

国外计算机科学经典教材

---

# MATLAB 原理与应用

(第 7 版)

## 工程问题求解与科学计算

[美] 布莱恩·D. 哈恩(Brian D. Hahn) 著  
丹尼尔·T. 瓦伦丁(Daniel T. Valentine)  
杨海陆 席亮 译

清华大学出版社

北 京

Essential MATLAB for Engineers and Scientists, Seventh Edition

Brian D. Hahn Daniel T. Valentine

ISBN: 9780081029978

Copyright © 2019 Elsevier Inc. All rights reserved.

Authorized Chinese translation published by Tsinghua University Press Limited.

《MATLAB 原理与应用(第7版) 工程问题求解与科学计算》(杨海陆 席亮 译)

ISBN: 9780081029978

Copyright © Elsevier Inc. and Tsinghua University Press Limited. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from Elsevier (Singapore) Pte Ltd. Details on how to seek permission, further information about the Elsevier's permissions policies and arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: [www.elsevier.com/permissions](http://www.elsevier.com/permissions).

This book and the individual contributions contained in it are protected under copyright by Elsevier Inc. and Tsinghua University Press Limited.

This edition of Essential MATLAB for Engineers and Scientists is published by Tsinghua University Press Limited under arrangement with ELSEVIER INC.

This edition is authorized for sale in China only, excluding Hong Kong, Macau 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.

本版由 ELSEVIER INC. 授权清华大学出版社在中国大陆地区(不包括香港、澳门以及台湾地区)出版发行。

本版仅限在中国大陆地区(不包括香港、澳门以及台湾地区)出版及标价销售。未经许可之出口, 视为违反著作权法, 将受民事及刑事法律之制裁。

本书封底贴有 Elsevier 防伪标签, 无标签者不得销售。

#### 注意

本书涉及领域的知识和实践标准在不断变化。新的研究和经验拓展我们的理解, 因此须对研究方法、专业实践或医疗方法作出调整。从业者和研究人员必须始终依靠自身经验和知识来评估和使用本书中提到的所有信息、方法、化合物或本书中描述的实验。在使用这些信息或方法时, 他们应注意自身和他人的安全, 包括注意他们负有专业责任的当事人的安全。在法律允许的最大范围内, 爱思唯尔、译文的原文作者、原文编辑及原文内容提供者均不对因产品责任、疏忽或其他人身或财产损失及/或损失承担责任, 亦不对由于使用或操作文中提到的方法、产品、说明或思想而导致的人身或财产损失及/或损失承担责任。

北京市版权局著作权合同登记号 图字：01-2020-3288

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：第 7 版：工程问题求解与科学计算 / (美) 布莱恩·D. 哈恩 (Brian D. Hahn), (美) 丹尼尔·T. 瓦伦丁 (Daniel T. Valentine) 著；杨海陆，席亮 译. —北京：清华大学出版社，2020.7

(国外计算机科学经典教材)

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

ISBN 978-7-302-55823-1

I. ①M… II. ①布…②丹…③杨…④席… III. ①Matlab 软件—教材 IV. ①TP317

中国版本图书馆 CIP 数据核字(2020)第 107802 号

责任编辑：王 军

封面设计：孔祥峰

版式设计：思创景点

责任校对：成凤进

责任印制：丛怀宇

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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈：010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者：北京鑫丰华彩印有限公司

经 销：全国新华书店

开 本：170mm×240mm

印 张：19

字 数：615 千字

版 次：2020 年 8 第 1 版

印 次：2020 年 8 月第 1 次印刷

定 价：79.00 元

---

产品编号：082787-01

# 译者序

MATLAB 是 matrix 和 laboratory 两个词的组合,意为矩阵工厂(矩阵实验室),是由美国 MathWorks 公司发布的主要面对科学计算、可视化以及交互式程序设计的高科技计算环境。它将数值分析、矩阵计算、科学数据可视化以及非线性动态系统的建模和仿真等诸多强大功能集成在一个易于使用的视窗环境中,为科学研究、工程设计以及必须进行有效数值计算的众多科学领域提供了一种全面的解决方案,代表了当今国际科学计算软件的先进水平。

MATLAB 的基本数据单位是矩阵,它的指令表达式与数学、工程中常用的形式十分相似,可以进行矩阵运算、绘制函数和数据、实现算法、创建用户界面、连接其他编程语言的程序等,主要应用于工程计算、控制设计、信号处理与通信、图像处理、信号检测、金融建模设计与分析等领域。在数学类科技应用软件中, MATLAB 在数值计算方面首屈一指。

《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 7 版。本书整体风格通俗易懂,系统讲解了 MATLAB 基本环境和操作方法,介绍了最新的 MATLAB 功能,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,并分章阐述了 MATLAB 基础、程序设计与算法开发、MATLAB 函数与数据导入/导出工具、逻辑向量、矩阵和阵列、函数 M 文件、循环、MATLAB 图形、作为阵列的向量以及其他数据结构、错误和陷阱等内容。

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,详细介绍 MATLAB 在动力系统、仿真、数值方法、信号处理、Simulink 工具箱、Symbolic 工具箱的具体使用方法和技巧。基础知识和经典案例相结合,知识掌握更容易,学习更有目的性。

本书非常重视动手能力的培养,提供了大量实例和习题,鼓励读者在实践过程中发现问题和加深理解。阅读本书不需要任何计算机编程经验,但书中的示例(尤其是第 II 部分)需要一些高等数学和统计学知识。本书可作为高校学生系统学习 MATLAB 的书籍,也可作为广大科研和工程技术人员在工作中使用 MATLAB 的参考书。

在这里要感谢清华大学出版社的编辑,他们为本书的翻译投入了巨大的热情并付出了很多心血。没有他们的帮助和鼓励,本书不可能顺利付梓。

对于这本经典之作,译者本着“诚惶诚恐”的态度,在翻译过程中力求“信、达、雅”,但是鉴于译者水平有限,错误和失误在所难免,如有任何意见和建议,请不吝指正。

译者





# 作者简介

Brian D. Hahn 曾是南非开普敦大学数学和应用数学系的教授。在他的职业生涯中，Brian 撰写了十几本书，为初学者讲授编程语言。

Daniel T. Valentine 是名誉教授，曾担任纽约波茨坦克拉克森大学机械和航空工程系的教授和系主任。他还是纽约 NASA 太空拨款联盟克拉克森太空拨款项目的副主任，该项目为本科生和研究生的研究提供支持。他的博士学位是美国天主教大学(CUA)的流体力学。他在美国罗格斯大学获得机械工程学士和硕士学位。



# 致 谢

感谢 Mary、Clara、Zoe Rae 和 Zash T 的支持和鼓励，并将《MATLAB 原理与应用(第 7 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 7 版) 工程问题求解与科学计算》主要是为了跟上 MATLAB 的升级进度(最新版本是 9.5 版 R2018b)。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。

