

21 世纪高等院校教材

数 学 实 验

焦光虹 主编

王希连 张云飞 尚寿亭 副主编

科 学 出 版 社

北 京

内 容 简 介

数学实验是借助数学软件,结合所学的数学知识解决实际问题的一门实践课.本书是国家工科数学教学基地之一的哈尔滨工业大学数学系,根据数学教学改革成果而编写的系列教材之一.包括数学软件 MATLAB 的入门知识,数学建模初步及运用高等数学、线性代数与概率论等相关知识的实验内容.亦尝试编写了几个近代数学应用的阅读实验,对利用计算机图示功能解决实际问题安排了相应的实验.实验选材贴近实际,易于上机,并具有一定的趣味性.

本书可作为工科大学本科生的数学课教材,亦可供参加大学生建模竞赛的学生参考.

图书在版编目(CIP)数据

数学实验/焦光虹主编. —北京:科学出版社,2006
(21 世纪高等院校教材)
ISBN 7-03-015462-2

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

中国版本图书馆 CIP 数据核字(2005)第 044014 号

责任编辑:赵 靖 / 责任校对:张 琪
责任印制:安春生 / 封面设计:陈 敬

科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

印刷

中国科学院印刷厂印刷

科学出版社发行 各地新华书店经销

*

2006 年 1 月第 一 版 开本:B5(720×1000)

2006 年 1 月第一次印刷 印张:19

印数:1—4 000 字数:356 000

定价:22.00 元

(如有印装质量问题,我社负责调换(环伟))

哈尔滨工业大学数学教学丛书编写委员会

主 任 王 勇

委 员 (按汉语拼音排序)

包革军 董增福 盖云英 高广宏 焦光虹

李道华 尚寿亭 田波平 王希连 吴勃英

谢鸿政 杨凤林 游 宏 张 彪 张池平

张传义 张云飞

前 言

数学建模,是应用数学的语言与方法刻画并反映实际问题规律性的一种科学方法.20 世纪 50~60 年代,许多数学模型因实验时庞大的数据计算量,使其应用受到限制,模型的验证与改进亦相当困难.进入 80 年代,尤其 90 年代中期,基于计算机技术及相应软件技术的高速发展,已相当程度地解决了上述问题.

数学实验课作为一门对数学理论与方法的实践课,侧重于以下几个方面.

1. 对具体数学模型提供的数据,学习处理与分析的方法.本教材主要借助于数学软件 MATLAB,在验证过程中要使学生学会确定所给模型的适合度,并提出修改模型的意见与方法.

2. 针对简单问题,提出数学模型并分析数据以修正模型.

3. 辅助于前两项工作的完成,同时掌握对计算机及软件 MATLAB 的使用,培养一定的编程能力.

由于建模与实验的关系,教程中提供了一些建模实例,其实例选取满足:

1. 贴近实际,直观有趣,使同学在理解这些例子的基础上对数学内涵有更深入的理解,并进一步促进同学们学习理论的热情;

2. 模型的数学形式便于上机编程验证其结果,所用数学方法涵盖所学的数学内容;

3. 适当选取几个建立在近代数学理论基础上的数学模型,以开阔同学们的眼界;

4. 对有些模型的建立,不追求理论的完整与逻辑的严密,重点强调背景与直观.

作为本门课程的主要工作平台, MATLAB 的一般使用方法将作为重要内容加以介绍. MATLAB 是现在流行的数学、科技应用软件中国际公认的最优秀的软件之一,其特点为:

1. 算法先进的数值计算与功能强大的符号演算;

2. 可编程的可视化运算与文字统一处理功能;

3. 外挂众多实用工具箱,为各种科研工作提供了相当程度的技术支持.

基于上述优点,在国外许多高校, MATLAB 的使用已成为大学生、硕士生乃至博士生必须掌握的基本技能.随着对 MATLAB 认识的深入,相信这一软件一定会成为同学们学习与科研的良师益友.

对在本书编写过程中给予热情支持的李容录教授、张宗达教授、薛小平博士、严质彬博士一并表示深深的谢意.

由于个人学术水平有限,在教材的选材与编写过程中出现疏漏与问题在所难免,恳请读者给予批评指正.

作 者

2004 年 8 月于哈尔滨

目 录

绪论	1
第 1 章 MATLAB 简介与入门	7
1.1 简介	7
1.2 应用入门	8
1.3 MATLAB 的语言程序设计简介	13
1.4 特殊量与常用函数	18
1.5 图形功能	21
1.6 M 文件	24
1.7 符号运算与应用	26
第 2 章 微分方程建模初步	35
2.1 模式与若干准则	35
2.2 阅读与理解	37
2.3 几个例子	38
2.4 一阶微分方程定性解的图示	42
第 3 章 平面线性映射的迭代	53
3.1 线性函数迭代	54
3.2 平面线性映射的迭代	56
第 4 章 微分方程数值解	64
4.1 算法	64
4.2 欧拉与龙格-库塔方法	66
4.3 模型与实验	69
第 5 章 曲线拟合	76
5.1 磨光公式	76
5.2 修正与误差	78
5.3 进一步讨论的问题	80
第 6 章 图的着色	81
6.1 一个时间安排问题	82

6.2	数学思想的导出	83
6.3	一般的计数问题	84
6.4	进一步探索的问题	87
第 7 章	敏感问题的随机调查	91
7.1	阅读与理解	91
7.2	直觉的定义	93
7.3	统计思想的一个基本原理	93
7.4	随机应答调查	93
7.5	估计的基本性质	100
7.6	估计的其他性质	104
第 8 章	数学建模	110
8.1	投篮角度问题	110
8.2	壳形椅的讨论与绘图	113
8.3	独家销售商品广告问题	114
8.4	售报策略	119
8.5	Galton 钉板问题	120
第 9 章	优化问题	122
9.1	优化工具箱	122
9.2	优化函数的使用	123
9.3	污水控制	126
第 10 章	图像增强	129
10.1	图像及操作	129
10.2	直接灰度调整	132
10.3	直方图处理	138
10.4	空域滤波增强	139
10.5	频域增强	145
第 11 章	数学曲面	149
11.1	MATLAB 语言的预备知识	149
11.2	几种有趣的数学曲面	154
11.3	默比乌斯曲面族	164
第 12 章	阅读实验一 泛函分析初步	171
12.1	一个例子	171

12.2	距离空间简介·····	173
12.3	应用·····	174
12.4	线性空间与 Hilbert 空间 ·····	176
12.5	例与问题·····	178
第 13 章	阅读实验二 群与应用 ·····	182
13.1	背景与阅读·····	182
13.2	抽象群·····	184
13.3	应用·····	187
第 14 章	阅读实验三 积分教学中的几点注释 ·····	191
14.1	阅读与理解·····	191
14.2	理论阐述·····	192
第 15 章	建模竞赛真题 ·····	199
15.1	非典数学模型的建立与分析·····	199
15.2	西大直街交通最优联动控制·····	209
15.3	股票全流通方案数学模型的创新设计·····	230
附录	·····	258
A	数学实验课实验教学大纲 ·····	258
B	数学实验报告 ·····	260

绪 论

20 世纪 70 年代末 80 年代初,英国一些高校率先开设了称之为“数学实验”的一门新课.其后,美国一些重点大学及至 90 年代末国内几所重点院校亦相继开设了同类课程,但模式不尽相同.

在现代数学教育中,国内外同行逐渐形成的共识是:讲授数学知识不能再局限于几个世纪形成的经典理论,教师在讲授数学知识的同时,还应培养其应用数学知识的技能,特别是数学建模和计算机模拟的本领,在强调抽象思维的同时,亦应强调形象思维与几何方法.借助计算机的图示功能,培养想像力和创造力,并强调能较好理解计算机的运算结果.

正是由于上述教育观念的转变,铺垫了开设数学实验课的内因,而现代计算机软、硬件技术的发展,则为开设此类课程提供了相应的硬件支持.

本门课程的开设以培养学生应用数学知识的能力为主线,借助计算机的计算与图示功能,使学生在用数学中学数学,更深层次地体会所学数学知识的内涵,培养他们学习数学的热情,启迪创新意识.

此类课程的开设模式,按国内外我们所见教材的形式,认为可大体分为三类.

(1) 以计算方法(包括传统计算方法、优化问题、概率统计中的计算问题等)为主要教学内容,配置大量的实际相关背景,使学生在应用新知识的同时应用其解决实际问题.教学手段的重要辅助成分包括建模、上机、验证并分析结果等.但总的来说,仍以学习新知识和方法为主,只是教学手段已发生了革命性的变化.教学与上机时数比约为 2 : 1.

(2) 以建立数学模型为主线,利用所学过的数学知识,培养加强处理实际问题的技巧与能力.适当引入一些新的数学方法是必要的,但所占比例不大,其教学过程仍以教师为主,重点讲解经典数学模型建立的思想、方法.一般分为如下几个步骤:提出问题;分析并转化为数学问题(建模);上机并分析数据,验证结果并与实际问题相比较;修改模型(如有必要);再重复上机,最后如有可能,还应依据模型给出更深刻的结论性理论分析结果.教学与上机时数比约为 1 : 1.

(3) 以加深理解已学数学知识为主,通过对一些简单实际问题的接触,增加学生应用已学知识的意识与兴趣.其主要精力放在对所布置实验在计算机上产生的数据及图像的分析与理解.例如,在线性变换的实验中,考虑线性方程

$$A \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}, \quad (0.1)$$

其中

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

(0.1)式给出了一个平面点到平面点的线性变换 $f: (x_1, y_1) \rightarrow (x_2, y_2)$, 那么, 当 f^n 作用于 (x_1, y_1) 即

$$A^n \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = A^{n-1} \left(A \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \right)$$

时, 问 (x_1, y_1) 的轨迹是怎样的, 或问点列

$$A \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, A^2 \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = A \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}, \dots, A^n \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \dots$$

的发展趋势如何?

讨论的主要方式是对 A 的 4 个参数选择不同的值, 计算其特征值与特征向量, 并观察理解相应点列的变化规律. 之后要求学生对平面一个图形作线性变换, 要求在指定方向将图形拉长, 而在另一方向将图形压缩, 以体会计算机动态生成过程中的基本数学原理, 更深入的结论性内容在实验中亦有说明.

因涉及的数学内容基本为已学知识, 而提出的应用背景又很直观, 本着留下更多时间给学生的原则, 课堂教学与上机时数比为 1 : 2 或 1 : 3.

本教材的编写模式以方式(3)为主.

无论以何种模式开设实验课, 提供实际背景, 建立数学模型都是必要的. 亦考虑到各层次大学生的建模竞赛, 在教材中增加相当数量的建模内容, 对提高同学们的科研能力有积极的意义. 本书中, 有一章选自哈尔滨工业大学往届大学生建模竞赛的作品, 以供参赛同学参考.

下面介绍建模竞赛的规则与评奖标准.

(1) 竞赛期间, 参赛队员可以使用各种图书资料, 计算机和软件及网上资源, 但不得与队外任何人讨论, 包括指导教师.

(2) 评奖以假设的合理性、建模的创造性、结果的正确性、文字表述的清晰程度为主要评分标准.

下面介绍两个建模实例, 其中第二个实例讨论较详细, 并给出相应 MATLAB 程序. 进一步的深入讨论, 在实验正文中给出.

例 0.1 “椅子能否放平”问题.

这里的放平应理解为椅子的所有腿能否同时着地, 而地面显然不应假设为几何平面. 对三腿椅子, 在一般地面均能放平, 这是已知的几何事实, 因而提出如下假设:

- (1) 椅子有 4 条腿;
- (2) 地面光滑;

(3) 三腿椅子总能放平.

依据上述假设建立数学模型.

(1) 建立坐标系如图 0.1 所示, A, B, C, D 为 4 腿初始位置, 令椅子绕 z 轴旋转.

(2) 设 $f(\theta), g(\theta)$ 分别为转角等于 θ 时 A, C 腿和 B, D 腿与地之间的距离之和. 当 $\theta=0$ 时, 4 腿位置如图 0.1 所示. 令

$$F(\theta) = f(\theta) - g(\theta),$$

则由地面的光滑性知 F 是 θ 的连续函数. 由假设, 三腿总能着地, 所以 $f(\theta), g(\theta)$ 中至少一个为 0, 若 $f(\theta) = g(\theta) = 0$, 意味着 4 腿着地. 不妨设初始状态 ($\theta=0$) 下 $f(0) > 0, g(0) = 0$, 则有

$$F(0) = f(0) - g(0) > 0.$$

当 $\theta = \frac{\pi}{2}$ 时, $f\left(\frac{\pi}{2}\right) = 0, g\left(\frac{\pi}{2}\right) > 0$, 此时

$$F\left(\frac{\pi}{2}\right) = f\left(\frac{\pi}{2}\right) - g\left(\frac{\pi}{2}\right) < 0.$$

由介值定理, $\exists \xi \in \left[0, \frac{\pi}{2}\right]$, 使得

$$F(\xi) = f(\xi) - g(\xi) = 0,$$

从而 $f(\xi) = g(\xi) = 0$. 这因为对任意转角 θ , f 与 g 至少一个为 0, 此时, 椅子 4 腿着地. 在图 0.1 中椅子为方形, 考虑椅子是长方形时, 上述讨论是否仍然正确.

例 0.2 设敌方飞机 A 沿 y 轴正向以速度 v 飞行. 过 $O(0,0)$ 点时, 位于 $M_0(16,0)$ 处的导弹 B 起飞, 其飞行速度矢量始终指向 A , 大小是 A 速度大小的 2 倍. 求 B 的追踪曲线 $y=f(x)$ 与 B 击中 A 的位置.

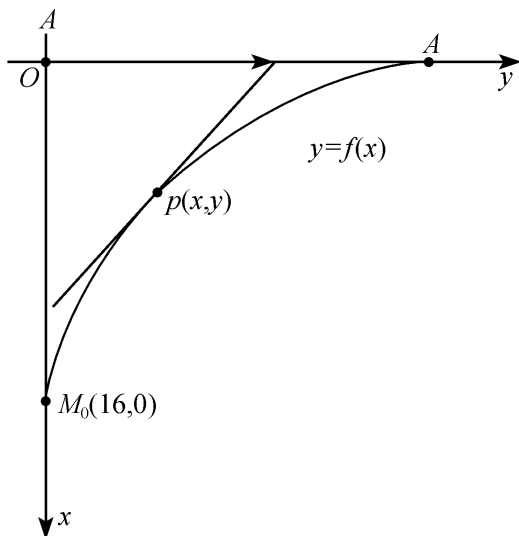


图 0.2

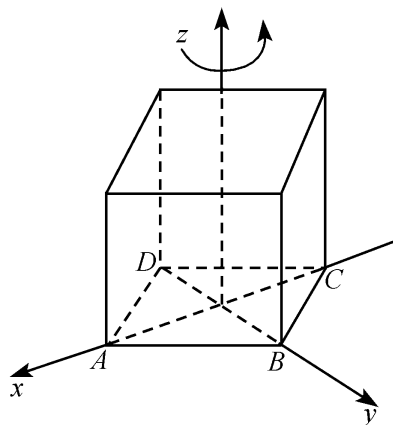


图 0.1

解 (图 0.2) 通过分析, 建立方程

$$y'' = \frac{\sqrt{1+y'^2}}{2x}.$$

这是一个缺 y 型可降阶的微分方程, 化为一阶后, 分离变量, 可得 $y=f(x)$ 的解析解, 从而击中点易求. 下面的问题是, 当 A 发现导弹 B 时, 不再沿直线 $x=0$ 飞行, 而是沿曲线 l

$$l: \begin{cases} X = X(t), \\ Y = Y(t), \end{cases} \quad t \geq 0 \quad (0.2)$$

飞行, 此时导弹 B 的追踪轨迹是什么? 假设条件为:

- (1) B 的飞行速度大小为 $k|v|$, v 为 A 的飞行速度矢量, k 为大于 1 的常数;
- (2) B 的飞行方向为: 任意时刻 t , B 始终指向 A .

进一步考虑如下问题.

(1) 若当飞机飞至 $O(0,0)$ 点时, 改变航向, 直向 M_0 点飞来, B 击中 A 的时间最短; 当 A 沿 y 轴飞行时, B 击中 A 需较长时间. 那么, 当 A 沿什么曲线飞行时, B 击中 A 时间最长?

不妨称这一曲线为摆脱曲线. 有同学可能认为摆脱曲线为沿 x 轴负向飞行的直线, 但仔细想想, 又似乎并非如此.

(2) 若要求在限定时间内使 B 追上 A , k 值至少应为多少?

同学们自己还可再提出一些问题, 至于最终能否解决, 取决于诸多因素. 例如, 我们提出的进一步问题目前虽无法解决, 但还是提出以供思考. 下面的讨论集中于当 A 沿路线 (0.2) 式飞行时 B 的追踪路线. 为简化问题, 先假设 A, B 之间的距离保持不变. 此后, 请同学以此为基础, 分析 A, B 之间距离不断变小, 最终追上 A 时 B 的飞行轨迹.

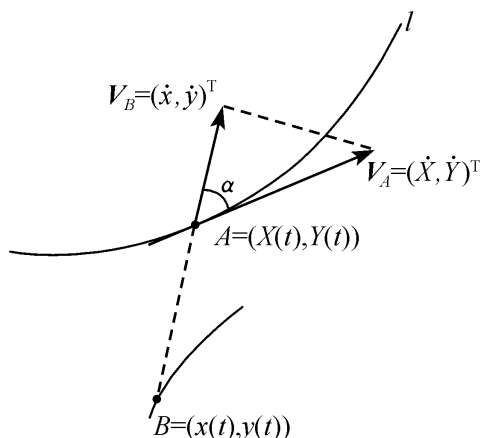


图 0.3

设 t 时刻, A 的位置函数为 $(X(t), Y(t))$, 飞行曲线为图 0.3 中的 l , B 在 t 时刻的位置函数为 $(x(t), y(t))$, 故 t 时刻 B 的飞行方向为

$$\mathbf{V} = (X - x, Y - y)^T = \begin{bmatrix} X - x \\ Y - y \end{bmatrix},$$

且

$$(X - x)^2 + (Y - y)^2 = S^2,$$

其中 S 为一个与时间 t 无关的常数. 所以 B 的飞行速度 $\mathbf{V}_B$ 为

$$\mathbf{V}_B = (\dot{x}, \dot{y})^T = r(X - x, Y - y)^T.$$

注意: B 的速度依赖于 A 的速度矢量 $\mathbf{V}_A = (X, Y)^T$ 的方向, 若 A 沿半径为 S 的圆飞行, 而 B 在圆心时, 则 B 根本不运动. 由图 0.3 知, $|\mathbf{V}_B| = |\mathbf{V}_A| \cos \alpha$, 若设 $\mathbf{w} = \mathbf{V}/|\mathbf{V}|$, 又

$$\mathbf{V}_A^T \cdot \mathbf{w} = (X, Y) \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = |\mathbf{V}_A| |\mathbf{w}| \cos \alpha = |\mathbf{V}_A| \cos \alpha,$$

所以

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = |\mathbf{V}_B| \mathbf{w} = (\mathbf{V}_A^T \cdot \mathbf{w}) \mathbf{w}. \quad (0.3)$$

一般, 微分方程 (0.3) 关于 $(x(t), y(t))$ 的解析解不存在, 我们只能求其数值解. 下面的程序将完成这一工作, 并对特定的 (0.2) 式给出飞机与导弹的飞行路线.

```
function xs=f(t,x)
```

```

[X,Xs,Y,Ys]=A(t);
V=[Xs;Ys];
w=[X-x(1);Y-x(2)];
w=w/norm(w);           %norm(w)=|w|
xs=(v' * w) * w;       %V'=V^T
function [X,Xs,Y,Ys]=A(t)
X=5 * cos(t);   Xs=-5 * sin(t);
Y=5 * sin(t);   Ys=5 * cos(t);
%main 1.m
y0=[10,0]';           %B 的初始位置
[t,x]=ode45('f',[0,100],y0);
hold on
axis([-6,10,-6,10]);   %设定 x,y 轴范围
plot(x(:,1),x(:,2))
%以下程序在同一图中加入 A 的飞行路线
t=0:0.05:6.3;
[X,Xs,Y,Ys]=A(t);
plot(X,Y,':');
hold off;

```

练习 设 A 的飞行路线为

$$l: \begin{cases} x(t) = t, \\ y(t) = 5\sin t, \end{cases} \quad \begin{cases} x(0) = 0, \\ y(0) = 0, \end{cases}$$

求 B 的追踪路线的数值解并画图.

学习本门课程使用的数学软件为 MATLAB,作为产品推出始于 1984 年.国外,尤其美国的许多大学,早于 20 世纪 80 年代末,将其列入本科生与研究生的必修课和实验环节中必须掌握的技术工具.这里强调的是:实验课不是程序语言课,要求同学通过自学必须掌握 MATLAB 的一般编程技能,在许多实验中,要求自己编写程序.实践证明,大多数同学在教师指导下能完成这一工作.在一些实验中,只给出编程伪码,同学们需将伪码转换成程序代码,否则实验无法进行,我们觉得这样做是必要的.实验中使用的主要工具箱包括符号工具箱,统计工具箱,优化工具箱等.

对 MATLAB 中数据输入,重点强调以下几点.

(1) 分号“;”的作用.

① 在括号[]内时,是矩阵行与行之间的分隔符,可以用回车键↵代之.

② 可用作指令与指令间的分隔符.

③ 在赋值指令后加分号“;”时,其结果将不在屏幕显示.

(2) 逗号“,”的作用.

① 在方括号“[]”内时,可作为同一行内元素与元素的分隔符,亦可用空格键代之.

② 可用作指令与指令间的分隔符.

③ 当其存在于赋值指令之后时,该指令执行后的结果在屏幕上显示.

④ 当位于函数 $f(\quad)$ 的小括号内时,其作用是充当参数与参数之间的分隔符.

(3) 输入向量或矩阵数据的几点注意.

① 输入矩阵必须以方括号“[]”为其首尾.

② 矩阵中行与行之间必须用“;”或回车键隔开.

③ 每一行元素之间需用逗号“,”或空格键分开.

(4) 简单数组创建.

创建包含指定元素的行向量,用记号“ $\gg$ ”表示键入.

```
 $\gg$  x=[22*pi,sqrt(2),2-3j];
```

创建第一个元素为 first,第二个元素为 first+1,……,最后一个为 last 的行向量.

```
x=first:last;
```

```
x=first:increment:last;
```

增量 increment 可正可负,如 $x=1:-0.5:-3$.

```
x=linspace(first,last,n);
```

创建从 first 开始到 last 结束的 n 个元素的行向量,在 first 与 last 之间均匀插入 $n-2$ 个点.可由下列函数生成特殊矩阵,包括:

```
zeros(n,m);ones(n,m);eye(n,m);
```

```
rand(n,m);randn(n,m);
```

在第 1 章中,要较详细介绍 MATLAB 的使用,但还是在这里开列了一些最常用的指令,以引起关注.

在第二个建模例子中给出了一段完整的程序,希望通过运行这一程序,增加同学们的感性认识与兴趣,也希望通过教师对程序的说明使同学们能体会出其中的数学味道,了解掌握基本数学原理对编写程序代码的重要性.

在第 2 章中,用较大篇幅讨论微分方程的建模问题,能对研究生考试有所帮助也是编写该章目的之一.

第 1 章 MATLAB 简介与入门

1.1 简介

MATLAB 是矩阵实验室 matrix laboratory 的简写,其基本数据单位为矩阵,它的指令表达式与数学及工程中的书写习惯十分类似,如矩阵方程 $\mathbf{b} = \mathbf{A}\mathbf{x}$,在 MATLAB 中写为 $\mathbf{b} = \mathbf{A} * \mathbf{x}$,欲求 $\mathbf{x}$,只需输入指令 $\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$ 即可,这比 C 语言或其他语言要方便得多.

MATLAB 经过多年发展,现已成为线性代数、自控理论、数理统计、数字信号处理、时间序列分析、动态系统仿真等现代课程的基本教学软件.在科研设计部门, MATLAB 更是广泛应用于各个领域而成为不可缺少的助手.

值得一提的是,除内部函数外, MATLAB 所有主包文件与各工具箱文件都是可读可改的源文件,这无疑极大方便了读者的使用与学习.

还需说明的是,在 MATLAB 中所有数据单位作为矩阵处理带来的便利,使算法统一且便捷,这是 C 语言与其他语言所不具备的.例如,对实数 x 可看成 1×1 矩阵,对多项式

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

其系数为 $1 \times (n+1)$ 阶矩阵,从而对其根的求解即可转化为求其伴随矩阵的特征值问题,关于矩阵求特征值的方法皆可使用.理论已证明此种求根方式无论精度还是可靠性都高于经典的迭代方式.

计算结果的可视化是 MATLAB 的重要特征.学生在学习一元函数微积分学时就有这样一个愿望,若对所给函数都能看到伴随它的图形该有多好,而这一愿望随着对多元函数微积分学的学习而变得愈加强烈.

从事数据整理与计算结果分析的科研人员往往要面对一大堆离散数据,而这些数据的内在关系的表现,莫过于图形手段.

同学们通过使用 MATLAB 会惊喜地发现,它的图形功能大大超出你的想像,其奇妙的近乎完美的图形会使你惊叹,进而想到,它的内核编程是如何做的,自己若也能编出这样的程序该多好!

在各专业领域, MATLAB 提供了许多功能强大的工具箱,开列一些以供参考.动态仿真(simulink),控制(control),优化(optimization),神经网络(neural network),样条(spline),系统识别(system identification),符号演算(symbolic math),模糊控制(fuzzy control)等,关于它们的使用,已有很专业的书籍问世予以说明.

MATLAB 为初学者提供了大量的演示例程,同学们可通过帮助窗口获得,亦可在命令窗口内直接键入要演示的例程文件名,这些例程展示了 MATLAB 的功能,亦是初学者的入门向导.

应注意的是,MATLAB 对大、小写敏感,而其所有命令的键入均以小写形式给出.

1.2 应用入门

MATLAB 6.5 版是一个高度集成的语言环境,在其命令窗口中(MATLAB command window)可编写、运行程序,其执行方式类似于 BASIC 程序的执行方式,但更快捷简便,命令窗口本身又是一个功能强大的计算器,输入要计算的表达式,按回车键即得结果.下面分几段介绍 MATLAB 的基本应用并约定用“↵”表示按下回车键.

1. 启动

假设快捷方式已经建立,只需用鼠标即可方便启动.若未建立快捷启动图标,可在目录 c:\matlab\bin 下运行 matlab.exe 文件.

2. 对输入指令的编辑(表 1.1)

表 1.1

键 名	作 用	键 名	作 用
↑	前寻式调回已输入指令行	Home	光标移至当前行首
↓	略	End	光标移至当前行尾
←	当前行中左移光标	Delete	删去光标右边字符
→	略	Backspace	删去光标左边字符
Page UP	前寻翻阅窗口中内容	ESC	清除当前行全部内容
Page Down	略	ctrl+c	中止命令执行

MATLAB 工作窗中部分通用指令见表 1.2.

表 1.2

键入的指令	作 用
quit	退出 MATLAB
clc	擦除工作窗中所有显示内容
clf	擦除当前图形窗中的图形
clear	清除内存中的变量与函数
dir	列出当前目录下的文件与子目录清单
cd	改变当前工作目录

续表

键入的指令	作用
disp	(在运行中)显示变量或文件内容
type	显示指定文件的全部内容
echo	控制文件指令是否显示的开关
hold on(hold off)	控制当前图形窗口对象是否刷新

3. MATLAB 的帮助功能

MATLAB 的命令很多,功能各异,为帮助用户使用,它提供了广泛的在线帮助功能,主要指令为 Help,Lookfor 及交互式的 help 菜单.

help 命令

如果用户要寻求已知标题的帮助,可键入如下命令:

```
help sqrt
```

其中 sqrt 是你想使用的一个函数名,但使用方法不详,即可借助 help 命令,具体显示内容.同学们可上机实习.需要说明的是,帮助信息中 sqrt 的形式为大写的 SQRT,具体使用时要用小写.

在大多数情形下当你不知道具体标题时,若键入

```
help
```

你会发现这一命令形式会帮助你找到所需的确切标题.而命令

```
help general
```

返回 MATLAB 的一般命令列表.

lookfor 命令

lookfor 命令提供的帮助方式是通过搜索所有 MATLAB help 标题及 MATLAB 路径中 M 文件的第一行,返回包含指定关键词的那些项.最重要的是关键词不必是 MATLAB 的命令.如下面的指令:

```
lookfor complex
```

关键词 complex 不是 MATLAB 命令,但在显示的 6 条内容中都会有这个词,此时再结合命令 help 即可显示某一特定指令的帮助.

菜单驱动的帮助

对 PC 机,用户可直接选择 help 菜单中的 Help Desk 项获得,此时会打开一个帮助窗口,用户可在显示列表中双击鼠标以选择任何标题或函数. MATLAB 帮助窗口采用标准格式,允许用户搜索主题,设置书签,注释主题及打印帮助屏幕.

4. 数学运算

在 MATLAB 指令窗口下,可方便进行演草纸式的数学运算,分别介绍以下

几种.

算术运算

包括 $a+b$; $a-b$; $a*b$, $a\backslash b$, a^b ; 最后一式表示 a 的 b 次方, a, b 代表使运算有意义的实数.

复数运算

$a+bi$ 为复数基本运算单位, 其中 $i = \sqrt{-1}$, 例如, 在窗口中键入

$x=1+2i$; $y=3+5*i$;

$c=x*y$;

$c=-7+11i$

说明 复数 x 亦可记为 $x=r*\exp(\varphi*i)$, 这里 φ 为幅角, r 是其模. 关于输入指令的方式简要说明, 几条语句可在同一行书写, 语句与语句间可由“,”或“;”隔开, “;”指明运算结果不显示, 要注意虚部的输入可写为 ai (a 与 i 之间无空格), 或 $a*i$. 如果表达式很长, 一行写不下, 可键入“...”(续行号)后回车在下一行续写, 如

$$S=1-1/2+1/3-1/4+1/5-\dots$$

$$1/6+1/7-1/8+1/9-1/10$$

在任何情形中, 同时按下 $\text{ctrl}+\text{c}$ 键, 都将终止程序运行.

在复数运算中, 类似于求模运算, 求复角运算, 取虚、实部的运算方法要借助 MATLAB 所提供的数学函数. 表 1.3 开列了一些常用的数学函数.

表 1.3

符 号	功 能	符 号	功 能
$\text{abs}(x)$	绝对值或复数的模	$\text{floor}(x)$	对 $-\infty$ 方向取整数
$\text{acos}(x)$	反余弦	$\text{gcd}(x, y)$	整数 x, y 的最大公约数
$\text{acosh}(x)$	反双曲余弦	$\text{imag}(x)$	复数虚部
$\text{angl}(x)$	取复数相角	$\text{lcm}(x, y)$	x, y 的最小公倍数
$\text{asin}(x)$	反正弦	$\text{log}(x)$	自然对数
$\text{asinh}(x)$	反双曲正弦	$\text{log10}(x)$	常用对数
$\text{atan}(x)$	反正切	$\text{real}(x)$	复数实部
$\text{atanh}(x)$	反双曲正切	$\text{rem}(x, y)$	x/y 的余数
$\text{ceil}(x)$	对 $+\infty$ 方向取整数	$\text{round}(x)$	四舍五入到最近的整数
$\text{conj}(x)$	复数共轭	$\text{sign}(x)$	符号函数
$\text{cos}(x)$	余弦	$\text{sin}(x)$	正弦
$\text{cosh}(x)$	双曲余弦	$\text{sinh}(x)$	双曲正弦
$\text{exp}(x)$	指数函数 e^x	$\text{sqrt}(x)$	平方根
$\text{fix}(x)$	对 0 方向取整数	$\text{tan}(x)$	正切

向量与矩阵的输入与运算, 因向量与矩阵所含元素较多, 一个一个地输入不但麻烦而且容易出错, MATLAB 提供了一些有效的输入方法, 现简介如下.

在 MATLAB 中,冒号“:”是一个重要字符,如键入 $x=1:4$; 则产生一个首元素为 1,尾元素为 4,步长是 1 的含 4 个元素的行向量.此时, $x=[1,2,3,4]$. 同学们在输入指令时省略分号看看会出现什么情形.再输入 $y=0:0.5:3$,产生的行向量 $y=[0,0.5,1,1.5,2,2.5,3]$,当然可直接输入 y ,但要注意,作为行向量输入,元素之间要用逗号,或空格分隔.下面给出一个例子,说明一些问题.键入 $x=(0:0.2:2.0)'$; $y=\exp(-x) \cdot \sin(x)$; $[x,y]$ ↵ 显示:

$$\text{ans} = \begin{bmatrix} 0 & 0 \\ 0.2000 & 0.1627 \\ 0.4000 & 0.2610 \\ 0.6000 & 0.3099 \\ 0.8000 & 0.3223 \\ 1.0000 & 0.3096 \\ 1.2000 & 0.2807 \\ 1.4000 & 0.2430 \\ 1.6000 & 0.2018 \\ 1.8000 & 0.1610 \\ 2.0000 & 0.1231 \end{bmatrix}$$

说明 $x=(0:0.2:2.0)$ 是一个行向量,“'”表示转置,故 $x=(0:0.2:2.0)'$ 是列向量,第一行为 0,步长为 0.2,最后一行数为 2.000, x 是一个 11 维列向量. $\exp(-x)$ 是对列向量 x 每项取负,再对每项进行指数运算而得到一个新的 11 维列向量, $\sin(x)$ 同样,之后两个列向量作 $\cdot$ 运算,即处于相同行的元素相乘组成新的 11 维列向量 y . 第三条语句 $[x,y]$ 构成一个 2 列 11 行的矩阵.再注意一些细节,在前两句输入,结束时有“;”,所以不显示结果,第三句后无分号,所以显示了计算结果,又第三行不是输入 $z=[x,y]$,而是 $[x,y]$,故 MATLAB 自动将结果赋给内置变量 ans ,并显示之,关于其他一些内置变量见表 1.4.

表 1.4

记 号	含 义
pi	π 的近似值 $\pi=3.14159265358979$
inf	正无穷大,定义为 $\frac{1}{0}$
NaN	在 IEEE 运算中产生 $\frac{0}{0}, \frac{\infty}{\infty}, 0 \times \infty$ 等运算
i, j	虚数单位,定义为 $i=j=\sqrt{-1}$

内置变量是 MATLAB 启动时自动定义的,不会被 clear 清除,最好不要将这些变量再重新定义为其值,尽管可以重新定义.

下面给出几种矩阵的赋值形式

```
A=[1 2 3;4 5 6;7,8,9]↙
```

```
A=
```

```
1 2 3
4 5 6    (*)
7 8 9
```

说明 各元素间可用空格分隔,亦可用逗号“,”分隔,分号表示换行,(*)式为显示的结果.

```
av=1:9; % 产生 9 个元素行向量.
```

```
C=[av(7:9);av(4:6);av(1:3)]↙
```

```
C=
```

```
7 8 9
4 5 6
1 2 3
```

无论对矩阵还是向量,都可借助下标访问,如 $C(1,1)=7$, $C(3,3)=3$, $av(1)=1$, $av(4)=4$. 顺便说明,“%”后的文字部分为注释语句,且一个“%”只管一行.若注释语句过长需要换行,则必须再加“%”.

```
B=C(1:2,3)↙
```

```
B=9
```

```
6
```

说明 $C(1:2,3)$ 表示第一行第三个元素与第二行第三个元素组成的二维列向量,而 $C(2:3,2:3)$ 则表示矩阵

$$\begin{bmatrix} 5 & 6 \\ 2 & 3 \end{bmatrix}.$$

$A(:,3)$ 代表第三列元素组成的子矩阵, $A(1:2,:)$ 表示 A 中前两行组成的子矩阵,这些内容同学们可方便地上机验证.

关于矩阵的运算,仅给出同学们熟悉的一些基本指令,如表 1.5 所示.

表 1.5

运算指令	含 义
A'	转 置
$A \pm B$	矩阵相加减
$s * A$	数乘矩阵, s 是一个数值
$\text{inv}(A)$	求逆运算
A^n	矩阵 A 的 n 次幂
$\det(A)$	矩阵 A 的行列式值
$[L, U] = \text{lu}(A)$	矩阵 A 的 LU 分解
$[Q, R] = \text{qr}(A)$	矩阵 A 的 QR 分解

续表

运算指令	含 义
<code>rank(A)</code>	矩阵 A 的秩
<code>[X,D]=eig(A)</code>	X 为 A 的特征向量, D 为特征值
<code>poly(A)</code>	求 A 的特征多项式
<code>size(A)</code>	返回 A 的大小

最后给出一个利用下标索引矩阵的技术小结表(表 1.6),其中 A 表示一个矩阵.

表 1.6

输入形式	含 义
<code>A(r,c)</code>	A 中第 r 行第 c 列元素
<code>A(r,:)</code>	A 中第 r 行构成的行向量
<code>A(:,c)</code>	A 中第 c 列构成的列向量
<code>A(:)</code>	对 A 按列看作一个列向量
<code>A(i)</code>	表示列向量 $A(:,)$ 中第 i 个元素

说明 对 $A(:,)$,若 A 为一个 3×3 矩阵如下

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},$$

则 $A(:,) = [1 \ 4 \ 7 \ 2 \ 5 \ 8 \ 3 \ 6 \ 9]'$, $A(4) = 2$.

1.3 MATLAB 的语言程序设计简介

熟悉 BASIC 语言的读者都知道编程的基本语句是由赋值、数值运算、关系运算、逻辑运算与程序流控制语句几部分组成. MATLAB 亦不例外,赋值与数值运算前边已有介绍,下面简介余下几部分,之后给出几个小程序.

1. 关系运算

共有 6 个关系操作符,如表 1.7 所示.

表 1.7

关系操作符	注 释
<code><</code>	小于
<code>></code>	大于
<code><=</code>	小于等于
<code>>=</code>	大于等于
<code>==</code>	等于
<code>~=</code>	不等于

函数 `find()` 在关系运算中很有用,如 y 是一个向量,语句 `k=find(y>3.0)`,
`x=y(k)`. k 表示 y 中所有大于 3 的元素的下标,而 x 是 y 中大于 3 的值组成的新的
 向量.

2. 逻辑运算

共有 4 种逻辑运算,分别为“&”、“|”、“~”、“xor”,分别代表与、或、非、异或.
 如果两个标量 a 和 b 参加运算,因为在逻辑运算中只有两个量,0 与 1,任何非零数
 均为 1,所以在表 1.8 的运算规则表中,输入 a, b 的值,或者为 0,或者为 1.

表 1.8

输 入		与	或	异或	非
a	b	$a \& b$	$a b$	$\text{xor}(a, b)$	$\sim a$
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

说明 “&”, “|”, “xor”用于比较两个同阶矩阵(包括标量与向量),具体到矩
 阵中的元素,如 A, B 是两个同阶矩阵,那么 $A \& B$ 表示 A, B 中处于相同行、列中相
 应元素作逻辑“与”运算而得到的另一个 0-1 矩阵.比较特殊的是标量 a 与矩阵 A
 作逻辑运算,则将 a 和 A 中元素逐一进行逻辑运算,得到一个与 A 同阶的 0-1
 矩阵.

需要强调的是:在逻辑操作符、关系操作符与运算操作符三者中,逻辑操作符
 优先级最小,而在逻辑操作符中,“与”、“或”有相同的优先级,且级别最小,“非”的
 优先级最高,这些规则很难记忆,最好的方法是加括号“()”,以确定运算的先后
 顺序.

“~”为一元操作符,如 $A = [0 \ 2; 3 \ 0]$, $\sim A = [1 \ 0; 0 \ 1]$,故无论 A 为何
 值, $A | (\sim A)$ 返回 1, $A \& (\sim A)$ 返回 0.

3. 关系与逻辑运算函数

除前面介绍的关系与逻辑运算符外, MATLAB 还提供了一些关系与逻辑运
 算函数,如表 1.9 所示.

表 1.9

函 数	说 明
<code>xor(x, y)</code>	异或运算, x 或 y 非 0, 返回 1, 其余返回 0
<code>any(x)</code>	向量 x 中有任一元素非 0 返回 1
<code>all(x)</code>	向量 x 中所有元素非 0 返回 1, 矩阵 x 中的每一列所有元素非 0 返回 1

续表

函 数	说 明
isinf(x)	元素为无穷大,返回 1
isnan(x)	元素为不定值时返回 1
finite(x)	元素有限返回真值 1

说明 any 与 all 命令对于矩阵,总是按列处理并返回一个带有处理列所得结果的一个行向量,故 any(any(A))总是将矩阵 A 化为一个标量条件,类似函数还有一些,不再一一介绍.

4. 控制语句

for 循环

以具体例子说明.键入:

```
x=zeros(1,5)           %预定义 x 为 1 行 5 列 0 向量
                        %以加快循环速度

for n=1:5
    x(n)=sin(n*pi/10);  %“;”不显示结果
end
```

键入:

x ↙

显示:

```
x=0.3090  0.5878  0.9511  1.000  0.9511
```

说明 1:5 是 MATLAB 标准的赋值语句,n 为行向量[1 2 3 4 5],第一次循环中,取 $n(1)=1, \dots$,第五次循环中取 n 中第五个元素 $n(5)=5$,之后循环结束,故而循环内不能对 n 重新赋值,如在 end 语句前加 $n=5$ 以结束循环.

类似 C 语言的循环嵌套,对 MATLAB 也是适用的,仅以一例说明.键入:

```
for n=1:5
    for m=5:-1:1          %嵌套中适当的缩进格式使
        A(n,m)=n^2+m^2;  %程序清晰
    end
    disp(n);              %显示 n 的值
end
```

键入:

A ↙

A=

```

2    5    10   17   26
5    8    13   20   29
10   13   18   25   34
17   20   25   32   41
26   29   34   41   50

```

读者可依据显示 A 的值自己考虑 A 的生成过程.

while 循环

for 循环在循环之初就已定下其循环次数,而 while 循环则以不定次数执行 while 与 end 间的语句,确切的循环次数依赖于对一个表达式真伪的判别.看下例.键入:

```

num=0;EPS=1;
while(1+EPS)>1
    EPS=EPS/2;           %分号不显示结果
    num=num+1;           %num 记录循环次数
end

```

键入:

```

num  ↙
num=53

```

说明 变量 EPS 不断被 2 除,当 EPS 小至 $1+EPS=1$ 时(由计算机精度所定),循环条件为假,循环结束,又因 eps 为内置变量,所以用大写 EPS.

if-else-end 结构

这一结构的最简单形式为

```

if 表达式
    指令

```

```

end

```

当表达式条件为真,执行 if-end 之间命令段,否则跳过 if-end 结构.

两种选择的结构形式为

```

if      表达式
    命令段 1           %若表达式真,执行此段
else
    命令段 2           %若表达式假,执行 2 段
end

```

当有三个或三个以上选择时,具有类似形式,仅以三个选择为例说明.

```

if      表达式 1
    命令段 1

```



```
elseif 表达式 2
    命令段 2
elseif 表达式 3
    命令段 3
else    命令段 4
end
```

MATLAB 将依次检查表达式 1,2,3 等,只执行第一个表达式为真的命令段,而略过余下的表达式不予检验,若所有表达式为假,则执行命令段 4,而 else 语句可缺省,若不存在 else 语句而所有表达式之值为 0 时,程序跳过 if-end 间所有语句而按顺序执行 end 后的语句.

可利用 if-else-end 结构给出跳出或中断 for 与 while 循环的合理方法如下例. 键入:

```
EPS=1;
for num=1:1000
    EPS=EPS/2;           %要有分号
    if(1+EPS)<=1
        EPS=EPS * 2
        break
    end
end
EPS=2.2204e-016
```

键入:

```
num ↙
num=53
```

if-else-end 结构检验 EPS 是否变得足够小.若是执行 $EPS * 2$,且用 break 中断 for 循环,这时 $num=53$,而非进行 1000 次循环.

应注意,当执行 break 语句时,MATLAB 跳出 for 循环,并执行“EPS=”语句,在本例情形下,在返回提示符同时并显示 EPS 的值.如果 break 语句出现在一个嵌套 for 或 while 结构中时,MATLAB 只跳出 break 所在循环而不是整个嵌套结构.

最后给出一个 M 文件实例,通过注解与文件的运行,可了解一些有关程序设计的方法.

```
function f=mmono(x)           %要调用的函数
% MMONO 测试单调向量
% 如果 x 是严格单调增加,返回值 2
```

```

% 如果 x 非减,返回 1
% 若 x 非增,返回 -1
% 若 x 是严格单调减,返回 -2
% 其他情形返回 0
x=x(:)           %使 x 成为一个列向量,维数不定
y=diff(x)         %在相继元素之间作差
if all(y>0)        %先测试严格增
    f=2;
elseif all(y>=0)
    f=1;
elseif all(y<0)    %测试严格减
    f=-2;
elseif all(y<=0)
    f=-1;
else
    f=0;           %其余情形
end

```

试运行上述函数.键入:

```

mmono(1:12)       %严格单增
ans=2
mmono([1,3,2,-1]) %非单调
ans=0
mmono([12:-1:0,0,-1]) %非增
ans=-1

```

要编写一个应用程序,还应简要知道 M 文件与 M 函数及它们的调用及存盘方式,亦应知道一些有关编程方面的功能语句及 MATLAB 所提供的大量内部函数,这需要读者去查阅有关 MATLAB 编程问题的相关书籍.本章最后亦有简单介绍.

1.4 特殊量与常用函数

1. 几个特殊量

- (1) 最小浮点数 eps;
- (2) 正无穷大 inf;

(3) 不定值 NaN, 特指 $\frac{0}{0}$.

例 1.1 `a=[0,1,0];b=[1 0 0];c=a./b` ✓

Warning Divide by Zero

`C=0 inf NaN`

说明 “./”表示两个向量的右除.`a./b`的结果是一个与 `a,b` 同维数的向量,各分量按如下方式得到:`b` 的第一个分量去除 `a` 的第一个分量得 `c` 的第一个分量等.“.\”则表示向量的左除,`a.\b`的结果为 $\begin{bmatrix} \frac{1}{0} & \frac{0}{1} & \frac{0}{0} \end{bmatrix} = (\text{inf}, 0, \text{NaN})$.

2. 常用函数

(1) 标量函数 `round`: 四舍五入取整;

(2) `floor`: 向 $-\infty$ 方向取整;

(3) `ceil`: 向 $+\infty$ 方向取整;

(4) `fix`: 向 0 方向取整;

(5) `rats`: 有理逼近.

这些函数本质上是作用于标量的,当作用于矩阵与数组时,则作用于其每一个元素.

例 1.2 键入:

`a=[-3.5,4.6];`

`b=round(a),c=floor(a),d=ceil(a),...`

`e=fix(a),f=rats(a)` ✓

`b=-4 5`

`c=-4 4`

`d=-3 5`

`e=-3 4`

`f=-7/2 23/5`

另一个计算函数值的命令是 `feval('F',x)`, `F` 是表示函数名的字符串.

例 1.3 键入:

`x=(0:0.2:1)*pi;`

`y=feval('sin',x)` ✓

`y=`

0 0.5878 0.9511 0.9511 0.5878 0.0000

3. 向量函数

有些函数只有作用于行或列向量时才有意义,这样的函数称为向量函数.向量

函数亦可作用于矩阵,此时它产生一个行向量,行向量的每个元素是函数作用于矩阵相应列的结果.常用的有 `max`; `min`; `sum`(和); `length`(长度); `mean`(平均值); `median`(中值); `prod`(乘积); `sort`(从小到大排列).

例 1.4 键入:

```
a=[4,3.1,-1.2,0,6];
b=min(a),c=sum(a),d=median(a),e=sort(a)
b=-1.2
c=11.9
d=3.100
e=-1.200  0  3.1000  4.000  6.000
```

4. 矩阵函数

- (1) `zeros`:生成 0 矩阵;
- (2) `ones`:生成 1 阵;
- (3) `eye`:生成单位阵;
- (4) `rand`:生成(0,1)均匀分布随机矩阵;
- (5) `randn`:生成标准正态分布矩阵;
- (6) `hilb`:生成 Hilbert 矩阵;
- (7) `magic`:生成幻方矩阵;
- (8) `triu`:生成或提取上三角阵;
- (9) `tril`:生成或提取下三角阵;
- (10) `diag`:生成或提取对角阵;
- (11) `norm(A)`:给出矩阵 A 的范数.

例 1.5 键入:

```
w=zeros(2,3) %生成 2×3 零矩阵
w=0  0  0
    0  0  0
>> u=ones(3) %3×3 阶 1 矩阵
u=1  1  1
    1  1  1
    1  1  1
v=eye(3,4) %3×4 阶对角线为 1 的矩阵
v=1  0  0  0
    0  1  0  0
    0  0  1  0
```

```
x=rand(1,3)✓ %1×3 阶的(0,1)均匀分布随机矩阵
x=
    0.2311    0.8913    0.0185
```

1.5 图形功能

1. 二维图形

MATLAB 常用于绘制二维图形的命令是 plot, 看几个实例. 键入:

```
y=[0,0.58,0.70,0.95,0.83,0.25];
plot(y)✓
```

读者可自行上机试验. 生成的图形是以 1, 2, 3, ..., 6 为横坐标, 数组 y 的数值为纵坐标画出的折线. 键入:

```
x=linspace(0,2*pi,30);
y=sin(x);
plot(x,y)✓
```

函数 linspace(a, b, n) 生成从 a 到 b 共 n 个数值的等差数组, 上述语句生成 $[0, 2\pi]$ 上由 30 个点连成的光滑正弦曲线.

在同一个画面上, 可画出多条曲线. 键入:

```
x=0:pi/15:2*pi;
y1=sin(x);
y2=cos(x);
plot(x,y1,x,y2)✓
```

或者将第二句至第四句改写为:

```
y=[sin(x);cos(x)];plot(x,y)✓
```

二者绘图效果相同. 键入:

```
y1=sin(x);
y2=cos(x);
plot(x,y1,'b:',x,y2,'g-')✓
```

与下面 plot 生成图形比较, 键入:

```
plot(x,y1,'b:',x,y2,'g-',x,y1,'+',x,y2,'*')✓
```

说明 格式(线性与颜色)说明是一个字符串, 故一定要用单引号 ' ' 括起来.

在一个图形上可以加网格、标题、x 轴标记、y 轴标记, 用下列命令完成这些工作. 键入:

```
x=linspace(0,2*pi,30);
```

```

y=[sin(x);cos(x)];
plot(x,y);
grid                                %加网格线
xlabel('independent variable x')   %加 x 轴标记
ylabel('dependent variable y')
title('sine and cosine curves')    %加标题
可以在图形的任何一个位置上加一个字符串,如用命令:
text(2.5,0.7,'sinx')

```

表示在坐标 $x=2.5, y=0.7$ 处加上字符串 $\sin x$.更方便的是用鼠标来确定字符串的位置,如输入指令:

```
gtext('sinx'),gtext('cosx')
```

在图形窗口,“十”字线的交点是字符串的位置,用鼠标一点,即可将字符串放在那里.

关于坐标轴的定制,可用 `axis` 命令控制,常用的方式有:

```

axis([xmin,xmax,ymin,ymax])
axis equal 或 axis('equal')    %x,y 轴单位长相同
axis square 或 axis('square')  %图形呈方形
axis off 或 axis on            %清除及恢复坐标刻度

```

此外还有 `axis auto`; `axis image`; `axis xy`; `axis ij`; `axis normal`; `axis on`; `axis(axis)`.用法参考在线帮助.

欲在同一屏上建立多个坐标系,可用命令 `subplot`, `subplot(m,n,p)` 将一个画面分成 $m \times n$ 个图形区域, p 代表当前区域号,在每个区域中可分别画一个图.键入:

```

x=linspace(0,2*pi,30);
y=sin(x);z=cos(x);
u=2*sin(x).*cos(x); %“.”*”两向量对应元素乘积作为新向量对应元素
v=sin(x)./(cos(x)+eps);
subplot(2,2,1),
plot(x,y),axis([0 2*pi -1 1]),title('sin(x)')
subplot(2,2,2),
plot(x,z),axis([0 2*pi -1 1]),title('cos(x)')
subplot(2,2,3),
plot(x,u),axis([0 2*pi -1 1]),title('2sin(x)*cos(x)')
subplot(2,2,4),
plot(x,v),axis([0 2*pi -20 20]),title('sinx/cosx')

```

得到 2×2 共 4 幅图.

还有一些其他画二维图形的命令如下: