计算科学丛书

The Numerical Analysis Class Library for VC++ and BC++

VC++和BC++数值分析类库

粟塔山　编著
Su Tashan

清华大学出版社

计 算 科 学 丛 书 / Series in Computational Science

王东明 主编

The Numerical Analysis Class Library for VC++ and BC++

VC++和BC++数值分析类库

粟塔山 编著

Su Tashan

清华大学出版社

北 京

内 容 简 介

本书是所随带的 VC++ 和 BC++ 数值分析类库光盘的使用手册。此 VC++ 和 BC++ 数值分析类库涵盖了数值分析领域中大部分常见算法，还包括线性和非线性最优化问题的多种算法以及概率统计中的一些基本算法。此类库中将矩阵和向量当成如 char，int，double 一样的基本变量类型，为矩阵和向量提供了几乎是随心所欲的操作函数。因此，可以在此数值类库的基础上进行二次开发。类库的各项功能均经过严格的检测，并与 MATLAB 作了比较，结果准确无误，效率不相上下。书中以菜单的方式对数值分析类库的每项功能作了详尽的解释，给出了调用方法的示例，对某些算法还提供了相应的数学背景知识。

如果你经常需要使用计算机求解科学与工程中的数值计算问题，特别是希望使用VC++ 或BC++ 做出独立于 MATLAB 的应用软件，本类库是不错的选择。

图书在版编目(CIP)数据

VC++ 和BC++ 数值分析类库/粟塔山编著. 北京：清华大学出版社，2005.11
ISBN 7-302-11943-0

Ⅰ. V… Ⅱ. 粟… Ⅲ. 电子计算机－数值计算 Ⅳ. TP301.6

中国版本图书馆 CIP 数据核字(2005)第 114685 号

出 版 者：清华大学出版社　　地　　址：北京清华大学学研大厦
http://www.tup.com.cn　　邮　　编：100084
社 总 机：010-62770175　　客户服务：010-62776969
组稿编辑：刘　颖
文稿编辑：王海燕
印 装 者：北京鑫海金澳胶印有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×230　印张：16.75　字数：356 千字
版　　次：2005 年 11 月第 1 版　2005 年 11 月第1 次印刷
书　　号：ISBN 7-302-11943-0/O·498
印　　数：1～4000
定　　价：29.80 元（附光盘 1 张）

前 言

数值计算是计算机应用永恒的话题。众多的工程技术人员、高等院校理工科专业的学生都要学习各种各样的数值计算方法，在计算机上编制或大或小的计算程序，以求得他们所研究问题的数值解答。这意味着众多的人耗费着他们宝贵的时间，重复地做着那些相同或类似而又不得不做的事情。可以想像一下，有多少人不止一次地用三重循环编制两个矩阵相乘的程序段。MATLAB 集成环境为我们提供了一个很方便的计算平台，然而，很多情况下，我们需要在 Visual C++ 或 Borland C++ 环境下开发应用软件。尽管在 C++ 环境中能够调用 MATLAB 数学库的资源，但必须在运行环境中安装 MATLAB(或者从 MATLAB 中抽出数学库)。这样一来，做不出独立的应用软件。再者，MATLAB 与C++ 毕竟是两类隔离的开发环境，它们之间不可能真正做到“无缝链接”，在C++ 中调用 MATLAB，要通过 MATLAB 引擎在两个环境中往返传递数据，这势必降低程序的运行效率。

作者在工作中日积月累，尝试做出这个C++ 环境下的数值分析类库(在随书附带的光盘中)，但愿对那些使用VC++ 或BC++ 开发应用程序的人能有所裨益。这个数值分析类库覆盖了数值分析领域中的大部分常见问题，还兼有线性和非线性最优化问题的多种算法以及概率统计中的基本问题(如产生服从某些常见分布的随机数)。各项功能在 Visual C++ 6.0 和 Borland C++ Builder 5.0 环境下经过了多次测试(例如，求 100 阶矩阵的特征根，解 500 个未知量的线性方程组)，并与 MATLAB 作了相应的比较，结果准确无误，效率不相上下。