为了与作者 Brian D. Hahn 在之前版本中的目标保持一致,本书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从未使用过该工具。

MATLAB 是 Matrix Laboratory 的缩写,它是基于矩阵的概念实现的。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。本书主要面向科学家和工程师,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,在开始介绍 for 和 if 结构时,和很多其他书籍不同,本书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。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。

本书应该和 MATLAB 软件结合使用,因此读者需要先安装 MATLAB 软件,完成书中练习,从而理解 MATLAB 是如何完成任务的。任何工具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	5
1.1.3 数学函数	6
1.1.4 函数和命令	6
1.1.5 向量	6
1.1.6 线性方程组	8
1.1.7 教程和演示	9
1.2 Desktop	9
1.2.1 使用 Editor 和运行脚本	10
1.2.2 帮助、发布和视图	12
1.2.3 活动脚本中的符号	14
1.2.4 APPS	16
1.2.5 附加特性	16
1.3 示例程序	18
1.3.1 剪切和粘贴	18
1.3.2 保存程序：脚本文件	19
1.3.3 程序实战	20
1.4 本章小结	21
1.5 本章练习	21
第 2 章 MATLAB 基础	23
2.1 变量	23
2.2 工作空间	24
2.3 阵列：向量与矩阵	25
2.3.1 初始化向量：显式列表	25
2.3.2 初始化向量：冒号运算符	26
2.3.3 linspace 和 logspace 函数	26
2.3.4 转置向量	27
2.3.5 下标	27
2.3.6 矩阵	27
2.3.7 捕获输出	28

2.3.8 结构规划	28
2.4 重力作用下的垂直运动	29
2.5 运算符、表达式和语句	30
2.5.1 数字	31
2.5.2 数据类型	31
2.5.3 算术运算符	31
2.5.4 运算符的优先级	32
2.5.5 冒号运算符	32
2.5.6 转置运算符	33
2.5.7 阵列的算术运算	33
2.5.8 表达式	34
2.5.9 语句	34
2.5.10 语句、命令和函数	35
2.5.11 公式向量化	35
2.6 输出	38
2.6.1 disp 语句	38
2.6.2 format 命令	38
2.6.3 比例因子	39
2.7 for 循环	40
2.7.1 用牛顿法计算平方根	40
2.7.2 阶乘!	41
2.7.3 数列的极限	41
2.7.4 基本 for 结构	42
2.7.5 单行中的 for 语句	43
2.7.6 更加一般化的 for 语句	43
2.7.7 通过向量化避免使用 for 循环	43
2.8 判断	45
2.8.1 单行 if 语句	45
2.8.2 if-else 结构	46
2.8.3 单行 if-else 语句	47
2.8.4 elseif	47
2.8.5 逻辑运算符	48
2.8.6 多个 if 与 elseif 的对比	49
2.8.7 嵌套 if	50
2.8.8 是否将 if 向量化	50
2.8.9 switch 语句	50

2.9	复数	51	6.1.7	删除行和列	96
2.10	本章小结	52	6.1.8	初等矩阵	97
2.11	本章练习	53	6.1.9	特殊矩阵	97
<b>第3章</b>	<b>程序设计与算法开发</b>	<b>59</b>	6.1.10	对矩阵使用 MATLAB 函数	98
3.1	程序设计流程	59	6.1.11	操纵矩阵	99
3.2	MATLAB 函数编程	65	6.1.12	对矩阵进行阵列运算	99
3.2.1	内联对象: 谐振子	65	6.1.13	矩阵和 for 循环	99
3.2.2	MATLAB 函数: $y=f(x)$	66	6.1.14	矩阵的可视化	100
3.3	本章小结	68	6.1.15	将嵌套的 for 循环向量化: 贷款 偿还表格	100
3.4	本章练习	68	6.1.16	多维阵列	102
<b>第4章</b>	<b>MATLAB 函数与数据导入导出 工具</b>	<b>71</b>	6.2	矩阵运算	102
4.1	常用函数	71	6.2.1	矩阵乘法	102
4.2	导入和导出数据	75	6.2.2	矩阵求幂运算	103
4.2.1	load 和 save 命令	76	6.3	其他矩阵函数	104
4.2.2	导出文本(ASCII)数据	76	6.4	种群增长: 莱斯利矩阵	104
4.2.3	导入文本(ASCII)数据	76	6.5	马尔可夫过程	106
4.2.4	导出二进制数据	76	6.6	线性方程	108
4.2.5	导入二进制数据	77	6.6.1	MATLAB 中的解法	109
4.3	本章练习	77	6.6.2	残量	109
<b>第5章</b>	<b>逻辑向量</b>	<b>79</b>	6.6.3	超定方程组	109
5.1	示例	80	6.6.4	欠定方程组	110
5.1.1	不连续图	80	6.6.5	病态	110
5.1.2	避免除零	80	6.6.6	矩阵除法	111
5.1.3	避免无穷	81	6.7	稀疏矩阵	112
5.1.4	对随机数进行计数	82	6.8	本章小结	113
5.1.5	掷骰子	82	6.9	本章练习	113
5.2	逻辑运算符	83	<b>第7章</b>	<b>函数 M 文件</b>	<b>115</b>
5.2.1	运算符的优先级	84	7.1	示例: 再看牛顿法	115
5.2.2	危险	84	7.2	基本规则	116
5.2.3	逻辑运算符和向量	84	7.2.1	子函数	120
5.3	将逻辑向量作为下标	85	7.2.2	私有函数	120
5.4	逻辑函数	86	7.2.3	P-code 文件	120
5.5	用逻辑向量代替 elseif 阶梯	87	7.2.4	使用分析工具提高 M 文件的 性能	120
5.6	本章小结	89	7.3	函数句柄	120
5.7	本章练习	89	7.4	命令/函数对偶性	122
<b>第6章</b>	<b>矩阵和阵列</b>	<b>91</b>	7.5	函数名解析	122
6.1	矩阵	91	7.6	调试 M 文件	122
6.1.1	具体示例	91	7.6.1	调试脚本文件	122
6.1.2	创建矩阵	92	7.6.2	调试函数	124
6.1.3	下标	93	7.7	递归	124
6.1.4	转置	93	7.8	本章小结	125
6.1.5	冒号运算符	93	7.9	本章练习	125
6.1.6	复制行和列	96			

