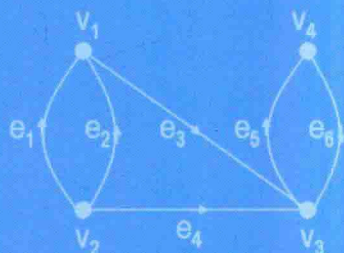


数值代数实验指导

Numerical linear algebra using matlab

周解勇◎编著



$$M(D) = \begin{matrix} & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} -1 & -1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \end{matrix}$$

$$(x' \ y' \ z' \ 1) = (x \ y \ z \ 1) \begin{bmatrix} \cos \gamma & \sin \gamma & 0 & 0 \\ -\sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

数值代数实验指导

周解勇 编著

上海财经大学出版社

图书在版编目(CIP)数据

数值代数实验指导/周解勇编著. —上海:上海财经大学出版社,
2017.8

ISBN 978-7-5642-2755-5/F · 2755

I. ①数… II. ①周… III. ①线性代数计算法
IV. ①O241.6

中国版本图书馆 CIP 数据核字(2017)第 120318 号

☐ 责任编辑 何苏湘

☐ 封面设计 张克瑶

SHUZHAI DAISHU SHIYAN ZHIDAO

数值代数实验指导

周解勇 编著

上海财经大学出版社出版发行

(上海市中山北一路 369 号 邮编 200083)

网 址: <http://www.sufep.com>

电子邮箱: webmaster@sufep.com

全国新华书店经销

上海华教印务有限公司印刷装订

2017 年 8 月第 1 版 2017 年 8 月第 1 次印刷

787mm×1092mm 1/16 6.25 印张 102 千字

定价:33.00 元

前 言

数值代数是信息与科学计算本科专业的一门核心方向课程,涉及线性方程组求解、最小二乘问题、特征值求解等理论与方法,对于培养学生的理论和实践应用能力具有十分重要的意义。这门课程在进行理论教学的同时,需要同步进行实验教学,以加深学生对于理论知识的理解和运用。因此,本实验指导书以徐树芳先生等人编写的《数值线性代数》为参照,对本书的理论知识进行归纳,并辅以大量算例。特别地,本书还对Krylov子空间方法的部分进行扩充,培养学生处理大规模问题的能力,以适应大数据时代的需求。作者希望该书能尽量做到独立使用,作为信息与科学计算专业、应用数学专业及相关专业的本科生与研究生、工程技术人员的参考用书。

因作者水平有限,时间仓促,本书多有不足,请同行及读者指正。

本书的写作得到上海财经大学数学学院、上海财经大学教务处和上海财经大学实验教学中心的支持,作者对他们表示感谢。本书的责任编辑何苏湘女士,我的研究生余宏伟同学、蓝日辉同学也为本书付出了大量的劳动,作者对他们表示感谢。

编者

2017年4月

目 录

前言	1
1 Matlab应用基础	1
§ 1.1 Matlab简介	1
§ 1.2 Matlab开发环境	2
§ 1.3 Matlab的矩阵计算功能	3
§ 1.4 Matlab的常用控制语句和绘图语句	8
§ 1.5 Matlab的常用绘图语句	11
§ 1.6 Matlab中M文件的编写及使用	18
§ 1.7 Matlab中的help命令	19
2 线性方程组的直接解法	21
§ 2.1 概念回顾	21
2.1.1 三角形方程组和三角分解	21
2.1.2 选主元三角分解	24
2.1.3 平方根法	25
§ 2.2 算例分析	26
2.2.1 Gauss消去法的稳定性	26
2.2.2 方程组算例	28
本章习题	32

