



经典译丛



人工智能与智能系统

Data-Driven Modeling and Scientific
Computation: Methods for Complex Systems
and Big Data

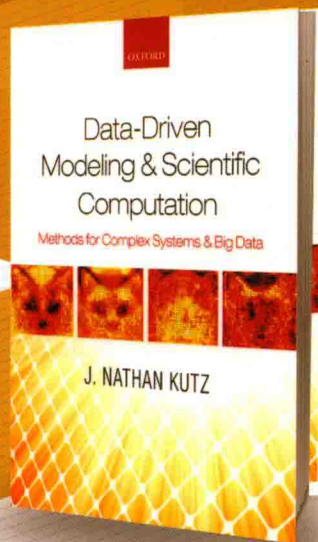
数据驱动建模 及科学计算

——复杂系统和大数据处理方法

Data-Driven Modeling and Scientific Computation
Methods for Complex Systems and Big Data

【美】 J. Nathan Kutz 著

吕丽刚 王立华 黄红坡 于晓 等译
任鸿 审校



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

经典译丛·人工智能与智能系统

数据驱动建模及科学计算

——复杂系统和大数据处理方法

Data-Driven Modeling and Scientific Computation
Methods for Complex Systems and Big Data

[美] J. Nathan Kutz 著

吕丽刚 王立华 黄红坡 于晓 等译

任鸿 审校

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

由于数据在各个科学领域的增值, 新兴的数据分析技术正在以难以置信的速度发展。大数据集目前通常在科学上用于激励发展数学技术和计算方法, 用来帮助分析、解释和释疑数据在科学应用环境中的意义。本书的特定目的是集成标准的科学计算方法和数据分析技术。通过这种方式, 本书还引入了统计学、时频分析和降维处理等方面的重要思想。全书共分四部分(26章), 前三部分详细讲解各类数学运算与分析方法, 第四部分重点讲解如何应用数学方法进行动态复杂系统分析与大数据处理。其中, 第一部分讨论数学、矩阵分析和概率论的主要数据计算方法及结果可视化; 第二部分讨论微分方程计算与建模; 第三部分讨论各种数值分析与计算方法并进行比较, 引入动态复杂系统概念; 第四部分讲解复杂系统与大数据分析方法和处理模型的建立。

该书适合作为本科生的参考阅读书籍, 也适合作为科学计算分析等相关专业研究生的教材。从事数据计算方法、计算与建模等相关工作的研究者也能从中有所获益。

Copyright © 2013 by J. Nathan Kutz.

This translation of *Data-Driven Modeling and Scientific Computation: Methods for Complex Systems and Big Data*, First Edition, originally published in English in 2013, is published by arrangement with Oxford University Press and is for sale only in the territories of Mainland China not including Hong Kong SAR, Macau SAR and Taiwan.

本书中文简体版专有版权由美国 Oxford University Press 授权电子工业出版社。未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。此版本仅限在中国大陆发行与销售。

版权贸易合同登记号 图字: 01-2014-5134

图书在版编目(CIP)数据

数据驱动建模及科学计算: 复杂系统和大数据处理方法 / (美)库茨(Kutz, J. N.)著; 吕丽刚等译.

北京: 电子工业出版社, 2017.1

(经典译丛·人工智能与智能系统)

书名原文: *Data-Driven Modeling and Scientific Computation: Methods for Complex Systems and Big Data*

ISBN 978-7-121-26596-9

I. ①数… II. ①库… ②吕… III. ①数据库系统—建立模型 ②数据库系统—科学计算 IV. ①TP311.13

中国版本图书馆 CIP 数据核字(2015)第 155835 号

策划编辑: 马 岚

责任编辑: 马 岚 特约编辑: 马晓云

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 29.5 字数: 755 千字

版 次: 2017 年 1 月第 1 版

印 次: 2017 年 1 月第 1 次印刷

定 价: 89.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010)88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: classic-series-info@phei.com.cn。

译者序

大数据的飞速发展彻底改变了我们的工作、学习和生活。未来几年，各种数据源仍会持续爆炸式增长，对大数据的研究也进入了快速发展时期。大数据研究的目的是发展大数据技术并将其应用到相关领域，通过处理巨量数据促进其突破性发展。及时掌握国内外相关领域研究成果和应用技术，迫切需要内容全面、权威的教材。本书为合适之选。

本书以作者多年的科学研究和工程实践为基础，对 MATLAB 软件、数据处理理论和数据分析技术等专业知识进行了较为全面系统的阐述，并就各种专业知识在实际科研、工程技术应用方面进行了讲解，提供了一些解决现实工程技术问题必要的软件工具，便于学习研究。

全书共四个部分(26章)。其中第一部分(第1章至第6章)介绍 MATLAB 语言基础、编程思想和解决基本问题的方法；第二部分(第7章至第11章)介绍了常微分和偏微分方程数值计算方法中高层次的理论，系统地为广大科学计算爱好者们提供了应用程序的基础理论框架；第三部分(第12章至第23章)从独特的视角提供了数据分析方法，即利用数据统计、时频分析、降维运算这三个思想来解决各种复杂问题；第四部分(第24章至第26章)是科学应用部分，其核心思想是运用有限差分、谱方法、有限元方法等数值方法，求解三体问题、量子力学等实际的复杂问题。

本书的翻译、审校工作得到相关领域专家的关注与指导。邵志平教授、孙鹏教授和张少华教授等专家学者均对书稿的翻译做了大量指导工作，提出了宝贵的修改意见。在此，译者表示衷心的感谢。

本书翻译的具体分工如下：吕丽刚翻译了前言、学习指南、术语对照表等内容，并负责全书的统稿和校对；王立华、周宏升和丛培荣翻译了第一部分和第二部分，常广建、赵铭和于晓翻译了第三部分的部分章节；黄红坡、刘永锋和刘丽巧翻译了第三部分的部分章节和第四部分。全书由任鸿审校。

大数据为目前最新研究领域之一，无论在理论研究还是工程技术应用方面，都有很多问题有待进一步探讨解决。加之译者水平有限，翻译时间比较仓促，本书的不当甚至错误之处在所难免，恳请各位专家和广大读者批评指正。

前 言

科学计算普遍存在于物理、生物和工程领域。今天，专业的计算方法对研究分析能力有重要影响。尽管数学家们提供了大量的数据分析教程，但此类教程往往更适合数字专业的人看懂。可视化数据分析技术仍是必不可少的，因为它为数据分析教材与课程提供了基础。可视化数据分析被熟练地应用于实践，并解决实际问题，传统的数学计算方式需要一年的工作才能达到与其相当的水平。

MATLAB 已经在科学计算软件领域确立了其主导地位。其内嵌的运算规则使得科学计算从技术细节转换到整体的算法实现和解决方案上。重复繁重的计算任务产生了 MATLAB 线性代数运算包、快速傅里叶变换函数运算包和有限元(偏微分方程)运算包。这些函数是解决技术问题的关键所在。当然，提供简单的说明是任何函数应用所必需的，作为应用程序的参考材料，可以为用户提供查阅服务。

本书的最终目的是为学生开发一套可信的解决计算问题的工具。特别要提出的是，在本书结束部分，学生应该能够解决椭圆、双曲线或抛物线类型的一维至三维的所有问题，起码他们会有大量关于如何解决问题的知识、信息和参考文献的辅助信息。类似地，就目前的数据分析框架而言，为统计学、时序分析和降维处理(奇异值分析)等以数据处理为核心的问题提供了更便捷的解决途径。

总而言之，本书可以为读者提供一种通过数据驱动建模来解决各种复杂问题的思路。

致谢

本书第一部分内容来源于与 Dave Muraki 的一系列谈话，然后逐步发展成为科学计算课程的主要内容，其目的在于为生物和物理工程学的学生提供通用、有效的课程。过去两年中，本书内容得到进一步丰富，包括数据分析方法，并支撑了本书的主要内容。正如大家所熟知的，本书数据分析的主要部分极大地受益于 Emmanuel Candes, Yannis Kevrekidis 和 Clancy Rowley 的出色工作。我还获益于早期与 James Rossmanith, 以及与 Peter Blossey 和 Sorin Mitran 的大量讨论，以及近期与 Steven Brunton, Edwin Ding, Jsshua Proctor, Peter Schmid, Eli Shlizerman, Jonathan Tu 和 Matthew Williams 关于维度降低方法的讨论。同时，Leslie Butson, Sarah Hewitt 和 Jennifer O'Neil 也为编辑本书给予了大量帮助，正是由于他们的这些努力，增加了本书的可读性、适用性和准确性。另外，特别需要感谢大量学生读者，正是他们提出如此多的批评性建议和重要的反馈意见，才增加了本书的发行量、可读性和正确率。当然，由于水平有限且时间仓促，书中错误在所难免，敬请谅解。