<b>第 8 章 循环</b> .....	127	9.3.4 图形对象创建函数	154
8.1 使用连续 for 语句的确定循环	127	9.3.5 指定父对象	154
8.1.1 二项式系数	127	9.3.6 定位图形	154
8.1.2 更新过程	128	9.4 编辑绘图	155
8.1.3 嵌套 for 语句	129	9.4.1 绘图编辑模式	155
8.2 使用连续 while 语句的不确定循环	129	9.4.2 属性编辑器	155
8.2.1 猜谜游戏	129	9.5 动画	156
8.2.2 while 语句	130	9.6 颜色等属性	158
8.2.3 投资翻倍的时间	130	9.6.1 色图	158
8.2.4 质数	131	9.6.2 曲面绘图的颜色	160
8.2.5 抛射体轨迹	132	9.6.3 Truecolor(真彩)	160
8.2.6 break 和 continue 语句	133	9.7 光照和镜头	161
8.2.7 菜单	134	9.8 保存、打印和导出图形	161
8.3 本章小结	134	9.8.1 保存和打开图像文件	161
8.4 本章练习	135	9.8.2 打印图形	161
<b>第 9 章 MATLAB 图形</b> .....	139	9.8.3 导出图形	162
9.1 基本二维图形	139	9.9 本章小结	162
9.1.1 标签	140	9.10 本章练习	163
9.1.2 在相同的坐标轴上绘制多个图形	140	<b>第 10 章 作为阵列的向量以及其他数据结构</b> .....	167
9.1.3 线型、标记和颜色	141	10.1 更新过程	167
9.1.4 坐标轴限制	141	10.1.1 单位时间步长	167
9.1.5 在一幅图中绘制多个图形: subplot	142	10.1.2 非单位时间步长	169
9.1.6 figure、clf 和 cla 函数	143	10.1.3 使用函数	170
9.1.7 图形输入	143	10.1.4 精确解	171
9.1.8 对数作图	143	10.2 频率、柱状图和直方图	172
9.1.9 极坐标作图	144	10.2.1 随机漫步	172
9.1.10 绘制快速变化的数学函数: fplot	145	10.2.2 直方图	173
9.1.11 属性编辑器	145	10.3 排序	173
9.2 三维作图	145	10.3.1 冒泡排序	173
9.2.1 plot3	145	10.3.2 MATLAB 中的 sort 函数	175
9.2.2 使用 comet3 绘制三维动画	146	10.4 结构体	175
9.2.3 网面	146	10.5 元胞阵列	177
9.2.4 等高线图	148	10.5.1 将数据赋给元胞阵列	177
9.2.5 使用 NaN 剪切曲面	148	10.5.2 访问元胞阵列中的数据	178
9.2.6 可视化向量场	149	10.5.3 使用元胞阵列	178
9.2.7 矩阵的可视化	150	10.5.4 显示和可视化元胞阵列	179
9.2.8 三维图形的旋转	150	10.6 类和对象	179
9.3 句柄图形	151	10.7 本章小结	179
9.3.1 获得句柄	151	<b>第 11 章 错误和陷阱</b> .....	181
9.3.2 图形对象的属性和修改方法	152	11.1 语法错误	181
9.3.3 句柄向量	153	11.1.1 向量的大小不匹配	181
		11.1.2 名称屏蔽	182
		11.2 逻辑错误	182
		11.3 舍入误差	182

11.4	本章小结	183	14.6	龙格-库塔法	219
11.5	本章练习	183	14.6.1	单个微分方程	219
<b>第 II 部分 实践应用</b>					
<b>第 12 章</b>	<b>动力系统</b>	<b>187</b>	14.6.2	差分方程组: 混沌	219
12.1	悬臂梁	188	14.6.3	将额外参数传递给 ODE 求解 程序	221
12.2	电流	189	14.7	偏微分方程	222
12.3	自由落体	191	14.8	复数变量和保角映像	224
12.4	摩擦力作用下的投射体问题	197	14.9	其他数值方法	226
12.5	本章小结	200	14.10	本章小结	227
12.6	本章练习	200	14.11	本章练习	227
<b>第 13 章</b>	<b>仿真</b>	<b>201</b>	<b>第 15 章</b>	<b>信号处理</b>	<b>229</b>
13.1	随机数的生成	201	15.1	谐波分析	230
13.2	旋转硬币	202	15.2	快速傅里叶变换(FFT)	233
13.3	投掷骰子	202	<b>第 16 章</b>	<b>Simulink 工具箱</b>	<b>237</b>
13.4	细菌分裂	203	16.1	弹簧-质块-阻尼器动力系统	241
13.5	随机游走	203	16.2	弹跳球动力系统	243
13.6	交通流量	204	16.3	范德波尔振子	244
13.7	正态(高斯)随机数	206	16.4	杜芬振子	245
13.8	总结	207	16.5	本章练习	246
13.9	本章练习	207	<b>第 17 章</b>	<b>Symbolic 工具箱</b>	<b>249</b>
<b>第 14 章</b>	<b>数值方法入门</b>	<b>209</b>	17.1	代数	250
14.1	方程组	209	17.1.1	多项式	250
14.1.1	牛顿法	209	17.1.2	向量	252
14.1.2	二分法	211	17.1.3	矩阵	253
14.1.3	fzero 函数	212	17.2	微积分	255
14.1.4	roots 函数	212	17.3	拉普拉斯变换和 Z 变换	257
14.2	积分	212	17.4	广义函数*	258
14.2.1	梯形法则	212	17.5	微分方程	259
14.2.2	辛普森法则	213	17.6	funtool、MuPAD 和帮助文档的 使用	260
14.2.3	quad 函数	214	17.6.1	funtool	260
14.3	数值微分	214	17.6.2	MuPAD 记事本和帮助文档	260
14.4	一阶微分方程	215	17.7	本章练习	262
14.4.1	欧拉方法	215	<b>附录 A</b>	<b>语法: 快速参考</b>	<b>263</b>
14.4.2	示例: 细菌生长	216	<b>附录 B</b>	<b>运算符</b>	<b>267</b>
14.4.3	另一种下标表示法	217	<b>附录 C</b>	<b>命令与函数: 快速参考</b>	<b>269</b>
14.4.4	预估-校正法	218	<b>附录 D</b>	<b>部分练习的答案</b>	<b>277</b>
14.5	线性常微分方程(LODE)	218			

# 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。







### 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 R2018b 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 特性)

MATLAB 是一个处理科学和工程计算问题的强大的科技计算系统。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极大地简化矩阵计算。矩阵  $A$  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:

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

从表示元素位置的行和列的索引号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。

本书假设你是 STEM(科学、技术、工程和数学)领域的工程师、科学家或研究生,STEM 领域的学生和从业人员在走进大学校门之前,就在数学课上学过矩阵。另一方面,本书假定你以前从未使用过 MATLAB 解决工程或科学问题,但对发现这个工具的科学计算能力很有兴趣,你能够熟练操作计算机键盘,并了解所使用的操作系统(如 Windows、UNIX 或 macOS)。你唯一需要已经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,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,只要输入一两个命令——MATLAB 就会完成大部分工作。

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。计算机编程就是编写一组指令,在

计算机执行这些指令时，就会完成特定的任务。本书将使用 MATLAB 的一些功能进行技术计算，以介绍编程。

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；还能调整 MATLAB 的外观，修改与之交互的方式，开发一个自己的工具箱，帮助解决感兴趣的问题。换句话说，可以根据经验定制 MATLAB 工作环境。

