 所示的错误提示对话框

errordlg('Invalidation Operation !','Error Dialog')

📣 Error	Dialog			
$\bigcirc$	Invalidati	on Operatic	on !	
		OK		

### 图 8-25 错误对话框

错误提示对话框的大小根据它显示的错误信息字符串长短自动设置,用户不能改变其大小

	(2)helpdlg 函数	
	该函数调用格式为	
	▶ helpdlg 建立一个缺省帮助对话框,如果已有帮助对话框存在,则把该对语	舌框弹出到前面来显
	示。	
	➢ helpdlg(helpstring') 建立一个包含字符串"helpstring"的帮助对话框。	
	➢ helpdlg('helpstring','dlgname') 建立名为"dlgname",包含字符串"helpstring)	ıg"的帮助对话框。
	▶ h=helpdlg() 返回对话框的句柄。	
	例如,可以使用如下命令建立一个如图	8-26 所示的帮助对
话框		
	helpdlg('Use Ctrl+d to draw graph', 'My	Help Dialog')
	按下"OK''按钮对话框消失,否则将一	直显示。
	≥ 帮助对话框的大小根据它显示的帮	助字
	符串长短自动设置,用户不能改变	其 大
	(3) inputdlg 函数	
	该函数的调用格式为:	
	➤ answer=inputdlg(prompt) 建立一个输入灯话框,用尸输入返回到"answer" 字符串。	"," prompt " 是提示
	➤ answer=inputdlg(prompt,title) 带有标题"title"的输入对话框。	
	➢ answer=inputdlg(prompt,title,lineNo) "lineNo"用宋指定用户输入值的行数	•0
	➢ answer=inputdlg(prompt,title,lineNo,defAns) "defAns"用于指定每个输入口	页的缺省值,格式为
	字符串。	
	➢ answer=inputdlg(prompt,title,lineNo,defAns,Resize) "Resize"指示对话框能	否被改变大小,取
	值包括" on "和" off", 分别表示可以改变和不能改变。缺省情况下输入对话	i框大小不能改变,
	它是有模式对话框;如果选中了 " on ", 那么输入对话框自动成为无模式对话	i框。
	所谓有模式对话框,指在对话框被关闭之前,用户无法在程序的其它地方进行工作,比	;如" OpenFile " 对话
框京	就是典型的有模式对话框;而无模式对话框在保留在屏幕上的同时,用户还可以在程序的	的其它窗口中进行工
ſF,	,"Find"对话框是典型的无模式对话框。	

例如,可以使用如下命令建立一个如图 8-27 所示的输入对话框

```
prompt={'Enter Graph Title : ','Enter X Label : ','EnterYLabel : '}
```

title=['Setup Graph Label...']

line=1

×

def={'Graph1','X axis','Y axis'};

glabel=inputdlg(prompt,title,line,def);

📣 Setup Graph Label…		×
Enter Graph Title:		
Enter X Label:		
X axis		_
EnterYLabel: Y axis		
	ОК	Cancel



在按下该对话框的"OK"或"Cancel"按钮之前,对话框会一直显示在屏幕上,而且由于输入对话框缺省

情况下是有模式对话框,这是用户无法进行其他操作。

(4) pagedlg 函数

该函数的调用格式为:

- ▶ pagedlg 显示当前图形窗口的页面位置对话框。
- ▶ pagedlg(fig) 显示指定句柄的图形窗口页面位置对话框。

如果当前窗口中没有图形窗口,那么该函数会自动建立一个缺省的图形窗口。

例如,可以使用如下命令建立一个如图 8-28 所示的一个已有的图形窗口的页面位置对话框 pagedlg(gcf)

剩 Page Position: Figure 1 : My Gl	I IIIX
Paper Size: [8.5 11]	
Paper Units: Jinches	
Paper Orientation: portrait	
Match Paper Area to Figure Area	
Set Paper Position Explicitly	
Paper Position:	[0.25 2.5 8 6]
Default Paper Settings	Fill
Center	Fill(Fixed Aspect Ratio)
Help Print	Cancel Done

图 8-28 页面位置调整对话框

(5) printdlg 函数

该函数的调用格式为

- ▶ printdlg 显示当前图形窗口的打印对话框。
- > printdlg(fig) 建立指定句柄图形窗口的打印对话框。
- ➢ printdlg('-crossplatform', fig) 显示标准的跨平台打印对话框,而不是 windows 平台内置的打印 对话框,该对话框不受操作系统平台限制。

