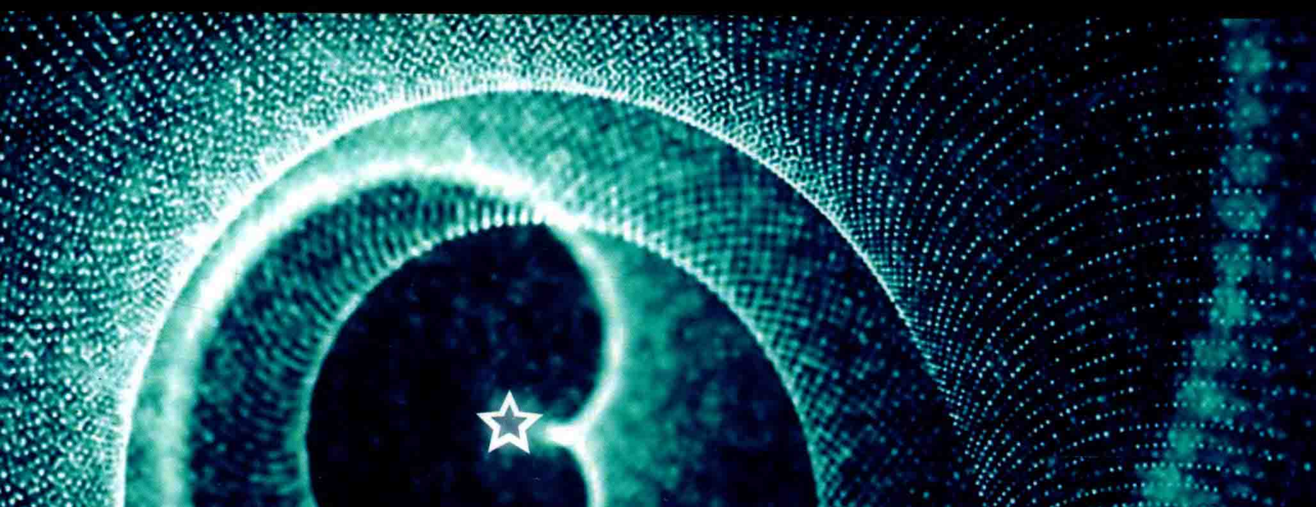
多次修订，精益求精，算法新；方便实用，数值计算的好伴侣

常用算法程序集

（C++描述）（第6版）

徐士良◎编著

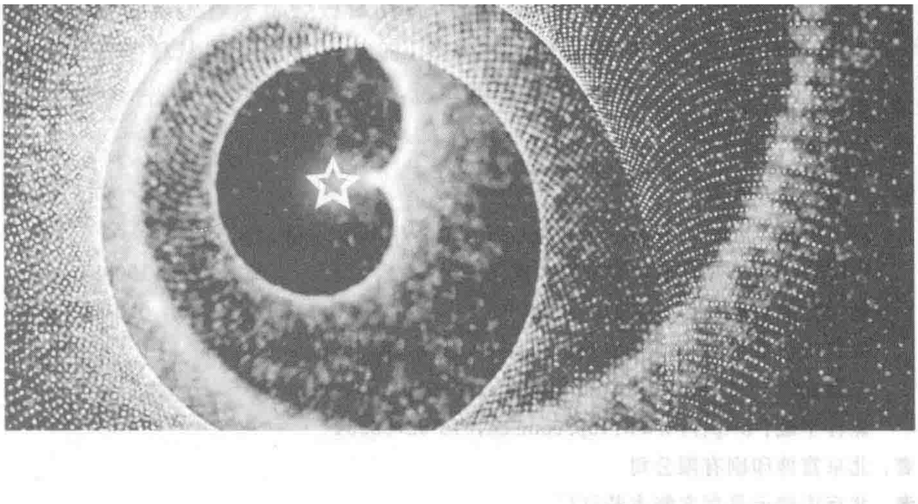


清華大學出版社

常用算法程序集

（C++描述）（第6版）

徐士良◎编著



清華大學出版社

北京

内 容 简 介

本书是针对工程中常用的行之有效的算法而编写的，其主要内容包括封装的四个基本运算类（复数运算类、实系数与复系数多项式运算类以及产生随机数类），矩阵运算，矩阵特征值与特征向量的计算，线性代数方程组，非线性方程与方程组的求解，插值与逼近，数值积分，常微分方程组的求解，数据处理，极值问题的求解，数学变换与滤波，特殊函数的计算，排序等。

书中所有的算法程序均用 C++ 描述，源代码可从清华大学出版社网站(www. tup. com. cn)下载。

本书可供广大科研人员、工程技术人员及管理工作者阅读使用，也可作为高等院校师生的参考书。

图书在版编目(CIP)数据

常用算法程序集：C++ 描述/徐士良编著. —6 版. —北京：清华大学出版社，2019
ISBN 978-7-302-50542-6

Ⅰ. ①常…　Ⅱ. ①徐…　Ⅲ. ①工程计算程序－程序设计 ②C 语言－程序设计　Ⅳ. ①TP319 ②TP312.8

中国版本图书馆 CIP 数据核字(2018)第 141944 号

责任编辑：白立军　张爱华
封面设计：杨玉兰
责任校对：梁　毅
责任印制：杨　艳

出版发行：清华大学出版社
网　　址：http://www. tup. com. cn，http://www. wqbook. com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup. tsinghua. edu. cn
质量反馈：010-62772015，zhiliang@tup. tsinghua. edu. cn
课件下载：http://www. tup. com. cn，010-62795954
印 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：35.5　　**字　　数**：838 千字
版　　次：1989 年 11 月第 1 版　2019 年 5 月第 6 版　　**印　　次**：2019 年 5 月第 1 次印刷
定　　价：89.00 元

产品编号：080113-01

前　言

在本次修订中，所有的算法程序均采用 C++ 语言描述，并逐个进行了重新调试，对原来的程序做了较大的修改。对于有些问题，为了便于读者直接使用，在使用面向过程的 C++ 语言描述基础上，还使用了面向对象的 C++ 语言描述，将若干同类算法封装在一个类中。例如，在本书的第 1 章中，分别将复数运算封装成一个类，实系数多项式运算封装成一个类，复系数多项式运算封装成一个类，产生随机数运算封装成一个类；在第 12 章和第 13 章中分别将特殊函数与数据排序封装成类。由于在第 1 章中定义了复数运算类，因此在 2.1 节中矩阵相乘包括了实矩阵与复矩阵的相乘，2.2 节中的矩阵求逆包括了实矩阵与复矩阵的求逆。

本书是针对工程中常用的行之有效的算法而编写的，并且根据算法的分类以及使用特点做了精心的组织和安排。本书具有以下特点。

(1) 书中除收集了传统的算法外，还根据作者的工作经验和近年来数值计算的发展，选取了一些新的、实用的算法。可以说，书中各章几乎都有一些新的算法。

(2) 书中所有的算法程序都经过认真的调试(在 Visual C++ 6.0 环境下)。

(3) 书中收集的算法都是行之有效的，基本可以满足解决工程中各种实际问题的需要。

限于作者水平，书中难免有疏漏之处，恳请读者批评指正。

作　者

2018 年 3 月

目录

第7章 数值积分 256

第8章 常微分方程组的求解 302

第9章 数据处理 386

第 1 章

基本运算类

1.1 复数运算类

【功能】

1. 计算两个复数的和
2. 计算两个复数的差
3. 计算两个复数的乘积
4. 计算两个复数的商
5. 计算复数的整数次幂
6. 计算复数的 n 次方根
7. 计算复数的指数
8. 计算复数的自然对数
9. 计算复数的正弦值
10. 计算复数的余弦值
11. 计算复数的模
12. 计算复数的幅角

在本节中，$\mathrm{j}=\sqrt{-1}$。

【方法说明】

1. 计算两个复数的和

设

$$u+\mathrm{j}v=(a+\mathrm{j}b)+(c+\mathrm{j}d)$$

则

$$u=a+c,\quad v=b+d$$

2. 计算两个复数的差

设

$$u+\mathrm{j}v=(a+\mathrm{j}b)-(c+\mathrm{j}d)$$

则

$$u = a - c, \quad v = b - d$$

3. 计算两个复数的乘积

通常，计算两个复数的乘积需要 4 次实数乘法，下面的算法只需要 3 次乘法。

设

$$u + \mathrm{j}v = (a + \mathrm{j}b)(c + \mathrm{j}d)$$

令

$$p = ac, \quad q = bd, \quad s = (a + b)(c + d)$$

则

$$u = p - q, \quad v = s - p - q$$

4. 计算两个复数的商

设

$$u + \mathrm{j}v = (a + \mathrm{j}b)/(c + \mathrm{j}d)$$

令

$$p = ac, \quad q = -bd, \quad s = (a + b)(c - d), \quad w = c^2 + d^2$$

则

$$u = (p - q)/w, \quad v = (s - p - q)/w$$

5. 计算复数的整数次幂

设

$$u + \mathrm{j}v = (x + \mathrm{j}y)^n$$

令

$$z = x + \mathrm{j}y = r(\cos\theta + \mathrm{j}\sin\theta)$$

其中

$$r = \sqrt{x^2 + y^2}, \quad \theta = \arctan\frac{y}{x}$$

则

$$u + \mathrm{j}v = z^n = (x + \mathrm{j}y)^n = r^n(\cos n\theta + \mathrm{j}\sin n\theta)$$

即

$$u = r^n \cos n\theta, \quad v = r^n \sin n\theta$$

6. 计算复数的 *n* 次方根

设复数为

$$z = x + \mathrm{j}y = r(\cos\theta + \mathrm{j}\sin\theta)$$

其中

$$r = \sqrt{x^2 + y^2}, \quad \theta = \arctan\frac{y}{x}$$

则

$$\begin{aligned} u_k + \mathrm{j}v_k &= z^{\frac{1}{n}} = (x + \mathrm{j}y)^{\frac{1}{n}} \\ &= r^{\frac{1}{n}}\left(\cos\frac{2k\pi + \theta}{n} + \mathrm{j}\sin\frac{2k\pi + \theta}{n}\right), \quad k = 0, 1, \cdots, n - 1 \end{aligned}$$

其中

$$u_k = r^{\frac{1}{n}} \cos \frac{2k\pi + \theta}{n}, \quad v_k = r^{\frac{1}{n}} \sin \frac{2k\pi + \theta}{n}$$

7. 计算复数的指数

设复数为

$$z = x + \mathrm{j}y$$

则

$$u + \mathrm{j}v = \mathrm{e}^z = \mathrm{e}^{x+\mathrm{j}y} = \mathrm{e}^x(\cos y + \mathrm{j}\sin y)$$

其中

$$u = \mathrm{e}^x \cos y, \quad v = \mathrm{e}^x \sin y$$

8. 计算复数的自然对数

设复数为

$$z = x + \mathrm{j}y$$

则

$$u + \mathrm{j}v = \ln z = \ln(x + \mathrm{j}y) = \ln \sqrt{x^2 + y^2} + \mathrm{j}\arctan \frac{y}{x}$$

其中

$$u = \ln \sqrt{x^2 + y^2}, \quad v = \arctan \frac{y}{x}$$

9. 计算复数的正弦值

设复数为

$$z = x + \mathrm{j}y$$

则

$$\begin{aligned} u + \mathrm{j}v &= \sin z = \sin(x + \mathrm{j}y) = \sin x \cos(\mathrm{j}y) + \cos x \sin(\mathrm{j}y) \\ &= \left(\frac{\mathrm{e}^y + \mathrm{e}^{-y}}{2}\right)\sin x + \mathrm{j}\left(\frac{\mathrm{e}^y - \mathrm{e}^{-y}}{2}\right)\cos x \end{aligned}$$

其中

$$u = \left(\frac{\mathrm{e}^y + \mathrm{e}^{-y}}{2}\right)\sin x, \quad v = \left(\frac{\mathrm{e}^y - \mathrm{e}^{-y}}{2}\right)\cos x$$

10. 计算复数的余弦值

设复数为

$$z = x + \mathrm{j}y$$

则

$$\begin{aligned} u + \mathrm{j}v &= \cos z = \cos(x + \mathrm{j}y) = \cos x \cos(\mathrm{j}y) - \sin x \sin(\mathrm{j}y) \\ &= \left(\frac{\mathrm{e}^y + \mathrm{e}^{-y}}{2}\right)\cos x - \mathrm{j}\left(\frac{\mathrm{e}^y - \mathrm{e}^{-y}}{2}\right)\sin x \end{aligned}$$

其中

$$u = \left(\frac{\mathrm{e}^y + \mathrm{e}^{-y}}{2}\right)\cos x, \quad v = -\left(\frac{\mathrm{e}^y - \mathrm{e}^{-y}}{2}\right)\sin x$$

11. 计算复数的模

设复数为

$$z = x + \mathrm{j}y$$

则复数模为

$$r = |x + \mathrm{j}y| = \sqrt{x^2 + y^2}$$

12. 计算复数的幅角

设复数为

$$z = x + \mathrm{j}y$$

则复数的幅角为

$$\theta = \arctan \frac{y}{x}$$

【数据成员与函数成员】

类名：complex

数据成员	说明
double R	复数的实部
double I	复数的虚部

函数成员	说明
complex(double, double)	构造函数
void prt()	复数输出。形式为(实部，虚部)
complex operator+(complex&)	复数加法。重载运算符+
complex operator-(complex&)	复数减法。重载运算符-
complex operator*(complex&)	复数乘法。重载运算符*
complex operator/(complex&)	复数除法。重载运算符/
complex cpower(int)	复数乘幂
void croot(int, complex *)	复数的 n 次方根
complex cexp()	复数指数
complex clog()	复数对数
complex csin()	复数正弦
complex ccos()	复数余弦
double cfabs()	复数模
double angle()	复数幅角

【程序】

```
//复数运算类.h
#include <iostream>
#include <cmath>
using namespace std;
```

```
class  complex
{
private:
        double  R;
        double  I;
public:
        complex(double real=0, double imag=0)   //构造函数
        { R =real;  I =imag;  }
        void prt()                              //复数输出
        {
           cout <<"(" <<R <<", " <<I <<")" ;
                                                //输出形式为(实部, 虚部)
           return;
        }
        double cfabs()                          //复数模
        {
           double y;
           y=sqrt(R * R+I * I);
           return y;
        }
        double angle()                          //复数幅角
        {
           double y;
           y =atan2(I, R);
           return(y);
        }
        complex operator + (complex& c2)        //复数加法
        {
           complex c;
           c.R =R +c2.R;  c.I =I +c2.I;
           return c;
        }
        complex operator - (complex& c2)        //复数减法
        {
           complex c;
           c.R =R -c2.R;  c.I =I -c2.I;
           return c;
        }
        complex operator *  (complex& c2)       //复数乘法
        {
           complex c;
           double p, q, s;
           p =R * c2.R; q =I * c2.I;
           s = (R+I) * (c2.R+c2.I);
```

```
        c.R =p -q;  c.I =s -p -q;
        return c;
    }
    complex operator / (complex& c2)                //复数除法
    {
        complex c;
        double p, q, s, w;
        p =R * c2.R; q =-I * c2.I;
        s = (R+I) * (c2.R-c2.I);
        w = (c2.R) * (c2.R) + (c2.I) * (c2.I);
        if (w +1.0 !=1.0)
        {
            c.R = (p -q)/w;  c.I = (s -p -q)/w;
        }
        else
        {
            c.R =1e+300 ;  c.I =1e+300 ;
        }
        return c;
    }
    complex cpower (int n)                          //复数乘幂
    {
        complex  c;
        double r, q;
        q =atan2(I, R);
        r =sqrt(R * R +I * I);
        if (r+1.0 !=1.0)
        { r =n * log(r);  r =exp(r); }
        c.R =r * cos(n * q);  c.I =r * sin(n * q);
        return  c;
    }
    void croot (int n, complex * p)                 //复数的 n 次方根
    {
        complex c;
        int k;
        double r, q, t;
        if (n <1) return;
        q =atan2(I, R);
        r =sqrt(R * R +I * I);
        if (r+1.0 !=1.0)
        { r = (1.0/n) * log(r);  r =exp(r); }
        for (k=0; k<n; k++)
        {
            t = (2.0 * k * 3.1415926 +q)/n;
```

```
            c.R =r * cos(t);  c.I =r * sin(t);
            p[k] =c;
        }
    }
    complex cexp()                                   //复数指数
    {
        complex c;
        double p;
        p =exp(R);
        c.R =p * cos(I);  c.I =p * sin(I);
        return c;
    }
    complex clog()                                   //复数对数
    {
        complex c;
        double p;
        p =R * R +I * I;
        p =log(sqrt(p));
        c.R =p;  c.I =atan2(I, R);
        return c;
    }
    complex csin()                                   //复数正弦
    {
        complex c;
        double p, q;
        p =exp(I); q =exp(-I);
        c.R =sin(R) * (p+q)/2;
        c.I =cos(R) * (p-q)/2;
        return c;
    }
    complex ccos()                                   //复数余弦
    {
        complex c;
        double p, q;
        p =exp(I); q =exp(-I);
        c.R =cos(R) * (p+q)/2;
        c.I =-sin(R) * (p-q)/2;
        return c;
    }
};
```

【例】 首先输入两个复数，分别执行加、减、乘、除运算，并输出结果；然后重新输入一个复数，分别求该复数的－3次方、5次方根、指数、对数、正弦和余弦，并输出结果。

主函数程序如下：

```
//复数运算类例.cpp
#include <iostream>
#include <cmath>
#include  "复数运算类.h"
using namespace std;
int  main()                                    //主函数
{
    int i;
    double a, b;
    complex  c1, c2, c3, c, p[5];
    cin >>a >>b;                               //输入复数 c1 的实部与虚部
    c1 =complex(a, b);  cout <<"c1 ="; c1.prt(); cout <<endl;
    cin >>a >>b;                               //输入复数 c2 的实部与虚部
    c2 =complex(a, b);  cout <<"c2 ="; c2.prt(); cout <<endl;
    cin >>a >>b;                               //输入复数 c3 的实部与虚部
    c3 =complex(a, b);  cout <<"c3 ="; c3.prt(); cout <<endl;
    c =c1 +c2;
    cout <<"c1 +c2 =";  c.prt(); cout <<endl;
    c =c1 -c2;
    cout <<"c1 -c2 =";  c.prt(); cout <<endl;
    c =c1 * c2;
    cout <<"c1 * c2 =";  c.prt(); cout <<endl;
    c =c1 / c2;
    cout <<"c1 / c2 =";  c.prt(); cout <<endl;
    c =c3.cpower(-3);
    cout <<"c3 的-3 次方 =";  c.prt(); cout <<endl;
    cout <<"c3 的 5 次方根为: " <<endl;
    c3.croot(5, p);
    for (i=0; i<5; i++)
    {
        p[i].prt(); cout <<endl;
    }
    c =c3.cexp();
    cout <<"cexp(c3) =";  c.prt(); cout <<endl;
    c =c3.clog();
    cout <<"clog(c3) =";  c.prt(); cout <<endl;
    c =c3.csin();
    cout <<"csin(c3) =";  c.prt(); cout <<endl;
    c =c3.ccos();
    cout <<"ccos(c3) =";  c.prt(); cout <<endl;
    return 0;
}
```

运行结果为(其中第1行为键盘输入)