在本章的其余部分，将介绍一些简单的示例。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，就可以掌握第 2 章和 MATLAB 提供的帮助文件中的很多其他功能了。这将有助于解决更有趣和更具实质性的问题。本章的最后一节将快速浏览一下 MATLAB 桌面。

## 1.1 使用 MATLAB

请确认已在计算机上安装了 MATLAB，或者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(R2018b 版)。

要从 Windows 启动，请双击 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 macOS 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。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 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 保存的所有东西都将保存在该文件夹中。现在可在 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以激活 Command Window。

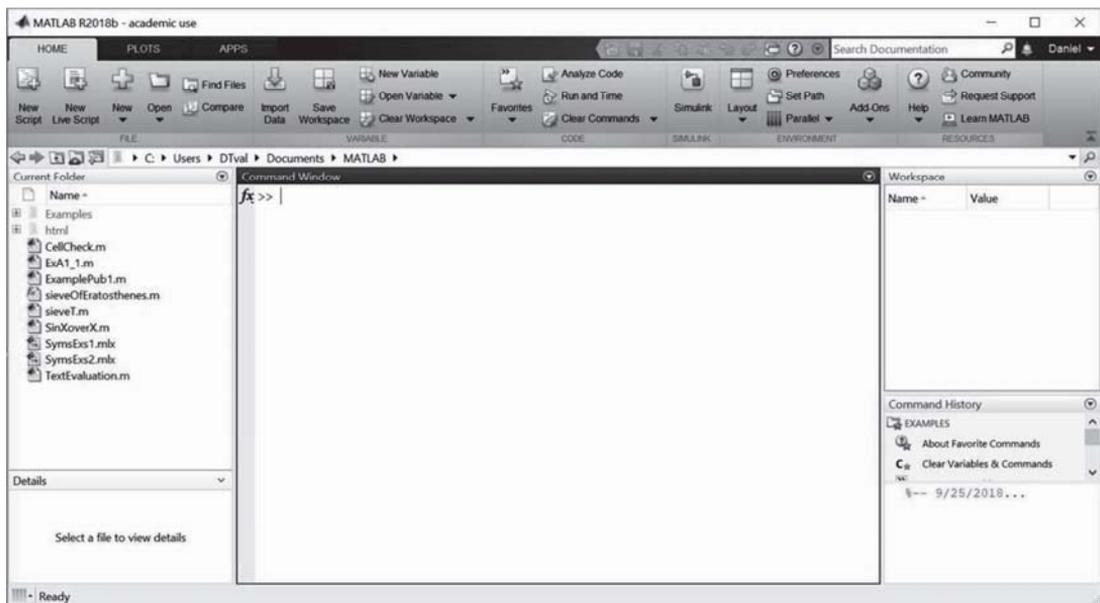


图 1-1 MATLAB 桌面显示 Home 任务栏(2018b 版本)

### 1.1.1 算术

由于熟悉算术，因此这里就用算术检验 MATLAB 是否能正确地运算。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？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(在之后处理数字阵列时将解释为什么需要这些符号)。

下面是创建和编辑命令行的一些提示:

- 包含提示符>>的行称为命令行。
- 在按 Enter 键之前，可以组合使用 Backspace 键、左方向键、右方向键和 Del 键以编辑 MATLAB 命令。这个有用的特性称为命令行编辑。
- 可以通过使用上方向键和下方向键选择(和编辑)输入的命令。记得按 Enter 键执行命令(即运行或执行命令)。
- MATLAB 有一个实用编辑功能，称为智能召回(smart recall)，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。

MATLAB 会如何处理 0/1 和 1/0？试试吧。如果坚持在计算中使用 $\infty$ ，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。

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

### 1.1.2 变量

现在给变量赋值，执行变量的算术运算。首先输入命令(在编程术语中称为语句) $a = 2$ 。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。

- 用命令  $\text{sqrt}(\pi)$  求解  $\sqrt{\pi}$ 。答案应该是 1.7725。注意,由于  $\pi$  是 MATLAB 的众多内置函数之一,因此 MATLAB 知道  $\pi$  的值。
- 诸如  $\sin(x)$  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  $\sin(90^\circ)$ 。答案应该是  $1(\sin(90 * \pi/180))$ 。
- 在 MATLAB 中,用  $\text{exp}(x)$  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>
```

注意,单独执行 `clear` 命令会清除工作区中的所有局部变量(工作区负责存储在命令行上定义的局部变量;参见默认桌面右边的 `Workspace` 面板); `>>clear pi` 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`,结果将显示对内置 `pi` 的重新定义:一个  $1 \times 1$  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。

### 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。它也有许多命令,如 `clc`(clear command window 的缩写), `help` 是一个经常使用的命令(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

1.1.2 节中使用的  $a$  和  $b$  之类的变量称为标量(scalar),它们是单个数值。MATLAB 还会处理向量(vector,通常称为数组),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  $x$ , 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>
```

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, 因此向量  $w$  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>
```

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

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

这幅图像会出现在一个单独的图像窗口中(见图 1-2)。要绘制出如图 1-2 所示  $\sin(x)$  函数的图像, 只需要用如下代码替换前面代码的最后一行:

```
>> plot(x,y,'-rs','LineWidth',2,'MarkerEdgeColor','k','MarkerSize',5),grid<Enter>
```

```
>> xlabel(' x '), ylabel(' sin(x) ') <Enter>
```

```
>> whitebg('y') <Enter>
```

可以通过单击 Command Window 或图像窗口中的任何地方, 对它们进行选择。Windows 下拉菜单在两种窗口中都可以使用。

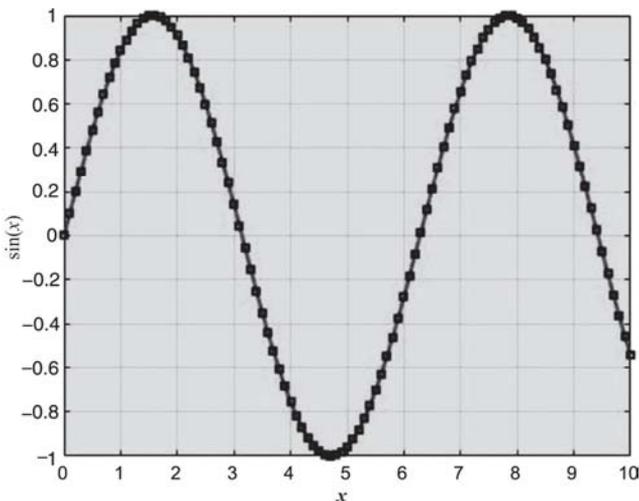


图 1-2 图像窗口

注意, 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。

修改图 1.2 中的 plot 函数, 这个函数有许多可用的选项。这里选择了 4 个选项。变量 y 后跟一个逗号, 再后面是 '-rs', 表示用红色实线连接正弦曲线上计算出来的点, 这些点在图中用方块标记出来, 线宽增加到 2, 标记的边线是黑色(k), 大小为 5。轴标记和背景色用 plot 命令后面的语句来改变(背景色、对象颜色等的其他修改可以用图像属性 Editor 进行, 该编辑器可以在图像工具栏的 Edit 下拉菜单中找到。本书中图像的颜色已经用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。注意, 在 ezplot 中, x 的默认取值范围不是 0~10。

tab 自动补全是 Command Window 中一个有用的编辑特性: 输入 MATLAB 名称开头的几个字母, 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。下面是一个求解两个联立方程的简单示例:

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

这两种求解方法。

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 solve 函数。输入以下命令(如实输入):

```
>> syms x y; [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)。左击 `Getting Started`，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。单击 `View more MATLAB examples`，向下滚动到 `Animations` 和 `Images`，学习 MATLAB 生成动画的特性，作为分析各种不稳定问题的一种方式。新特性会连续不断地添加到 MATLAB 中，所有新特性都会在 MATLAB 网站中报告。MathWorks 公司销售 MATLAB 和 SIMULINK，支持对它们的持续开发和改进，它们还为科学、工程、技术和数学团体提供了各种工具箱。

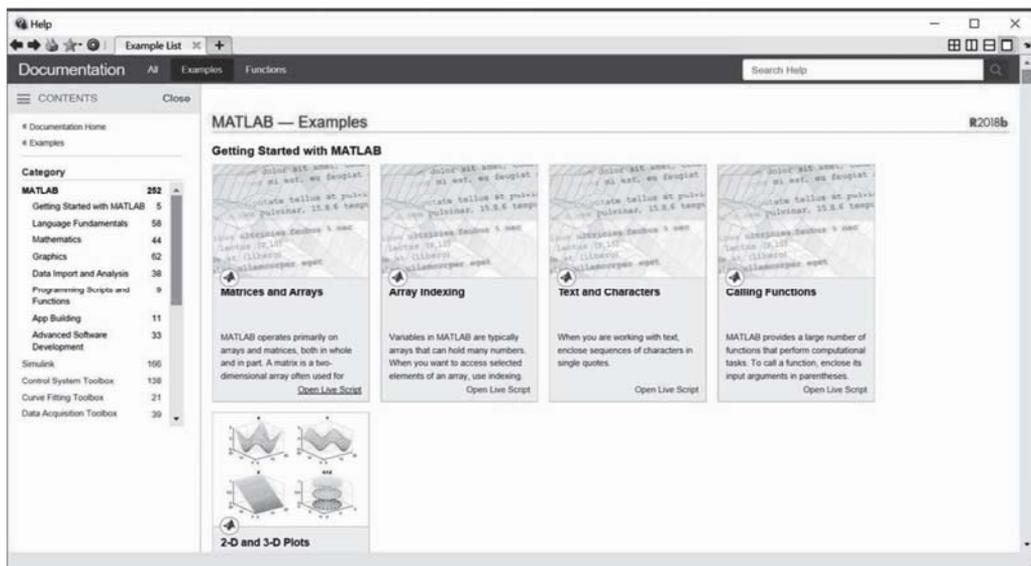


图 1-3 MATLAB 示例的帮助文档

## 1.2 Desktop

MATLAB R2018a 有个非常有用的特性，当第一次打开它时，它会在 Documents 文件夹中创建名为 MATLAB 的文件夹(如果尚不存在)。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 Documents 文件夹中，这个新的文件夹是默认的工作文件夹，保存创建的所有文件。`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。在顶行的左边是正在运行的 MATLAB 版本名称。这里的版本是 MATLAB R2018a。在顶行的右侧有三个按钮，它们是：下划线按钮，用于将 Desktop 最小化；矩形

按钮，用于将 Desktop 最大化；×按钮，用于关闭 MATLAB(见图 1-4)。

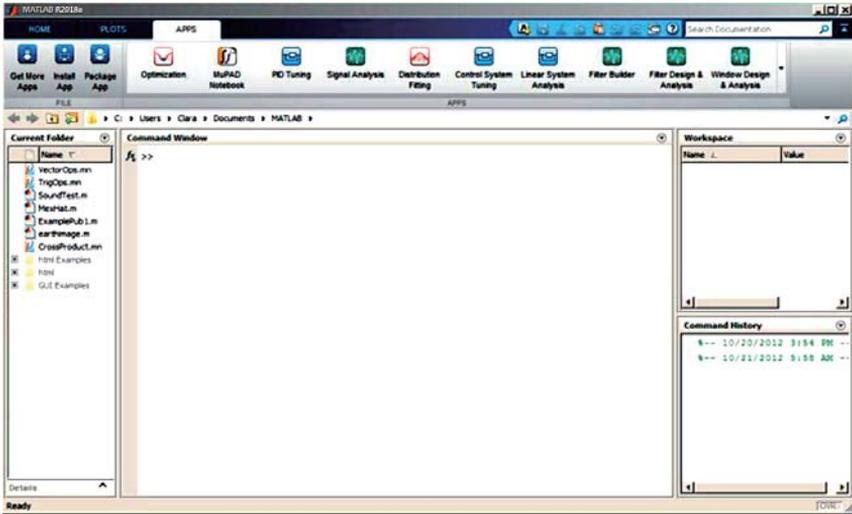


图 1-4 MATLAB 2018a 中全新的桌面工具栏

在 Desktop 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的运行方式与早期版本的 MATLAB 一样。另外两个标签是 PLOTS 和 APPS。这些特性可以通过鼠标指针指向并单击以使用 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。这样做可在 Desktop 的中心打开 Editor，如图 1-5 所示。注意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。VIEW 标签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。