### 🔉 打印对话框是有模式对话框

例如,可以使用如下命令建立一个如图 8-29 所示的一个已有的图形窗口的 windows 打印对话框 printdlg(gcf);

打印 打印机	? ×
名称 (M): HP LaserJet 5L PCL	▼ 属性 (2)
状态: 默认的打印机:就绪	
型号: HP LaserJet 5L PCL	and the second
位置: \\Tom\hp5	
备注:	打印到文件(L)
打印范围	┌份数
● 全部 (&)	份数 (C): 1 📑
○ 页数 ④ 从 ⑧: 到 ⑪):	
○ 选择的范围 (5)	
	<b>補</b> 定 取消

图 8-29 windows 内置打印对话框

如下命令建立标准的跨平台打印对话框,如图 8-30 所示

printdlg('-crossplatform',gcf) ;

Print: Figure 1: My GUI		×
_ Destination	and the second second	
Printer: \\Tom\hp5		
C File:		Browse
Driver: Windows		Options
<ul> <li>Figure Size on Printed Page</li> <li>○ Same size as screen</li> <li>● 8 by 6 inches</li> </ul>	Axes Limits and Tid C Same as on sc C Allow MATLAB 1	cks reen to select
Help.,	ОК	Cancel

图 8-30 标准跨平台打印对话框

# (6) questdlg 函数

该函数的调用方式为:

- ▶ button=questdlg('qsstring') 建立提问对话框,提示问题"qstring"。对话框有三个缺省的按钮 "Yes"、"No"和"Cancel"。"buUon"包含按下的按钮的名字。
- ➢ button=questdlg('qstring','title') 为提问对话框加上标题"title"。
- ▶ button=questdlg('qstring','title','default') 在 " default " 中指定当按下回车键时的缺省按钮。
- ▶ button=questdlg('qstring','title','strl','str2','default')
   建立有两个指定按钮的对话框,按钮的标注为 "strl"和"str2",缺省按钮在"default"中指定。
- ▶ button=questdlg('qstring','title','strl','strT,'str3','default')
   建立有三个按钮的对话框,按钮的标注为
   "strl "、"str2 "和 "str3 ",缺省按钮为 "default "。

## 🔉 提问对话框是有模式对话框

例如,可以使用如下命令建立一个提问对话框

queststring={'This operation may take a long time!';' Are you sure?'};

title='Are you sure?';

button=questdlg(queststring,title,'Yes','No','Why','No');

## 按下不同的按钮就会返回不同的按钮名给"button"。

(7) warndlg 函数

## 该函数的调用格式如下

- wamdlg 建立一个缺省的警告对话框,对话框标题为"Warning Dialog",显示的警告字符串为
   "This is the default warning string "。
- ➤ warndlg('warningstring') 用 "warningstring "指定警告字符串。
- ➢ warndlg('warningstring','dlgname') 用 "wamingstring"指定警告字符串,用"dlgname"

## 指定对话框标题。

▶ h=warndlg(...) 返回对话框句柄。

## 例如,下面的命令建立如图 8-31 所示的警告对话框

warnstr='This operation may take a long time!!';

title='WarningI!!';

warndlg(warnstr, title);

刘 WarningI!! 📃 🗖 🖪	×
This operation may take a long time!	
ОК	•

运 提问对话框也可以发出警告信息,它和警告对话框的区别是:警告对话框仅是提示信息,不对下 一步的操作做出反应,而提问对话框可以对不同按钮被按下的事件进行不同的操作

(8)msgbox 函数

该函数的调用格式为:

- msgbox(message) 建立一个消息框,显示"message"中的信息。"message"可以是字符串的向量、矩阵或数组。消息框带有一个"OK"按钮。
- ▶ msgbox(message,title) 用 " title " 指定消息对话框的标题。
- msgbox(message,title,'icon') 用 " icon " 指定在消息框中使用的图标。" icon " 的取值可以为 : " none "、" error "、" help "、" warn " 或 " custom ", 分别对应于不显示图标、错误图标(错误对话框 中的图标)、帮助图标(帮助对话框中的图标)、警告图标(警告对话框中的图标)或自定义图标。缺 省情况是 " none "。
- msgbox(message,title,'custom',iconData,iconCmap)
   自定义一个自己的图标, "iconData"是定义
   图标的数据, "iconCmap"是图标采用的颜色映象。
- msgbox(...,'createMode') 指定消息框是有模式还是无模式对话框。" createMode " 的取值包括 " modal " (有模式)和 " non-modal " (无模式)。不指定参数 " createMode " 时消息框是无模式对话 框。
- ▶ h=msgbox(...) 返回消息框的句柄。

