


普通高等教育“十三五”规划教材

数学实验教程

Introduction to Mathematic Experiments

王宏洲 李学文 闫桂峰 李炳照 © 编著

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

数学实验教程

王宏洲 李学文 闫桂峰 李炳照 编著

 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书主要介绍如何通过 MATLAB 等数学软件工具,验证各种数学理论、概念及其特性,快速输出公式计算、数值计算的结果,并以直观的图形形式展现出来,从而帮助学习者深入认识和理解数学理论和方法。

全书共分 5 章:第 1 章简单介绍 MATLAB 的基本操作;第 2 章介绍与微积分、常微分方程等有关的一些数学实验;第 3 章介绍线性代数中的常见运算实验;第 4 章介绍一系列涉及概率论与统计学的实验;第 5 章介绍常见的最优化问题、解法和数值求解实验。

本书可供选修北京理工大学制作的在线开放课程“数学实验”的同学作为教材使用,也可作为高等数学、线性代数、概率与统计、最优化方法等大学基础数学课程教学的辅助教材。由于本书主要介绍如何用 MATLAB 来验证数学理论及其运算规则,因此基本不涉及 MATLAB 的编程技巧,学习者只需要掌握基本的命令和编程语言即可。

版权专有 侵权必究

图书在版编目(CIP)数据

数学实验教程/王宏洲等编著. —北京:北京理工大学出版社, 2019. 1

ISBN 978-7-5682-6611-6

I. ①数… II. ①王… III. ①高等数学-实验-高等学校-教材 IV. ①O13-33

中国版本图书馆 CIP 数据核字(2019)第 005775 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

(010) 82562903 (教材售后服务热线)

(010) 68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 10.75

字 数 / 260 千字

版 次 / 2019 年 1 月第 1 版 2019 年 1 月第 1 次印刷

定 价 / 32.00 元

责任编辑 / 王美丽

文案编辑 / 孟祥雪

责任校对 / 周瑞红

责任印制 / 李志强

图书出现印装质量问题,请拨打售后服务热线,本社负责调换

在信息技术飞速发展的今天，我们经常听到人们对高校教学内容和方式、方法的质疑。比如，在计算机能够解决绝大多数计算问题的情况下，高校的基础数学课程还有没有必要在解题方法和技巧上花费大量时间？对于这样的问题，我们既需要从高等教育的固有规律出发坚持某些做法，也需要从现实需要的角度出发做一些改变。

从高校基础数学课程的教学来看，教学内容中蕴含的数学思想、基础理论、基本运算规则在任何时候都是不可或缺的。与此同时，在各行各业都越来越依赖计算机软件解决问题的情况下，适当在基础课教学中引入一定的软件求解、编程计算等内容，无疑更有利于培养大学生的综合能力。这也是国内开展数学实验课程教学探索的一个重要目的。

在近 20 年，围绕数学实验的教学内容和教学方法，国内积累了大量实践经验。一般来说，各个高校都认为数学实验教学不仅可以通过计算机和数学软件，让学生自己动手，切身体会所学数学知识的基本原理和各种变化，而且可以有力地支持数学建模教学、竞赛活动。

本书的主要作用是辅助高等数学、线性代数、概率和统计、最优化方法等大学数学主干课程教学，通过介绍相关的 MATLAB 命令、算法和程序等，使大学生可以在理论学习之余，初步具备软件求解、编程计算等基本能力。对于上述数学基础课的教师而言，可以在课堂教学过程中同步使用本教材，要求学生用数学软件来了解有关定义、定理、例题在不同情形下的特点，从而达到举一反三的效果。相关的内容无须占用课堂时间来讲解，学生可以在中国大学 MOOC 网站 (<https://www.icourse163.org/>) 上注册并搜索“数学实验（北京理工大学）”，通过我们制作的同步在线开放课程来自学。