学习指南

本书分为三个数值方法部分和一个应用部分。实际上，本书各部分针对不同读者设计，不同的用户会关注适合他的部分。本书所有部分均经过本科毕业生和许多学生仔细校对和检查，以便提高整体可读性和实用性；本书的应用部分可以帮助学生解决更广泛的现实问题。就我个人而言，感觉很多其他的 MATLAB 书籍设计的问题和对关键问题的阐述缺乏真正的实用性和指导性。本书特点详述如下。

本科课程中的科学计算

本书第一部分汇集了目前华盛顿大学一二年级工科和物理专业的学生们最为关注的资料。鉴于在工程项目中使用的是 MATLAB 语言，我们首先介绍该语言的基本概念和理论思想。其次，介绍利用该语言编译的程序实例，如最小二乘、导数、积分等。最后讲解如何用 MATLAB 处理较为常见的简单问题，如导入/导出数据等。因此第 1 章至第 6 章介绍 MATLAB 语言基础、基础编程 (if 和 for 循环语句) 思想和解决基本问题的思路。最后在第 6 章阐述了微分方程。快速高效地解决微分方程是许多高年级学生在航空航天(三体问题)工程、电子工程等方面的主要课程。因此，本书第一部分适合初学科学计算的读者。

研究生课程中的科学计算

本书还为研究生(或者高年级本科生)提供科学计算方法的基础介绍和高级概述。数学分析在传统研究生课程中，重点通常都集中在数学基础理论与实现方面，而目前许多数据分析课程极少有面向工程、生物和物理专业学生的讲解。本书的第二部分，第 7 章到第 11 章为读者介绍了解决微分和偏微分方程计算方法中高层次的理论，系统地为广大科学计算爱好者们提供了应用程序的基础理论框架，其中考虑了有限差分、光谱和有限元等关键问题。第 7 章是研究生学习的起点。一方面，学生可以根据个人情况跳过这本书的第一部分，从研究生阶段开始学习科学计算。另一方面，本书的第一部分包含了部分研究生的简单工程设计的内容，比如创建影像、导入/导出数据或提供编程架构。因此，本书的第一部分对研究生具有研究参考价值。

研究生课程中数据分析的计算方法

本书第一、二部分提供了一套科学计算的标准化处理流程，用大量简单的实例说明初级和中级数值方法问题的偏微分方程等解决方案。本书第三部分(第 12 章至第 23 章)则从更独特的视角提供了数据分析方法。该部分的目的是向研究生(或高年级本科生)介绍迅猛发展的数据分析学科，该学科几乎应用于所有的科学研究领域，发展极为迅速。各学科遇到的大数据问题极大地鼓舞了他们对旧理论的解释及对新理论的设想。这一部分总结了其中有代表性的思想：(1) 数据统计；(2) 时频分析；(3) 降维运算。提供有意义的数据分析方法是当今任何科学领域需要面对的问题。这是一个令人振奋的学科，这一部分也由许多实例组成，这三个思想可用于指导解决各种问题。与第二部分一样，本书第一部分补充了学生的背景知识，是全书总体结构不可或缺的重要部分。

计算方法参考指南

本书除了作为大学科学计算及数据分析的教材外，也可作为参考指南。这本书提供了大量应用实例和一些关键、经典问题的另类解决思路。使应用程序和理论相结合，提供了一个可以更加高效快捷提高自己相关技能的解决方案。在与读者们交流的过程中，我听到最多的就是学生们发现本书有助于完成作业。我认为这是由于他们在使用本书后，既获得了 MATLAB 的理论知识，又能在实际应用中借鉴本书中的代码，从而降低了使用该语言的门槛。另外，本书也介绍了很多高级 MATLAB 子例程，对这些子例程的应用，可以使读者们在编程方面少走弯路。