如下命令将建立如图 8-32 所示的消息框

msgbox('This is my test messagebox . ','Test msgbox','help');

单击"OK''按钮消息框消失。



图 8-32 消息框

8.4 小结

本章介绍了图形用户界面(GUI)——包括用户界面菜单、用户界面控件和各种对话框的设计方法。涉及的内容比较全面,至此,大家应该可以通过组合各个部分编写出自己的用户界面程序了。对于用到的各种对象的属性有一个全面的了解,这样会启发自己灵活运用各种属性编写出高水平的用户界面来。当然,界面本身只相对于一个框架,框架里面的内容需要根据自己的需要进一步填充,这里面涉及的知识请大家参考相关部分。

304

图 8-31 注意对话框

# 附录 MATLAB 函数命令索引表

函数/命令	功能
abs0	绝对值函数
acos()	反余弦函数
acosh0	反双曲余弦函数
acot()	反余切函数
acoth()	反双曲余切函数
acsc()	反余割函数
acsch()	反双曲余割函数
airfoil()	NASA 翼面稀疏矩阵显示
all()	测试向量中所有元素是否为真
angle()	相角函数
any()	测试向量中是否有为真元素
ans()	返回最新结果
arith()	MATLAB 的各种算术运算符信息
asec()	反正割函数
asech()	反双曲正割
asin()	反正弦函数
asinh()	反双曲正弦
atan()	反正切函数
atan ² ()	四个象限内反正切
atanh()	反双曲正切
auread()	读声音文件
auwrite()	写声音文件
axes()	坐标轴任意形式的设定
axis()	坐标轴标度设定
balance()	改进特征精度的均衡变换
bar()	绘制条形图
bench()	MATLAB 测试基准问题
blanks()	设置一个由空格组成的字符串
bone	带有蓝色的灰度颜色表
break	中断循环执行的语句
brighten()	使图形色调变亮
bucky	Buckminister Fuller 拱形演示
caxis()	伪颜色坐标轴设定
cd	改变当前的工作目录
cdf2rd()	复块对角矩阵到实块对角阵转换
ceedit	设置命令行编辑与回调的参数
ceil()	对+ 方向取整数
census	2000 年美国人口普查预测
chol()	Cholesky 分解
cla	清除当前坐标轴
clabel()	等高线剖面标志
clc	清除命令窗口显示
clear	删除内存中的变量与函数
clf	清除当前图形窗口
clock	时钟
close	关闭图形窗口
clommd()	最小列的阶次
colon	冒号表达式的帮助信息
colormap()	设定颜色可查表
colormenu	颜色表演示
colperm()	由非零数据的计数来排列各列
comet()	彗星状轨迹绘制
V	

函数/命令	
comet3()	
compan()	
compass()	绕行曲线绘制
computer()	计算机类型测试
cond()	求矩阵的条件数
cond()	共轭复数函数
condest()	估算范数
contour()	等高线图形绘制
contour3()	三维等高线绘制
contourc()	等高线绘图计算
contrast()	灰度对比度设置
conv()	求多项式乘法的卷积
conv2()	二维卷积
cool	天蓝粉色基色颜色表
copper	
corrcoef()	相关函数系数
cos()	余弦函数
cosh()	双曲余弦
cot()	余切函数
coth()	双曲余切
cov()	协方差矩阵
cplxdemo	复变量承数映射承数演示
cplxpair()	将数据按共轭复数对重新排序
cputime0	所用的 CPU 时间
	余割函数
csch()	双曲余割函数
cusum()	各元素累加积
cylinder()	
date()	日期
dbclear	清除跟踪调试断点
dbcont	跟踪调试恢复执行
dbdown	改变局部工作空间内容
dbquit	退出跟踪调试模式
dbstack	列出函数调用关系
dbstatus	列出所有的断点情况
dbstep	跟踪调试单步执行
dbstop	设置跟踪调试断点
dbtype	列出带有命令行标号的.M 文件
dbup	改变局部工作空间内容
deblank	消除字符串中的空格
dec2hex()	十进制到十六进制的转换
deconv()	因式分解与多项式除法
del2()	离散 Laplace 变换
delete	删除文件
delsqdemo	各种域上的有限差分演示
demo	运行 MATLAB 演示程序
det()	求矩阵的行列式
diag()	建立对角矩阵或获取对角向量
diary	将 MATLAB 运行的命令存盘
diff()	差分函数与近似微分
diffuse()	图像柔焦处理
dir	列出当前目录的内容
disp()	显示矩阵或文本
dmperm()	Dulmage-Mendelsohn 分解
drawnow	刷新绘图指令队列
earthmap	地球拓扑图形的显示
echo	显示文件中的 MATLAB 命令
eig()	求矩阵的特征值与特征向量

