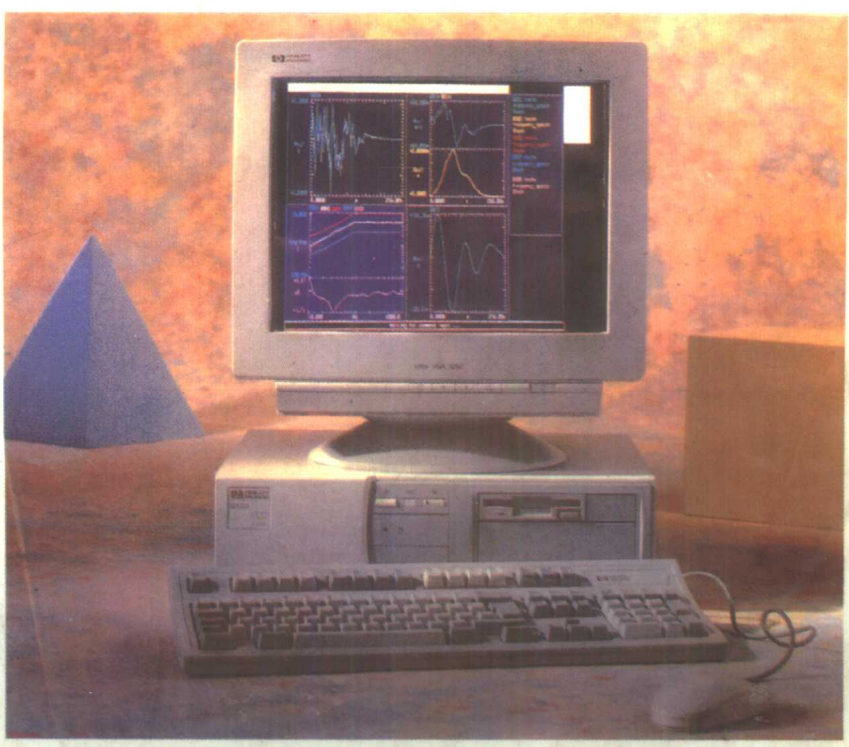
计算机应用入门丛书

科学与工程计算常用算法程序库 - C 语言版

KEXUE YU GONGCHENG JISUAN CHANGYONG SUANFACHENGXUKU - C YUANBAN

陈 进 梁 健 杨 扬 娄臻亮 编著



上海交通大学出版社

(沪)新登字 205 号

内 容 提 要

本书是一本由 C 语言编写的科学与工程计算常用算法程序库，汇集了计算方法中与科学与工程计算有关的较为常用和有效的算法，包含复数运算、插值法、数值积分与微分、线性方程组求解、代数方程与超越方程求解、拟合与平滑、矩阵分析、数理统计中的回归分析、数学变换等方面的内容。其中的绝大部分算法都经过了仔细的筛选，可以满足科学与工程计算的一般需要。

本书既是一本参考书，也是一本实用的工具书。书中所给出的全部算法程序均已收入软盘，并都经过了严格的调试考核。软盘中的内容又分为两大部分：第一部分包括算法源程序、例题程序和例题数据；第二部分为可供直接调用的函数库。需要此软盘的读者可与本社计算机编辑室联系。

读者对象：科研人员、工程技术人员、大专院校师生及计算工作者。

科学与工程计算常用算法程序库——C 语言版

出版：上海交通大学出版社
（上海市华山路 1954 号　邮政编码：200030）
发行：新华书店上海发行所　　版次：1995 年 7 月　第 1 版
印刷：上海交通大学印刷厂　　印次：1995 年 7 月　第 1 次
开本：787×1092(毫米)1/16　　印数：1－3000
印张：25.75　字数：640000　　科目：344－276

ISBN 7－313－01413－9/TP・262　　定价：29.50 元

前　言

当今世界，计算机尤其是微型计算机的普及已经无需更多的文字描述与形容。你只要步入学校、机关、研究所、工厂，你只要踏进办公室、实验室、车间控制室，你就会看到计算机那井井有条、快速准确的工作，你就会赞叹计算机那不可思议的处理能力。