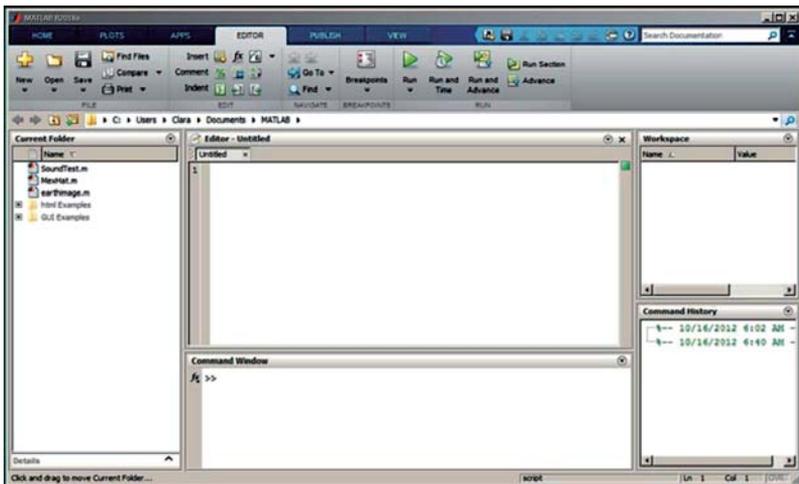


图 1-5 在默认位置打开的 Editor，位于桌面的中央

我们先考虑 Editor 的使用。在 Editor 中输入以下脚本：

```
% Example of one of the matrix inversion methods available in MATLAB
clear;clc
% 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
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 =
    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 矩阵相应条目进行逻辑对比时，上例中的 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显示在工作区中。

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 R2018a 中其他新特性的示例。

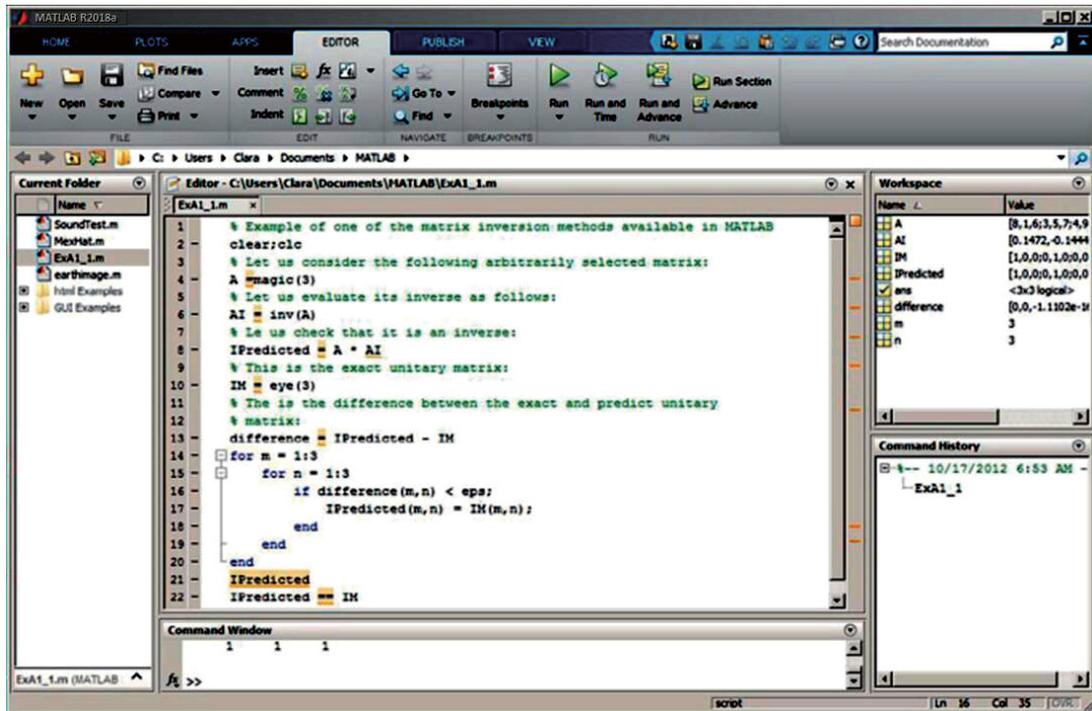


图 1-6 本节第一个示例创建并执行的示例脚本

## 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 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。要打开帮助文件，将鼠标移到 Desktop 顶部的问号上。左键单击问号？，Help 窗口就会打开。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 R2018a 中可搜索文档的新格式。将这个文档和那种可以自己发布的文档进行对比。通过下面这个简单示例，将看到创建 MATLAB 文档是多么容易。

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:
%
```

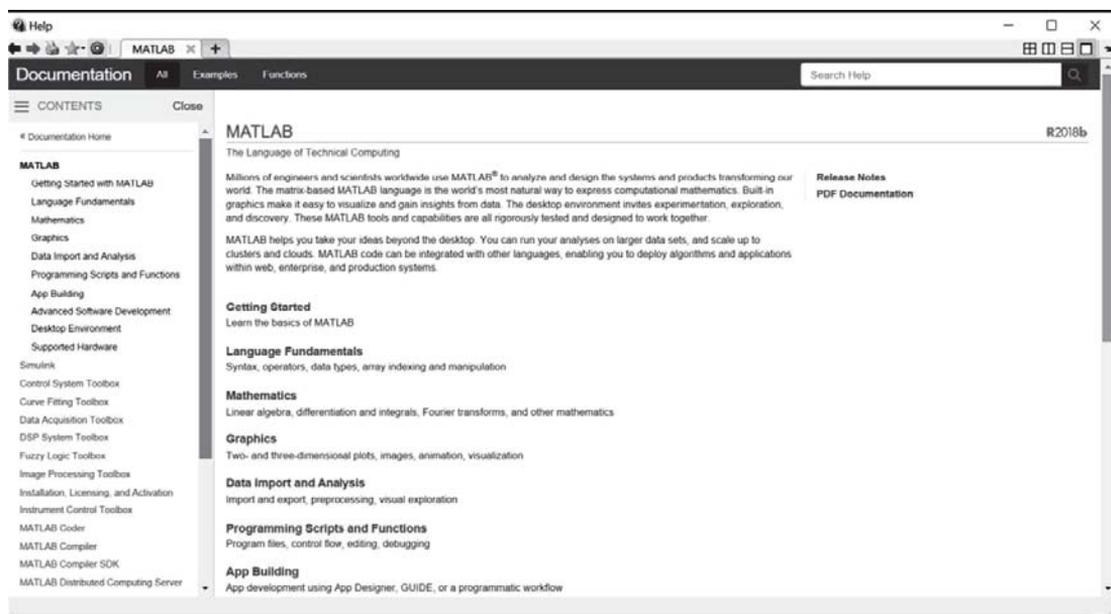


图 1-7 MATLAB 在线文档中的一页

然后, 单击位于 Insert Inline Markup 组中的  $\Sigma$  Inline LaTeX 按钮。这会将式子  $x^2 + e^{\pi i}$  加入脚本中。在这个方程的后面, 添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。