3	线性方程组的敏度分析与消去法的舍入误差分析	34
§ 3.1	概念回顾	34
3.1.1	向量范数	34
3.1.2	矩阵范数	35
3.1.3	线性方程组的敏度分析	36
3.1.4	计算解的精度估计和迭代改进	37
§ 3.2	算例	38
	本章习题	39
4	最小二乘问题的解法	40
§ 4.1	概念回顾	40
4.1.1	最小二乘问题	40
4.1.2	正交变换	41
4.1.3	正交化方法	44
§ 4.2	算例	46
	本章习题	50
5	线性方程组的古典迭代解法	51
§ 5.1	概念回顾	51
5.1.1	古典迭代格式	51
5.1.2	收敛性分析	52
§ 5.2	算例	54
	本章习题	58
6	Krylov子空间方法简介	59
§ 6.1	子空间投影法简介	59
§ 6.2	为什么使用Krylov子空间方法	60
§ 6.3	Arnoldi过程	62

§ 6.4	GMRES方法	65
6.4.1	基本的GMRES方法	65
6.4.2	重开始的GMRES方法	72
§ 6.5	对称Lanczos算法	72
§ 6.6	共轭梯度法	74
§ 6.7	收敛性分析	78
§ 6.8	预条件方法	78
6.8.1	预处理的CG方法	79
6.8.2	预处理的GMRES	80
§ 6.9	数值算例	81
	本章习题	82
7	非对称特征值问题的计算方法	83
§ 7.1	概念回顾	83
7.1.1	幂法	84
7.1.2	反幂法	85
7.1.3	QR方法	86
§ 7.2	数值算例	89
	本章习题	90
	参考文献	91

1 Matlab应用基础

§1.1 Matlab简介

Matlab是美国Mathworks公司推出的用于数值计算和图形处理计算系统环境。虽然Cleve Moler 教授开发它的初衷是为了更简单、更快捷地解决矩阵运算,但Matlab 现在的发展已经使其成为一种集数值运算、符号运算、数据可视化、图形界面设计、程序设计、仿真等多种功能于一体的集成软件。具有卓越的数值计算能力,还提供专业水平的符号计算、文字处理、可视化建模仿真和实时控制等功能。Matlab的指令表达式与数学、工程中常用的形式十分相似,是国际公认的优秀数学应用软件之一。

整个Matlab系统由两个部分构成,即Matlab内核和辅助工具箱。Matlab语言有两种基本的数据运算量:数组和矩阵。单从形式上,它们之间是不容易区分的,包括控制语句、函数、数据结构、输入输出及面向对象等特点的高级语言,它具有以下功能:

- (1) 语言简洁,编程效率高;
- (2) 交互性好,使用方便;
- (3) 强大的绘图能力,便于数据可视化;
- (4) 学科众多、领域广泛的工具箱;
- (5) 开放性好,易于扩充;
- (6) 与C 语言和Fortran 语言有良好的接口。

§1.2 Matlab开发环境

Matlab 的主界面是一个高度集成的工作环境, 有4个不同职责分工的窗口, 它们分别是命令窗口(Command Window)、历史命令(Command History)窗口、当前目录(Current Directory)窗口和工作空间(Workspace)窗口。除此之外, Matlab 6.5 之后的版本还添加了开始按钮(Start)。我们现介绍这些窗口功能如下:

(1) 主窗口: 该窗口不能进行任何计算任务的操作, 只用来进行一些整体的环境参数的设置。

(2) 命令窗口(Command Window): 可输入的对象除Matlab命令之外, 还包括函数、表达式、语句以及M 文件名或MEX 文件名等, 同时, 还在这个窗口里显示命令执行结果。

(3) 历史命令(Command History)窗口: 历史命令窗口是Matlab用来存放曾在命令窗口中使用过的语句。它借用计算机的存储器来保存信息。其主要目的是为了便于用户追溯、查找曾经用过的语句, 利用这些既有的资源节省编程时间。

(4) 当前目录(Current Directory)窗口: Matlab 借鉴Windows 资源管理器管理磁盘、文件夹和文件的思想, 设计了当前目录窗口。利用该窗口可组织、管理和使用所有Matlab 文件和非Matlab 文件, 例如新建、复制、删除和重命名文件夹和文件。甚至还可用此窗口打开、编辑和运行M程序文件以及载入MAT 数据文件等。当然, 其核心功能还是设置当前目录。