数值计算问题的类型层出不穷，这个数值分析类库提供的功能不可能满足所有人的所有要求。然而，它有个特点能部分地弥补这一不足。在这个数值分析类库中，内存动态分配形成的矩阵和向量成为一种基本的数据类型，使用它们就像使用C/C++ 中的基本类型 char，int，double 一样。例如，可以很方便地动态定义各种维数的矩阵和向量，免去内存动态分配及内存回收的烦琐操作。数值库还对矩阵和向量提供了几乎是随心所欲的操作(比 MATLAB 更丰富)。因此，用户可以进一步对该数值分析类库进行二次开发，比较轻松地编出自己所需要的数学算法。

本书实际上是该数值分析类库的使用手册。它对数值库的每一项功能(共 270 多项)作了详尽的解释，并给出调用方法的示例。对某些算法还给出相应的数学背景知识。读

者无需熟悉C++ 程序设计，只要知道在C++ 环境下如何编辑源程序并把它们编译链接成可执行程序，就能使用它。本书后面的附录 1 主要是为使用过 C 编程环境但没有使用过 C++ 编程环境的读者编写的。对这部分读者而言，只要读过了这段附录，就足以方便自如地在C++ 环境下使用该数值库编制数值计算程序。附录 2 列出了正确高效地使用数值库应该注意的一些事项。当读者使用数值库的某个函数出现了疑惑的结果，首先应仔细阅读该函数的使用说明，或者，能在附录 2 中找到问题的答案。

本书以菜单的方式列出数值分析类库的各项功能，从目录的每一个小标题，读者大致能了解每一项功能的用途。所以，阅读本书的最好方式是先浏览几遍书的目录，对数值库提供了什么功能做到心中有数。如果对某个小标题的含义仍有疑惑，可以翻到该页大致了解一下它的使用示例，以后需要时再仔细阅读。然后，到计算机上熟悉一下第 1 章、第 2 章和第 3 章的前几项基本功能，这主要包括如何定义矩阵和向量，如何给矩阵和向量的元素赋值，如何执行矩阵和向量的加法、减法和乘法运算。如果在你的应用程序中要对矩阵和向量做某项操作，应该先查阅一下数值库是否提供了你需要的功能（通常你不会失望），或者是否可以组合数值库的几项功能以达到你的目的。尽量避免对矩阵和向量的下标作循环，因为这会降低程序的运行效率。

在数值分析类库中，矩阵和向量作为两种不同的数据类型。有些读者可能会感到疑惑，为什么不把向量看成特殊的矩阵，将它们合二为一呢？数值库之所以把二者分开，目的是既提高运行效率又方便用户。因为矩阵是二级指针结构，向量是一级指针结构。如果矩阵和向量合二为一，向量就必须保持二级指针结构，这既降低了存储效率也降低了运行效率。此外，当向量保持二级指针结构时，“行向量”与“列向量”在内存中的布局是完全不同的，如此，用户就必须小心地区分一个向量 x 究竟是行向量还是列向量，这加重了用户的思维负担。事实上，“行向量”和“列向量”的概念只是在数学运算表达式中才显现出来，在内存中是无须区分的。例如，假设 A 是 n 阶方阵，x 是 n 维向量，而且是你心目中的“列向量”，那么，数学表达式 $y=Ax$ 产生 n 维“列向量”y；数学表达式 $y=x^{\mathrm{T}}A$ 产生 n 维“行向量”y；数学表达式 $a=x^{\mathrm{T}}Ax$ 产生标量 a。这在数值分析类库中如何体现呢？看，程序表达式

```
y = A * x
```

说明现在是把 x 当成“列向量”进行运算，它相当于数学表达式 $y=Ax$；程序表达式

```
y = x * A
```

说明现在把 x 当成“行向量”进行运算，它相当于数学表达式 $y=x^{\mathrm{T}}A$；程序表达式

```
a = x * A * x
```

说明既可以把 x 当成“行向量”，也可以把 x 当成“列向量”，它相当于数学表达式 $a=$

$x^{T}Ax$。由此可见，对于数值库中的“向量”，想让它是“行”，它就是“行”，想让它是“列”，它就是“列”。用户的意图完全可以通过相应的程序表达式得以实现。

数值分析库中矩阵和向量的元素都是 double 类型。某些细心的读者也许会问，该数值库中向量和矩阵的最大尺寸能定义到多大？作者认为，使用C++ 编程，那么操作系统应该是 Win32 的(Win95 以后)。在 Win32 下，一个整数占 4 字节，它的最大正值能达到 $N=2147483647$，也就是说，向量的维数可以达到 N，矩阵的维数可以达到 $N\times N$。在微机上求解的问题，远远不会达到这样的规模。

在数值分析类库中，函数取名以易记为原则。例如，设 x 是一个向量，你能猜出语句

```
x.SmallToBig( );
```

的作用吗？它将 x 的元素按从小到大的次序重新排列。语句

```
x.BigToSmall( );
```

的作用就不用说了。读者别见笑，这种中国式的 English 让你过目不忘。

顺便说明一点，数值分析库中若干算法带有一步随机摄动(这是算法所需要的)，这个摄动会使每次的计算结果有很细微的差异，即使是对于初值数据为确定的问题也如此。

在成书的过程中，清华大学的刘颖先生提出了很好的建议，使得本书对算法背景知识的介绍恰到好处，提纲挈领，又不过于细琐。作者在此谨表谢意。

如果读者在使用数值库时出现了无法解释的结果，或者你对数值库有其他的要求和设想，很感谢你能告诉我，以便作者对数值分析库进一步修正和补充，使它更好地为你服务。

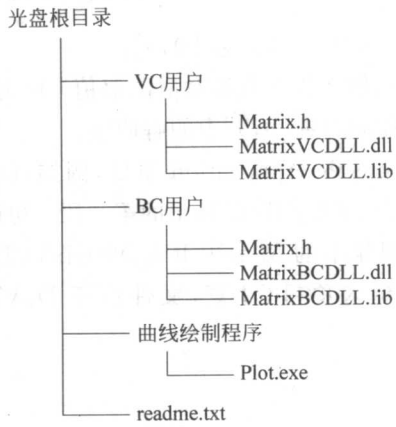
作　者

2003 年 12 月 于长沙国防科技大学

nudtsts@163.com

如何使用C++数值分析类库

首先，要把光盘上的一些文件复制到用户的计算机中。光盘上的内容如下：



这个C++ 数值库分别为 Visual C++ 用户和 Borland C++ 用户提供了不同的版本，两者不能相互替代。当然，你可能在同一台计算机中既使用VC++ 又使用BC++ ，那么两个版本的文件都要复制。

如果是 Visual C++ 用户，请将光盘上“VC 用户”目录下的 MatrixVCDLL. dll 文件复制到 Windows 目录下(或者 Windows\System 目录下)。Borland C++ 用户可以作平行的操作。无论哪一类用户，都要将光盘上“曲线绘制程序”目录下的 Plot. exe 文件复制到 Windows 目录下(或者 Windows\System 目录下)。上述动态链接库文件(*. dll)是数值分析库的主体，它里面包含了除绘制曲线以外的所有功能；应用程序 Plot. exe 为绘制曲线提供支持，它被 *. dll 中的某些函数调用，不能单独运行。如果用户的应用程序中不需要绘制曲线，可以不要它。

需要说明一点：对于BC++ 用户来说，如果要使用绘制曲线功能，必须同时复制 matrixBCDLL. dll 和 matrixVCDLL. dll 两个文件。

光盘上另外的四个文件是“VC 用户”目录下的 Matrix. h，MatrixVCDLL. lib 和“BC 用户”目录下的 Matrix. h，MatrixBCDLL. lib(两个目录下的 Matrix. h 是有差别的)。文件 *. h 和 *. lib 具体到创建某个应用程序时再复制，这两个文件只是在编译链接时才需

要，编译链接完成后，应用程序就不需要它们了。头文件 Matrix. h 给出了数值库中所有函数的声明，凡是使用了数值库函数的源文件都要包含 Matrix. h 这个头文件。MatrixVCDLL. lib(或者 MatrixBCDLL. lib)称为入口库文件，它是应用程序与数值分析动态链接库 *. dll 的中介。编译链接时，它在应用程序调用了 *. dll 中函数的地方插入一个路标，指示应用程序到 *. dll 中哪个地方找到相应的函数。

下面举一个简单的例子，说明如何使用这个C++ 数值分析库(以 Visual C++ 为例)。假设我们需要在区间[0,5]上求解一阶微分方程初值问题：

$$\begin{cases}\dfrac{dx}{dt} = \sin(tx) + t\cos(x\sqrt{t}) + t, \\ x(0) = 0, t \in [0,5],\end{cases}$$

显示 $x(t)$的数值解，再把数值解（节点值和对应的解值）作为文件“diff. dat”存到硬盘上(以后可以读取它)，并绘制出 $x(t)$在[0,5]中的解曲线。

第一步：创建一个 Win32 Console Application 项目，例如，D:\TestMatrix(没有使用过 C++ 编程环境的读者，请先阅读附录 1 的第一段：初识C++)。

第二步：将光盘“VC 用户”目录下的 Matrix. h 和 MatrixVCDLL. lib 复制到 TestMatrix 目录下，并将它们插入项目(注意，文件位于 D:\TestMatrix 下并不意味着插入)。

第三步：编辑源程序。

```
#include "math.h"
#include "matrix.h"              //使用了数值分析库，必须包含此头文件
double f(double t,double x)      //函数定义，它是微分方程右边那个二元函数
{
      return sin(t * x) + t * cos(sqrt(t) * x) + t;
}
void main(void)
{
    matrix A;                    //定义一个矩阵，用来存放微分方程的数值解
    A.RK(f,0,0,5,1E-3);          //用龙格-库塔法解微分方程
    A. Print( );                 //在屏幕上显示微分方程数值解
    A.Save("D:\\diff.dat");      //将微分方程数值解存储到 D 盘上
    A.Plot( );                   //绘制解曲线
}
```

第四步：按 F7 键编译链接，按 Ctrl+F5 键运行程序，得到结果如图 0-1 所示。

图 0-1 的左半部分是 $x(t)$的数值解，其中第一列是 t 的节点值，第二列是相应的 $x(t)$的值，因为读者没法拖动书上的滚动条，所以下面的数据看不到。图 0-1 的右半部分是解曲线，窗口中间的水平线是 t 轴，左端点对应 $t=0$，右端点对应 $t=5$。

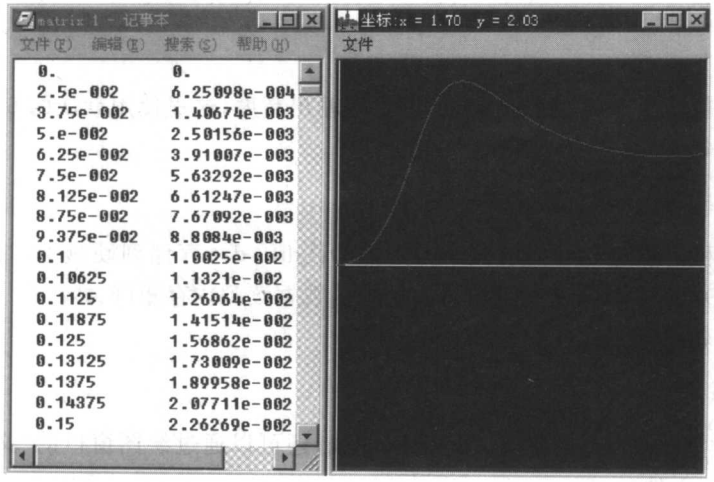


图 0-1

当鼠标进入绘图窗口，窗口标题条上会指示鼠标当前所在位置的坐标，例如，当把鼠标指向曲线的最高点位置，绘图窗口的标题条上会显示“坐标：x=1.70 y=2.03”，这说明微分方程的解曲线在区间[0,5]中的最大值 $y=2.03$，在 $x=1.70$ 处取到该最大值。当然，这里的 x，y 值都是近似值。

数值分析库用了五条语句完成所有的任务。下面解释一下这五条语句。

第一条语句

```
matrix A;
```

定义了一个矩阵 $\boldsymbol{A}$，用来存放 $x(t)$ 的数值解，矩阵的行列数不用用户设置。实际上，$\boldsymbol{A}$ 是两列多行矩阵，$\boldsymbol{A}$ 的第一列存放了 t 的节点值，第二列存放了相应的 $x(t)$ 值。节点的位置和个数取决于求解的微分方程和指定的允许误差。因为数值分析库使用的是变步长龙格-库塔法，它在曲线变化急剧的区间段用小步长，在曲线变化平缓的区间段用大步长，这样能够控制计算误差(如果你需要在指定点的解值，再使用数值库提供的样条插值功能)。

第二条语句

```
A.RK(f,0,0,5,1E-3);
```

使用变步长龙格-库塔法求解给定的微分方程，并将计算结果存放在 $\boldsymbol{A}$ 中。RK(…)中的第一个参数 f 就是微分方程右边的那个二元函数名；第二个参数对应初始条件 $t=0$；第三个参数对应初始条件 $x(0)=0$；第四个参数对应解区间的右端点 $T=5$；第五个参数保证数值解与真实解在节点上的最大相对误差不超过 1E−3。

第三条语句

```
A.Print( );
```

用记事簿窗口打印出 $x(t)$ 的数值解，可以指定输出精度，这里使用默认的 6 位精度。

第四条语句

```
A.Save("D:\\diff.dat");
```

将 t 的节点值和相应的 $x(t)$ 的解值作为文件 D:\diff.dat 存储到硬盘上。以后只要用语句 A.Read("D:\\diff.dat")就可以将文件中的数据读到内存矩阵 $\boldsymbol{A}$ 中。

第五条语句

```
A.Plot( );
```

根据 $\boldsymbol{A}$ 中的数据$(t,x(t))$绘制出 $x(t)$ 的解曲线，还可以通过绘图窗口的“文件”菜单将它保存为 *.bmp 文件。