此外，书中还以例题的形式引入了大量数学建模案例，有意参加数学建模竞赛的同学可以借此学习数学建模的基本思路和软件求解方法。

本书共分5章：第1章简单介绍 MATLAB 的基本操作；第2章介绍与微积分、常微分方程等有关的一些数学实验；第3章介绍线性代数中的常见运算实验；第4章介绍一系列涉及概率论与统计学的实验；第5章介绍常见的最优化问题、解法和数值求解实验。其中，第1章、第2章前7节、第4章前5节由王宏洲编写，第2章后8节、第5章由李学文编写，第3章由闫桂峰编写，第4章后5节由李炳照编写。

编 者

目 录

CONTENTS

第 1 章 MATLAB 简介	001
1.1 关于 MATLAB 软件	001
1.2 MATLAB 中的数组及其运算	003
1.3 MATLAB 中的矩阵及其运算	004
1.4 循环和判断语句	007
1.5 MATLAB 中的函数绘图 (1)	009
1.6 MATLAB 中的函数绘图 (2)	015
习题 1	020
第 2 章 微积分实验	023
2.1 极限	023
2.2 导数	026
2.3 多元函数的偏导数	029
2.4 导数的应用 (1)	032
2.5 导数的应用 (2)	035
2.6 常微分方程的精确解	037
2.7 常微分方程 (组) 的数值解	040
2.8 数值微分	043
2.9 不定积分	046
2.10 定积分与广义积分	048
2.11 数值积分	051
2.12 重积分	055

2.13	曲线积分	061
2.14	曲面积分	064
2.15	数列求和与级数的敛散性	067
	习题 2	070
第 3 章	线性代数实验	072
3.1	矩阵的引入与矩阵的初等行变换	072
3.2	向量与向量空间	076
3.3	线性方程组	082
3.4	线性方程组的几何意义与应用	088
3.5	行列式及其应用	096
3.6	方阵的特征值和特征向量	099
3.7	方阵的相似对角化	103
3.8	二次型	109
	习题 3	114
第 4 章	概率论与统计学实验	117
4.1	古典概型与离散型随机变量	117
4.2	连续型随机变量	121
4.3	随机变量逆累计分布函数	123
4.4	随机数与随机模拟	125
4.5	常用统计量	129
4.6	矩估计	133
4.7	极大似然估计	135
4.8	区间估计	137
4.9	假设检验 - 参数检验	138
4.10	假设检验 - 非参数检验	142
	习题 4	145
第 5 章	最优化方法实验	148
5.1	线性规划求解	148
5.2	无约束优化求解	153
5.3	二次规划求解	155
5.4	约束优化问题求解	156
5.5	整数规划 (0-1 规划)	158

习题 5	159
习题答案	160
习题 1	160
习题 2	161
习题 3	161
习题 4	164
习题 5	164

第 1 章

MATLAB 简介

1.1 关于 MATLAB 软件

MATLAB 是美国 MathWorks 公司推出的数学软件，可以用来进行算法开发、数据可视化、数据分析以及数值计算。MATLAB 将数据分析、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功能集成在一个易于使用的视窗环境中，有效简化了编程的复杂性。实际上，用户只需简单地列出数学表达式，其结果就以数值或图形方式显示出来。

要获得 MATLAB 软件，只需登录 MathWorks 的官方网站“<https://cn.mathworks.com/>”，在网页上找到“获取免费试用版”链接，如图 1-1 所示。



图 1-1

按照提示，依次填写个人信息，即可在 30 天内免费试用最新版本的 MATLAB 软件。

MATLAB 的用户界面很友好，启动之后会自动创建命令行窗口（Command Window），如图 1-2 所示。

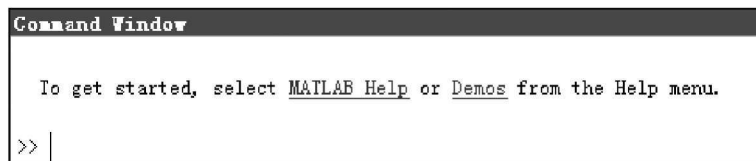


图 1-2

大多数计算、绘图任务，直接在 Command Window 中的提示符“>>”后面输入命令即可完成。

比如在“>>”后面直接输入运算公式：

```
>> (2 * 100 - 34 / 5) / 2 * 100
```

按回车键即可得到结果：

```
ans =  
9660
```

曾经输入过的指令，如果需要再次使用，可以直接单击方向键“↑”或“↓”，“>>”符号后面就会出现此前输入过的指令。输入指令后，Command Window 可能会长时间没有反馈结果，这时可以使用 Ctrl + C 快捷键来中止指令的执行。