(5) 工作空间(Workspace)窗口: 工作空间窗口的主要目的是为了对Matlab 中用到的变量进行观察、编辑、提取和保存。从该窗口中可以得到变量的名称、数据结构、字节数、变量的类型甚至变量的值等多项信息。工作空间的物理本质就是计算机内存中的某一特定存储区域, 因而工作空间的存储表现亦如内存的表现。

§1.3 Matlab的矩阵计算功能

矩阵是Matlab数据存储的基本单元，在Matlab语言系统中几乎一切运算均是以对矩阵的操作为基础的。

1.3.1 矩阵的生成

(1) 直接输入法

从键盘上直接输入矩阵是最方便、最常用的创建数值矩阵的方法，尤其适合较小的简单矩阵。在用此方法创建矩阵时，应当注意以下几点：

- 输入矩阵时要以“[]”为其标识符号，矩阵的所有元素必须都在括号内。
- 矩阵同行元素之间由空格或逗号分隔，行与行之间用分号或回车键分隔。
- 矩阵大小不需要预先定义。
- 矩阵元素可以是运算表达式。
- 若“[]”中无元素表示空矩阵。

另外，在Matlab语言中冒号的作用是最为丰富的。首先，可以用冒号来定义行向量，如图1.1所示。

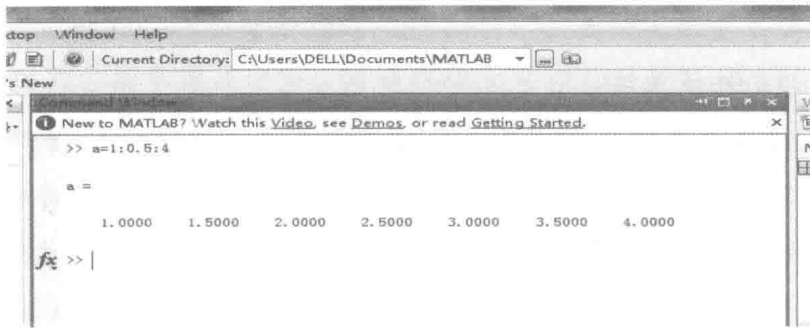
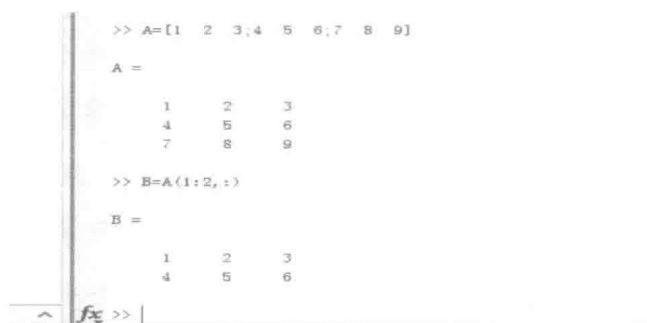


图 1.1 矩阵的生成

其次,通过使用冒号,可以截取指定矩阵中的部分,如图1.2所示。



```
>> A=[1 2 3;4 5 6;7 8 9]
A =
     1     2     3
     4     5     6
     7     8     9

>> B=A(1:2,:)
B =
     1     2     3
     4     5     6

f>> |
```

图 1.2 矩阵的生成

通过上例可以看到B是由矩阵A的1到2行和相应的所有列的元素构成的一个新的矩阵。在这里,冒号代替了矩阵A的所有列。

Matlab中,也采用特殊的矩阵向量表示序列,但Matlab序列下标默认从1开始递增,因此表示离散信号 $\{x(n)\} = \{\cdots, x(-1), x(0), \cdots\}$,一般采用两个向量分别对信号的自变量和因变量进行描述。如 $x(n) = \{2, 1, -1, 0, 1, 4, 3, 7\}$, $n = -3, \cdots, 4$ 。

在Matlab中表示为:

$n = [-3, -2, -1, 0, 1, 2, 3, 4];$ %自变量取值

$x = [2, 1, -1, 0, 1, 4, 3, 7];$ %因变量取值

注意:分号“;”表示不回显表达式的值;“%”表示其后内容为注释对象。

(2) 外部文件读入法

Matlab语言也允许用户调用在Matlab环境之外定义的矩阵。可以利用任意的文本编辑器编辑所要使用的矩阵,矩阵元素之间以特定分断符分开,并按行列布置。另外也可以利用Load函数,其调用方法为:Load+ 文件名[参数]。