目 录

前言 …… I

如何使用C++数值分析类库 …… V

第1章 矩阵的操作 …… 1

1.1 矩阵的定义、元素访问、重置、销毁与显示 …… 1

1.1.1 动态矩阵及元素的访问 …… 1

1.1.2 关于矩阵的作用域 …… 2

1.1.3 空矩阵与矩阵尺寸的重新设置 …… 3

1.1.4 矩阵的提前销毁 …… 3

1.1.5 矩阵的显示 …… 4

1.2 矩阵的整体赋值与均匀分布随机矩阵 …… 5

1.2.1 使用初始化函数 …… 5

1.2.2 用另一矩阵初始化或整体赋值 …… 6

1.2.3 初始化(或调整)为单位矩阵 …… 7

1.2.4 让矩阵元素服从$[a,b]$区间上的均匀分布 …… 8

1.2.5 产生整数值随机矩阵 …… 9

1.3 矩阵的加法、减法、乘法、转置、反号和置零 …… 9

1.3.1 矩阵相加 …… 9

1.3.2 矩阵的累加 …… 11

1.3.3 矩阵相减 …… 12

1.3.4 矩阵的累减 …… 12

1.3.5 矩阵相乘 …… 13

1.3.6 矩阵的累乘 …… 14

1.3.7 矩阵乘以标量 …… 15

1.3.8 矩阵累乘标量 …… 16

1.3.9 矩阵倍加另一个矩阵 …… 17

1.3.10 矩阵的转置 …… 18

1.3.11 矩阵反号 …… 19
1.3.12 矩阵置零 …… 19
1.3.13 去除矩阵的垃圾元素 …… 19
1.4 矩阵的初等变换 …… 20
1.4.1 矩阵交换两行 …… 20
1.4.2 矩阵交换两列 …… 21
1.4.3 矩阵倍乘一行 …… 22
1.4.4 矩阵倍乘一列 …… 23
1.4.5 矩阵行倍加 …… 23
1.4.6 矩阵列倍加 …… 24
1.5 矩阵行、列的添加、插入和删除 …… 25
1.5.1 矩阵添加一零行 …… 25
1.5.2 矩阵添加一零列 …… 25
1.5.3 矩阵插入一零行 …… 26
1.5.4 矩阵插入一零列 …… 27
1.5.5 矩阵删除一行 …… 28
1.5.6 矩阵删除一列 …… 28
1.5.7 获取矩阵的行、列数 …… 29
1.6 矩阵取子块与矩阵拼接 …… 29
1.6.1 取矩阵的任意子块 …… 29
1.6.2 取矩阵的四角块 …… 30
1.6.3 取矩阵的连续若干行 …… 32
1.6.4 取矩阵的连续若干列 …… 33
1.6.5 矩阵的填补(1) …… 34
1.6.6 矩阵的填补(2) …… 35
1.6.7 矩阵的横向拼接(1) …… 36
1.6.8 矩阵的横向拼接(2) …… 37
1.6.9 矩阵的竖向拼接(1) …… 38
1.6.10 矩阵的竖向拼接(2) …… 39
1.7 矩阵的存盘与读取 …… 40
1.7.1 矩阵存储为磁盘文件 …… 40
1.7.2 读取磁盘文件矩阵 …… 41
1.8 矩阵与C/C++ 数组交换数据 …… 44
1.8.1 矩阵串行为C/C++ 数组 …… 44

1.8.2 C/C++ 数组排列成矩阵 …… 45
1.9 其他 …… 46
1.9.1 方阵的对角线加常量 …… 46
1.9.2 矩阵的所有元素加常量 …… 47
1.9.3 方阵的迹 …… 47
1.9.4 矩阵元素的平均值 …… 48
1.9.5 由一个矩阵产生的协方差矩阵 …… 49
1.9.6 矩阵的绝对值最大元素及定位 …… 50
1.9.7 矩阵的最大元素及定位 …… 51
1.9.8 矩阵的绝对值最小元素及定位 …… 52
1.9.9 矩阵的最小元素及定位 …… 53

第 2 章 向量的操作 …… 55

2.1 向量的定义、元素访问、重置、销毁与显示 …… 55
2.1.1 向量的定义、元素访问及作用域 …… 55
2.1.2 空向量与向量长度的重置 …… 56
2.1.3 向量的提前销毁 …… 56
2.1.4 向量的显示 …… 57
2.2 向量的整体赋值与随机向量 …… 58
2.2.1 使用初始化函数 …… 58
2.2.2 用另一向量初始化或整体赋值 …… 59
2.2.3 将向量初始化或设置为单位向量 …… 59
2.2.4 使向量的所有元素都相同 …… 60
2.2.5 使向量的元素为区间的等分点 …… 61
2.2.6 一元函数在若干坐标点上的值构成的向量 …… 61
2.2.7 $[a,b]$上均匀分布的随机数构成的向量 …… 63
2.2.8 $[-|N|,|N|]$范围内的随机整数值构成的向量 …… 63
2.2.9 服从正态分布的随机数构成的向量 …… 64
2.2.10 服从 Γ 分布的随机数构成的向量 …… 64
2.2.11 服从 β 分布的随机数构成的向量 …… 65
2.2.12 向量数据的频率 …… 66
2.3 向量的加、减、乘运算及置零 …… 66
2.3.1 向量相加 …… 66
2.3.2 向量的累加 …… 67

2.3.3 向量相减 …… 68
2.3.4 向量的累减 …… 69
2.3.5 向量的内积 …… 69
2.3.6 向量乘以标量 …… 69
2.3.7 向量累乘标量 …… 70
2.3.8 向量每个元素加上同一标量 …… 71
2.3.9 向量倍加另一向量 …… 72
2.3.10 向量置零 …… 73
2.3.11 两向量的欧氏距离 …… 73
2.3.12 去除向量的垃圾元素 …… 73
2.4 向量元素的添加、插入和删除 …… 74
2.4.1 向量添加一元素 …… 74
2.4.2 向量插入一元素 …… 75
2.4.3 向量删除一元素 …… 76
2.4.4 获取向量的维数 …… 77
2.5 向量的拼接、截取和填补 …… 77
2.5.1 向量的拼接(1) …… 77
2.5.2 向量的拼接(2) …… 78
2.5.3 截取向量的左段 …… 79
2.5.4 截取向量的中段 …… 80
2.5.5 截取向量的右段 …… 80
2.5.6 向量的填补 …… 81
2.6 向量的存盘与读取 …… 82
2.6.1 向量存储为磁盘文件 …… 82
2.6.2 读取磁盘文件向量 …… 84
2.7 向量与C/C++ 数组交换数据 …… 86
2.7.1 向量转换为C/C++ 数组 …… 86
2.7.2 C/C++ 数组转换成向量 …… 87
2.8 其他 …… 88
2.8.1 向量元素的均值 …… 88
2.8.2 向量元素的方差 …… 88
2.8.3 向量的绝对值最大元素及定位 …… 88
2.8.4 向量的绝对值最小元素及定位 …… 89
2.8.5 向量的最大元素及定位 …… 90

2.8.6　向量的最小元素及定位 …… 91
2.8.7　向量元素按升序排列 …… 91
2.8.8　向量元素按降序排列 …… 92
2.8.9　一个实数的区间定位 …… 93
2.8.10　计算 n 次二项展开式的系数 …… 94
2.8.11　向量的逆转 …… 95
2.8.12　向量的移位 …… 96

第 3 章　矩阵与向量的关联操作 …… 98

3.1　矩阵添加和插入指定的行、列 …… 98
3.1.1　矩阵添加指定行 …… 98
3.1.2　矩阵添加指定列 …… 99
3.1.3　矩阵插入指定行 …… 99
3.1.4　矩阵插入指定列 …… 100
3.2　矩阵行、列的设置与提取 …… 101
3.2.1　替换矩阵的一行 …… 101
3.2.2　替换矩阵的一列 …… 102
3.2.3　提取矩阵的一行 …… 103
3.2.4　提取矩阵的一列 …… 103
3.3　矩阵与向量相乘 …… 104
3.3.1　列向量右乘矩阵 …… 104
3.3.2　行向量左乘矩阵 …… 105
3.3.3　行、列向量同时左右乘矩阵 …… 106
3.3.4　两向量相乘产生矩阵 …… 107
3.4　其他 …… 108
3.4.1　产生一系列多维正态随机向量 …… 108
3.4.2　提取方阵的对角线构成向量 …… 109
3.4.3　设置方阵的对角线 …… 110
3.4.4　方阵的对角线加向量 …… 111
3.4.5　方阵的对角线减向量 …… 111
3.4.6　矩阵的各行累加构成向量 …… 112
3.4.7　矩阵的各列累加构成向量 …… 112
3.4.8　矩阵的元素串行成向量 …… 113
3.4.9　向量排列成矩阵 …… 113

第 4 章 矩阵的数值分析 …… 115

4.1 矩阵的行列式、秩、值空间和核空间、范数及条件数 …… 115
4.1.1 方阵的行列式 …… 115
4.1.2 矩阵的秩 …… 116
4.1.3 矩阵的值空间 …… 117
4.1.4 矩阵的核空间 …… 118
4.1.5 矩阵的 1-范数 …… 119
4.1.6 矩阵的∞-范数 …… 119
4.1.7 矩阵的 2-范数 …… 119
4.1.8 矩阵的条件数 …… 120
4.2 矩阵分解 …… 121
4.2.1 对称正定矩阵的楚列斯基分解 …… 121
4.2.2 一般对称矩阵的强迫正定楚列斯基分解 …… 123
4.2.3 "高型"矩阵的 QR 分解 …… 125
4.2.4 任意矩阵的奇异值分解 …… 127
4.3 矩阵的特征值和特征向量 …… 129
4.3.1 对称矩阵的所有特征值及特征向量 …… 129
4.3.2 一般方阵的所有特征值(包括复特征值) …… 131
4.3.3 指定方阵的一个实特征值,求相应的一个实特征向量 …… 133
4.3.4 指定方阵的一个复特征值,求相应的一个复特征向量 …… 134
4.4 矩阵的逆与伪逆(广义逆) …… 135
4.4.1 矩阵求逆或者判断不可逆 …… 135
4.4.2 矩阵的伪逆(广义逆) …… 137
4.5 解线性方程组 …… 138
4.5.1 系数矩阵为三对角矩阵(追赶法) …… 138
4.5.2 系数矩阵为对称正定矩阵(平方根法) …… 140
4.5.3 系数矩阵为一般的非奇异矩阵(高斯法,高斯-塞德尔迭代法) …… 141
4.5.4 系数矩阵非方阵的最小二乘解或最小范数解 …… 143
4.5.5 系数矩阵为任意矩阵的广义解 …… 146

第 5 章 函数的数值分析 …… 149

5.1 一元函数的基本问题 …… 149
5.1.1 绘制一元函数曲线 …… 149

5.1.2　一元函数的一阶导数、二阶导数 …… 152
5.1.3　一元函数的零点 …… 153
5.1.4　一元函数的定积分(龙贝格法,高斯法) …… 155
5.1.5　一元函数的含参变量积分(带单参数) …… 157
5.1.6　一元函数的含参变量积分(带多参数) …… 160
5.1.7　一元函数的局部极小点 …… 162
5.2　一元实系数多项式 …… 164
5.2.1　多项式的表示与求值、求导、求积分 …… 164
5.2.2　多项式的加法、减法和乘法 …… 166
5.2.3　多项式的除法 …… 167
5.2.4　多项式的所有根(包括复根) …… 168
5.2.5　已知多项式的所有根(包括复根),求多项式系数 …… 169
5.2.6　多项式在闭区间上的最大值点和最小值点 …… 170
5.2.7　矩阵多项式 …… 171
5.2.8　四种著名的正交多项式 …… 173
5.3　样条插值、离散数据的求导和求积分 …… 175
5.3.1　样条函数与插值 …… 175
5.3.2　离散数据求导 …… 178
5.3.3　离散数据求积分 …… 180
5.4　函数逼近、离散数据最小二乘拟合、快速傅里叶变换和向量卷积 …… 182
5.4.1　连续函数的多项式最佳平方逼近 …… 182
5.4.2　连续函数的多项式最佳一致逼近 …… 185
5.4.3　离散数据的多项式最小二乘拟合 …… 187
5.4.4　快速傅里叶变换 …… 190
5.4.5　两个向量的线性卷积 …… 192
5.5　常微分方程和方程组、线性定常系统及二阶线性边值问题 …… 194
5.5.1　一阶常微分方程 …… 194
5.5.2　一阶常微分方程组 …… 196
5.5.3　线性定常系统 …… 200
5.5.4　二阶线性微分方程的边值问题 …… 202
5.6　多元函数的梯度、二阶导数矩阵及雅可比矩阵 …… 205
5.6.1　多元函数在指定点的梯度 …… 205
5.6.2　多元函数在指定点的二阶导数矩阵 …… 206
5.6.3　一组多元函数在指定点的雅可比矩阵 …… 208