在进行数学实验时，经常需要定义各种各样的参变量。MATLAB 对变量的命名有特定的规则，需要特别注意的有以下 4 点：

(1) 变量名必须是不含空格和标点符号的字符组合，像 x 、 y 、 $x23$ 、 $density3$ 都是合理的变量名，而 $x+3$ 、 $y:x$ 都不符合命名规则。

(2) 必须以字母打头，之后可以是字母、数字或下划线，像 $d123$ 、 $data_234$ 都是允许的。

(3) 变量名区分字母的大小写，即 x 与 X 、 $Cat12$ 与 $cat12$ 被视为不同的变量。

(4) 变量名最多不超过 19 个字符。

需要注意，在完成一部分运算后，如果准备重新为某些变量赋值，必须先清除此前的变量定义。具体做法是在 Command Window 下输入 `clear` 并按回车键，即可将此前定义的所有变量全部清除，所有变量、字符都可以重新定义。

MATLAB 中的数学运算符多数符合人们日常的习惯用法，比如：

(1) 加减乘除： $+$ ， $-$ ， $*$ ， $/$ 。

(2) 两个同型矩阵对应位置元素相乘： $.*$ 。

(3) 乘幂： $^$ ，比如 3^2 ，表示 3 的平方。

更多的运算符可参见有关介绍 MATLAB 的书籍。

在 MATLAB 的指令中，要特别注意正确使用标点符号。每条命令后若没有标点符号或者有一个逗号，按回车键就会直接显示命令的结果。若命令后为分号，则不会显示命令结果，可以继续输入其他命令。在编写程序代码时，以“%”打头的命令行，后面所有文字均为注释。

MATLAB 中定义了大量的数学函数，但是其记号和表示规则与数学教材上略有不同。比如反三角函数 $\arcsin x$ 、 $\arccos x$ 在 MATLAB 中的记号为 $\text{asin}(x)$ 、 $\text{acos}(x)$ ， $\ln x$ 的记号为 $\text{log}(x)$ 。具体的信息可查阅 MATLAB 的有关介绍性书籍，或者查阅在线帮助 `helpwin` 中 `MATLAB \ elfun` 或 `help elfun`。

虽然在 Command Window 中输入命令、调用函数比较方便，但在执行一些比较复杂的运算时，可能需要输入很多命令，重复执行同类运算时会很不方便。这时可以考虑将计算过程编写成一个程序，保存为后缀为 `m` 的文件，这就是 MATLAB 中的 M 文件。重复执行同类运算时，直接调用这个新函数即可。

根据调用方式的不同，MATLAB 中的 M 文件可以分为命令脚本文件 (Script File) 和函数文件 (Function File) 两类。更早的 MATLAB 版本只有一种 M 文件 (M-file)。函数形式的 M 文件的第一行必须以 `function` 开始。具体格式为：

`function` 因变量名 = 函数名 (自变量)

比如打开程序编辑窗口，输入如下代码：

```
function f=new(x)
```

```
f=x-1+(x-1)^2+(x-1)^3
```

然后保存为 `new.m`。随后就可以在 Command Window 中直接调用，比如输入：

```
>> x = 4;
>> new (x)
```

即可获得 $f(4)$ 的函数值。

在初步了解了 MATLAB 软件之后，注意经常复习常见的运算命令和函数，为后续数学实验内容的学习打好基础。本章接下来会更详细地介绍数学实验中需要用到的 MATLAB 功能和命令。