理论联系实际

本书最重要的部分是科学应用部分(第 24 章至第 26 章)，该思想的核心是用理论解决实际问题。有时我不得不开发复杂的程序以解决复杂的问题，而教材提供的理论简化了这些复杂的问题，更容易进行问题分析。教材中使用这种方法是可以理解的，因为这就是教学目的。我们最终的目标是培养构建代码的专业人员，他们终究会遇到复杂的问题。所以实际应用问题的选择要足够广泛，教师们也应该有条件选择自己感兴趣的東西。另外，本部分中的每个问题都有精心布置的背景，使得这些问题更好地与实际应用相联系。这些问题的难度依次增加，但这些问题仍然适用于任何读者，无论是低年级本科生、高年级本科生还是研究生都可以从中获益。本部分对于不同层次的读者唯一的区别在于他们可以读到本书的哪一部分。如果初学者想要更深入地了解，也可以读更深层次的部分。另外，本部分还提出很多经典问题，如：三体问题、量子力学等。希望这类问题的提出能让学生和教师们一样感兴趣，故本书可以作为理论联系实际的计算工具。

目 录

第一部分 基本计算和可视化

第 1 章	MATLAB 概述	2
1.1	向量和矩阵	2
1.2	逻辑、选择和循环	6
1.3	迭代法: Newton-Raphson 方法	9
1.4	函数调用, 输入/输出及调试	12
1.5	绘图和数据的导入/导出	16
第 2 章	线性系统	22
2.1	直接方法求解 $Ax=b$	22
2.2	迭代法求解 $Ax=b$	25
2.3	梯度下降(最速下降)法求解 $Ax=b$	28
2.4	特征值、特征向量和可解性	32
2.5	特征值、特征向量应用与人脸识别	36
2.6	非线性系统	41
第 3 章	曲线拟合	45
3.1	最小二乘拟合法	45
3.2	多项式拟合和样条插值	48
3.3	基于 MATLAB 的数据拟合	51
第 4 章	数值微积分	57
4.1	数值微分	57
4.2	数值积分	61
4.3	数值微分和积分计算	64
第 5 章	基本优化	69
5.1	无约束最优化	69
5.2	无约束最优化(微分方法)	73
5.3	线性规划	77
5.4	单纯形法	81
5.5	遗传算法	84
第 6 章	可视化	88
6.1	定制图形和基本的二维绘图	88
6.2	高级二维和三维绘图	92
6.3	电影及动画	96

第二部分 常微分方程和偏微分方程

第 7 章 常微分方程初边值问题	100
7.1 初值问题: 欧拉方法、Runge-Kutta 方法和 Adams 方法	100
7.2 时间步进算法的误差估计	105
7.3 高级时间步进算法	109
7.4 边值问题: 打靶法	111
7.5 打靶法的实现和收敛性研究	116
7.6 边值问题: 直接求解与松弛	120
7.7 使用 MATLAB 求解边值问题	122
7.8 线性算子及谱的计算	125
第 8 章 有限差分方法	131
8.1 有限差分离散	131
8.2 求解线性方程组 $Ax = b$ 的高级迭代方法	135
8.3 快速泊松解子: 傅里叶变换	136
8.4 线性方程组求解技术的比较: 经验法则	138
8.5 克服计算困难	142
第 9 章 时间和空间步进方式: 线性法	145
9.1 基本时间步进方法	145
9.2 时间步进方法: 显式和隐式方法	148
9.3 稳定性分析	151
9.4 比较时间步进方法	154
9.5 算子分裂技术	157
9.6 优化计算性能: 粗略估计	159
第 10 章 谱方法	164
10.1 快速傅里叶变换和余弦、正弦变换	164
10.2 切比雪夫多项式和切比雪夫变换	167
10.3 谱方法的实现	170
10.4 带滤波的伪谱方法	171
10.5 边界条件和切比雪夫变换	175
10.6 实现切比雪夫变换	178
10.7 计算谱: Floquet-Fourier-Hill 方法	182
第 11 章 有限元法	187
11.1 有限元法基础	187
11.2 有限元离散和边界	191
11.3 使用 MATLAB 求解偏微分方程	194
11.4 MATLAB 偏微分方程工具箱	198