函数/会会	<b>       Th能</b>
	对我们你是你们就会是
else	
elseit	与 If 一起使用的转移语句
end	结束控制语句块的命令
eps	浮点相对差限
error()	显示错误信息并中断函数
errorbar()	误差条型图绘制
etree()	矩阵消元树结构
etreeplot()	绘制消元路径
etime()	所用时间的函数
eval()	执行 MATLAB 语句构成的字符串
exist()	检验变量或文件是否已经定义
expm()	矩阵指数函数
expml()	expm()函数的 M 文件实现
expm2()	Taylor 级数求矩阵指数
expm3()	与你有特征向量法求拓防指数
exp()	指数函数
eve()	
falosa()	
foother	
teof()	测试又仵是省结束
terror()	
reval()	执行字符串指定的文件
fft()	离散 Fourier 变换
fft2()	二维离散 Fourier 变换
fftdemo	快速 Fourier 变换演示
ffshift()	去掉谱分析中的直流分量
fegtl()	从文件读入一行数据(忽略换行)
fgets()	从文件读入一行数据(保留换行)
figure()	生成绘图窗口
fill()	绘制充填的二维多边形
fill3()	绘制充填的三维多边形
filter()	一维数字滤波
filter2()	
find()	
findstr()	由一个字符串中查找
finite()	
fitdemo	
fiv()	
fliplr()	
flipud()	
flagr()	按上下力円翻转起阱兀糸 时会工家之白丽敷数
	刈贝尤为方回取登数
Tops()	学点运算计数器
fmin()	▲ 単
fmins()	多变量最优化函数
fopen()	打开文件
for	循环语句
format	
fourier	Fourier 级数展开图形演示
fplot()	给定函数绘图
fplotdemo	函数图形绘制演示
fptintf()	有格式地向文件写入数据
fread()	从文件读入二进制数据
frewind()	将文件指针至文件开头
fscanf()	从文件有格式地读入数据
fseek()	设置文件位置指针
ftell()	获得文件位置指针

承数/命令	功能
full()	由稀疏矩阵变换常规矩阵
function	
funm()	新生产的任金函数
fwrite()	
fzero()	前二近间数据与八天日
gallery()	生成一些小的测试拓阵
gca()	了一 <u>工况。三小小次说这些种</u>
gcf()	获得当前至你福的名称
get()	获得当前因为的窗口 5 倍
getenv	—————————————————————————————————————
getframe()	
ginput()	由鼠标器作图像输入
gllobal	定义全局变量
gplot()	<u> </u>
grav	34件 加度 颜 色 表
graymon()	将图形窗口设置成灰度默认值
grid	给图形加网格线
griddata()	插值用数据网络生成
gradient()	近侧梯度计算
gtext()	在鼠标指定的位置加文字说明
hadamard()	
hankel()	生成 Hankel 矩阵
help	户动联机都助文件显示
hess()	求取 Hessenberg 标准型
hes2num()	十六进制到 IFFF 浮占数的转换
hex2dec()	十六进制到十进制的转换
hidden	网络图陷全线设置开关
hilh()	生成 Hibert 矩阵
hist()	直方图绘制
hold	当前图形保护模式
home	将光标移动到左上角位置
hostid	
hot	型灯带白其色颜色表
hsv	血度物和值(HSV)颜色表
hsv2rgb()	HSV 对 RGB 颜色的结换
if	条件转移语句
ifft()	
ifft2()	□ 维察散 Fourier 逆变换
imag()	求取虑部函数
image	创建图像
Imagedemo	MATLAB4_0版图形处理功能演示
inf	
info	显示 MATLAB 与 MathWorks 信息
input()	带有提示的键盘输入函数
int2str()	整数转换为字符串
interpl()	
interp2()	
interp1()	
intro	MATLAB 引言信息
inv()	
invhilb()	生成逆 Hibert 矩阵
isempty()	若参数为空矩阵,则结果为直
isglobal()	若参数为全局变量则为真
ishold()	若屏幕处于保护状态则为真
isieee()	若有 IEEE 算术标准则为真
isinf()	若参数为 Inf,则结果为真
isletter()	若字符串为字母组成则为直
isnan()	若参数为 NaN,则结果为直

函数/会会	<b> </b>
	—————————————————————————————————————
	石疋阵万怖疏衣小则万具
isstr()	右参数为子符串,则结果为具
jet	HSV色调的变化型
keyboard	启动键盘管理程序
knot	围绕三维结的柱形显示
kron()	Kronecke 乘积函数
lasterr()	查询上的一条错误信息
lengty()	查询向量的维数
life	Conway 生命假设的 MATLAB 版
linspace()	构造线性分布的向量
load	从文件中读如变量
log()	自然对数函数
logl0()	常用对数函数
loglog()	全对数坐标图绘制
logm()	矩阵的对数
logspace0	构造等对数分布的向量
lookfor	对 HELP 信息中的关键词查找
lorenz	Lorenz 混沌吸引子的曲线
lower()	将一个字符串内容转换为小写
lscov()	局小 ^一 乖方差
magic()	
magle()	上
	后
median()	
membrane	产生 MathWorks 公司标志
menu()	
mesh()	三维网格图形
meshc()	带有等高线的网格图形
mgshgrid()	用 x,y 阵列构造三维图形
meshz()	带有零平面的三维网格图形
min()	求向量中最小元素
more	控制命令窗口的输出页面
movie()	播放存储的"电影"幅面
moviein()	初始化 " 电影 " 各幅图像内存
mu21in()	声音文件对线性标度文件的转换
NaN	不定式
nargchk()	函数输入输出参数个数检验
nargin	函数中实际输入变量个数
nargout	函数中实际输出变量个数
newplot	Nextpolt 特性的.M 文件前缀
nextpow2()	找出下一个 2 的指数
nnls()	
nnz()	非零元麦个数
nonzeros()	
norm()	<u>小 9/0 宗</u> 求拓阵的范数
normest()	人名英格兰 名英格兰人名英格兰人名英格兰人名英格兰人名英格兰人名英格兰人名英格兰
null()	七東穴向
num2str()	1177年119
nzmax()	行奴俱积次刀士的甲
ode23()	
ode22p()	
oue25p()	
00045()	佩尔力在尚阶数值解法
odedemo	吊佩分力程通示
ones()	产生元素全部为1的矩阵
orient()	设置打印纸方向
orth	正交空间

	тh能
nack	教理工作空间内左
patch()	—————————————————————————————————————
path	山级填光夕边形绘制函数 设署式杏询 MATLAR 的败汉
paren	及且以旦问 MAILAD 时始任 冬釉长导的本海信自
pascal()	口竹泊与的旦闷后态 生成 Dascal 拓陈
pause()	工从 I ascal 尼件 新启函数
pcolor()	台厅函数
neaks	西峦兽的修住函数演云
penny	個十種币的各个角度加肉
ni	
pink	
piny()	のここにののことで、「のここの」のことののでは、「のここの」のことののことで、「のここの」のでは、「のここの」ので、「のここの」ので、「のここの」ので、「のここの」ので、「のここの」ので、「のここの」ので、「のここの」ので、
plot()	线性坐标图形绘制
plot3()	绘制三维线或占型图形
polar()	极坐标图形绘制
Poly()	求矩阵的特征多项式
polyder()	多项式求导
polyfit()	数据的多项式拟合
polyval()	多项式求值
polyvalm()	
print()	打印图形或将图形存盘
printopt()	
prism	光谱颜色表
prod()	
punct	各种标点符号的查询信息
	矩阵的正交三角化(OR)分解
qrdelete()	QR 分解中删除一列
qrinsert()	QR 分解中插入一列
quad()	低阶数值积分算法
quad8()	高阶数值积分算法
quaddemo	自适应变步长数值演示
quake	Loma Prieta 地震模型
quit	退出 MATLAB 环境
quiver()	箭头图形
qz()	广义特征值问题求解(QZ 算法)
rand()	产生随机矩阵
randn()	产生正态分布随机阵
randperm()	随机置换向量
rank()	求矩阵的秩
rbbox()	擦除框
rcond()	L1NPACK 倒数条件估计
real()	求取实部函数
realmex	最大浮点数值
realmin	最小浮点数值
relop	各种关系符号的查询信息
rem()	除法的余数
reset()	
retur	返回到主调函数的命令
rgb2hsv()	RGB 対 HSV 颜色的转换
rgbplot()	绘制颜色图
roost()	水多坝式的根
rose()	极坐标(角度)直万图绘制
rosser()	典型的对称矩阵特征值问题测试
rot90()	将矩阵兀索旋转 90 度
round()	一
rret()	矩阵的行阶梯型头坝
rretmovie	消元法解万栏过栏演示
rsi2csi()	头状对用阵转移复状对用阵

函数/命令	功能
save	
savis()	高音坐标轴处理
surfnorm()	户目主你神父廷   主面图彭抑范化
schur()	衣田宮形然记10 Selver 公祝
scrint	Schur 万胜
script	MAILAB 诺可及义件信息
sec()	止割凶数
sech()	
semilogx()	x 轴半对数坐标图形绘制
semilogy()	y 轴半对数坐标图形绘制
sepdemo	有限元网格图演示
set()	
setstr()	将数值转换为字符串
shading	阴影模式
sigdemol	离散 Fourier 变换演示
sigdem02	连续 Fourier 变换演示
sign()	符号函数
in()	正弦函数
sinh()	以曲正弦
size()	查询矩阵的维数
slash	
slice()	容量可视图形
sort()	
sound()	
soundemo	
spallog()	MAILAD 时户自功能演示 绘北索二美字位在梯穴间
sparse()	从吊规矩阵转换佈城矩阵
sparsity	柿城矩阵排序效应演示
spaugment()	建立最小
spconvert()	田桸疏矩阵外部格式进行转换
spdiags()	稀疏对角矩阵
specular()	反射
speye()	稀疏单位矩阵
spfunF()	对稀疏矩阵处理的非线性函数
sphere()	产生球面
spinmap()	使颜色旋转
spline2rd	二维样条函数演示
spenes0	将原稀疏矩阵非零元素用1取代
spparms0	设置稀疏矩阵参数
sprank()	结构秩数
sprandn0	稀疏随机矩阵
sprandsym()	稀疏对称随机矩阵
sprintf0	按照 C 语言格式书写字符串
spv()	绘制稀疏矩阵结构
sademo	招一次维面的显示
sqrt()	亚方根函数
sartm0	57.000000000000000000000000000000000000
scon()	
stoirs	22220日日1日15日1日日日日日日日日日日日日日日日日日日日日日日日日日日日
startup	
statup	
stam()	│ 水門里屮合兀糸仦准力左 │
stell()	商取净约附认图形绘制
str2mat0	
str2num()	子付串转换为头型数据
strcmf()	
s trmgs	关于 MATLAB 字符串的帮助信息
subplot	将图形窗口分成若干个区域
subscribe	
subspac()	子空间

函数/命令	功能
sum0	
sunspots	大阳黑子活动模拟
surf()	三维表面图形
surface()	
surfc0	世有等高线的三维表面图形
sum0	
wd()	
symbfacO	
symmed	1950320万計
symrcm()	· 消标取分析次 一 逆
tan().	正扣函数
tanh0	
terminal	□
text()	
title0	/ 方动行我们的留
toenlitz0	失税少役们的留 大成 Tooplitz 拓陈
trace()	主版 Toephiz 把件
tranz()	水龙門凹型     梯形注式物估扣公
treelayout()	
treenayout()	一份1公约9
tril()	四山刀刮焰住的图形   坦亚拓防的工艺会如公
un()	
uicontroll)	
uncontribe()	建立用厂乔闻控制的函数    与次法典文件名处理对关键
uigethe()	你准误盈义什石处理对话性
uisetfont()	你准颜巴说直对话性
uppe + )	
Vander()	主成 vandermonde 起降
Ver	
vibos	业小 MAILAD 版本与
vibes	L 空振汤切画 二维图形现色指定
view()	
viewinitx0	亚小坐你受换起阵 
waterrail()	
what	
which	
which	找击图数与义件所住的日家石
white	個环信円 
whitebg	付图形图山设直风口巴月京
who	
whos	详细列出上作全间受重名
why	
wiikiiison wiahal()	土成 Wilkinson 行任11測试矩阵
xiabei()	│
	[逻辑异 <u>现</u>
ylabel()	
zero&mo	水根演示
zeros()	
zlabel()	│ 给图形加 Z 轴标注