1.2 MATLAB 中的数组及其运算

现代科学计算都基于多维空间，因此 MATLAB 中的数值计算都是以向量或矩阵的形式来定义的。需要注意的是，MATLAB 中定义的数组（向量）运算，无论是运算符还是运算规则，都与我们熟悉的线性代数中的定义有一定差异。

MATLAB 中数组（向量）和矩阵的定义非常简便，比如在命令窗口输入：

$$x = [1\ 2\ 3\ 4\ 5\ 6]$$

或

$$x = [1, 2, 3, 4, 5, 6]$$

即可定义一个 6 维行向量或 1×6 的矩阵。输入命令：

$$x = [1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9]$$

得到的是一个 3×3 的矩阵，括号内的“;”表示分行。

MATLAB 中可以定义多种形式的数组（向量）。设 $m < n$ ，命令如表 1-1 所示，

表 1-1

命令	将 x 定义为数组
$x = m : n$	$(m, m+1, m+2, \dots, n)$
$x = m : k : n$	$(m, m+k, m+2k, \dots, n)$
$x = \text{linspace}(m, n, k)$	$(m, m + (n-m)/k, m+2(n-m)/k, \dots, m + (k-1)(n-m)/k, n)$

其中， $\text{linspace}(m, n, k)$ 是将 $[m, n]$ 进行 k 等分，取其端点。

上面介绍的都是如何定义行向量，要定义列向量可以用命令：

$$x = [1; 2; 3; 4; 5]$$

如果已有行向量 x ，则可以对其转置获得列向量，转置命令为：

$$y = x'$$

在 MATLAB 中，数组与数字之间可以定义加减乘除。比如设 $X = [a, b, c, d, e]$ ， q 为标量，则有如下计算规则：

$$X + q = [a + q, b + q, c + q, d + q, e + q]$$

$$X * q = [a * q, b * q, c * q, d * q, e * q]$$

$$X / q = [a / q, b / q, c / q, d / q, e / q]$$

$$X . \setminus q = [q / a, q / b, q / c, q / d, q / e]$$

务必要特别注意上述代码的差异，很多编程错误都缘于这些基本的运算符输入有误。

设 $X = [a, b, c, d, e]$, q 为标量, 数组的幂运算代码及其计算规则为

$$X.^q = [a^q, b^q, c^q, d^q, e^q]$$

$$q.^X = [q^a, q^b, q^c, q^d, q^e]$$

设 $X = [a, b, c, d]$, $Y = [e, f, g, h]$, 则 MATLAB 定义的数组之间的运算规则如下:

$$X + Y = [a + e, b + f, c + g, d + h]$$

$$X .* Y = [a * e, b * f, c * g, d * h]$$

$$X ./ Y = [a / e, b / f, c / g, d / h]$$

$$X . \setminus Y = [e / a, f / b, g / c, h / d]$$

$$X.^Y = [a^e, b^f, c^g, d^h]$$

显然, MATLAB 中定义的数组 (向量) 运算种类比线性代数课程中的要多, 尤其要注意其中向量点乘 $X .* Y$ 、向量点除 $X ./ Y$ 和 $X . \setminus Y$ 以及向量指数运算 $X.^Y$ 的正确代码和意义。

1.3 MATLAB 中的矩阵及其运算

MATLAB 中的很多运算都会涉及矩阵, 因此需要对 MATLAB 中关于矩阵的定义方式、特殊矩阵的生成、运算代码等有详细的了解。

首先, 在输入矩阵时, 注意用逗号或空格来区分同一行的不同数字, 用分号来区分不同的行。一行输入完毕后, 也可以按回车键来开始新的一行。

```
>> a = [1 2 3 4; 5, 6, 7, 8]
```

a =

```
1 2 3 4
5 6 7 8
```

需要注意的是, 输入矩阵时, 所有的行都必须有相同数量的元素。

```
>> b = [ 1 2 3 4
5 6 7 8
1, 2, 2, 2]
```

b =

```
1 2 3 4
5 6 7 8
1 2 2 2
```

对于零矩阵、单位矩阵等一些特殊的矩阵, 可以由 MATLAB 自动生成。比如生成 $m \times n$ 阶零矩阵可以使用命令 $A = \text{eye}(m, n)$ 。

```
>> A = eye (2, 4)
```

```
A =
```

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

生成 $m \times n$ 阶单位矩阵可以使用命令 `B = zeros (m, n)`。

```
>> B = zeros (2, 4)
```

```
B =
```

```
    0    0    0    0
    0    0    0    0
```

我们知道，矩阵就是数据表格，有时可能需要提取其中的部分数据，或者对数据表格进行切割、合并等操作，这些都可以借助一些 MATLAB 命令迅速生成相应的结果。提取矩阵中属于第 i 行到第 j 行、第 m 列到第 n 列的部分元素组成新的矩阵，可以用命令 `A (i: j, m: n)`。

```
A =
```

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

```
>> A (1: 2, 2: 3)
```

```
ans =
```

```
    0    0
    1    0
```

删去矩阵中的部分行，剩余元素构成新的矩阵可以用命令 `A (i: j, :)`。删去部分列，剩余元素构成新的矩阵可以用命令 `A (:, m: n)`。

```
A =
```

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

```
>> A (1: 2, :)
```

```
ans =
```

```

1     0     0     0     0
0     1     0     0     0

```

```
>> A (:, 2:3)
```

```
ans =
```

```

0     0
1     0
0     1

```

将 A 和 B 两个矩阵合并为一个矩阵，可以用 $[A \ B]$ 和 $[A; B]$ 命令，前者表示 A 在左而 B 在右，后者表示 A 在上而 B 在下。

```
>> A = [1, 2; 3, 4]; B = [1, 0; 0, 1];
>> [A B]
```

```
ans =
```

```

1     2     1     0
3     4     0     1

```

```
>> [A; B]
```

```
ans =
```

```

1     2
3     4
1     0
0     1

```

此外，线性代数课程中讲过的矩阵运算也可以很方便地实现。比如两个同型矩阵 A 和 B 相加，可以用 $A+B$ 命令，两个矩阵相乘可以用 $A*B$ 命令。常见的矩阵运算命令如表 1-2 所示。

表 1-2

命令	运算规则
$A+B$	两个同型矩阵 A 和 B 相加
$A*B$	矩阵 A 、 B 相乘
$\text{inv}(A)$	求矩阵 A 的逆矩阵
$\text{det}(A)$	求矩阵 A 的行列式
A/B	已知矩阵 A 、 B ，求 $X*B=A$ 的解 X
$A \setminus B$	已知矩阵 A 、 B ，求 $A*X=B$ 的解 X

矩阵运算在科学与工程计算中很常见，在本书后续的线性代数部分会介绍更多的矩阵运算。

1.4 循环和判断语句

在完成一些比较复杂的运算时，往往需要用到循环和分支等逻辑语言结构，这就是 MATLAB 中的控制流结构。MATLAB 提供了 for 循环、while 循环等循环语句，以及逻辑判断结构 if - else - end 等。

首先介绍 for 循环。for 循环用于让某个语句或一组语句被重复执行指定次数。比如对 1 到 50 之间的数字求和，可以使用如下循环语句实现：

```
>> clear
>> s = 0;
>> for i = 1: 50;
s = s + i;
end
>> s
```

```
s =
```

```
1275
```

在上面的代码中，首先为变量 s 赋予初始值 0，每次增量为 1，循环 50 次，输出结果就是 1 到 50 之间的数字之和。

再比如计算 $n = 1, 2, 3, 4, 5$ 时的 $\sin(\pi/n)$ 值，输入如下代码：

```
>> for n = 1: 5;
x (n) = sin (pi/n);
end
```

即可算出所有结果。如果需要输出单个计算结果，输入命令 $x(n)$ 即可， n 为 1 至 5 中的某个数字，比如求 $x(2)$ ：

```
>> x (2)
```

```
ans =
```

```
1
```