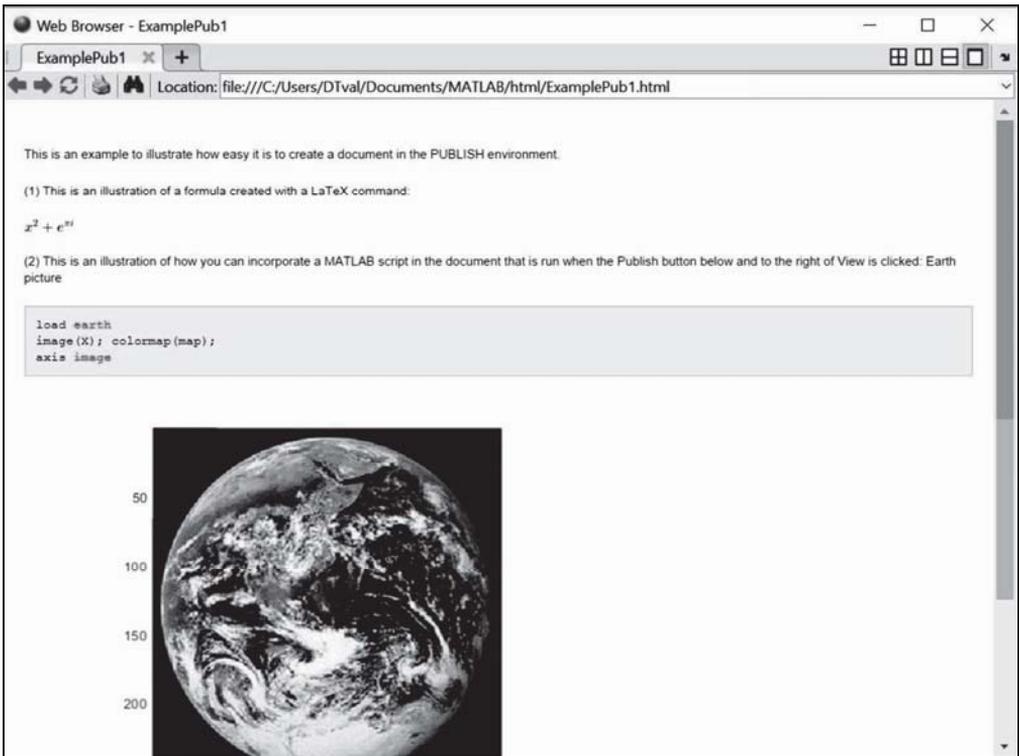


图 1-8 在发布环境中创建的示例文档

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 活动脚本中的符号

Symbolic Math Toolbox (符号数学工具箱)是一款非常有用的工具，有助于进行符号数学分析。它很方便，如果在 Live-Script 环境中使用这个工具箱，结果就更容易显示出来。在这类文件中输入代码的部分，实际上与 Work Space 或普通脚本文件中输入的代码一样。但是，其显示比在 Desktop 的 Work Space 窗口中显示的输出赏心悦目得多。本节的示例会证明这一点。

在第一组示例中，展示了如何使用 Symbolic Tools 先对一个函数求导，然后对结果进行积分。首先在 Home 工具栏中的 Home 下方单击 Live Editor 图标，打开一个 Live Editor 文件，Live Editor 就会显示在 Desktop 的中间，如图 1-9 所示。通过所创建文件顶部的灰框，把 MATLAB 脚本输入到这个编辑器中。

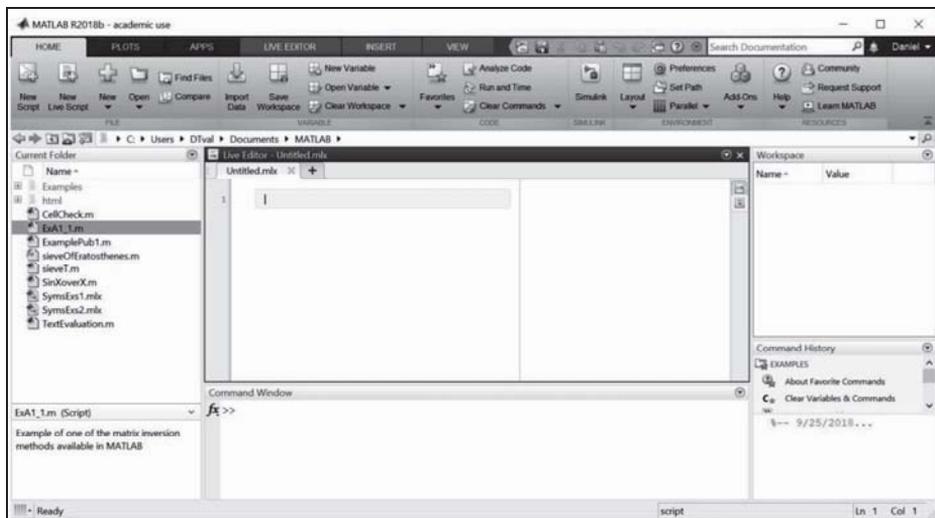


图 1-9 Live Editor 窗口

图 1-10 中的简单示例演示了 Symbolic Tools 进行微积分的应用。图 1-11 中的示例演示了如何利用 MATLAB 中的工具功能研究工作中可能遇到的函数图形。一个是  $\text{sech}^2(x)$  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。把这个函数选作示例，是因为作者在工作中遇到的第二个函数就是它。展示它的目的是为了说明在工程问题的计算分析中，很容易探索其中相当复杂的函数的一些特性。作为练习，打开帮助文档，找到其他函数、教程和示例。自 MATLAB 首次推出以来的许多年里，帮助文档有了很大改进。