第三部分 数据分析计算方法

第 12 章 统计方法及其应用	204
12.1 概率论基本概念	204
12.2 随机变量和统计概念	208
12.3 假设检验及其统计意义	215
第 13 章 时频分析: 傅里叶变换与小波理论	220
13.1 傅里叶级数及傅里叶变换	220
13.2 FFT 的应用: 雷达探测和滤波	226
13.3 FFT 的应用: 雷达探测与平均法	230
13.4 时频分析: 窗口傅里叶变换	235
13.5 时频分析与小波理论	239
13.6 多分辨率分析与小波基函数	245
13.7 MATLAB 中的谱图及 Gábor 变换	248
13.8 MATLAB 滤波器设计和小波工具箱	253
第 14 章 图像分析处理	261
14.1 图像分析基本概念	261
14.2 图像降噪的线性滤波	264
14.3 散度及图像处理	268
第 15 章 线性代数及其奇异值分解	273
15.1 奇异值分解基础	273
15.2 广义 SVD	276
15.3 主成分分析 (PCA) 简介	281
15.4 主成分分析, 对角化及 SVD	283
15.5 主成分及适当正交模型	286
15.6 稳定 PCA 模型	292
第 16 章 独立成分分析	299
16.1 独立成分的概念	299
16.2 图像分离问题	303
16.3 图像分离及 MATLAB 应用	307
第 17 章 图像识别: 基本的机器学习	312
17.1 识别猫狗	312
17.2 SVD 和线性判别分析	315
17.3 MATLAB 识别猫狗	320
第 18 章 压缩感知理论基础	323
18.1 最小二乘拟合之外的 L^1 范数	323
18.2 信号重构和规避奈奎斯特	328
18.3 稀疏采样的数据 (图像) 重构	333

第 19 章	偏微分方程降维	339
19.1	偏微分方程的模态扩展技术	339
19.2	PDE 动力学的正确(最优)基	342
19.3	PDE 全局范数的分叉结构	346
19.4	POD 方法及其对称性/不变性	353
19.5	POD 中使用稳定 PCA	358
第 20 章	动态模式分解	363
20.1	动态模式理论	363
20.2	动态特性上 DMD 与 POD 的比较	366
20.3	DMD 应用	369
第 21 章	数据同化方法	373
21.1	数据同化理论	373
21.2	数据同化、采样和卡尔曼滤波	376
21.3	洛伦兹方程的数据同化	378
第 22 章	方程自由建模	384
22.1	多尺度物理学: 方程自由方法	384
22.2	方程自由建模的提升和限制	387
22.3	方程自由时空动力学特征	391
第 23 章	复杂动力学系统: 降维合并、压缩感知和机器学习	394
23.1	复杂系统数据合并方法	394
23.2	实现一个动力学系统工具库	397
23.3	圆柱绕流: 一个典型案例	403

第四部分 科学应用

第 24 章	微分方程	410
24.1	神经科学和霍奇金-赫胥黎模型	410
24.2	天体力学和三体问题	413
24.3	大气运动和洛伦兹方程	415
24.4	量子力学	418
24.5	电磁波导	420
第 25 章	偏微分方程的应用	422
25.1	波动方程	422
25.2	锁模激光	424
25.3	玻色-爱因斯坦凝聚体	429
25.4	平流传播和大气动力学	432
25.5	扩散-反应系统介绍	436
25.6	螺旋桨上的稳态流	440

第 26 章 数据分析应用	443
26.1 乐谱分析和 Gábor 函数转换	443
26.2 通过过滤和扩散进行图像降噪	444
26.3 振荡量和降维	446
26.4 音乐风格识别	447
参考文献	449
MATLAB 命令索引	454
术语对照表	457

第一部分 基本计算和可视化

第 1 章 MATLAB 入门	1
1.1 初识 MATLAB	1
1.2 创建脚本	1
1.3 运行脚本	1
1.4 帮助系统	1
1.5 保存与加载	1
1.6 小结	1

第1章 MATLAB 概述

第一部分 基本计算和可视化

第1章 MATLAB 概述

第2章 线性系统

第3章 曲线拟合

第4章 数值微积分

第5章 基本优化

第6章 可视化

第 1 章 MATLAB 概述

本书前 6 章内容主要介绍数据操作、逻辑表示和数据绘图的基本知识，其他章节重点介绍如何使用好这些基本技能。

1.1 向量和矩阵

熟练掌握矩阵和向量的操作至关重要，是精通 MATLAB 的前提和基础。本节介绍矩阵和向量的结构组成和基本操作。我们首先来理解一下行向量和列向量的结构。通俗地说，向量就是矩阵的一个子集，它只有一列或只有一行，是一种特殊形式的矩阵。

通过输入下列命令，即可生成一个行向量。

```
>>x = [1 3 2]
```

执行这条命令，就可以生成一个行向量。

$$\mathbf{x} = (1 \ 2 \ 3) \quad (1.1.1)$$

行向量的数据水平方向排列。相反，列向量的数据垂直方向排列，要生成一个列向量，有两种方法，可以输入下列两条命令语句中的任意一条。

命令语句中的分号表示终止当前行，并换到下一新行。例如，

```
>>x = [1; 3; 2]
```

此外，在 MATLAB 中，输入回车符也可创建一个新行。下列命令的作用与上一句是等同的。

```
>>x = [1  
      3  
      2 ]
```

以上这两种方法都可以生成一个列向量，即

$$\mathbf{x} = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} \quad (1.1.2)$$

所有的行向量和列向量都可以采用以上方法生成。

向量还可以通过使用冒号生成。例如，下一命令行就可以生成一个从 0 到 10，步长为 1 的行向量。

```
>>x = 0:1:10
```

生成的行向量如式 1.1.3 所示。

$$\mathbf{x} = (0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10) \quad (1.1.3)$$

注意，若输入命令：

```
>>x = 0:2:10
```

就会生成一个从 0 到 10，步长为 2 的行向量。生成的行向量如下式所示：

$$\mathbf{x} = (0 \ 2 \ 4 \ 6 \ 8 \ 10) \quad (1.1.4)$$

数值递进的步长不一定是整数值。可以输入下述命令生成从 0 到 1，步长间隔为 0.2 的行向量。

```
>>x = 0:0.2:1
```

生成的行向量如下式所示：

$$\mathbf{x} = (0 \ 0.2 \ 0.4 \ 0.6 \ 0.8 \ 1) \quad (1.1.5)$$

综上所述，可以发现使用冒号操作符生成向量的基本语句结构为：

```
>>x = a:h:b
```

其中， a 表示初始值， b 表示末尾值， h 表示增量步长。

需要注意的是参数 b 和参数 h 的关系问题，有可能从初始值 a 开始以增量步长为 h 的值递增，但末尾值却不是 b 。例如：

```
>>x = 0:2:9
```

执行命令后，会生成一个行向量，

$$\mathbf{x} = (0 \ 2 \ 4 \ 6 \ 8) \quad (1.1.6)$$

从初始值 $a=0$ 开始，步长为 $h=2$ 递增，无论如何也达不到末尾值 $b=9$ 。因此，参数 b 的数值可以看成这个向量的上限值，而不是末尾值。

最后一种情况，如表达式中没有标出步长 h 这个参数，则默认其步长值为 1，例如，

```
>>x = 0:4
```

生成的向量为

$$\mathbf{x} = (0 \ 1 \ 2 \ 3 \ 4) \quad (1.1.7)$$

若无特别说明，步长 h 的数值默认为 1。在整数运算和循环语句中，每一次迭代或循环，就相当于变量 a 加 1，从而变为 $a+1$ 。

矩阵的创建也很简单，下面看一个指定行数 (N) 和列数 (M) 的矩阵，这个矩阵称为 $N \times M$ 维矩阵，比如 3×3 矩阵

$$\mathbf{A} = \begin{pmatrix} 1 & 3 & 2 \\ 5 & 6 & 7 \\ 8 & 3 & 1 \end{pmatrix} \quad (1.1.8)$$

可以通过输入分号操作符创建矩阵，如下列命令可创建矩阵 \mathbf{A} 。

```
>>A = [1 3 2; 5 6 7; 8 3 1]
```

也可以利用回车符创建 \mathbf{A} 。

```
>>A = [1 3 2
      5 6 7
      8 3 1]
```

如果矩阵的行数 N 和列数 M 相等 ($N = M$)，这个矩阵就称为方阵。

若想获取一个指定矩阵或向量中的元素，就必须知道该元素在矩阵或向量中所处的位置。利用指定元素所在的行和列，就可以获得该位置所对应的元素数值。比如，下式就代表矩阵 A 的第 i 行、第 j 列的元素：

$$A = (i, j) \quad (1.1.9)$$

其中， i 表示该元素所在的行， j 表示该元素所在的列。如果要获取式 (1.1.8) 中矩阵的第二行、第 3 列的元素 (该元素对应的数值为 7)，可以使用命令

```
>>x = A[2,3]
```

该命令执行后，输出结果 $x = 7$ 。用冒号操作符可获得一整行的元素：

```
>>x = A[2,:]
```

执行该命令，可以获取整个第 2 行的元素，返回一个行向量，

$$\mathbf{x} = (5 \ 6 \ 7) \quad (1.1.10)$$

同理，也可以获取某一列的向量，

```
>>x = A[:,3]
```

执行该命令，可以获取整个第 3 列的元素，返回一个列向量：

$$\mathbf{x} = \begin{pmatrix} 2 \\ 7 \\ 1 \end{pmatrix} \quad (1.1.11)$$

这里所用到的冒号操作符，其功能与前面相同，都是用来创建向量的。实际上，冒号是 MATLAB 中功能最为强大的操作符之一。

通过冒号操作符，可以实现更多、更强大的数据获取和操作功能。下面以一个 4×4 矩阵来说明这些技巧的使用。

$$B = \begin{pmatrix} 1 & 7 & 9 & 2 \\ 2 & 3 & 3 & 4 \\ 5 & 0 & 2 & 6 \\ 6 & 1 & 5 & 5 \end{pmatrix} \quad (1.1.12)$$

```
>>x = B[2:3,2]
```

上一行命令代表从矩阵 B 中抽取出第 2 列中第 2 行到第 3 行的元素，创建一个新的列向量，

$$\mathbf{x} = \begin{pmatrix} 3 \\ 0 \end{pmatrix} \quad (1.1.13)$$

执行命令：

```
>>x = B[4,2:end]
```

其作用就是从矩阵 B 中抽取第 4 行的第 4 列到最后一列的所有元素，并创建一个新的行向量。

$$\mathbf{x} = (1 \ 5 \ 5) \quad (1.1.14)$$

我们也可以从矩阵中指定要获取的行的范围和列的范围。输入命令语句

```
>>C = B[1:end-1,2:4]
```


其功能为从矩阵 B 中读取从第 1 行到倒数第 2 行的从第 2 列到第 4 列的元素，这样就创建了一个新的矩阵

$$C = \begin{pmatrix} 7 & 9 & 2 \\ 3 & 3 & 4 \\ 0 & 2 & 6 \end{pmatrix} \quad (1.1.15)$$

最后，我们再来学习如何使用转置符号，将一个行向量转为列向量。同理，使用该操作符也可以将一个列向量转为行向量。

输入命令语句

```
>>D = [B(1,2:4); B(1:3,3).']
```

创建一个新的矩阵 D ，该矩阵的第一个行向量由矩阵 B 的第 1 行的第 2 列到第 4 列元素组成；该矩阵通过分号操作符生成第二个行向量，是对矩阵 B 的第 3 列的第 1 行到第 3 行元素所组成向量进行转置 (') 生成的，这样就得到新的矩阵 D 为

$$D = \begin{pmatrix} 7 & 9 & 2 \\ 9 & 3 & 2 \end{pmatrix} \quad (1.1.16)$$

对于转置运算，需要关注的很重要的一点就是转置符的顺序问题。尤其对于包含复数元素的向量，这些元素必须写在转置符号 (') 之前。

$$x = \begin{pmatrix} 3+2i \\ 1 \\ 8 \end{pmatrix} \quad (1.1.17)$$

例如，如果对式 (1.1.17) 的列向量进行转置运算，该列向量中元素 $3+2i$ 的 i 表示虚数单位， $i=\sqrt{-1}$ ，输入命令

```
>>y = x.'
```

执行此命令后，创建一个新的行向量

$$y = (3+2i \ 1 \ 8) \quad (1.1.18)$$

然而，若输入以下命令：

```
>>y = x'
```

执行此命令后，将创建一个新的行向量

$$y = (3-2i \ 1 \ 8) \quad (1.1.19)$$

因为单独使用转置符 (') 是求该向量的共轭值，也就是说，它会更改复数虚部的正负号。

再来看一个关于向量和(或)矩阵操作的一个实例，如下式中的向量：

```
x = [-1 2 3 5 -2];
```

这个行向量中既有正数，也有负数。对于这种向量，下列命令在实际应用中十分有效，例如可以对这个向量设置元素数值的阈值规则。