



数值与非数值分析

VC++类库

粟塔山 编著



科学出版社

国防科技大学研究生数学公共课程系列教材

数值与非数值分析 VC++类库

粟塔山 编著

科学出版社

北京

内 容 简 介

本书是《VC++和BC++数值分析类库》的增补版。《VC++和BC++数值分析类库》包括矩阵、向量的操作运算和数值分析各种算法,读者几乎可以随心所欲地操作处理矩阵和向量,功能比MATLAB更丰富;数值分析功能涵盖了该学科的各分支。本书除增补了矩阵向量操作和数值分析功能外,还增加了6项功能:字符串数学表达式解析;数据结构(链表、堆栈、队列)模板;信号基本分析工具箱;大整数、分数、分数矩阵、向量运算;复数、复数矩阵、向量运算;网络图操作与优化。本书提供了动态库和静态库,静态库使得用户能编译生成完全独立的应用程序。

本书读者对象为理工类本科生、研究生,尤其是习惯使用VC++编程环境的学生。该类库与VC++环境无缝连接,如果在你们的VC++应用程序中含有大量的数值分析和计算,使用该类库是不错的选择。

图书在版编目(CIP)数据

数值与非数值分析,VC++类库/桑培培编著. —北京:科学出版社,2015.5

国防科技大学研究生数学公共课教材

ISBN 978-7-03-041275-5

I. ①数… II. ②桑… III. ①数值分析—程序设计—研究生—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第109285号

责任编辑:李静科 赵彦超 / 责任校对:邹慧卿

责任印制:徐晓晨 / 封面设计:耕者工作室

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

北京京华虎彩印刷有限公司印刷

科学出版社总发行 各地新华书店经销

*

2015年6月第 一 版 开本:720×1000 B5

2015年6月第一次印刷 印张:25 3/4

字数:514 000

定价:98.00元(含光盘)

(如有印装质量问题,我社负责调换)

前 言

本书是《VC++和BC++数值分析类库》(清华大学出版社)的增补版,并改名为《数值与非数值分析 VC++类库》.新增内容包括两方面,其一是在原有的matrix类和vector类上新增的功能,作为第6章;其二是新增了如下的类,作为第7章.

- (1) 字符串数学表达式解析类
- (2) 数据结构(链表、堆栈、队列)模板类
- (3) 信号基本分析工具箱
- (4) 大整数、分数、分数矩阵、分数向量类
- (5) 复数、复数矩阵、复数向量基本运算类
- (6) 网络图类

本书所附光盘中包含本书所有示例的C++源程序.另外,本书不但提供了动态链接库,也提供了静态链接库,后者使得用户能制作完全独立的应用程序(但静态库没有曲线绘制功能).

matrix、vector类新增功能简介如下:

- (1) 多元函数全局最优计算
- (2) 一元函数全局最优计算
- (3) 共轭投影梯度法
- (4) 变量带上界的线性规划
- (5) 混合整数线性规划
- (6) 0-1线性规划
- (7) 运输问题
- (8) 指派问题
- (9) 离散余弦变换
 - ① II型离散余弦变换
 - ② IV型离散余弦变换
- (10) 函数的傅里叶系数与傅里叶展开
- (11) 阿克玛插值法
- (12) 张成子空间上的投影矩阵
- (13) 方阵的Doolittle分解
- (14) cuboid(长方体)数据类型及操作

- (15) 新增两个离散数据曲线绘制功能
- (16) 无约束最优化算法梯度表达式输入接口
- (17) 一元函数求导步长修改为可调参数
- (18) Gauss 法积分无须用户切割区间
- (19) 线性规划计算可获得最优基逆、基指标、非基指标
- (20) 傅里叶变换定义的微调
- (21) 向量、矩阵的基本操作
 - ① 向量的循环移位、循环矩阵
 - ② 向量的循环卷积
 - ③ 抽取向量的奇项、偶项
 - ④ 向量的交错插入
 - ⑤ 步进方式产生向量
 - ⑥ 从键盘读入矩阵、向量
 - ⑦ 矩阵的对角连接
 - ⑧ n 元函数在矩阵上的函数值向量
 - ⑨ 二元函数在平面格点上的函数值矩阵
 - ⑩ 矩阵的行移动、列移动
 - ⑪ 矩阵的行翻转、列翻转
 - ⑫ 矩阵元素设置为常数
 - ⑬ 矩阵对角设置功能增强

关于新增的类：

(1) 字符串数学表达式解析类

给出一个符合一定法则的字符串数学表达式，数值库能动态生成一个计算表达式并求其值。对一元或多元函数表达式，还能计算它在指定点的一阶导数值、二阶导数值、梯度向量和二阶导数矩阵。字符串数学表达式由运算符 $+$ 、 $-$ 、 $*$ 、 $/$ 、 $($)、 \min 、 \max 以及基本初等函数的任意复合构成。当字符串表达式不符合数学语法时能报告错误。

(2) 数据结构(链表、堆栈、队列)模板类

链表、堆栈、队列是三种常用的数据结构。本类库的链表、堆栈、队列的数据类型可以是用户自定义的任意 C++ 合法类型。链表是双向链表，队列是循环队列，堆栈、队列无容量限制。

(3) 信号基本分析工具箱

信号分析的两个核心是信号频谱与信号滤波。利用该工具箱读者可以轻松便利地做数值实验，绘制信号的频谱、相谱、频相谱，并搭建完美重构滤波器组，观察滤波

过程每一个环节信号频谱的变化. 这样的直观感受特别有益于概念、结论的理解和认知.

(4) 大整数、分数、分数矩阵、分数向量类

计算机中能使用的最大正整数为 $2^{32} - 1$ (32 位系统), 但是某些问题涉及的整数可能远远超过此范围. 本类库的大整数类可以生成“足够大”的整数, 并进行大整数的四则运算和逻辑判断.

以大整数为基础构造了分数类, 并进行分数的四则运算和逻辑判断.

以分数类为基础构造了分数矩阵、分数向量类, 并提供了与 `matrix` 和 `vector` 的基本操作相平行的功能.

(5) 复数、复数矩阵、复数向量基本运算类

矩阵、向量的元素可以为复数, 并对复数矩阵、向量提供了与 `matrix` 和 `vector` 的基本操作相平行的功能.

(6) 网络图类

根据网络图的邻接矩阵生成网络图, 能将网络图保存为磁盘文件并读取文件. 网络可以是有向网络或无向网络, 顶点可以带权. 可以任意添加顶点、删除顶点、添加边、删除边、修改边权、顶点权. 可以求网络图的连通子图、任意顶点的可达集和最短路程、网络图的最小支撑树、任意两点之间的最短路径. 读者不难进一步扩展其他功能.

为使读者对本书有更全面的了解, 下面附上《VC++ 和 BC++ 数值分析类库》的前言节选.

数值计算是计算机应用永恒的话题. 众多的工程技术人员、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. 可以想象一下, 有多少人不止一次地用三重循环编制两矩阵相乘的程序段. 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, 就做不出独立的应用软件, 例如, 把算法嵌入硬件 (MATLAB5.3 数学库高达 12M). 再说, 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 阶矩阵的线性方程组),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,你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人手续.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,你可以进一步对该数值分析类库作二次开发,比较轻松地编出所需要的数学算法.

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)做了详尽的解释,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.你无须熟悉 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 但没有使用过 C++ 编程环境的读者.对这部分读者而言,只要读过了这段附录 1,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 列出了为正确高效地使用数值库应该注意的一些事项.当你使用数值库的某个函数出现疑惑时,首先应仔细阅读该函数的使用说明,或者,也许能在附录 2 中找到问题的答案.

本书以菜谱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,可大致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,再翻到该页大致了解一下(以后需要时再仔细阅读使用示例).然后,再到计算机上熟悉一下第 1, 2, 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减乘运算.如果在应用程序中要对矩阵和向量作某项操作,应该先查阅一下数值库是否提供了你要的功能(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,矩阵和向量是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。再则,当向量保持二级指针结构,那么“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凸现出来,在内存中是无须区分的。举个例子,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 $y = x^T A$ 产生 n 维“行向量” y ;数学表达式 $a = x^T A x$ 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

$$y = A * x$$

说明你现在是把 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

$$y = x * A$$

说明你现在是把 x 当成“行向量”进行运算,它相当于数学表达式 $y = x^T A$;程序表达式

$$a = x * A * x$$

说明你既可以把 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 $a = x^T A x$ 。由此可见,对于数值库中的“向量”,你想让它是“行”,它就是“行”,你想让它是“列”,它就是“列”。你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,你既然用 VC++ 或 BC++ 编程,那么你的操作系统应该是 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。

本书的工作得到了国防科技大学理学院研究生创新教育系列教材出版基金的资助,在此表示感谢!

如果读者在使用数值库时出现了无法解释的结果,或者对数值库有其他的要求和设想,欢迎来信告诉我 nudtsts@163.com,以便作者对数值分析库进一步修正和补充,让它更好地为读者服务。

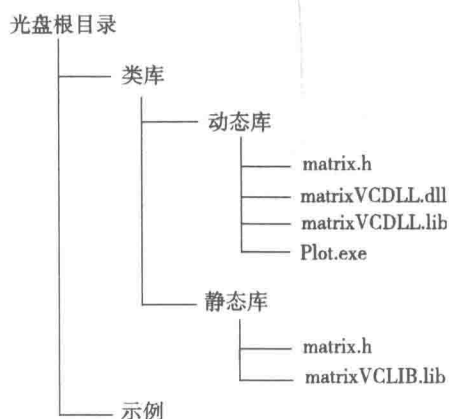
作 者

2015 年 3 月

于长沙国防科技大学

如何使用本书 VC++ 类库

首先,要把光盘上的一些文件复制到你的计算机中,光盘上的内容如下:



动态库与静态库各有特点. 编译链接时,动态库代码不会嵌入你的程序,而是在程序运行时才把调用的函数代码调入内存,如果你的程序在另一台机器上运行,必须伴随动态库 `matrixVCDLL.dll`. 而静态库会嵌入你的程序,这使得你的程序完全独立,但也可能使你的程序变得臃肿. 欲使用本类库的曲线绘制功能或者示例程序,必须使用动态库.

(1) 动态库使用方法:将类库/动态库目录下的 `matrixVCDLL.dll` 和 `Plot.exe` 文件复制到 Windows 目录下(或者 `Windows\System32` 目录下). 将类库/动态库目录下的 `matrix.h` 和 `matrixVCDLL.lib` 复制到你的项目(project)下,并将 `matrixVCDLL.lib` 插入项目.“插入”菜单项如图:



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` 称为入口库文件,它是应用

程序与动态链接库 matrixVCDLL.dll 的中介。编译链接完成后,应用程序就不需要它了。应用程序 Plot.exe 为绘制曲线提供支持,它被 matrixVCDLL.dll 中的某些函数调用,不能单独运行。如果你的应用程序中不需要绘制曲线,可以不要它。

(2) 静态库使用方法:将类库/静态库目录下的 matrixVCLIB.lib 和 matrix.h 复制到你的项目下,并将 matrixVCLIB.lib 插入项目。

(3) 示例使用方法(假设已复制 matrixVCDLL.dll 和 Plot.exe):将示例子目录复制到硬盘,打开示例/示例工场目录,启动项目(示例工场.dsw)。将某示例 cpp 文件插入项目(不需要复制),按 F7 编译链接,按 Ctrl+F5 运行程序。如果要换另一个示例,先点击 FileView 中的示例 cpp 文件,按 Del 键删除该文件,并关闭该文件。再插入另一个示例 cpp 文件。

下面举一个例子,说明如何使用这个 VC++ 数值库。假设我们需要在区间 $[0, 5]$ 上求解一阶微分方程初值问题:

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

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 Visual C++)。

第二步:将类库/动态库目录下的 matrix.h 和 matrixVCDLL.lib 复制到 TestMatrix 目录下,并将 matrixVCDLL.lib 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:

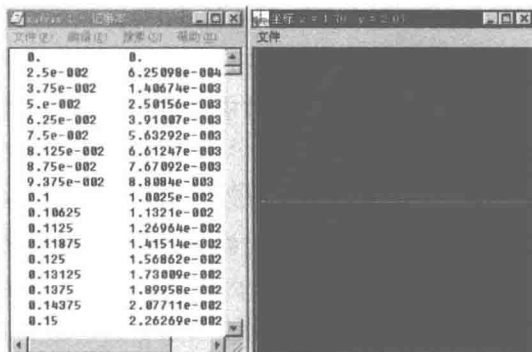
```
# include "math.h"
# include "matrix.h" //使用了数值分析库,必须包含此头文件
double f(double t,double x) //函数定义,它是微分方程右边那个二元函数
{
    return sin(t*x)+ t*cos(sqrt(t)*x)+ t;
}
void main(void)
{
    matrix A; //定义一个矩阵,用来存放微分方程的数值解
    A.RK(f,0,0,5,1E-3); //用龙格库塔法解微分方程
    l.Print(); //在屏幕上显示微分方程数值解
```

```

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 编译链接,按 Ctrl+F5 运行程序,得到结果



上图的左半部分是 $x(t)$ 的数值解,其中第一列是 t 的节点值,第二列是相应的 $x(t)$ 值,因为没法拖动书上的滚动条,所以下面的数据看不到.上图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 $t=5$.

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 A ,用来存放 $x(t)$ 的数值解.实际上, A 是两列多行矩阵. 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 A 中. $RK(\dots)$ 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( );
```

用记事本窗口打印出 $x(t)$ 的数值解, 可以指定输出精度, 这里使用缺省的 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 A 中.

第五条语句

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

根据 A 中的 $(t, x(t))$ 数据绘制出 $x(t)$ 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(16 色位图, 需要用户自己加扩展名. bmp).

目 录

前言

如何使用本书 VC++ 类库

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 $[a, b]$ 区间上均匀分布	7
1.2.5 产生整数值随机矩阵	8
1.3 矩阵的加、减、乘、转置、反号、置零.....	9
1.3.1 矩阵相加	9
1.3.2 矩阵的累加.....	10
1.3.3 矩阵相减.....	11
1.3.4 矩阵的累减.....	12
1.3.5 矩阵相乘.....	12
1.3.6 矩阵的累乘.....	14
1.3.7 矩阵乘标量.....	14
1.3.8 矩阵累乘标量.....	15
1.3.9 矩阵倍加另一矩阵.....	16
1.3.10 矩阵的转置	17
1.3.11 矩阵反号	18
1.3.12 矩阵置零	18

1.3.13	去除矩阵的垃圾元素	18
1.4	矩阵的初等变换	19
1.4.1	矩阵交换两行	19
1.4.2	矩阵交换两列	20
1.4.3	矩阵倍乘一行	20
1.4.4	矩阵倍乘一列	21
1.4.5	矩阵行倍加	22
1.4.6	矩阵列倍加	22
1.5	矩阵行、列的添加、插入与删除	23
1.5.1	矩阵添加一零行	23
1.5.2	矩阵添加一零列	24
1.5.3	矩阵插入一零行	25
1.5.4	矩阵插入一零列	25
1.5.5	矩阵删除一行	26
1.5.6	矩阵删除一列	27
1.5.7	获取矩阵的行、列数	27
1.6	矩阵取子块与矩阵拼接	27
1.6.1	取矩阵的任意子块	27
1.6.2	取矩阵的四角块	29
1.6.3	取矩阵的连续若干行	30
1.6.4	取矩阵的连续若干列	31
1.6.5	矩阵的填补(一)	32
1.6.6	矩阵的填补(二)	33
1.6.7	矩阵的横向拼接(一)	34
1.6.8	矩阵的横向拼接(二)	35
1.6.9	矩阵的竖向拼接(一)	35
1.6.10	矩阵的竖向拼接(二)	36
1.7	矩阵的存盘与读取	37
1.7.1	矩阵存储为磁盘文件	37
1.7.2	读取磁盘文件矩阵	38
1.8	矩阵与 C/C++ 数组交换数据	41
1.8.1	矩阵串行为 C/C++ 数组	41
1.8.2	C/C++ 数组排列成矩阵	41

1.9 其他	42
1.9.1 方阵的对角线加常量	42
1.9.2 矩阵的所有元素加常量	43
1.9.3 方阵的迹	44
1.9.4 矩阵元素的平均值	44
1.9.5 由一个矩阵产生的协方差矩阵	45
1.9.6 矩阵的绝对值最大元素及定位	46
1.9.7 矩阵的最大元素及定位	47
1.9.8 矩阵的绝对值最小元素及定位	48
1.9.9 矩阵的最小元素及定位	49
第2章 向量的操作	50
2.1 向量的定义、元素访问、重置、销毁与显示	50
2.1.1 向量的定义、元素访问及作用域	50
2.1.2 空向量与向量长度的重置	51
2.1.3 向量的提前销毁	51
2.1.4 向量的显示	52
2.2 向量的整体赋值与随机向量	53
2.2.1 使用初始化函数	53
2.2.2 用另一向量初始化或整体赋值	54
2.2.3 将向量初始化或设置为单位向量	55
2.2.4 使向量的所有元素都相同	55
2.2.5 使向量的元素为区间的等分点	56
2.2.6 一元函数在若干坐标点上的值构成向量	57
2.2.7 $[a, b]$ 均匀分布的随机数构成向量	58
2.2.8 $[- N , N]$ 范围内的随机整数值构成向量	59
2.2.9 服从正态分布的随机数构成向量	60
2.2.10 服从伽马分布的随机数构成向量	61
2.2.11 服从贝塔分布的随机数构成向量	61
2.2.12 向量数据的频率	62
2.3 向量的加、减、乘运算及置零	63
2.3.1 向量相加	63
2.3.2 向量的累加	64
2.3.3 向量相减	65

2.3.4	向量的累减	65
2.3.5	向量的内积	66
2.3.6	向量乘标量	66
2.3.7	向量累乘标量	67
2.3.8	向量每个元素加上同一标量	68
2.3.9	向量倍加另一向量	69
2.3.10	向量置零	69
2.3.11	两向量的欧氏距离	70
2.3.12	去除向量的垃圾元素	70
2.4	向量元素的添加、插入、删除	71
2.4.1	向量添加一元素	71
2.4.2	向量插入一元素	71
2.4.3	向量删除一元素	73
2.4.4	获取向量的维数	73
2.5	向量的拼接、截取和填补	74
2.5.1	向量的拼接(一)	74
2.5.2	向量的拼接(二)	75
2.5.3	截取向量的左段	75
2.5.4	截取向量的中段	76
2.5.5	截取向量的右段	77
2.5.6	向量的填补	78
2.6	向量的存盘与读取	79
2.6.1	向量存储为磁盘文件	79
2.6.2	读取磁盘文件向量	80
2.7	向量与 C/C++ 数组交换数据	83
2.7.1	向量转换为 C/C++ 数组	83
2.7.2	C/C++ 数组转换成向量	84
2.8	其他	85
2.8.1	向量元素的均值	85
2.8.2	向量元素的方差	85
2.8.3	向量的绝对值最大元素及定位	86
2.8.4	向量的绝对值最小元素及定位	86
2.8.5	向量的最大元素及定位	87

2.8.6	向量的最小元素及定位	88
2.8.7	向量元素按升序排列	89
2.8.8	向量元素按降序排列	89
2.8.9	一个实数的区间定位	90
2.8.10	计算 n 次二项展开式的系数	91
2.8.11	向量的逆转	92
2.8.12	向量的移位	93
第3章	矩阵与向量的关联操作	95
3.1	矩阵添加和插入指定的行列	95
3.1.1	矩阵添加指定行	95
3.1.2	矩阵添加指定列	95
3.1.3	矩阵插入指定行	96
3.1.4	矩阵插入指定列	97
3.2	矩阵行列的设置与提取	98
3.2.1	替换矩阵的一行	98
3.2.2	替换矩阵的一列	98
3.2.3	提取矩阵的一行	99
3.2.4	提取矩阵的一列	100
3.3	矩阵与向量相乘	101
3.3.1	列向量右乘矩阵	101
3.3.2	行向量左乘矩阵	101
3.3.3	行、列向量同时左右乘矩阵	102
3.3.4	两向量相乘产生矩阵	103
3.4	其他	104
3.4.1	产生一系列多维正态随机向量	104
3.4.2	提取方阵的对角线构成向量	105
3.4.3	设置方阵的对角线	106
3.4.4	方阵的对角线加向量	107
3.4.5	方阵的对角线减向量	107
3.4.6	矩阵的各行累加构成向量	108
3.4.7	矩阵的各列累加构成向量	108
3.4.8	矩阵的元素串行成向量	109
3.4.9	向量排列成矩阵	109