Load函数将会从文件名所指定的文件中读取数据,并将输入的数据赋给以文件名命名的变量,如果不给定文件名,则将自动认为matlab.mat文件为操作对象,如果该文件在Matlab搜索路径中不存在时,系统将会报错。

```
1 1 1
```

例如: 事先在记事本中建立文件: 1 2 3, 并以data1.txt 保存。在Matlab

```
1 3 6
```

命令窗口中输入:

```
>> load data1.txt
```

```
>> data1
```

```
1 1 1
```

```
data1 = 1 2 3
```

```
1 3 6
```

(3) 特殊矩阵的生成

对于一些比较特殊的矩阵(单位阵、矩阵中含1或0较多), 由于其具有特殊的结构, Matlab 提供了一些函数用于生成这些矩阵。常用的有下面几个:

zeros(m) 生成m阶全0 矩阵

eye(m) 生成m阶单位矩阵

ones(m) 生成m阶全1 矩阵

rand(m) 生成m阶均匀分布的随机矩阵

randn(m) 生成m阶正态分布的随机矩阵

1.3.2 矩阵的基本数学运算

矩阵的基本数学运算包括矩阵的四则运算、与常数的运算、逆运算、行列式运算、秩运算、特征值运算等基本函数运算, 这里进行简单介绍。

(1) 四则运算

矩阵的加、减、乘运算符分别为“+、-、*”, 用法与数字运算几乎相同, 但计算时要满足其数学要求(如: 同型矩阵才可以加、减)。在Matlab中矩阵的除法有两种形式: 左除“\”和右除“/”。在传统的Matlab 算法中, 右除是先计算矩阵的逆再相乘, 而左除则不需要计算逆矩阵直接进行除运算。通常右除要快一点, 但左除可避免被除矩阵的奇异性所带来的麻烦。

(2) 与常数的运算

常数与矩阵的运算即是同该矩阵的每一元素进行运算。但需注意进行数除时, 常数通常只能做除数。

(3) 基本函数运算

矩阵的函数运算是矩阵运算中最实用的部分，常用的主要有以下几个：

$\det(a)$ 求矩阵 a 的行列式

$\text{eig}(a)$ 求矩阵 a 的特征值

$\text{inv}(a)$ 或 a^{-1} 求矩阵 a 的逆矩阵

$\text{rank}(a)$ 求矩阵 a 的秩

$\text{trace}(a)$ 求矩阵 a 的迹（对角线元素之和）

例如：

```
>> a = [2 1 -3 -1; 3 1 0 7; -1 2 4 -2; 1 0 -1 5];
```

```
>> a1 = det(a);
```

```
>> a2 = det(inv(a));
```

```
>> a1 * a2
```

```
ans =
```

```
1
```

注意：命令行后加“；”表示该命令执行但不显示执行结果。

1.3.3 矩阵的数组运算

我们在进行工程计算时常常遇到矩阵对应元素之间的运算。这种运算不同于前面讲的数学运算，为有所区别，我们称之为数组运算。

(1) 基本数学运算

数组的加、减与矩阵的加、减运算完全相同。而乘除法运算有相当大的区别，数组的乘除法是指两同维数组对应元素之间的乘除法，它们的运算符为“.”、“/”或“\”。前面讲过常数与矩阵的除法运算中常数只能做除数。在数组运算中有了“对应关系”的规定，数组与常数之间的除法运算没有任何限制。

另外，矩阵的数组运算中还有幂运算 $\wedge$ 、指数运算 $\exp$ 、对数运算 $\log$ 、和开方运算 sqrt 等。有了“对应元素”的规定，数组的运算实质上就是针对数组内部的每个元素进行的。

例如

```
>> a = [2 1 -3 -1; 3 1 0 7; -1 2 4 -2; 1 0 -1 5]
```

```
>> a^3;
```

```
ans =  
32 -28 -101 34  
99 -12 -151 239  
-1 49 93 8  
51 -17 -98 139
```

```
>> a.^3
```

```
ans =  
8 1 -27 -1  
27 1 0 343  
-1 8 64 -8  
1 0 -1 125
```

由上例可见矩阵的幂运算与数组的幂运算有很大的区别。

(2) 逻辑关系运算

逻辑运算是Matlab中数组运算所特有的一种运算形式，也是几乎所有的高级语言普遍适用的一种运算。它们的具体符号、功能及用法见表1.1。

表 1.1 逻辑运算

符号运算符	功能	函数名
==	等于	eq
=	不等于	ne
<	小于	lt
>	大于	gt
<=	小于等于	le
>=	大于等于	ge
&	逻辑与	and
	逻辑或	or
~	逻辑非	not

在表1.1中,

(a) 在关系比较中, 若比较的双方为同维数组, 则比较的结果也是同维数组。它的元素值由0和1组成。当比较双方对应位置上的元素值满足比较关系时, 它的对应值为1, 否则为0。

(b) 当比较的双方中一方为常数, 另一方为一数组, 则比较的结果与数组同维。在算术运算、比较运算和逻辑与、或、非运算中, 它们的优先级关系先后为: 比较运算、算术运算、逻辑与或非运算。

逻辑运算如图1.3所示。

```
>> a=[1 2 3; 4 5 6; 7 8 9];
>> x=5;
>> y=ones(3)+5;
>> xa=x<=a;

xa =

     0     0     0
     0     1     1
     1     1     1

>> b=[0 1 0; 1 0 1; 0 0 1];
>> ab=a&b;

ab =

     0     1     0
     1     0     1
     0     0     1

fx >> |
```

图 1.3 逻辑运算

§1.4 Matlab的常用控制语句和绘图语句

1.4.1 循环语句

Matlab提供两种循环方式, for循环和while循环。在循环语句中, 一组被重复执行的语句称为循环体。每循环一次, 都必须做出判断, 是继续循环执行还

是中止执行跳出循环, 这个判断的依据称为循环的中止条件。由于这些结构经常包含大量的Matlab命令, 故经常出现在M文件中, 而不是直接加在Matlab提示符下。

(1) for 循环

for循环允许一组命令以固定的和预定的次数重复。for 循环的一般形式是:

```
for    x = array
      commands
end
```

在for和end语句之间的commands按数组中的每一列执行一次。在每一次迭代中, x 被指定为数组的下一列, 即在第 n 次循环中, $x=array(:, n)$ 。使用for循环应该注意:

- (a) for循环不能用for循环内重新赋值循环变量 n 来终止。
- (b) 在for循环内接受任何有效的Matlab数组。
- (c) for循环可按需要嵌套。

(2) while 循环

与for循环以固定次数求一组命令的值相反, while 循环以不定的次数求一组语句的值。while循环的一般形式是:

```
while  expression
      commands
end
```

只要在表达式里的所有元素为真, 就执行while和end 语句之间的commands。通常, 表达式的求值给出一个标量值, 但数组值也同样有效。在数组情况下, 所得到数组的所有元素必须都为真。

1.4.2 选择语句

很多情况下, 命令的序列必须根据关系的检验有条件地执行。在编程语言里, 这种逻辑由某种If-Else-End结构来提供。最简单的If-Else-End结构是:

```
if    expression
      commands
end
```

如果表达式中的所有元素为真(非零), 执行if和end语言之间的commands。当有三个或更多的选择时, If-Else-End结构采用形式

```
if    expression1
      commands evaluated if expression1 is True
elseif expression2
      commands evaluated if expression2 is True
elseif expression3
      commands evaluated if expression3 is True
elseif expression4
      commands evaluated if expression4 is True
elseif ... ..
      ⋮
      ⋮
else
      commands evaluated if no other expression is True
end
```

这种形式, 只和所碰到的、与第一个真值表达式相关的命令被执行; 接下来的关系表达式不检验, 跳过其余的If-Else-End结构。最后的else命令可有可无。