也可以输出全部计算结果，输入命令 x ，然后按回车键即可。

```
>> x
```

```
x =
```

```
0.0000    1.0000    0.8660    0.7071    0.5878
```

接下来介绍 while 循环。前面介绍的 for 循环是让某个或一组语句执行指定次数，如果需要设定一个更宽泛的结束循环的条件，则可以选择 while 循环。比如，某国现有人口 4.5 亿人，年增长 3%，什么时候会达到 5 亿人？输入如下代码：

```
>> m=4.5;
>> t=0;
>> while m<5
t=t+1
m=m*(1+0.03)
end
```

输出结果为： $t=1$ 时 $m=4.6350$ ， $t=2$ 时 $m=4.7740$ ， $t=3$ 时 $m=4.9173$ ， $t=4$ 时 $m=5.0648$ 。因此，4 年后人口会达到 5 亿人。

很多编程语言中都存在类似“如果某个条件满足，则执行某个指令”的逻辑判断—执行语句，MATLAB 中也有类似的结构，这就是 if-end 结构。比如，在定义分段函数时，就可以运用 if-end 结构。

对于如下分段函数：

$$f(x) = \begin{cases} 2x + \sin x, & x < 0, \\ e^x - 1, & x \geq 0. \end{cases}$$

求其在定义域内任何一点的函数值，我们可以建立如下的 M 文件，并将其命名为 fdhs.m：

```
function f=fdhs(x)
if x<0
f=2*x+sin(x)
end
if x>=0
f=exp(x)-1
end
```

计算 $f\left(-\frac{\pi}{2}\right)$ 和 $f(0)$ ，可以在命令窗口输入 fdhs(-pi/2)，得到 ans = -4.1416；输入 fdhs(0)，得到 ans = 0。

如果选项比较多，则可以用多个 if-end 语句覆盖全部选项。如果函数只有两个选项，则可以采用 if-else-end 结构：

```
function f=fdhs(x)
if x<0
f=2*x+sin(x)
else
f=exp(x)-1
end
```

选项较多时，也可以使用 if-elseif-elseif-end 结构。比如对于分段函数：

$$f(x) = \begin{cases} -1, & x < -1, \\ 0, & x \in [-1, 1], \\ 1, & x > 1. \end{cases}$$

可以反复使用 if - end 语句:

```
function f = fdhs (x)
    if x < -1
        f = -1
    end
    if x >= -1 & x <= 1
        f = 0
    end
    if x > 1
        f = 1
    end
end
```

也可以运用 if - elseif - elseif - end 语句:

```
function f = fdhs (x)
    if x < -1
        f = -1
    elseif x >= -1 & x <= 1
        f = 0
    elseif x > 1
        f = 1
    end
end
```

两种语句输出的结果完全一致,后者需要输入的代码会少一些。

MATLAB 中还有其他控制流结构,在面对不同问题时各有各的优势,这里不再赘述。

1.5 MATLAB 中的函数绘图 (1)

绘制函数的图像,可以帮助我们更为直观地了解函数的各种性态。MATLAB 中提供了强大的绘图功能,因此 MATLAB 也是科学研究中常用的绘图工具。

如果没有什么特殊要求,对于形如 $y = f(x)$ 这样的一般函数,可以直接用命令 `plot (x, y, 'S')` 来绘图,其中 x 、 y 表示横坐标、纵坐标, 'S' 表示绘图时用的线型——不同颜色的实线、虚线等。

需要注意的是,用 MATLAB 绘图先要得到满足函数的一系列点的坐标,然后将该点集的横坐标、纵坐标分别赋予 x 和 y ,因此命令中的 x 和 y 都是向量,在输入函数关系 $y = f(x)$ 时注意使用向量的运算符号。

比如要绘制 $y = \sin x$ 在区间 $[0, 2\pi]$ 上的图形,输入命令:

```
>> x = [0: 0.01: 2 * pi];
>> y = sin (x);
>> plot (x, y)
```

可以得到图 1-3 所示图形。

也可以输入如下命令: