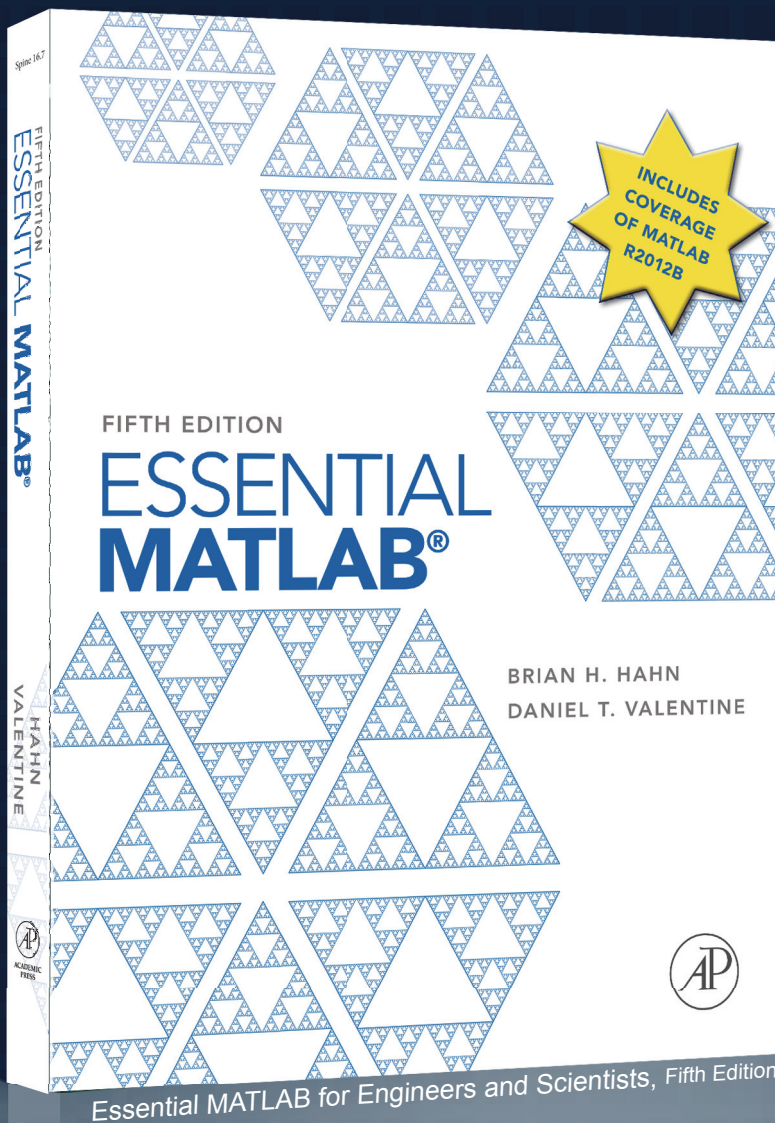


国外计算机科学经典教材

MATLAB原理与应用

(第5版)——工程问题求解与科学计算

[美] Brian H. Hahn Daniel T. Valentine 著
龙伟译



清华大学出版社

国外计算机科学经典教材

MATLAB 原理与应用

(第 5 版)

——工程问题求解与科学计算

[美] Brian H. Hahn 著
Daniel T. Valentine
龙 伟 译

清华大学出版社

北 京

Essential MATLAB for Engineers and Scientists, Fifth Edition

Brian H. Hahn, Daniel T. Valentine

ISBN: 978-0-12-394398-9

Copyright © 2013 by Elsevier. All rights reserved.

Authorized Simplified Chinese translation edition published by the Proprietor.

Copyright © 2014 by Elsevier (Singapore) Pte Ltd.

All rights reserved.

Published in China by Tsinghua University Press under special arrangement with Elsevier (Singapore) Pte Ltd. This edition is authorized for sale in China only, excluding Hong Kong SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书简体中文版由 Elsevier (Singapore) Pte Ltd. 授予清华大学出版社在中国大陆地区(不包括香港、澳门特别行政区以及台湾地区)出版与发行。未经许可之出口, 视为违反著作权法, 将受法律之制裁。

北京市版权局著作权合同登记号: 01-2013-2564

本书封面贴有 Elsevier 防伪标签, 无标签者不得销售。

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

图书在版编目(CIP)数据

MATLAB 原理与应用——工程问题求解与科学计算: 第 5 版 / (美) 哈恩(Hahn, B. H.), (美) 瓦伦丁(Valentine, D.T.) 著; 龙伟 译. —北京: 清华大学出版社, 2014

书名原文: Essential MATLAB for Engineers and Scientists, Fifth Edition

(国外计算机科学经典教材)

ISBN 978-7-302-37501-2

I. ①M… II. ①哈… ②瓦… ③龙… III. ①Matlab 软件—教材 IV. ①TP317

中国版本图书馆 CIP 数据核字(2014)第 176168 号

责任编辑: 王 军 李维杰

装帧设计: 思创景点

责任校对: 曹 阳

责任印制:

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

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

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

印 刷 者:

装 订 者:

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 22.5 字 数: 547 千字

版 次: 2014 年 8 月第 1 版 印 次: 2014 年 8 月第 1 次印刷

印 数: 1~4000

定 价: 58.00 元

产品编号:

出版说明

近年来，我国的高等教育特别是计算机学科教育，进行了一系列大的调整和改革，亟需一批门类齐全、具有国际先进水平的计算机经典教材，以适应我国当前计算机科学的教学需要。通过使用国外优秀的计算机科学经典教材，可以了解并吸收国际先进的教学思想和教学方法，使我国的计算机科学教育能够跟上国际计算机教育发展的步伐，从而培养出更多具有国际水准的计算机专业人才，增强我国计算机产业的核心竞争力。为此，我们从国外多家知名的出版机构 Pearson、McGraw-Hill、John Wiley & Sons、Springer、Cengage Learning 等精选、引进了这套“国外计算机科学经典教材”。

作为世界级的图书出版机构，Pearson、McGraw-Hill、John Wiley & Sons、Springer、Cengage Learning 通过与世界级的计算机教育大师携手，每年都为全球的计算科学高等教育奉献大量的优秀教材。清华大学出版社和这些世界知名的出版机构长期保持着紧密友好的合作关系，这次引进的“国外计算机科学经典教材”便全是出自上述这些出版机构。同时，为了组织该套教材的出版，我们在国内聘请了一批知名的专家和教授，成立了专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动，各位委员根据国内外高等院校计算机及相关专业的现有课程体系，并结合各个专业的培养方向，从上述这些出版机构出版的计算机系列教材中精心挑选针对性强的题材，以保证该套教材的优秀性和领先性，避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量，我们为这套教材配备了一批经验丰富的编辑、排版、校对人员，制定了更加严格的出版流程。本套教材的译者，全部由对应专业的高校教师或拥有相关经验的 IT 专家担任。每本教材的责编在翻译伊始，就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华，在经过翻译、排版和传统的三审三校之后，我们还请编审委员或相关的专家教授对文稿进行审读，以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限，该套教材在出版过程中很可能还存在一些遗憾，欢迎广大师生来电来信批评指正。同时，也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材，共同为我国高等院校计算机教育事业贡献力量。

清华大学出版社

国外计算机科学经典教材

编审委员会

主任委员：

孙家广 清华大学教授

副主任委员：

周立柱 清华大学教授

委员(按姓氏笔画排序)：

王成山	天津大学教授
王 珊	中国人民大学教授
冯少荣	厦门大学教授
冯全源	西南交通大学教授
刘乐善	华中科技大学教授
刘腾红	中南财经政法大学教授
吉根林	南京师范大学教授
孙吉贵	吉林大学教授
阮秋琦	北京交通大学教授
何 晨	上海交通大学教授
吴百锋	复旦大学教授
李 彤	云南大学教授
沈钧毅	西安交通大学教授
邵志清	华东理工大学教授
陈 纯	浙江大学教授
陈 钟	北京大学教授
陈道蓄	南京大学教授
周伯生	北京航空航天大学教授
孟祥旭	山东大学教授
姚淑珍	北京航空航天大学教授
徐佩霞	中国科学技术大学教授
徐晓飞	哈尔滨工业大学教授
秦小麟	南京航空航天大学教授
钱培德	苏州大学教授
曹元大	北京理工大学教授
龚声蓉	苏州大学教授
谢希仁	中国人民解放军理工大学教授

译者序

MATLAB是由美国 MathWorks 公司推出的一款数学类科技应用软件,其名称 MATLAB 是 Matrix Laboratory 的缩写,主要是因为它的一大特性就是以矩阵为基本数据单位,可以高效地完成各种矩阵运算。此外,其编程语言的语法和表达式与数学、工程中常用的形式十分相似,所以使用 MATLAB 来计算各种科技和工程问题要比用 C 之类的语言快捷直观得多。MATLAB 集诸多强大功能于一体,其所附带的工具箱支持 30 多个领域的计算、仿真等应用,涵盖工程计算、控制设计、信号处理、图像处理、信号检测、金融建模设计与分析等多个领域。可以说, MATLAB 是科技工作者攀登科学高峰的必备工具。

《MATLAB 原理与应用——工程问题求解与科学计算》是一本经典的 MATLAB 入门书籍,目前已经是第 5 版。本书整体风格通俗易懂,假定读者是初学者,甚至可以没有任何计算机编程经验。整体内容分为两部分:基础知识和实践应用。第 I 部分主要介绍 MATLAB 的基本语法和编程规范,第 II 部分通过一些具体问题的求解过程来展示利用 MATLAB 解决科学和工程问题的方法。本书非常重视动手能力的培养,提供大量的实例和习题,鼓励读者在实践的过程中发现问题和加深理解。

在本书的翻译过程中,清华大学出版社的编辑们给予了译者很大的帮助和鼓励,在这里首先要感谢他们的辛勤工作。此外,周英姿、靳晓辉、任远、谭方青、李荣生等人也参与了本书部分内容的翻译工作,在此也对他们表示感谢。限于水平和时间原因,书中难免有遗漏之处,敬请读者批评指正。

译者



致 谢

感谢 Mary、Clara 和 Zash 的支持，并将《MATLAB 原理与应用(第 5 版)——工程问题求解与科学计算》献给他们。

Daniel T. Valentine

前言

撰写《MATLAB 原理与应用(第 5 版)——工程问题求解与科学计算》的主要目的是为了跟上 MATLAB 的升级进度(最新版本是 R2012b)。与之前的版本一样,这一版也是将 MATLAB 作为解决问题的工具介绍给没有计算机编程经验的科学家、工程师以及相关领域的学生们。

为了与已故的作者 Brian D. Hahn 在之前版本中的目标保持一致,本书的第 5 版采用一种通俗的指南风格来实现“自学”的学习方法,读者将在使用 MATLAB 做实验的过程中掌握它的工作原理。本书假设读者在解决技术问题时从未使用过该工具。

MATLAB 是 Matrix Laboratory 的缩写,它是基于矩阵的概念来实现的。由于读者可能对矩阵并不熟悉,我们将根据上下文的需要逐步介绍矩阵的概念和结构。本书的主要读者是科学家和工程师,因此书中的示例(尤其是第 II 部分)都需要一些大学一年级的数学知识。但是,这些示例都是独立的,读者可以选择性地阅读,并不会影响读者编程技能的提升。可以通过两种不同的模式来使用 MATLAB。一种是在迫切需要得到即时的结果时,可以在 Command Window 中立即执行语句(或语句组)。另一种是在不那么迫切的情况下,可以利用脚本文件的方式来提供传统的编程方式。读者可以通过如下方法对两种模式善加利用:鼓励在前一种模式中使用剪切和粘贴的方式,以充分利用 Windows 操作系统中的交互环境;后者通过结构规划来强调编程原则和算法开发。

虽然本书的内容涵盖 MATLAB 的大部分基本(基础)特性,但它既不是一本完备的书籍,也不是一本系统的参考工具书。因为这和它通俗的风格不统一。例如,我们在开始介绍 for 和 if 结构的时候,和很多其他书本不同,并不总是使用它们的通用格式,而是在适当的地方自然而然地引入。即便如此,我们仍对 for 和 if 结构进行了透彻而全面的介绍,而不像其他书本只对其进行粗略介绍。如果读者想了解,可以在附录中找到实用的语法和函数快速索引。如下列表包含本书第 5 版的其他重点内容:

- 对于粗心的初学者可能碰到的很多陷阱给出警告。
- 给出大量来自科学技术(仿真、种群建模和数值方法)、商业和日常生活的示例。
- 强调编程风格的重要性,以生成清晰、可读性强的代码。
- 全面的章节总结。
- 丰富的章节练习(我们在附录中给出了很多练习的答案和解法)。

本书应该和 MATLAB 软件结合使用。所以读者需要先安装该软件，完成书中的练习，从而理解 MATLAB 是如何完成任务的。任何工具都只能通过动手练习的方法来学习。计算工具尤其如此，因为它们只有在接收到的命令和相应的数据正确且精确时，才能输出正确的答案。

目 录

第 I 部分 基础知识	
第 1 章 引言	3
1.1 使用 MATLAB.....	4
1.1.1 算术.....	5
1.1.2 变量.....	6
1.1.3 数学函数.....	7
1.1.4 函数和命令.....	7
1.1.5 向量.....	7
1.1.6 线性方程组.....	9
1.1.7 教程和演示.....	10
1.2 Desktop.....	11
1.2.1 使用 Editor 和运行脚本.....	12
1.2.2 帮助、发布和视图.....	14
1.2.3 符号与 MuPAD Notebook APP (记事本应用程序).....	17
1.2.4 其他 APP.....	20
1.2.5 附加特性.....	20
1.3 示例程序.....	22
1.3.1 剪切和粘贴.....	22
1.3.2 保存程序：脚本文件.....	23
1.3.3 程序实战.....	25
1.4 本章小结.....	26
1.5 本章练习.....	26
第 2 章 MATLAB 基础	27
2.1 变量.....	27
2.2 工作空间.....	28
2.3 阵列：向量与矩阵.....	29
2.3.1 初始化向量：显式列表.....	29
2.3.2 初始化向量：冒号运算符.....	31
2.3.3 linspace 和 logspace 函数.....	31
2.3.4 转置向量.....	32
2.3.5 下标.....	32
2.3.6 矩阵.....	32
2.3.7 捕获输出.....	33
2.3.8 结构规划.....	34
2.4 重力作用下的垂直运动.....	35
2.5 运算符、表达式和语句.....	36
2.5.1 数字.....	37
2.5.2 数据类型.....	37
2.5.3 算术运算符.....	37
2.5.4 运算符的优先级.....	38
2.5.5 冒号运算符.....	39
2.5.6 转置运算符.....	39
2.5.7 阵列算术运算.....	39
2.5.8 表达式.....	40
2.5.9 语句.....	41
2.5.10 语句、命令和函数.....	42
2.5.11 公式向量化.....	42
2.6 输出.....	45
2.6.1 disp 语句.....	45
2.6.2 format 命令.....	46
2.6.3 比例因子.....	47
2.7 for 循环.....	48
2.7.1 用牛顿法计算平方根.....	48
2.7.2 阶乘!.....	49

2.7.3	数列的极限	49	5.1.1	不连续图	96
2.7.4	基本 for 结构	50	5.1.2	避免除零	97
2.7.5	单行中的 for 语句	51	5.1.3	避免无穷	97
2.7.6	更加一般化的 for 语句	51	5.1.4	对随机数进行计数	98
2.7.7	通过向量化来避免使用 for 循环	52	5.1.5	掷骰子	99
2.8	判断	54	5.2	逻辑运算符	100
2.8.1	单行 if 语句	54	5.2.1	运算符的优先级	101
2.8.2	if-else 结构	56	5.2.2	危险	101
2.8.3	单行 if-else 语句	57	5.2.3	逻辑运算符和向量	102
2.8.4	elseif	57	5.3	将逻辑向量作为下标	102
2.8.5	逻辑运算符	58	5.4	逻辑函数	104
2.8.6	多个 if 与 elseif 的对比	59	5.5	用逻辑向量代替 elseif 阶梯	105
2.8.7	嵌套 if	60	5.6	本章小结	107
2.8.8	将 if 向量化?	60	5.7	本章练习	107
2.8.9	switch 语句	60	第 6 章	矩阵和阵列	111
2.9	复数	61	6.1	矩阵	111
2.10	本章小结	63	6.1.1	具体示例	111
2.11	本章练习	64	6.1.2	创建矩阵	113
第 3 章	程序设计与算法开发	71	6.1.3	下标	113
3.1	程序设计流程	72	6.1.4	转置	113
3.2	MATLAB 函数编程	79	6.1.5	冒号运算符	114
3.2.1	内联对象: 谐振子	79	6.1.6	复制行和列	116
3.2.2	MATLAB 函数: $y=f(x)$	81	6.1.7	删除行和列	117
3.3	本章小结	82	6.1.8	初等矩阵	118
3.4	本章练习	83	6.1.9	特殊矩阵	119
第 4 章	MATLAB 函数与数据导入 导出工具	85	6.1.10	对矩阵使用 MATLAB 函数	119
4.1	常用函数	85	6.1.11	操纵矩阵	120
4.2	导入和导出数据	90	6.1.12	对矩阵进行阵列(元素对元素) 运算	120
4.2.1	load 和 save 命令	90	6.1.13	矩阵和 for 循环	121
4.2.2	导出文本(ASCII)数据	90	6.1.14	矩阵的可视化	121
4.2.3	导出文本(ASCII)数据	91	6.1.15	将嵌套 for 循环向量化: 贷款偿还表格	121
4.2.4	导出二进制数据	91	6.1.16	多维阵列	124
4.3	本章练习	92	6.2	矩阵运算	124
第 5 章	逻辑向量	95	6.2.1	矩阵乘法	124
5.1	示例	96	6.2.2	矩阵求幂运算	126

6.3	其他矩阵函数	126	8.2.2	while 语句	161
6.4	种群增长: 莱斯利矩阵	126	8.2.3	投资翻倍的时间	161
6.5	马尔可夫过程	129	8.2.4	质数	162
6.6	线性方程	131	8.2.5	抛射体轨迹	163
6.6.1	MATLAB 中的解法	132	8.2.6	break 和 continue 语句	165
6.6.2	残量	133	8.2.7	目录	165
6.6.3	超定方程组	133	8.3	本章小结	166
6.6.4	欠定方程组	134	8.4	本章练习	166
6.6.5	病态	134			
6.6.6	矩阵除法	135	第 9 章	MATLAB 图形	171
6.7	稀疏矩阵	136	9.1	基本二维图形	171
6.8	本章小结	138	9.1.1	标签	172
6.9	本章练习	138	9.1.2	在相同的坐标轴上绘制多个图形	173
第 7 章	函数 M-文件	141	9.1.3	线型、标记和颜色	173
7.1	示例: 再看牛顿法	141	9.1.4	坐标轴限制	174
7.2	基本规则	143	9.1.5	在一幅图中绘制多个图形: subplot	175
7.2.1	子函数	147	9.1.6	figure、clf 和 cla 函数	176
7.2.2	私有函数	147	9.1.7	图形输入	176
7.2.3	P-code 文件	147	9.1.8	对数作图	176
7.2.4	使用分析工具提高 M-文件的性能	147	9.1.9	极坐标作图	177
7.3	函数句柄	148	9.1.10	绘制快速变化的数学函数: fplot	178
7.4	命令/函数对偶性	149	9.1.11	属性编辑器	179
7.5	函数名解析	150	9.2	三维作图	179
7.6	调试 M-文件	150	9.2.1	plot3	179
7.6.1	调试脚本文件	150	9.2.2	使用 comet3 绘制三维动画	180
7.6.2	调试函数	152	9.2.3	网面	180
7.7	递归	152	9.2.4	等高线图	182
7.8	本章小结	153	9.2.5	使用 NaN 剪切曲面	183
7.9	本章练习	154	9.2.6	可视化向量场	183
第 8 章	循环	157	9.2.7	矩阵的可视化	184
8.1	使用连续 for 语句的确定循环	157	9.2.8	三维图形的旋转	185
8.1.1	二项式系数	157	9.3	句柄图形	186
8.1.2	更新过程	158	9.3.1	获得句柄	186
8.1.3	嵌套 for 语句	159	9.3.2	图形对象的属性和修改方法	187
8.2	使用连续 while 语句的不确定循环	160	9.3.3	句柄向量	188
8.2.1	猜谜游戏	160	9.3.4	图形对象创建函数	189

9.3.5 指定父对象·····	189	第 11 章 错误和陷阱·····	221
9.3.6 定位图形·····	190	11.1 语法错误·····	221
9.4 编辑绘图·····	191	11.1.1 向量的大小不匹配·····	222
9.4.1 绘图编辑模式·····	191	11.1.2 名称屏蔽·····	222
9.4.2 属性编辑器·····	191	11.2 逻辑错误·····	222
9.5 动画·····	192	11.3 舍入误差·····	223
9.6 颜色等属性·····	195	11.4 本章小结·····	224
9.6.1 色图·····	195	11.5 本章练习·····	224
9.6.2 曲面绘图的颜色·····	196	第 II 部分 实践应用	
9.6.3 Truecolor(真彩)·····	197	第 12 章 动力系统·····	227
9.7 光照和镜头·····	198	12.1 悬臂梁·····	228
9.8 保存、打印和导出图形·····	198	12.2 电流·····	230
9.8.1 保存和打开图像文件·····	198	12.3 自由落体·····	232
9.8.2 打印图形·····	199	12.4 摩擦力作用下的投射体	
9.8.3 导出图形·····	199	问题·····	239
9.9 本章小结·····	199	12.5 本章小结·····	241
9.10 本章练习·····	200	12.6 本章练习·····	242
第 10 章 作为阵列的向量以及其他		第 13 章 仿真·····	243
数据结构·····	203	13.1 随机数的生成·····	243
10.1 更新过程·····	203	13.2 旋转硬币·····	244
10.1.1 单位时间步长·····	204	13.3 投掷骰子·····	245
10.1.2 非单位时间步长·····	206	13.4 细菌分裂·····	245
10.1.3 使用函数·····	207	13.5 随机游走·····	246
10.1.4 精确解·····	208	13.6 交通流量·····	247
10.2 频率、柱状图和直方图·····	209	13.7 正态(高斯)随机数·····	250
10.2.1 随机漫步·····	209	13.8 总结·····	250
10.2.2 直方图·····	210	13.9 本章练习·····	250
10.3 排序·····	211	第 14 章 数值方法入门·····	253
10.3.1 冒泡排序·····	211	14.1 方程组·····	253
10.3.2 MATLAB 中的 sort 函数·····	212	14.1.1 牛顿法·····	253
10.4 结构体·····	213	14.1.2 二分法·····	255
10.5 元胞阵列·····	215	14.1.3 fzero 函数·····	257
10.5.1 将数据赋给元胞阵列·····	215	14.1.4 roots 函数·····	257
10.5.2 访问元胞阵列中的数据·····	216	14.2 积分·····	257
10.5.3 使用元胞阵列·····	217	14.2.1 梯形法则·····	257
10.5.4 显示和可视化元胞阵列·····	218	14.2.2 辛普森法则·····	258
10.6 类和对象·····	218	14.2.3 quad 函数·····	259
10.7 本章小结·····	218		

14.3	数值微分	259	16.4	杜芬振子	294
14.4	一阶微分方程	260	16.5	本章练习	295
14.4.1	欧拉方法	261	第 17 章	Symbolic 工具箱	297
14.4.2	示例: 细菌生长	261	17.1	代数	298
14.4.3	另一种下标表示法	263	17.1.1	多项式	298
14.4.4	预估-校正法	264	17.1.2	向量	300
14.5	线性常微分方程(LODE)	264	17.1.3	矩阵	301
14.6	龙格-库塔法	264	17.2	微积分	304
14.6.1	单独的微分方程	265	17.3	拉普拉斯变换和 Z 变换	305
14.6.2	差分方程组: 混沌	265	17.4	广义函数*	307
14.6.3	将额外参数传递给 ODE 求解程序	268	17.5	微分方程	308
14.7	偏微分方程	269	17.6	funtool、MuPAD 和帮助文档 的使用	309
14.8	其他数值方法	271	17.6.1	funtool	309
14.9	本章小结	273	17.6.2	MuPAD 记事本和帮助 文档	309
14.10	本章练习	273	17.7	本章练习	311
第 15 章	信号处理	277	附录 A	语法快速参考	313
15.1	谐波分析	278	附录 B	命令与函数快速参考	317
15.2	快速傅里叶变换(FFT)	281	附录 C	ASCII 字符编码表	327
第 16 章	Simulink 工具箱	285	附录 D	部分练习的答案	329
16.1	弹簧-质块-阻尼器动力 系统	290			
16.2	弹跳球动力系统	291			
16.3	范德波尔振子	293			

第

I

部分

基础知识

第 I 部分主要涉及 MATLAB 和科技计算中读者所需要掌握的基础知识。因为这本书是一本指南，所以作者鼓励大家在阅读本书时，广泛地使用 MATLAB 中的各种功能。

引言

本章目标:

- 学会在 Command Window (指令窗口)中使用一些简单的 MATLAB 命令
- 介绍 MATLAB 的各种操作桌面和编辑特性
- 学习 MATLAB R2012b Desktop(操作桌面)的一些新特性
- 学会在编辑器中编写并运行脚本
- 学习一些与标签相关联的新特性(特别是 PUBLISH 和 APPS)

MATLAB 是一个强大的处理科学和工程计算问题的科技计算系统。名称 MATLAB 是 Matrix Laboratory (矩阵实验室)的缩写,这是因为设计者的目的是为了极大地简化矩阵计算。矩阵是按 m 行、 n 列排列的数字阵列。下面这个 $m \times n = 2 \times 3$ 的阵列就是一个例子:

$$A = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

借由表示元素位置的行和列的索引号,我们可以摘选出矩阵中的任何元素。本例中的元素可以摘选如下: $A(1,1)=1$ 、 $A(1,2)=3$ 、 $A(1,3)=5$ 、 $A(2,1)=2$ 、 $A(2,2)=4$ 、 $A(2,3)=6$ 。第一个索引是指从上至下计数的行号,第二个索引是指从左至右计数的列号。这是 MATLAB 中定位阵列信息的习惯用法。计算机擅长快速地进行大量计算,因此对于以阵列或矩阵的形式排列的大型数据集来说,其运算非常有效。

本书假设你以前从未使用计算机做过这类由 MATLAB 处理的科学计算,但是能够熟练操作计算机键盘,并了解你所使用的操作系统(例如 Windows、UNIX 或 MAC-OS)。除此之外,你唯一需要掌握的和计算机相关的技能就是一些非常基础的文本编辑。

MATLAB 有许多令人喜爱的特性,其中之一就是可以在使用过程中和它交互(这也是它有别于许多其他计算机编程系统的重要特征,例如 C++和 Java)。这意味着在 MATLAB 特殊的提示符后输入某些命令,就会立即得到结果。我们既可以用这种方式来解决非常简单的问题,比如求平方根,也可以解决非常复杂的问题,比如求解微分方程组。对于许多科学技术问题,你只要输入一个或两个命令——MATLAB 就会为你做大部分的工作。

要学会应用 MATLAB,有如下三个基本要求:

- 必须掌握编写 MATLAB 语句和使用 MATLAB 工具的准确规则。
- 必须了解与所要解决问题相关的数学知识。
- 必须制定合理的作战计划——算法——来解决特定的问题。

本章主要致力于第一个要求：学习一些基本的 MATLAB 规则。计算机编程是一门精确的科学(也有人说是一门艺术)；你必须严格地以正确的方式输入语句。在计算机程序员之间有一种说法：错进，错出。意思是如果你给 MATLAB 提供错误的指令，就将得到错误的结果。

随着经验的积累，你将能够设计、开发和实现计算及图形工具来解决相对复杂的科学和工程问题。你还能调整 MATLAB 的外观，修改与之交互的方式，开发一个你自己的工具箱，帮助你解决感兴趣的问题。换句话说，你可以根据经验定制 MATLAB 工作环境。

在学习 MATLAB 的基础知识，或者任何其他计算机工具时，切记：应用程序不会无缘无故地做任何事情。因此，当你使用 MATLAB 时，注意观察和研究它对你所执行的命令行操作的所有响应，以了解该工具的功能和应用范围。在研究 MATLAB 功能的初始阶段，由于我们的目的是评估该工具及其功能，我们将求解一些已知答案的相对简单的问题。这总是第一步。在学习 MATLAB 时，你还将学习编程，1) 创建自己的计算工具，2) 领会设计高效、鲁棒和精确的计算和图形工具(即计算机程序)的困难所在。

在本章的其余部分，我们将考虑一些简单的示例。即使你并未完全理解也不用担心。随着后续章节的深入学习，你将逐渐领悟。实践练习对于了解 MATLAB 的运行机制是非常重要的。在领会本章所述的基本规则之后，你就可以掌握下一章和 MATLAB 提供的帮助文件中的很多其他功能了。这将有助于你去解决更有趣和更具实质性的问题。在本章的最后一节中，你将快速浏览一下 MATLAB 桌面。

1.1 使用 MATLAB

请确认你已经在计算机上安装了 MATLAB，或者你能够访问一个可以使用 MATLAB 的网络。本书基于撰写时的 MATLAB 最新版本(R2012b 版)¹。

要从 Windows 启动，双击 Windows 桌面上的 MATLAB 图标。要从 UNIX 启动，请在操作系统的提示符后输入 matlab。要从 MAC-OS 启动，请打开 X11(即打开一个 X-终端窗口)，然后在提示符后输入 matlab。MATLAB 桌面打开后如图 1-1 所示。现在我们看到的桌面上的窗口是 Command Window，特殊的提示符>>出现在其中。此提示符表示 MATLAB 正在等待命令。你可以在任何时候以如下方式退出：

- 单击 MATLAB 桌面右上角的×(关闭框)。
- 在 Command Window 中的提示符后输入 quit 或 exit，然后按下 Enter 键。

MATLAB 在启动时会自动地在用户的 Document Folder (文档文件夹)中创建一个名为 MATLAB 的文件夹。这个特性非常方便，因为它是默认的工作文件夹。从 Command Window

1. R2012a 版也适用于本书所述的练习和问题。与以前发布的版本相比，R2012b 版有一些新的特性。幸运的是，使用以前版本开发的内容在最新版本中基本上能够以同样的方式工作。最新版本中包含的新功能，增强了 MATLAB 的易用性，因此提高了用户的工作效率。

保存的所有东西都将保存在该文件夹中。现在可以在 MATLAB 的 Command Window 中进行试验。如有必要，将光标移到 Command Window 并在窗口范围内的任意位置单击鼠标左键以激活 Command Window。

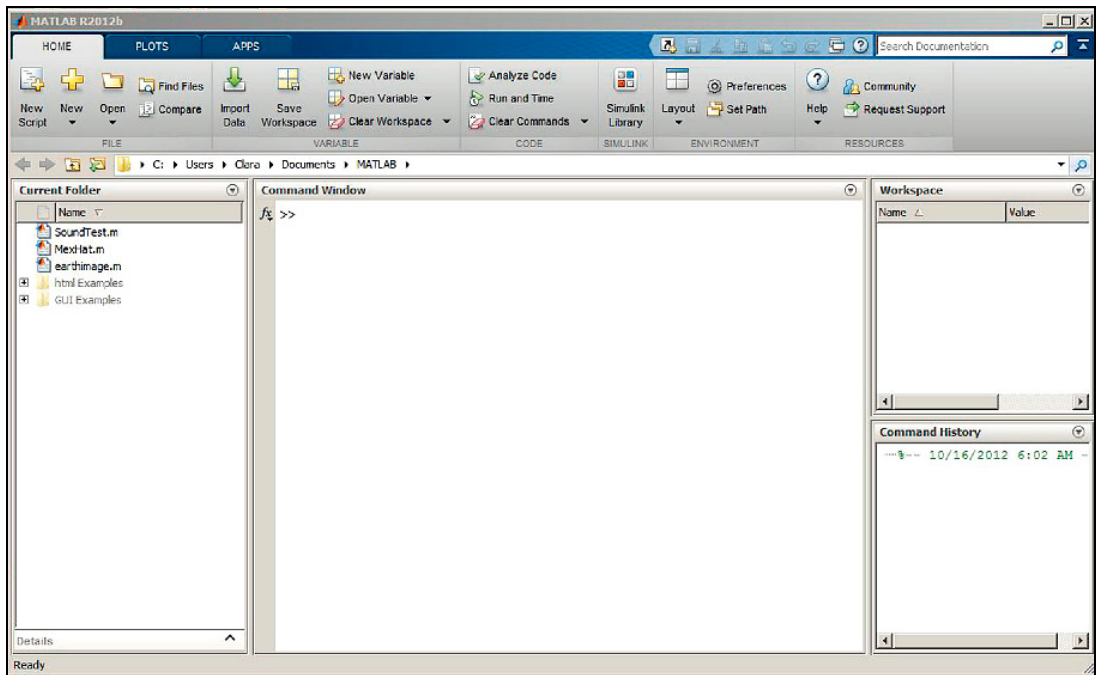


图 1-1 MATLAB 桌面

1.1.1 算术

由于熟悉算术，因此这里就用算术来检验 MATLAB 是否能正确地进行运算。这对于我们获取对使用任何工具的信心来说，是个必需的步骤。

在提示符`>>`后输入 `2+3`，然后按 Enter 键，在命令行中显示为`<Enter>`：

```
>> 2+3 <Enter>
```

命令只有在输入后才会被执行。在这种情况下，答案当然是 5。再试试：

```
>> 3-2 <Enter>
>> 2*3 <Enter>
>> 1/2 <Enter>
>> 2^3 <Enter>
>> 2\11 <Enter>
```

对于`(1)/(2)`和`(2)^(3)`会有怎样的结果？你知道符号`*`、`/`和`^`的意思吗？是的，它们是乘法、除法和乘方。反斜杠表示分母是在符号的左边，分子是在右边；最后一个命令的结果是 5.5。该运算等同于 $11/2$ 。

现在输入以下命令：

```
>> 2.*3 <Enter>
>> 1./2 <Enter>
>> 2.^3 <Enter>
```

在*、/和^的前面分别有一个句点，这不会改变运算的结果，因为这里的乘法、除法和乘幂都是针对单一数字进行的(我们将在之后处理数字阵列时解释为什么需要这些符号)。

这里有一些关于创建和编辑命令行的提示：

- 包含提示符>>的行被称为命令行。
- 在按 Enter 键之前，可以组合使用 Backspace 键、左方向键、右方向键和 Del 键来编辑 MATLAB 命令。我们称这个有用的特性为命令行编辑。
- 可以通过使用上方向键和下方向键来选择(和编辑)输入的命令。记得按 Enter 键来执行命令(即运行或执行命令)。
- MATLAB 有被称为智能召回(smart recall)的实用编辑功能，只需要输入要召回的命令的前几个字符。例如，输入字符 2*，然后按上方向键，就可以召回最近的以 2* 开始的命令。

你想知道 MATLAB 会如何处理 0/1 和 1/0 吗？试试吧。如果坚持在计算中使用 ∞ ，当然这也是合理的要求，输入符号 Inf(infinity 的简写)即可。试试 13+ Inf 和 29/Inf。

另一个你可能会遇到的特殊值是 NaN，这是 Not-a-Number 的缩写。诸如 0/0 之类的计算会得到这样的答案。

1.1.2 变量

现在我们给变量赋值，做变量的算术运算。首先输入命令(在编程术语中称之为语句)a=2。MATLAB 命令行应该如下所示：

```
>> a = 2 <Enter>
```

a 是变量。这条语句将数值 2 赋予变量 a(注意，此值在执行该语句后会立即显示)。现在尝试输入语句 a = a+7，然后在新的一行输入 a = a*10。你会认同 a 的最终值吗？我们是否认同它就是 90 呢？

现在输入语句：

```
>> b = 3; <Enter>
```

分号(;)将会阻止 b 值的显示。然而，b 仍然被赋予数值 3，可以通过输入不带分号的 b 看到 b 的值：

```
>> b <Enter>
```

将任何数值赋予两个变量 x 和 y。现在看看你是否可以在一条语句中将 x 与 y 的和赋予第三个变量 z。实现该赋值操作的一种方式：

```
>> x = 2; y = 3; <Enter>
```

```
>> z = x + y <Enter>
```

请注意上面的命令，除了执行赋值变量的算术运算之外，分号(或逗号)分隔的多个命令还可以放在同一行中。

1.1.3 数学函数

对于能在科学电子计算器上找到的常用数学函数，MATLAB中都有对应的函数，如 `sin`、`cos` 和 `log`(意为自然对数)。更多的示例见附录 C.5 节。

- 用命令 `sqrt(pi)` 求解 $\sqrt{\pi}$ 。答案应该是 1.7725。需要注意的是，由于 `pi` 是 MATLAB 的众多内置函数之一，因此 MATLAB 知道 `pi` 的值。
- 诸如 `sin(x)` 之类的三角函数的输入参数 `x` 为弧度。角度数乘以 $\pi/180$ 可以得到弧度。例如，使用 MATLAB 计算 `sin(90)`。答案应该是 `1(sin(90* pi/180))`。
- 在 MATLAB 中，用 `exp(x)` 来计算指数函数 e^x 。请据此求解 e 和 $1/e$ (2.7183 和 0.3679)。由于 MATLAB 中有大量的像 `pi` 或 `sin` 这样的内置函数，请务必注意用户自定义变量的命名。如无很大必要，变量名尽量不要与这些内置函数重复。这个问题可以举例说明如下：

```
>> pi = 4 <Enter>
>> sqrt(pi) <Enter>
>> whos <Enter>
>> clear pi <Enter>
>> whos <Enter>
>> sqrt(pi) <Enter>
>> clear <Enter>
>> whos <Enter>
```

需要注意的是，单独执行 `clear` 命令会清除工作区中的所有局部变量；`>>clear pi` 则只会清除局部变量 `pi`。换句话说，如果决定重新定义内置函数或命令，它们就会使用新的数值！执行 `whos` 命令，可以确定当前工作区中局部变量或命令的清单。执行上面示例中的第一条命令 `pi=4`，结果将显示出对内置 `pi` 的重新定义：一个 1×1 的双精度类型的数组。这意味在赋予 `pi` 数值时，就创建了这种数据类型(随着我们对 MATLAB 的继续研究，你将学习更多其他数据类型)。

1.1.4 函数和命令

MATLAB 有大量的通用函数。先试试 `date` 和 `calendar` 函数。它也有许多的命令，如 `clc`(*clear command window* 的缩写)，`help` 是一个你将会大量使用的命令(见下文)。函数和命令的不同之处在于，函数通常会返回一个值(如日期)，而命令倾向于以某种方式改变环境(如清理屏幕或将语句保存到工作区)。

1.1.5 向量

诸如 1.1.2 节中使用的 `a` 和 `b` 之类的变量被称为标量(*scalar*)，它们是单个的数值。MATLAB 还会处理向量(*vector*，通常被称为数组)，这是它具有许多强大功能的关键所在。

定义一个向量，其中的元素(组件)以相同的数量递增，最简单的方法是使用如下语句：

```
>> x = 0 : 10; <Enter>
```

在 0 和 10 之间有一个冒号(:)。在冒号的两侧留空格只是为了让程序更具可读性。可以看出， x 是一个向量，它是一个由 1 行和 11 列组成的行向量。输入以下命令可以验证它确实如此：

```
>> size(x) <Enter>
```

依照刚才定义的向量 x ，我们还可以定义(或创建)其他向量，这展现出 MATLAB 的强大功能。请尝试：

```
>> y = 2.* x <Enter>
```

```
>> w = y./ x <Enter>
```

和

```
>> z = sin(x) <Enter>
```

(没有分号)。注意，第一个命令行将因子 2 与 x 的每个元素相乘，产生向量 y 。第二个命令行是数组运算，通过将 y 的每一个元素除以 x 中的对应元素，产生向量 w 。由于 y 的每个元素是 x 相应元素的两倍，向量 w 是一个 11 个元素都等于 2 的行向量。最后， z 是一个以 $\sin(x)$ 为元素的向量。

要绘制出一幅 $\sin(x)$ 函数的漂亮图像，只需要输入以下命令：

```
>> x = 0: 0.1: 10; <Enter>
```

```
>> z = sin(x); <Enter>
```

```
>> plot(x,z), grid <Enter>
```

这幅图像会出现在一个单独的图像窗口中(见图 1-2)。可以通过单击 Command Window 或图像窗口里的任何地方，对它们进行选择。Windows 下拉菜单在两种窗口中都可以使用。

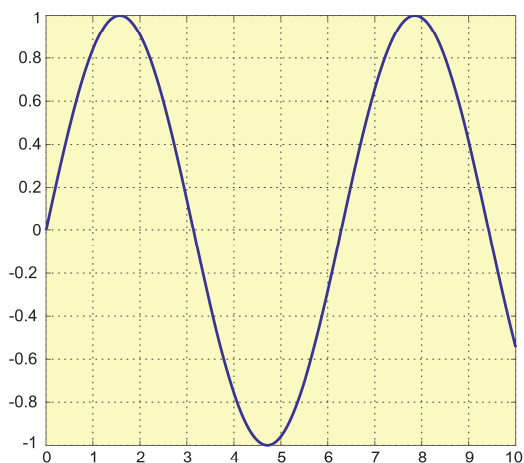


图 1-2 图像窗口

需要注意的是，上面的第一个命令行的等号后有3个数字。当3个数字以这种方式被两个冒号分开时，中间的数字代表递增的增量。选择增量为0.1，是为了给出一张平滑的图。最后一个命令行的逗号后的 `grid` 命令，是为了给图形增加网格背景(在图像窗口中，通过使用图像属性 `Editor`，可以改变背景色，它可以在工具栏的 `Edit` 下拉菜单中找到。本书中图像的颜色已经图像编辑工具处理过)。

如果想看到正弦曲线更多的周期，使用命令行编辑，将 `sin(x)` 变成 `sin(2*x)` 即可。

请尝试在同一定义域绘制一幅 `tan(x)` 函数的图像。你会发现该图像令人惊讶的方面。为了便于检查此函数，可以使用 `axis([0 10 -10 10])` 命令来改进该图形，如下：

```
>> x = 1:0.1:10; <Enter>
>> z = tan(x); <Enter>
>> plot(x, z), axis([0 10 -10 10]) <Enter>
```

使用以下命令是利用图形检查数学函数的另一种方式：

```
>> ezplot('tan(x)') <Enter>
```

在 `ezplot` 命令中，围绕 `tan(x)` 的单引号非常重要。需要注意的是，在 `ezplot` 中，`x` 的默认取值范围不是0到10。

`Tab` 自动补全是 `Command Window` 中一个有用的编辑特性：输入 `MATLAB` 名称的几个字母，然后按 `Tab` 键。如果该名称是唯一的，它会自动补全。如果它不是唯一的，第二次按 `Tab` 键可以看到所有可能的命令。请尝试在命令行中输入 `ta`，然后按 `Tab` 键两次。

1.1.6 线性方程组

线性方程组在工程和科学分析中非常重要。下面是一个简单的求解两个联立方程组的示例：

$$\begin{aligned}x + 2y &= 4 \\ 2x - y &= 3\end{aligned}$$

这里有两种求解方法。

矩阵法。输入以下命令(如实输入)：

```
>> a = [1 2; 2 -1]; <Enter>
>> b = [4; 3]; <Enter>
>> x = a\b <Enter>
```

参考结果是：

```
x =
     2
     1
例如， x = 2, y = 1
```

内置求解函数。输入以下命令(如实输入)：

```
>> [x, y] = solve('x+2*y=4', '2*x - y=3') <Enter>
>> whos <Enter>
```

```
>> x = double(x), y = double(y) <Enter>
>> whos <Enter>
```

`double` 函数将 `x` 和 `y` 从符号对象(MATLAB 中的另一种数据类型)转换为双精度数组(即与赋值数相关联的数值变量数据类型)。

在两种方法执行之后, 输入以下命令(如实输入)以检验结果:

```
>> x + 2*y % should give ans = 4 <Enter>
>> 2*x - y % should give ans = 3 <Enter>
```

%符号是一个标记, 它指示右边显示的所有信息不是命令的一部分, 而是注释(之后当学习开发命令行的编码程序时, 我们将说明注释的必要性)。

1.1.7 教程和演示

如果想要 MATLAB 提供的公开演示实例, 在命令行中输入 `demo` 即可。在输入该命令后, 帮助文档会在 MATLAB Examples 中打开(见图 1-3)。单击“Getting Started”, 你将看到可供随意使用的 MATLAB 应用程序教程和演示清单。单击任何其他主题, 可以学习更多 MATLAB 丰富的功能。你可能希望回顾一些与你正在研究的科学计算需求相关的主题教程。向下滚动到“New Features Video”, 就可以学习更多关于 Desktop 以及其他新特性的内容, 其中一些将在后面介绍。

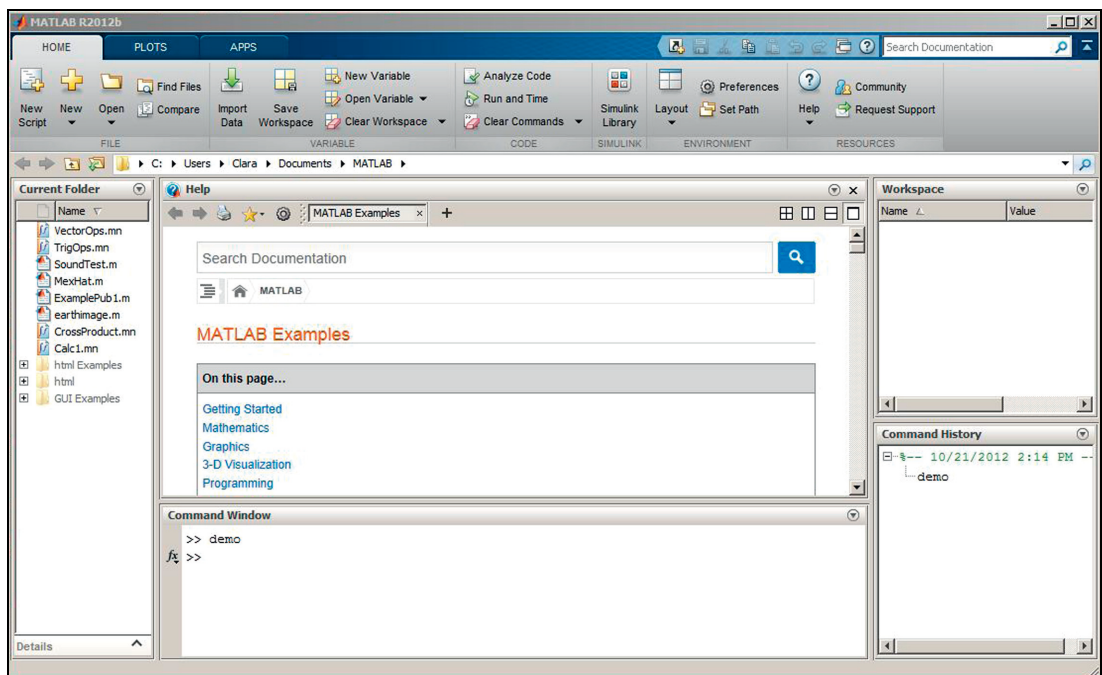


图 1-3 MATLAB 示例的帮助文档

1.2 Desktop

MATLAB R2012b 有个非常有用的特性，当第一次打开它时，它会在你的 Documents 文档中创建名为 MATLAB 的文件夹(如果它还不存在的话)。在第一次执行时，文件夹中没有条目，所以 Current Folder(当前文件夹)面板是空的。在你的 Documents 文档中，这个新的文件夹是默认的工作文件夹，它将保存你创建的所有文件。Command Window 上面的第一个工具栏里给出了这个文件夹的位置。该位置是 C: \Users\Clara\Documents\MATLAB。这个位置的格式是通过用鼠标指针指向并单击 Command Window 正上方的这一行来确定的。

让我们自上而下地看一下 Desktop。在顶行的左边可以看到正在运行的 MATLAB 版本名称。这里的版本是 MATLAB R2012b。在顶行的右侧有三个按钮，它们是：下划线按钮，可以用它将 Desktop 最小化；矩形按钮，可以用它将 Desktop 最大化；×按钮，可以用它将关闭 MATLAB(见图 1-4)。

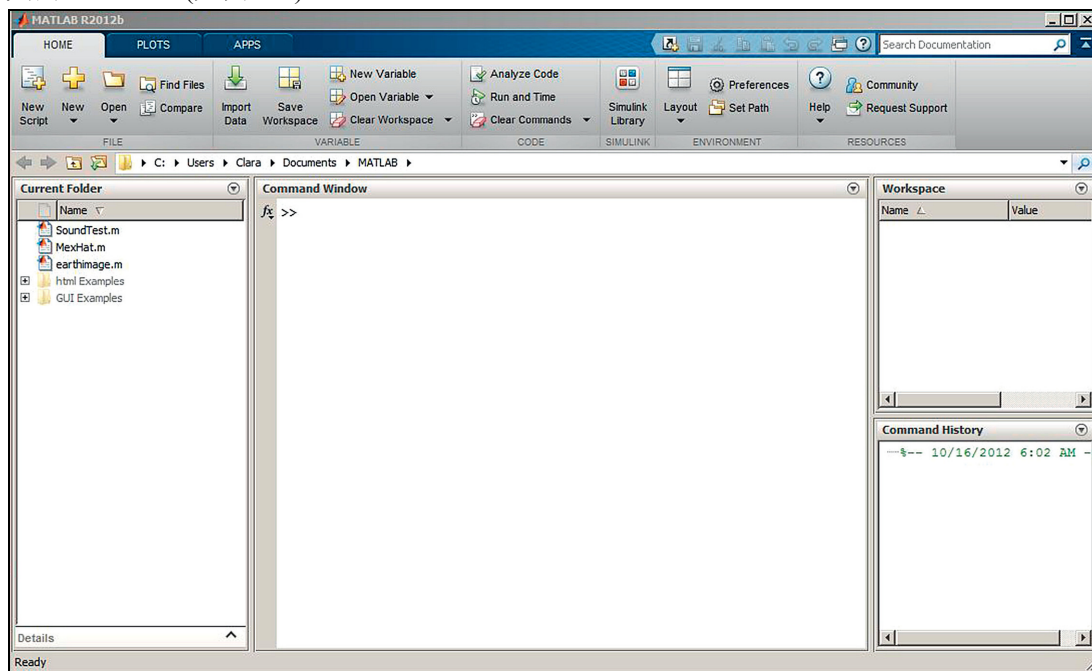


图 1-4 MATLAB 2012b 中全新的桌面工具栏

在 Desktop 的下一行的左侧有三个标签。图 1-4 中的第一个标签在最前面，因此桌面显示的是 Home 工具栏(标签和与标签相关的工具栏是该版 MATLAB 的主要新特性)。如果你已经熟悉以前版本的 MATLAB，就会发现，这些新特性明显提高了 MATLAB 的易用性。此外，所有以前开发的工具在早期版本的 MATLAB 中一样可以运行。另外两个标签是 PLOTS 和 APPS。这些特性使你可以通过鼠标指针指向并单击来使用 MATLAB 中的工具，因此，它们增强了 MATLAB 中工具和工具箱的易用性。此外，APPS 环境允许用户创建自己的应用程序(或 APPS)。

1.2.1 使用 Editor 和运行脚本

用鼠标指针指向并单击 Home 工具栏最左侧的 New Script (新建脚本)图标。这样做可以在 Desktop 的中心打开 Editor, 如图 1-5 所示。我们注意到出现了三个新的标签, 其中突出显示的标签是与编辑器相关联的 EDITOR 标签。另外两个标签是 PUBLISH 和 VIEW。后者在创建与科学计算工作相关的记事本或其他文件时很有用。这些工具的应用将通过后文中的一个示例来说明。

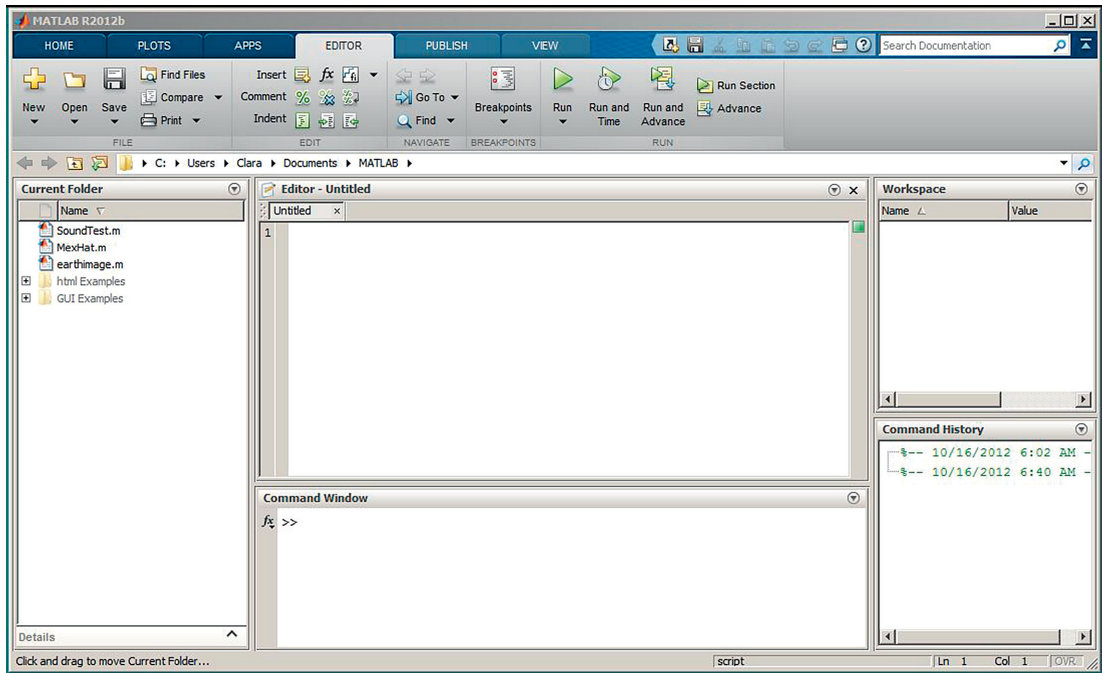


图 1-5 在默认位置打开的 Editor, 位于桌面的中央

我们先考虑 Editor 的使用。在 Editor 中输入以下脚本:

```
% Let us consider the following arbitrarily selected matrix:
A = magic(3)
% Let us evaluate its inverse as follows:
AI = inv(A)
% Let us check that it is an inverse:
IPredicted = A * AI
% This is the exact unitary matrix:
IM = eye(3)
..

% The is the difference between the exact and predict unitary
% matrix:
difference = IPredicted - IM
```

```

for m = 1:3
    for n = 1:3
        if difference(m,n) < eps;
            IPredicted(m,n) = IM(m,n);
        end
    end
end
IPredicted
IPredicted == IM

```

然后单击位于 View 标签正下方的 Run 按钮。第一次执行该脚本时需要命名该文件。在这个示例中使用的名称是 ExA1_1.m。如果所有行都输入正确(除了以符号“%”开始的行,因为它们都是注释,同脚本中的命令序列没有任何关系,只是帮助读者理解脚本的功能),Command Window 会显示如下内容:

```

A =
     8     1     6
     3     5     7
     4     9     2
AI =
    0.1472   -0.1444    0.0639
   -0.0611    0.0222    0.1056
   -0.0194    0.1889   -0.1028
IPredicted =
    1.0000         0   -0.0000
   -0.0000    1.0000         0
    0.0000         0    1.0000
IM =
     1     0     0
     0     1     0
     0     0     1
difference =

difference =
    1.0e-15 *
         0         0   -0.1110
   -0.0278         0         0
    0.0694         0         0
IPredicted =
     1     0     0
     0     1     0
     0     0     1
ans =
     1     1     1
     1     1     1
     1     1     1

```

IPredicted 矩阵应该是单位矩阵(Identity Matrix, IM)。IPredicted 矩阵是由矩阵 A 的逆 AI 与矩阵 A 相乘得到的。最后输出的 IPredicted 矩阵是原矩阵的修正版本;如果预测和实际的 IM 元素的差小于 $\text{eps}=2.2204e-16$, IPredicted 中的元素就会被改为 IM 中的对应元素。因为结果等于单位矩阵,所以该逆计算是正确的(至少在计算环境的计算误差范围内,即 $0 < \text{eps}$)。这个结论基于如下事实:在将经过调整的 IPredicted 矩阵中的所有条目与 IM 矩阵

相应条目做逻辑对比时，上面示例中的 ans 输出的逻辑结果都是 1(或真)。

练习进行到此时，Desktop 将如图 1-6 所示。该文件的名称是 ExA1_1.m。它出现在 Current Folder 中，同时也出现在 Command History (命令历史记录)中。注意该脚本创建的变量填充在工作区中。

到本节为止，我们完成了本书中大部分练习所需要的最重要工具的入门学习。在下一节中，我们会学习一个关于 MATLAB R2012b 中一些其他新特性的示例。

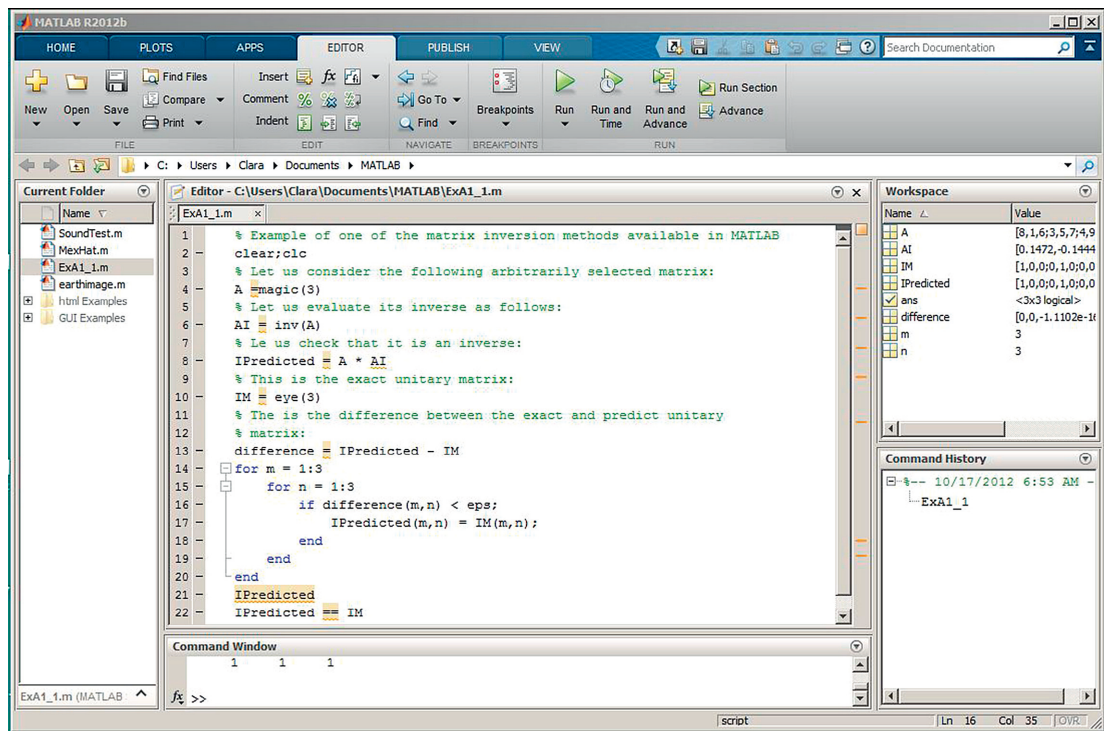


图 1-6 本节第一个示例创建并执行的示例脚本(一)

1.2.2 帮助、发布和视图

发布是一种以 HTML 格式创建记事本或其他文件的简单方法。MATLAB 将输入到 M-文件中的信息发布到一个文档中，该文档看起来就像全新的 Help 环境。我们打开帮助文件，首先将鼠标移到 Desktop 顶部的“Search Documentation”文本前面的问号上。左键单击问号“?”。Help 窗口会在桌面的中间打开。左键单击主题“MATLAB”，就会打开如图 1-7 所示的窗口。这也展示了 MATLAB R2012b 中可搜索文档的新格式。

对这个文档和那种你可以自己发布的文档进行对比。通过下面这个简单的示例，你将看到创建像 MATLAB 文档一样的文档是多么容易。

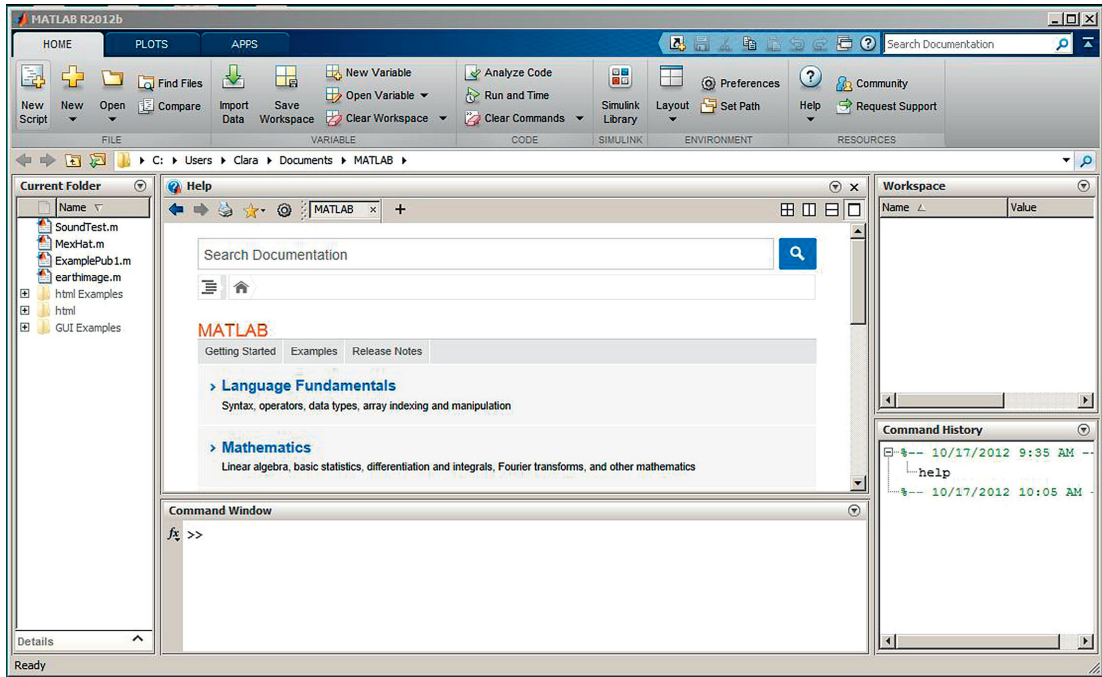


图 1-7 本节第一个示例中创建并执行的示例脚本(二)

单击 New Script 按钮，打开 Editor(或在 Command Window 命令提示符后输入 edit，然后按回车键)。此时 Editor 标签位于主任务栏上最靠前的位置。将光标置于 PUBLISH 上，单击鼠标左键，这会让 PUBLISH 工具栏在最前面显示。左键单击 Section with Title 按钮。用 PUBLISH example 替换 SECTION TITLE。接下来，将 DESCRIPTIVE TEXT 替换为：

```
% This is an example to illustrate how easy it is to create a document
% in the PUBLISH environment.
%
% (1) This is an illustration of a formula created with a LaTeX command.
%
```

然后，单击位于 Insert Inline Markup 组中的 Σ inline LaTeX 按钮。这会将式子 $x^2 + e^{\pi i}$ 加入到脚本中。在这个方程的后面，增加如下所示的最终脚本文件中的文本，该文本以“clicked:”结尾。在这之后是一个空行和一个命令脚本，该命令脚本用于说明如何将 MATLAB 命令合并到发布文档中。

```
% This is an example to illustrate how easy it is to create a document
% in the PUBLISH environment.
%
% (1) This is an illustration of a formula created with a LaTeX command:
%
%%
%  $x^2 + e^{\pi i}$ 
%
% (2) This is an illustration of how you can incorporate a MATLAB script
```

```
% in the document that is run when the Publish button below and to the  
% right of View is clicked:
```

```
% Earth picture  
load earth  
image(X); colormap(map);  
axis image
```

最后一步是左键单击位于 **View** 右下方的 **Publish** 按钮。出现的第一个窗口将要求你保存 M-文件。在这个示例中使用的名称是 `ExamplePub1.m`。M-文件在 **Current Folder** 中出现之后，就说明它被保存了。系统会自动创建一个名为 `html` 的文件夹，其中包含刚刚创建的 HTML 文件。该文档如图 1-8 所示。

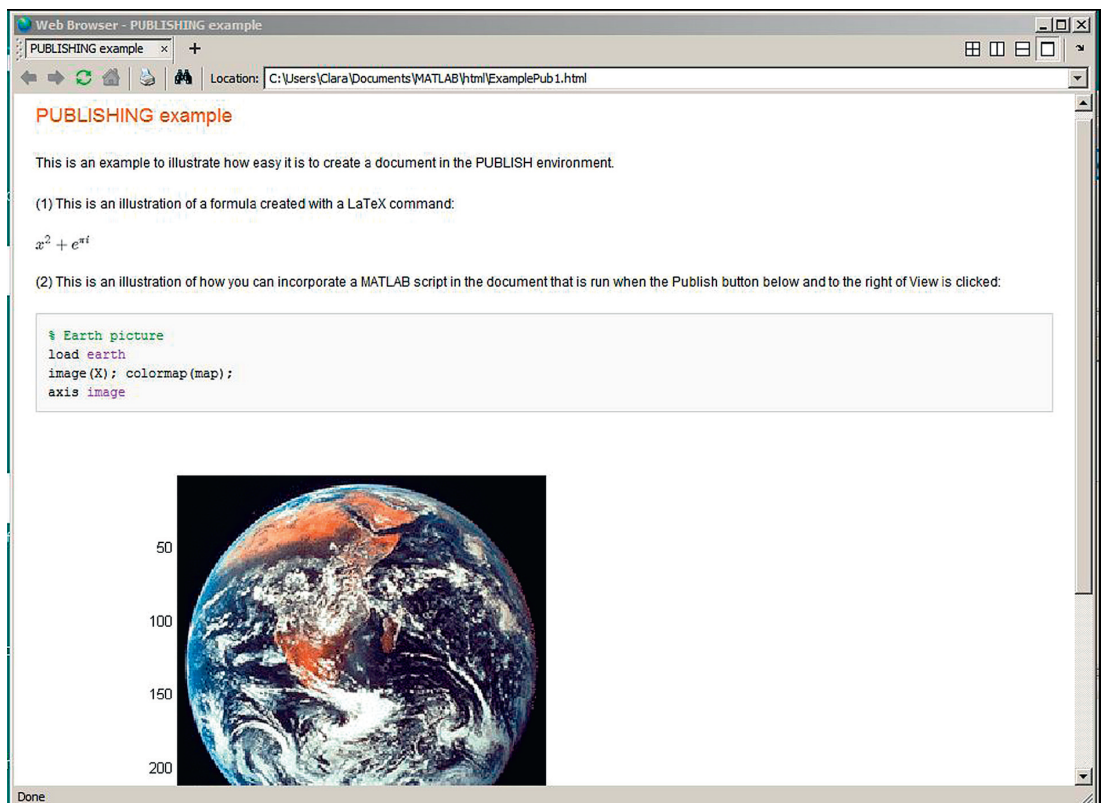


图 1-8 在发布环境中创建的示例文档

最后还有一点，可以使用 **VIEW** 选项卡的工具栏来更改 **Editor** 窗口的配置。作者认为，还是保持默认的编辑器环境为好，因为这是专门为刚开始使用 **MATLAB** 进行科学计算的用户设计的。当然，在 **MATLAB** 中定制工作环境是完全可行的。话说回来，学会如何使用默认的环境将有助于判断哪些更改可以更好地满足自己使用 **MATLAB** 的要求。

1.2.3 符号与 MuPAD Notebook APP(记事本应用程序)

Symbolic Math Toolbox (符号数学工具箱)是一款非常实用的工具,它可以帮助你做符号数学分析。MuPAD Notebook APP 使该工具箱变得更加易用。要启动该 APP,首先左键单击如图 1-4 所示桌面顶端的第二行的 APPS 标签。该操作会使 APPS 工具栏显示在最前面,如图 1-9 所示。

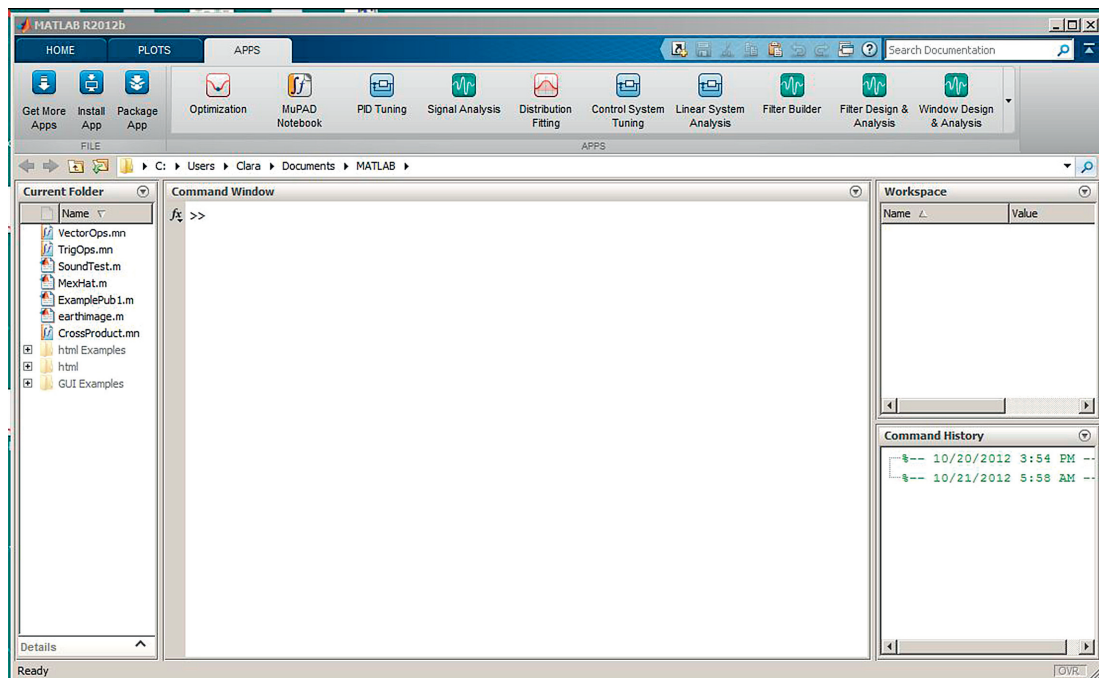


图 1-9 桌面上的 APPS 工具栏

左键单击 MuPAD Notebook APP。图 1-10 中的窗口是打开的记事本环境。白色记事板左上方的左方括号处是输入命令的地方,记事板右边的面板是 Command Bar(命令栏)。它为许多数学计算所需的命令提供快捷的使用方式,包括数学表达式的操作和计算,以及图形的绘制。记事本上方的两个工具栏提供了有用的工具,用以改善 MuPAD 的使用体验。将光标移至第二行工具栏条目的上方,就能看到每个按钮的用途。第一行工具栏上的条目则需要将光标移至其上方,再左键单击鼠标,打开下拉菜单。这个工具栏对 MATLAB 科学计算环境的所有窗口通用。下面来看一个简单的示例。

左键单击记事板中的左方括号。在这个位置可以开始输入文本(也就是即将在记事本中进行的数学运算的相关笔记)。图 1-11 展示了一个简单的示例,先对一个函数求导,然后对结果进行积分,以此来说明微分和积分计算的关系。该记事本保存为 Calc1.mn。请留意 MuPAD 中创建的 MATLAB 记事本文件的扩展名(*.mn)。在 Command Folder(命令文件夹)面板中双击带这个文件名的文件,打开该记事本(如图 1-11 中所示)。图 1-11 中提供了该例的细节。

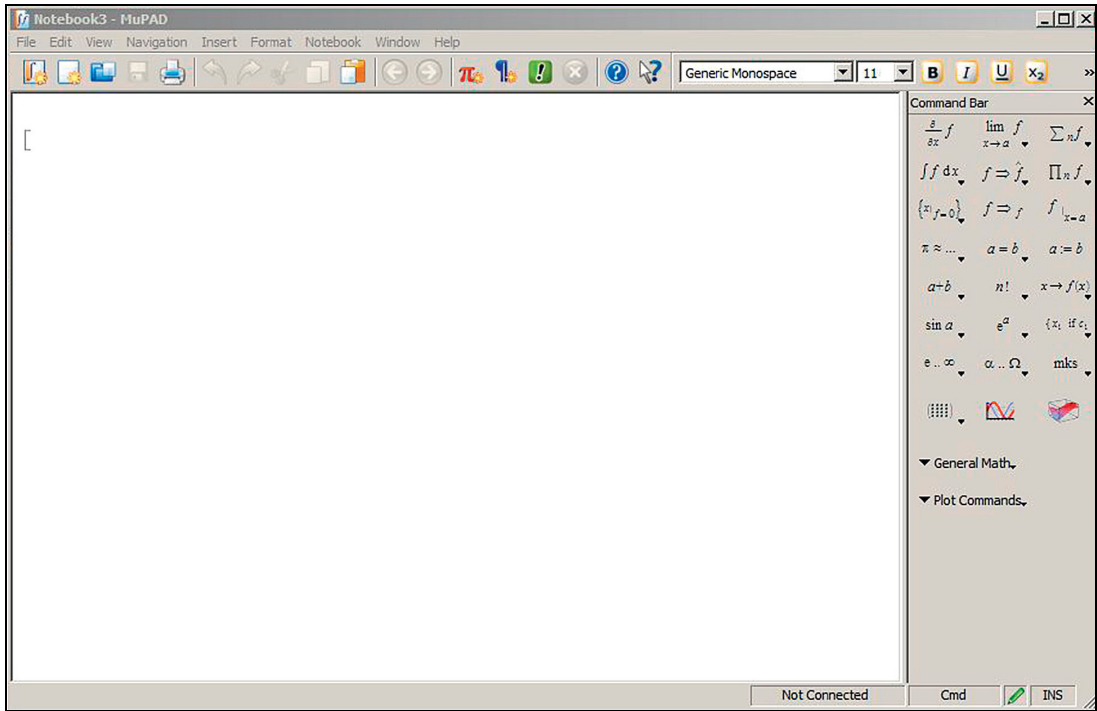


图 1-10 MuPAD 窗口或记事本

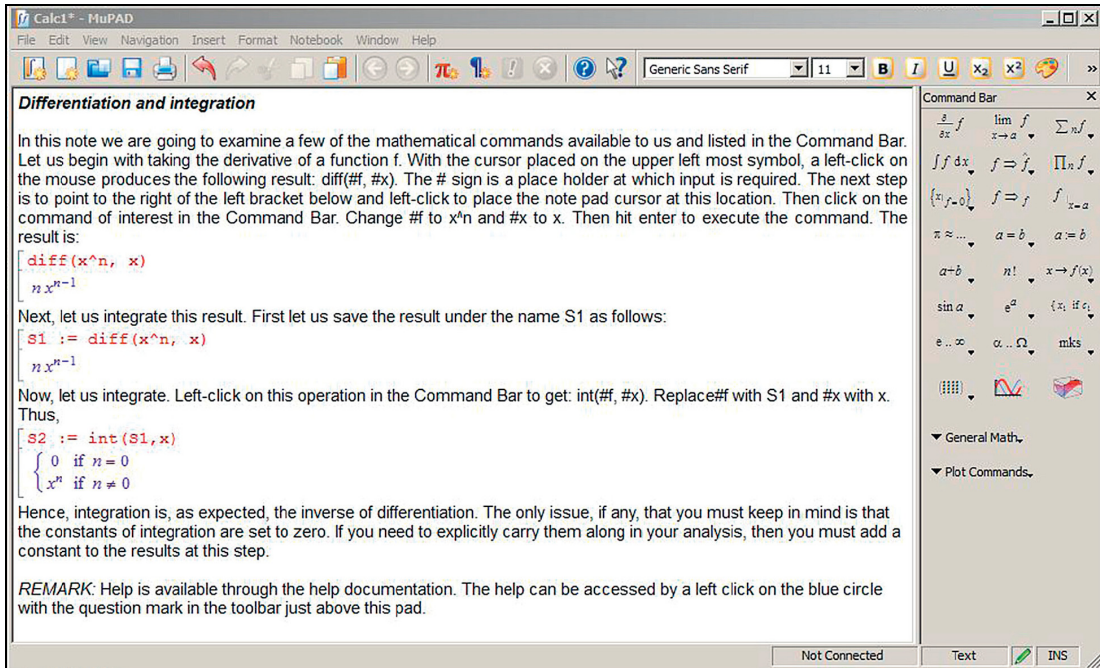


图 1-11 MuPAD 记事本 Calc1.mn

另一个示例是 MuPAD 的图形功能。在 MATLAB 中有很多其他的绘图工具，可以让你形象而快速地查看函数。MuPAD 环境尤其适合这项任务。假设你正在阅读一篇技术文章，遇到两个有趣的函数，你想知道它们的图形是什么样子。我们来看两个示例。一个是 $\operatorname{sech}^2(x)$ 函数，它在非线性波理论中起到非常重要的作用。另一个是第一类的完全椭圆积分，即：

$$K(m) = \int_0^{\pi/2} \frac{1}{\sqrt{1-m^2 \sin^2 \theta}} d\theta$$

其中， $m = \pi/4$ 。此积分在位势论中扮演重要角色。这些函数到底是什么样子的？图 1-12 中的 MuPAD 记事本展示了这些函数的形状。更多此类函数可以在帮助文档及其引用的文献中找到。

小结一下，本节简要介绍了一个 APP 以及 Symbolic Math Toolbox (符号数学工具箱) 的功能。

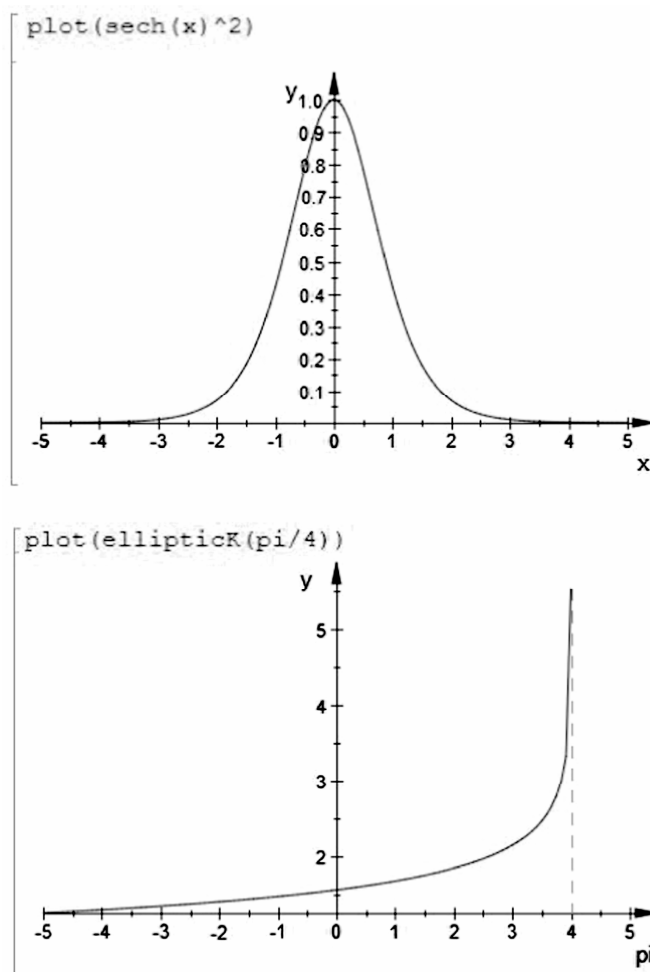


图 1-12 MuPAD 图形展示

1.2.4 其他 APP

在 MathWorks 中还有很多其他的 APP。此外，还可以创建自己的 APP。因此，我们在第一次体验 MATLAB 时就会了解到，需要学习的还有很多(甚至是终生学不完)，因为该科学计算环境中融入了非常丰富的技术。你可以开发属于自己的工具箱、自己的 APP，还能定制工作环境(桌面布置、背景颜色、字体、图形用户界面等)，这些特性为用户设计、创建有用的工具和撰写工作文档提供真正的机会和有用的体验。

1.2.5 附加特性

MATLAB 还有其他好用的特性。例如，可以通过执行命令 `magic(10)` 来生成 10 乘 10 的幻方(magic square)，其行、列和主对角线元素的和为同一数值。请读者自己尝试一下。一般来说，一个 n 乘 n 的幻方，其一行和一列的总和是 $n(n^2 + 1)/2$ 。

你甚至可以得到幻方中元素的等高线(contour)图。MATLAB 假设幻方中的各元素是地图上每个点的海拔高度，然后绘制等高线。`contour(magic(32))` 绘制出的图形看起来很有趣。

如果想看看著名的墨西哥帽(见图 1-13)，输入以下 4 行命令即可(小心，不要出现拼写错误)：

```
>> [x y] = meshgrid(-8 : 0.5 : 8) ; <Enter>
>> r = sqrt(x.^2 + y.^2) + eps; <Enter>
>> z = sin(r) ./ r; <Enter>
>> surf(z)
```

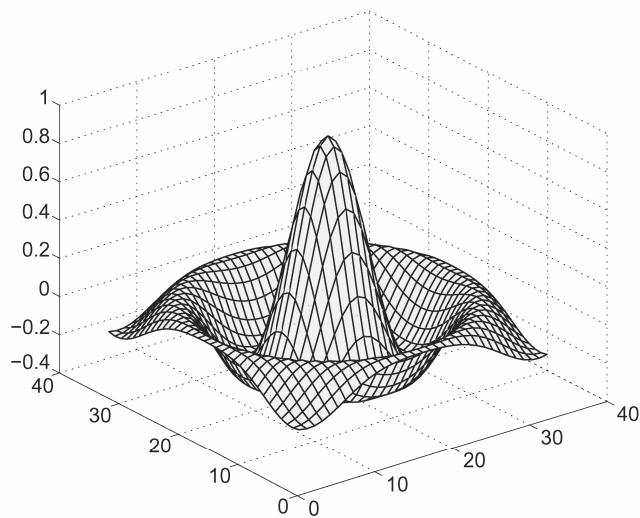


图 1-13 墨西哥帽

`surf(z)` 生成曲面(填充)视图。`surfc(z)` 或 `meshc(z)` 则会在曲面下绘制 2D 的等高线图。

```
>> surf(z), shading flat <Enter>
```

上面这条命令删除网格线，生成一幅漂亮的图。

下面的动画是图 1-13 所示墨西哥帽图形的扩展。它使用 `for` 循环，进行从 $n=-3$ 到 $n=2$ ，步长为 0.05 的重复计算。它以 `for n = -3:0.05:3` 命令开始，以 `end` 命令结束，这是编程中最重要的结构之一。本例中，`for` 和 `end` 之间的命令重复执行 121 次。`pause(0.05)` 命令在 `for` 循环中引入 0.05 秒的延迟，使动画放慢，因此这张图片每隔 0.05 秒会变化一次，直到计算结束。

```
>> [x y] = meshgrid(-8 : 0.5 : 8) ; <Enter>
>> r = sqrt(x.^2 + y.^2) + eps; <Enter>
>> for n = -3 : 0.05 : 3; <Enter>
>> z = sin(r. * n) ./r; <Enter>
>> surf(z), view(-37, 38), axis([0, 40, 0, 40, -4, 4]); <Enter>
>> pause(0.05) <Enter>
>> end <Enter>
```

可以在 MATLAB 中用很多种方式来检查声音。一种方式是听信号。如果计算机有扬声器，可以尝试如下命令来听一段 Handel 的哈利路亚合唱：

```
>> load handel <Enter>
>> sound(y,Fs) <Enter>
```

还可以尝试加载 `chirp`、`gong`、`laughter`、`splat` 和 `train` 来听不同的声音，为每个声音运行函数 `sound(y,Fs)`。

如果想从太空看地球，试试如下命令：

```
>> load earth <Enter>
>> image(x); colormap(map) <Enter>
>> axis image <Enter>
```

使用以下命令，在 MATLAB 中输入出现在本章开头的矩阵：

```
>> A = [1 3 5; 2 4 6] <Enter>
```

在下一行的命令提示符后，输入 `A(2,3)` 来摘选出第二行、第三列的数字。

在 MATLAB 中有一些搞笑的函数。请读者尝试一下 `why` (为什么不呢?)。然后再尝试输入 `why(2)` 两次。输入以下命令，可以看到实现上述操作的 MATLAB 代码：

```
>> edit why <Enter>
```

如果已经看到这个文件，单击 Editor 窗口的菜单栏下方的文件标签(`why.m`)右边的 × 按钮，即可关闭该文件。如果不小心输入了一些东西或修改了它，不要保存该文件。

稍后我们将使用 `edit` 命令来说明如何创建类似 `why.m` (`why` 命令所执行的文件的名称) 的 M-文件。在浏览过 MATLAB 桌面的基本特性之后，你就可以自己创建一个 M-文件了。更多关于在 MATLAB 环境中创建程序的细节，将在第 2 章介绍 Editor 时进行解释。