在众多的计算机应用中，科学与工程计算一直是计算机应用的一个极为重要的方面，是绝大多数从事教学和科研的教师、研究人员，从事工程设计与分析的技术人员，以及尚在攻读的研究生和大学生都会碰到或正面临的问题。然而，科学与工程计算却往往只是整个研究课题或整个工程项目的一个部分。在实践中，我们一般总是希望将主要的注意力和精力都集中在要解决的课题或项目上面，这样每当处理到某方面的具体计算问题时，自然就希望能有现成的，在无需加工或尽可能少加工的条件下就可以被采用的计算方法和相应的程序，而且还要求这些程序是经过提炼和总结被证明是有效的。因此，预备一套有关科学与工程计算常用算法就成了每一个科研和工程技术人员的共同愿望。本书正是根据这种广泛的需要而编写的。

本书中所给出的所有算法程序都是基于目前十分流行的，并且呈上升趋势的 C 语言(Microsoft C 5.0)而设计出来的。从计算机程序语言发展的趋势来看，由于 C 语言具有许多其他语言所没有的优秀特征，可以预计 C 语言在科学与工程领域中的应用必将更加广泛，更加深入。因此，我们将过去在工作中积累起来的 C 语言算法程序进行了整理、补充、精选和对比分析，组织构造了一个比较完整的 C 语言算法程序库，以利于广大的教师、科研人员、工程技术人员、研究生、本科生以及其他从事计算工作的人员对常用算法的共享，从而避免不必要的重复劳动，有效地保证程序设计质量，提高工作效率，将最主要的精力花在处理自己的专业问题上面。

书中所选用的算法均是科学与工程计算中最为常用的算法，覆盖了复数运算、插值法、数值积分与微分、线性方程组求解、代数方程与超越方程求解、拟合与平滑、矩阵分析、数理统计中的回归分析、数学变换等范围，基本上能够满足科学与工程计算的一般需要。在上述各方面的算法中，通常都有若干种算法，书中则只筛选了其中的一种或几种较为典型和具有代表性的算法，并且尽量阐明算法的特点和适用范围，以方便使用者的选用。

书中所给出的所有算法程序均在 IBM-PC 系列微机及其兼容机上调试通过，并收录于软盘之中。软盘中的程序共分为两大部分：第一部分是所有算法的源程序，包括例题及数据，使用者可以直接取出或稍加修改融入自己的程序之中加以利用；第二部分则是一个实用的运行联接函数库，就像 C 语言编译器所提供的函数库一样，使用者可以直接调用其中的函数，极为简单、方便。

在本书的编写过程中，得到了邓小嘉、王小铁两位同学的许多帮助。此外，本书的顺利出版也得到了上海交通大学出版社的大力支持和帮助，在此一并表示衷心的感谢。

由于水平所限，书中难免有错误与不妥之处，恳请广大读者批评指正。

编　著　者

1993 年 5 月

目　录

第一章　复数运算 …… 1
§1.1　复数的构成 …… 1
§1.2　复数的加、减、乘、除四则运算 …… 2
§1.3　复数的模与辐角 …… 6
§1.4　复数的共轭 …… 8
§1.5　复指数函数 e^z …… 9
§1.6　复数的乘方 z^n …… 10
§1.7　复数的 n 次方根 $z^{\frac{1}{n}}$ …… 12
§1.8　复数的三角函数 …… 14

第二章　插值法 …… 17
§2.1　一元全区间拉格朗日插值 …… 17
§2.2　一元三点拉格朗日插值 …… 19
§2.3　一元三点分段拉格朗日插值 …… 22
§2.4　埃特金逐步插值 …… 25
§2.5　埃尔密特插值 …… 28
§2.6　第一种边界条件的三次样条函数插值 …… 31
§2.7　第二种边界条件的三次样条函数插值 …… 35
§2.8　第三种边界条件的三次样条函数插值 …… 39
§2.9　二元三点拉格朗日插值 …… 44
§2.10　二元双三次样条函数插值 …… 48

第三章　微分方程(组)的求解与数值积分 …… 57
§3.1　定步长欧拉方法 …… 57
§3.2　定步长龙格－库塔法 …… 60
§3.3　变步长龙格－库塔法 …… 63
§3.4　变步长基尔方法 …… 67
§3.5　变步长默森方法 …… 72
§3.6　哈明方法 …… 77
§3.7　阿当姆斯预报－校正法 …… 84
§3.8　吉尔方法 …… 90
§3.9　变步长辛卜生求积法 …… 110
§3.10　自适应梯形求积法 …… 113

§3.11 龙贝格求积法…… 116
§3.12 高斯求积法…… 119
§3.13 高振荡函数求积法…… 122
§3.14 切比雪夫求积法…… 125
§3.15 变步长辛卜生二求重积分法…… 128
§3.16 计算多重积分的高斯方法…… 133

第四章 线性方程组求解…… 138
§4.1 列主元高斯消去法 …… 138
§4.2 全主元高斯消去法 …… 142
§4.3 列主元高斯－约当消去法 …… 146
§4.4 克劳特分解法 …… 151
§4.5 追赶法 …… 155
§4.6 对称线性方程组的(三角)分解法 …… 158
§4.7 对称正定方程组的平方根法 …… 162
§4.8 对称带型方程组的解法 …… 166
§4.9 一般线性带型方程组的解法 …… 170
§4.10 大型稀疏线性方程组的解法…… 176
§4.11 复系数线性方程组的解法…… 180
§4.12 高斯－赛德尔迭代法…… 185

第五章 代数方程与超越方程求根…… 189
§5.1 求实系数多项式实根的切线法 …… 189
§5.2 求实系数多项式根的劈因子法 …… 193
§5.3 二分法求根 …… 199
§5.4 牛顿求根法 …… 203
§5.5 改进的割线求根法 …… 206
§5.6 插值与二分法组合求根 …… 209
§5.7 多点迭代求根法 …… 216
§5.8 求复函数方程根的下山法 …… 220

第六章 拟合与平滑…… 227
§6.1 最小二乘曲线拟合 …… 227
§6.2 等距节点正交多项式曲线拟合 …… 231
§6.3 切比雪夫曲线拟合 …… 235
§6.4 最佳一致逼近的里米兹方法 …… 241
§6.5 五点三次平滑 …… 247

第七章　矩阵分析…… 251
§7.1　行列式的值 …… 251
§7.2　对称正定阵的求逆 …… 255
§7.3　一般实矩阵的秩和求逆 …… 258
§7.4　计算实对称阵特征值和特征向量的雅可比方法 …… 263
§7.5　广义特征值和特征向量的简化 …… 269
§7.6　一般实矩阵化为上赫申伯格阵的初等变换法 …… 279
§7.7　将一般实矩阵化为上赫申伯格阵的豪斯荷尔德变换 …… 282
§7.8　将一般复矩阵化为上赫申伯格阵 …… 286
§7.9　求一般实矩阵特征值的初等相似法 …… 291
§7.10　求实对称阵特征值及特征向量的 QL 算法…… 298
§7.11　求一般实矩阵的特征值及特征向量的 QR 算法…… 305
§7.12　复矩阵的特征值及特征向量求解…… 317
§7.13　一般实矩阵的奇异值分解…… 328

第八章　数理统计中的回归分析…… 346
§8.1　一元线性回归分析 …… 347
§8.2　多元线性回归分析 …… 351
§8.3　一元曲线回归分析 …… 356
§8.4　多元逐步回归分析 …… 362

第九章　几种快速数学变换…… 371
§9.1　基 2 的时间抽取型 FFT …… 371
§9.2　基 2 的频率抽取型 FFT …… 376
§9.3　基 4 的 FFT …… 380
§9.4　同时计算两组实数据的 FFT …… 386
§9.5　用 N 点变换来计算 $2N$ 个实数据点的 FFT …… 391
§9.6　快速沃尔什变换(FWT)…… 396
§9.7　快速逆拉普拉斯变换(FILT)…… 399

参考文献…… 404

第一章　复数运算

在科学与工程计算中，我们常会碰到有关复数的运算。然而，C 语言并不具备直接处理复数运算的功能，因此，在这一章中，首先给出有关复数的各种常用基本运算方法。

在后面的章节中也将引用复数运算。

§1.1　复数的构成

一、功能

用两个实数 x 和 y 分别作为一个复数的实部和虚部构成复数 z。

二、方法简介

复数 $z=x+iy$。在 C 语言中具有一种叫做“结构”的数据类型，而且按照 ANSI 标准，结构可以作为函数的参数进行传递，同时也可以作为函数的返回值。因此，我们可以方便地利用结构来表示复数，并定义复数运算。

三、使用说明

1. 复数数据类型定义

```
typedef    struct cmplx
           {
              double Re;
              double Im;
           } Complex;
```

于是，“Complex”就成为一种新的数据类型，即复数类型。

上面这个关于复数类型的定义，可以放在某一个“头文件”中，本书后面有关内容均假定该头文件为“Complex. h”，这样，在具体使用复数运算时，只需在程序头上加 #include “Complex. h”一行即可。

2. 函数说明

```
Complex  Cmplx __Make(x,y)
double   x,y;
```

3. 参数解释

x　　复数实部的一个实数。

y　　复数虚数的一个实数。

4. 返回值

返回一个复数。

四、源程序

1. 复数运算头文件 Complex.h

```
typedef struct cmplx
        {
            double Re;
            double Im;
        } Complex;

extern  Complex  Cmplx__Make(double, double);
extern  Complex  Cmplx__Add(Complex, Complex);
extern  Complex  Cmplx__Sub(Complex, Complex);
extern  Complex  Cmplx__Mul(Complex, Complex);
extern  Complex  Cmplx__Div(Complex, Complex);
extern  double   Cmplx__Abs(Complex);
extern  double   Cmplx__Arg(Complex);
extern  Complex  Cmplx__Cnj(Complex);
extern  Complex  Cmplx__Exp(Complex);
extern  Complex  Cmplx__Pow(Complex, double);
extern  Complex  Cmplx__Root(Complex, int, int);
extern  Complex  Cmplx__Tri(Complex, char);
```

2. 源程序

```
#include "complex.h"

Complex Cmplx__Make(x,y)
double x,y;
{
    Complex z;

    z.Re=x;
    z.Im=y;
    return(z);
}
```

§1.2 复数的加、减、乘、除四则运算

一、功能

计算两个复数 $z_1=x_1+iy_1$ 和 $z_2=x_2+iy_2$ 的和、差、积、商。

二、方法简介

1. 两个复数的和

$$z_1 + z_2 = (x_1 + x_2) + \mathrm{i}(y_1 + y_2)。$$

2. 两个复数的差

$$z_1 - z_2 = (x_1 - x_2) + \mathrm{i}(y_1 - y_2)。$$

3. 两个复数的积

$$z_1 \cdot z_2 = (x_1x_2 - y_1y_2) + \mathrm{i}(x_1y_2 + x_2y_1)。$$

4. 两个复数的商

$$z_1/z_2 = (x_1x_2 + y_1y_2)/(x_2^2 + y_2^2) + \mathrm{i}(x_2y_1 - x_1y_2)/(x_2^2 + y_2^2)。$$

为防止计算 x_2^2, y_2^2 时溢出，实际计算时

$$z_1/z_2 = \begin{cases} ((x_1 + y_1\dfrac{y_2}{x_2}) + \mathrm{i}(y_2 - x_1\dfrac{y_2}{x_2}))/(x_2 + y_2 \cdot y_2/x_2), & |x_2| \geqslant |y_2|, \\ ((x_1 \cdot \dfrac{x_2}{y_2} + y_1) + \mathrm{i}(y_1\dfrac{x_2}{y_2} - x_1))/(x_2 \cdot x_2/y_2 + y_2), & |x_2| < |y_2|。 \end{cases}$$

三、使用说明

1. 函数说明

(1)复数的加法

```
Complex  Cmplx __Add(z1,z2)
Complex  z1,z2;
```

(2)复数的减法

```
Complex  Cmplx __Sub (z1,z2)
Complex  z1,z2;
```

(3)复数的乘法

```
Complex  Cmplx __Mul(z1,z2)
Complex  z1,z2;
```

(4)复数的除法

```
Complex  Cmplx __Div(z1,z2)
Complex  z1,z2;
```

2. 参数解释

z1,z2　　分别代表参加运算的两个复数，在计算复数除法时，若|z2|=0，会给出出错警告，并返回一个实部和虚部均为“无穷大”的复数。

四、算例

1. 例题

设 z_1=2.1+i7.3，　　z_2=0.7+i3.5。

分别计算：

(1)$z=z_1+z_2$；

(2)$z=z_1-z_2$;

(3)$z=z_1 \cdot z_2$;

(4)$z=z_1/z_2$。

2. 例题程序

```
#include <stdio.h>
#include "complex.h"

main()
{
    Complex z,z1, z2;
    z1=Cmplx _Make(2.1,7.3);
    z2=Cmplx _Make(0.7,3.5);

    z=Cmplx _Add(z1,z2);
    printf("\nz=z1+z2=(%6.2f)+i(%6.2f)", z.Re, z.Im);
    z=Cmplx _Sub(z1,z2);
    printf("\nz=z1-z2=(%6.2f)+i(%6.2f)", z.Re, z.Im);
    z=Cmplx _Mul(z1,z2);
    printf("\nz=z1*z2=(%6.2f)+i(%6.2f)", z.Re, z.Im);
    z=Cmplx _Div(z1,z2);
    printf("\nz=z1/z2=(%6.2f)+i(%6.2f)", z.Re, z.Im);
}
```

3. 结果

```
z=z1+z2=(  2.80)  +i (10.80)
z=z1-z2=(  1.40)  +i ( 3.80)
z=z1*z2=(-24.08)  +i (12.46)
z=z1/z2=(  2.12)  +i (-0.18)
```

五、源程序

```
#include <float.h>
#include <math.h>
#include "complex.h"

Complex Cmplx _Add(z1,z2)
Complex z1,z2;
{
    Complex z;

    z.Re=z1.Re+z2.Re;
    z.Im=z1.Im+z2.Im;
    return(z);
}
```

```
Complex Cmplx _Sub(z1,z2)
Complex z1,z2;
{
    Complex z;

    z.Re=z1.Re-z2.Re;
    z.Im=z1.Im-z2.Im;
    return(z);
}

Complex Cmplx _Mul(z1,z2)
Complex z1,z2;
{
    Complex z;

    z.Re=z1.Re*z2.Re-z1.Im*z2.Im;
    z.Im=z1.Re*z2.Im+z1.Im*z2.Re;
    return (z);
}

Complex Cmplx _Div(z1,z2)
Complex z1,z2;
{
    Complex z;
    double w, w1;

    if ( z2.Re==0.0 && z2.Im==0.0)
    {
        z.Re=z.Im=1.0e308;
        return(z);
    }
    if( fabs(z2.Re) >=fabs(z2.Im))
    {
        w=z2.Im/z2.Re;
        w1=z2.Re+z2.Im*w;
        z.Re=(z1.Re+z1.Im*w)/w1;
        z.Im=(z1.Im-z1.Re*w)/w1;
    }
    else
    {
        w=z2.Re/z2.Im;
        w1=z2.Re*w+z2.Im;
```

```
        z.Re=(z1.Re*w=z1.Im)/w1;
        z.Im=(z1.Im*w-z1.Re)/w1;
    }
    return(z);
}
```

§1.3 复数的模与辐角

一、功能

计算复数 $z=x+iy$ 的模与主辐角。

二、方法简介

复数的模定义为

$$|z| = \sqrt{x^2+y^2}。$$

复数的辐角定义为

$$\mathrm{Arg}(z) = \arctan(y/x) + 2k\pi, (k = 0, \pm 1, \pm 2, \cdots)。$$

其中主辐角满足

$$0 \leqslant \mathrm{Arg}(z) < 2\pi。$$

三、使用说明

1. 函数说明

(1)复数的模

```
double  Cmplx __Abs(z)
Complex  z;
```

(2)复数的主辐角

```
double  Cmplx __Arg(z)
Complex  z;
```

2. 参数解释

z　参加运算的复数，当 $|z|=0$ 时，辐角的值没有定义，此时主辐角函数会给出警告，并以超过主辐角范围的 2π 作为返回值。

3. 返回值

(1)返回复数 z 的模 $|z|$。

(2)返回复数 z 的主辐角。

四、算例

1. 例题

$z=2.1+i7.3$。

2. 例题程序

```
#include <stdio.h>
#include <math.h>
#include "complex.h"

main()
{
    double    a, Arg;
    Complex   z;

    z=Cmplx__Make (2.1,7.3);
    a=Cmplx__Abs(z);
    Arg=Cmplx__Arg(z);
    printf("\n|z|      =%7.4f\nArg(z)=%7.4f",a,Arg);
}
```

3. 结果

$|z|=(7.5961)$

$Arg(z)=(1.2907)$

五、源程序

```
#include <float.h>
#include <math.h>
#include "complex.h"

#define   PI    3.141592653

double Cmplx__Abs(z)
Complex z;
{
    double a;

    a=sqrt(z.Re * z.Re+z.Im * z.Im);
    return(a);
}

double Cmplx__Arg(z)
Complex z;
{
    double a;

    if(z.Re==0.0 && z.Im==0.0)
    {
        a=2.0 * PI;
```

```
    }
    else
    {
        a=atan2(z.Im,z.Re);
    }
    return(a);
}
```

§1.4 复数的共轭

一、功能

计算复数 $z=x+iy$ 的共轭复数。

二、方法简介

根据定义复数 z 的共轭复数 z^* 为

$$z^* = x - iy。$$

三、使用说明

1. 函数说明

```
Complex  Cmplx__Cnj(z)
Complex  z;
```

2. 参数解释

z　参数运算的复数。

四、算例

1. 例题

已知 $z=2.1+i7.3$。求 z^*。

2. 例题程序

```
#include <stdio.h>
#include "complex.h"

main()
{
    Complex z;

    z=Cmplx__Make(2.1,7.3);
    z=Cmplx__Cnj(z);
    printf("\nz'=(%4.1f)+i(%4.1f)",z.Re, z.Im);
}
```

3. 结果

$z^* = (2.1) + i(-7.3)$

五、源程序

```
#include "complex.h"

Complex Cmplx__Cnj(z)
Complex z;
{
    Complex c;

    c.Re=z.Re;
    c.Im=-z.Im;
    return(c);
}
```

§1.5 复指数函数 e^z

一、功能

计算复指数函数 e^z，其中 $z=x+iy$。

二、方法简介

根据欧拉公式，有

$$e^z = e^{x+iy} = e^x\cos y + ie^x\sin y。$$

三、使用说明

1. 函数说明

   Complex Cmplx__Exp(z)
   Complex z;

2. 参数解释

   z 参数运算的复数指数。

3. 返回值

   返回复指数函数的结果，其结果为一复数。

四、算例

1. 例题

   已知 $z=1+i\frac{\pi}{4}$。计算 e^z。

2. 例题程序

```
#include <stdio.h>
#include "complex.h"
```

```
#define PI 3.1415926

main()
{
    Complex z;

    z=Cmplx__Make      PI/4.0);
    z=Cmplx__Exp(z);
    printf("\nExp(z)=(%7.4f)+i(%7.4f)",z.Re,z.Im);
}
```

3. 结果

Exp(z)=(1.9221)+i(1.9221)

五、源程序

```
#include <math.h>
#include "complex.h"

Complex Cmplx__Exp (z)
Complex z;
{
    Complex e;

    e.Re=exp(z.Re) * cos(z.Im);
    e.Im=exp(z.Re) * sin(z.Im);
    return(e);
}
```

§1.6 复数的乘方 z^n

一、功能

计算复数 z 的 n 次乘方 z^n，其中 $z=x+iy$，n 为实数。

二、方法简介

将复数 z 表示成指数形式，有

$$z = re^{i\theta}$$

于是，根据德·莫弗公式

$$z^n = r^n e^{rn\theta} = r^n(\cos n\theta + i\sin n\theta)。$$

这里：

$$r = \sqrt{x^2 + y^2} = |z|,$$

$$\theta = \arctan(y/x)。$$