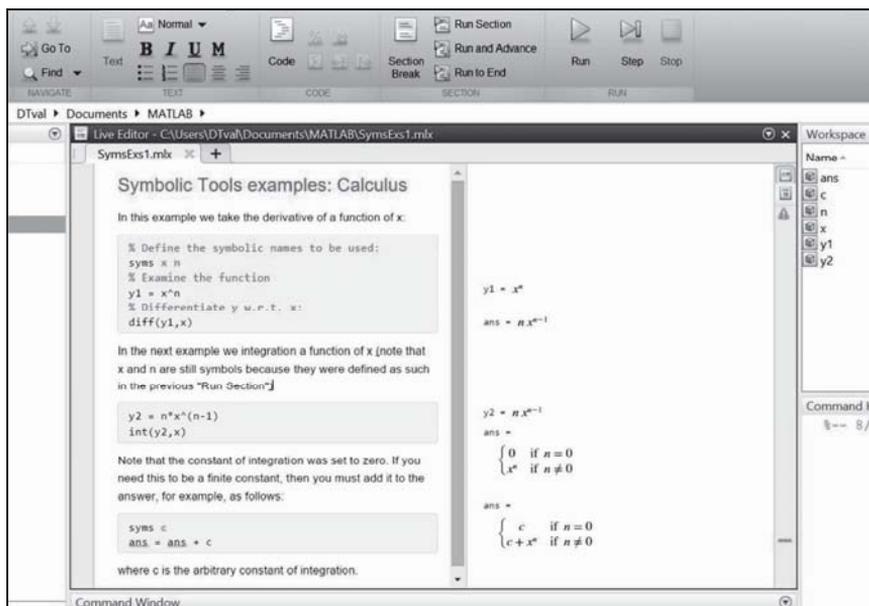


图 1-10 微积分问题的符号分析示例

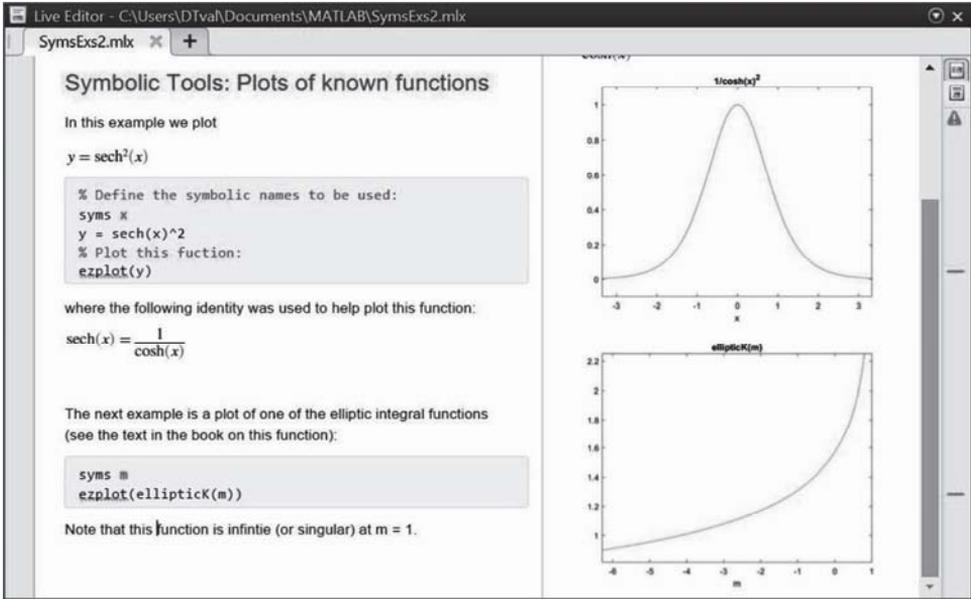


图 1-11 STEM 领域中各种重要函数的符号分析示例

## 1.2.4 APPS

在 EDITOR 标签左边单击 APPS 标签，就可以看到 MATLAB 工具套件附带的应用程序。在 MathWorks 中还有很多其他 APPS。此外，还可以创建自己的 APPS。因此，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。可以开发自己的工具箱、APPS，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)` 以生成 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  $n(n^2+1)/2$ 。

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 1-12)，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>
>> mesh(z); <Enter>
```

`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>
```

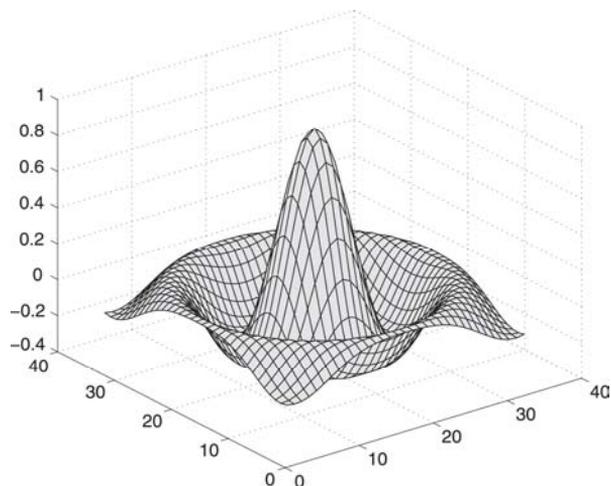


图 1-12 墨西哥帽

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 1-12 所示墨西哥帽图形的扩展。它使用 for 循环，进行从  $n=-3$  到  $n=3$ ，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 0.05s 的延迟，使动画放慢，因此这张图片每隔 0.05s 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 中用多种方式检查声音。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) 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, 通过下拉菜单, 单击 Editor 窗口顶部的 File, 再单击 Close Editor, 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。

稍后将使用 edit 命令说明如何创建类似 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, 就可以自己创建 M 文件。关于在 MATLAB 环境中创建程序的更多细节, 参见第 2 章。

## 1.3 示例程序

1.1 节列举了一些简单示例, 分析了如何通过 MATLAB 提示符后输入单独的命令或语句以使用 MATLAB。然而, 你可能需要求解一些 MATLAB 无法仅用一行命令解决的问题, 比如找出二次方程的根(考虑所有特殊情况)。解决这样问题的语句集合就称为程序。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——在第 2 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  $6\pi$  范围内的图像, 如图 1-13 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:

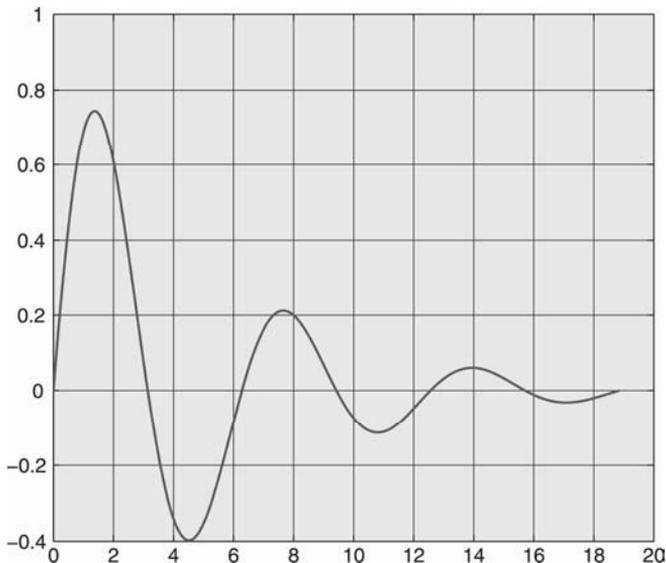


图 1-13  $e^{-0.2x}\sin(x)$

从 MATLAB 桌面选择 File | New | Script, 或者单击桌面工具栏的 New File 按钮(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 Untitled 窗口。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\*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'将绘制黑色图形, 以示区别。如果更喜欢红色, 可以将'k'改为'r'以生成红色图形。

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。这个过程称为拖放。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 的过程称为“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, 假如在银行存了 1000 美元。每年利息复合增长率为 9%。一年后, 银行的存款余额将会是多少? 如果想写一个 MATLAB 程序以计算存款余额, 那么原则上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 按钮。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
```

注意，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。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)设置为保存该脚本的文件夹(或目录)。可以通过如下操作改变当前文件夹，在工具栏中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)，然后选择一处新位置保存和执行文件(例如，如果要为不同的课程创建文件，可以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，然后右击。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

现在详细讨论复合利率程序是如何运行的。

MATLAB 系统在技术上称为解析器(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？

第 2 章会回答这些问题。在编写更完整的程序之前，还需要介绍其他一些基本概念。第 2 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以使用 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** 1.2.5 节中给出一个脚本，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后，尝试修改它。

- a) 将  $n$  的最大值从 3 改为 4，然后执行脚本。
- b) 将 `pause` 函数中的延迟时间从 0.05 改为 0.1。
- c) 将命令行 `z = sin(r.*n)./r;` 改为 `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(不仅是实数)。因此不应该将符号  $i$  用于表示索引或变量名。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 D 中。