1.3 示例程序

在 1.1 节我们看到了一些简单示例，了解了如何通过 MATLAB 提示符后输入单独的命令或语句来使用 MATLAB。然而，你可能需要求解一些 MATLAB 无法仅用一行命令来解决的问题，比如找出二次方程的根(考虑所有特殊的情况)。解决这样问题的语句集合就叫做程序。在本节中，我们看看两个简短程序的编写和运行流程，而不考虑它们是如何工作的——在下一章中将会详细介绍。

1.3.1 剪切和粘贴

假设想绘制函数 $e^{-0.2x}\sin(x)$ 在 0 到 6π 范围内的图像，如图 1-14 所示。Windows 环境下有非常好用且易于掌握的剪切和粘贴编辑操作。操作过程如下：

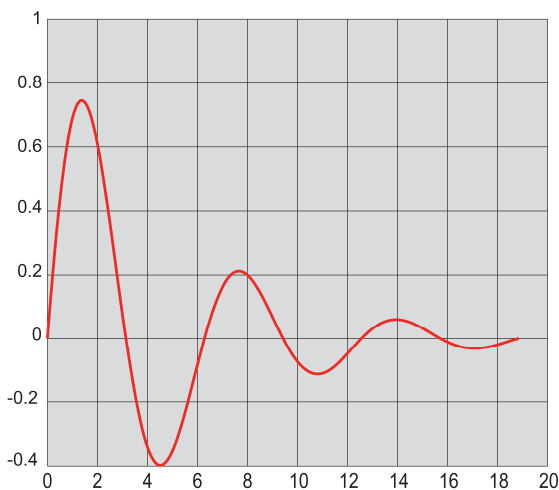


图 1-14 $e^{-0.2x}\sin(x)$

从 MATLAB 桌面选择 File | New | Script，或者单击桌面工具栏的 New File 按钮(你还可以在 Command Window 中输入 edit，然后按回车键)。该操作会在 Editor/Debugger 中打开一个无标题的窗口。你可以暂时认为这是一个可以在其中写程序的“便签”。现在，在 Editor 中准确无误地输入以下两行：

```
x = 0 : pi/20 : 6 * pi;  
Plot (x, exp(-0.2*x) .*sin(x), 'k' ),grid
```

顺便说一句，在第二行的第二个*的前面是个句点(句号)，之后会有更详细的解释！plot 函数中附加的参数'k'将绘制黑色的图形，以示区别。如果喜欢红色，可以将'k'改为'r'来生成红色的图形。

接下来，将鼠标指针(它现在看起来像非常细的大写字母 I)移到第一行中 x 的左边。按住鼠标左键，将鼠标指针移到第二行的末尾。这个过程叫做拖放。此时，这两行程序应该都高亮显示，可能是蓝色，以表明它们都被选中。

选择 Editor 窗口中的 Edit 菜单，然后单击 Copy(或使用键盘快捷键 Ctrl+C)。如果操作系统是 Windows，那么该操作会将高亮的文本复制到 Windows 的剪贴板中。

现在返回到 Command Window。确定光标位置在提示符>>处(如有必要，可以单击那里)。选择 Edit 菜单，然后单击 Paste(或用快捷键 Ctrl + V)。剪贴板中的内容将会被复制到命令窗口中。按 Enter 键执行这两行程序。图形应该在图像窗口中显示出来。

在 Editor 中加亮(选择)文本，然后将它们复制到 Command Window 的过程叫做“剪切和粘贴”(更准确地说，这里应该叫“复制和粘贴”，因为原始文本是从 Editor 中复制而不是剪切)。该操作值得好好练习直至熟练掌握。

如果需要修改程序，请返回 Editor，单击出错的位置(这一步是将插入点移到合适的位置)，进行修正，并再次剪切和粘贴。也可以通过编辑命令行来改正错误。还有另一种方法，可以从 Command History 窗口粘贴(还可以顺带追溯许多先前的会话)。在单击时长按 Ctrl 键，可以在 Command History 窗口中选择多行。

如果喜欢，可以直接在 Command Window 中输入多行命令。在输入完最后一行之前，为了防止程序运行，在每一行的后面使用 Shift+Enter 键来换行。最后按 Enter 键运行所有命令行。

来看另一个示例，假如你在银行存了 1000 美元。每年利息复合增长率为 9%。一年后，你在银行的存款余额将会是多少？如果想写一个 MATLAB 程序来计算存款余额，那么原则上你自己必须知道如何求解这个问题。即使一个像这样相对简单的问题，首先写下粗略的结构规划也是非常有帮助的：

- 在 MATLAB 中输入数据(初始余额和利率)
- 计算利息(1000 美元的 9%，即 90 美元)
- 将利息加到余额中(90 美元+1000 美元，即 1090 美元)
- 显示新的余额

返回到 Editor 中，清除以前所有的文本，照例通过拖放操作来选择它们(或使用 Ctrl+A 快捷键)，然后按 Del 键。顺便说一下，单击选择区以外的任何地方，都可以取消选择高亮的文本。输入如下程序，然后剪切并粘贴到 Command Window 中：

```
balance = 1000;
rate = 0.09;
Interest = rate * balance;
balance = balance + interest;
Disp('New balance:');
Disp(balance);
```

当按下 Enter 键运行它，应该能在 Command Window 中获得以下输出结果：

```
New balance:
    1090
```

1.3.2 保存程序：脚本文件

我们已经知道如何在 Editor 和 Command Window 之间剪切和粘贴，以编写和运行

MATLAB 程序。如果想在之后再次使用程序，显然需要保存这个程序。

在 Editor 菜单栏中选择 File | Save，保存 Editor 中的内容。此时会出现“Save file as”(另存为)对话框。选择一个文件夹，然后在“File name”(文件名)输入框中输入一个文件名，该文件名必须使用.m 扩展名，例如 junk.m。然后单击 Save。Editor 窗口现在有了标题 junk.m。如果在 Editor 中对 junk.m 进行修改，编辑器顶部的名字的旁边将出现一个星号，直到保存这些修改。

在 Editor(或任何其他 ASCII 文本编辑器)中保存的带.m 扩展名的 MATLAB 程序被称为脚本文件，或者简称为脚本。(MATLAB 函数文件的扩展名也是.m，因此我们统一把脚本和函数文件称为 M-文件)。脚本文件的特殊意义在于，如果在命令行提示符处输入脚本文件的名称，MATLAB 会执行脚本文件中的每条语句，就好像这些语句是在提示符处输入的一样。

脚本文件的命名规则和 MATLAB 中变量的命名一样(见接下来第 2 章的 2.1 节)。

举个例子，将上文的复合利率程序保存在名为 compint.m 的脚本文件中。然后在 Command Window 中的提示符处简单地输入这个名字：

```
compint
```

一旦按 Enter 键，compint.m 中的语句将会被准确执行，就像将它们复制到 Command Window 中执行一样。至此，你已经成功地创建了一个新的 MATLAB 命令，即 compint。

使用命令 type，可以将脚本文件的内容在 Command Window 中列出，例如：

```
type compint
```

(扩展名.m 可省略)

脚本文件提供了一种管理大型程序的有用方法。你没必要每次运行这些大型程序时，都将它们粘贴到 Command Window 中。

1. 当前目录

当运行脚本时，必须确保将 MATLAB 的当前文件夹(显示在 Current Folder 正上方的工具栏里)设置为保存着该脚本的文件夹(或目录)。可以通过如下操作来改变当前文件夹，在工具栏里输入新的当前文件夹的路径，或者从之前工作文件夹的下拉列表中选择文件夹，还可以单击“浏览”按钮(它是显示 Current Folder 路径的区域的左边第一个文件夹，带着绿色箭头)，然后选择一处新的位置来保存和执行文件(例如，如果要为不同的课程创建文件，你可能希望将工作保存到以正在学习的课程名字或编号命名的文件夹中)。

可以在命令行中用 cd 命令来更改当前文件夹，例如：

```
cd \mystuff
```

cd 命令自身会返回当前目录或文件夹的名称(在 MATLAB 的最新版本中亦是如此)。

2. 从当前文件夹浏览器运行脚本

如下是一种简便地运行脚本的方法。在 Current Folder 浏览器中选择一个文件，右键单

击。上下文菜单就会出现(上下文菜单是通用的桌面特性)。在上下文菜单中选择 Run, 运行结果将出现在 Command Window 中。如果想编辑该脚本, 在上下文菜单中选择 Open。

1.3.3 程序实战

现在我们将详细讨论复合利率程序是如何运行的。

MATLAB 系统在技术上被称为解析器(与编译器相反)。意思是它将提交给命令行的每一条语句翻译(解析)成计算机能更好理解的语言, 然后立即执行。

MATLAB 中的一个基本概念是如何将数据存储于计算机的随机存取存储器(Random Access Memory, RAM)中。如果一条 MATLAB 语句需要存储一个数字, RAM 会为其预留空间。可以将这部分存储器想象成一堆盒子或内存位置, 每一个只能同时容纳一个数字。这些内存位置由 MATLAB 语句中的符号名引用。因此语句:

```
balance = 1000
```

将数字 1000 分配到名为 balance 的内存位置。因为在会话期间 balance 的内容可能会改变, 所以它被称为变量。

因此, MATLAB 将程序中的语句解析为:

- 将数字 1000 放入变量 balance 中
- 将数字 0.09 放入变量 rate 中
- 用 rate 的内容乘以 balance 的内容, 然后把结果放入 interest 中
- 将 balance 的内容加到 interest 的内容中, 然后把结果放入 balance 中
- 显示单引号中的信息(在 Command Window 中)
- 显示 balance 的内容

似乎没必要强调这一点, 但这些经过解析的语句是按照从上到下的顺序来执行的。在程序运行完毕后, 所使用的变量的值如下:

```
balance    : 1090
interest   : 90
rate       : 0.09
```

注意 balance 原来的值(1000)消失了。

请尝试以下练习:

- (1) 运行原程序。
- (2) 改变程序中的第一条语句, 看看结果。

```
balance = 2000
```

确保程序在运行时你能理解发生了什么。

- (3) 略去下面这一行:

```
balance = balance + interest;
```

然后重新运行。你能解释发生了什么吗?

- (4) 尝试重新编写程序，让 `balance` 的原始值不消失。现在你可能会遇到一些问题，比如：
- 可以使用什么名称来命名变量？
 - 如何表示数字？
 - 如果一条语句不适合放在某一行，会发生什么？
 - 如何使输出的结果更整洁？下一章会回答这些问题。在我们编写一些更完整的程序之前，还需要介绍一些其他的基本概念。下一章也会介绍这些概念。

1.4 本章小结

- MATLAB 是基于矩阵的计算机系统，用于帮助解决科学和工程问题。
- 可以通过在 Command Window 的命令行中输入命令和语句来使用 MATLAB。MATLAB 将立即执行这些命令和语句。
- 可以使用 `quit` 或 `exit` 命令来关闭 MATLAB。
- `clc` 命令可以清空 Command Window。
- `help` 和 `lookfor` 命令可以提供帮助。
- `plot` 命令可以在图像窗口中绘制一幅 x-y 图形。
- `grid` 可以在图形上绘制网格线。

1.5 本章练习

1.1 在命令行中对变量 `a` 和 `b` 赋值，例如，`a=3` 和 `b=5`。编写一些语句来解变量 `a` 和 `b` 的和、差、积和商。

1.2 在 1.2.5 节中给出了一个脚本，用于绘制墨西哥帽子问题的动画。将该脚本输入到编辑器中，保存并执行。在完成调试并成功运行之后，尝试修改它。

- 将 `n` 的最大值从 3 变更为 4，然后执行脚本。
- 将 `pause` 函数中的延迟时间从 0.05 变更为 0.1。
- 将命令行 `z=sin(r.*n)./r;` 变更为 `z=cos(r.*n);`，然后执行脚本。

1.3 在命令行中对变量 `x` 赋值，例如 `x=4*pi^2`。`x` 的平方根是多少？`x` 的平方根的余弦是多少？

1.4 在命令行中对变量 `y` 进行如下赋值：`y=-1`。`y` 的平方根是多少？答案显示：

```
ans =  
    0 + 1.0000i
```

是的，MATLAB 还能处理复数(不仅仅是实数)。因此不应该将符号 `i` 用于表示索引或变量名。在默认情况下，`i` 等于 -1 的平方根(必要时，MATLAB 也将 `j` 用于表示 $\sqrt{-1}$)。因此，也不应将 `i` 用于表示索引或变量名)。请举一个例子加以说明，在截至目前所受的教育中，你是如何在数学和科学研究中使用复数的。很多练习的答案在附录 E 中。