C 语言数值算法

张圣华编著

海洋出版社

北京希望电脑公司计算机技术丛书

C语言数值算法

张圣华 编著

海洋出版社

1993年·北京

内 容 简 介

本书包括了162个工程上常用的数学数值算法的C语言程序。内容包括：复数运算，插值，拟合与平滑，数值积分，线性代数方程组的求解，行列式求值及矩阵求逆，特征值与特征向量的计算，多项式方程的求解，超越方程的求解，线性代数方程组的求解，常微分方程的数值积分，特殊函数，数学变换，概率统计等。

所有这些程序均在IBM-PC系列及其兼容机上用Microsoft C 6.0调试通过。

本书可供计算机工作者、科技人员及高校师生参考。

需要本书的用户，可直接与北京8721信箱联系，邮码：100080，电话：2562329.

(京)新登字087号

C语言数值算法

张圣华 编著

*

海洋出版社出版(北京市复兴门外大街1号)

海洋出版社发行 北京四季青印刷厂印刷

开本：787×1092 1/16 印张：38 字数：840千字

1993年8月第一版 1993年8月第一次印刷

印数：1—5000册

*

ISBN 7-5027-3528-3/TP·195 定价：35.00元

前　言

数值计算在计算机的应用领域起着重要的作用。它不仅在科技和工程领域中扮演着重要角色，在当前微机迅猛发展的形势下，很多非科技与工程领域的应用软件中也要用到这其中的算法。如最小二乘法、三次样条函数插值、曲线拟合等算法应用得就非常广泛。

80年代C语言取得了瞩目的发展，C语言从一个小的系统程序设计语言成长为广为人知的通用程序设计语言。C语言由于它的紧凑、高效、易于移植、模块化等特色已被广泛地应用于系统程序设计、软件工具、图像处理、科学计算、数值分析、人工智能、数据库管理系统设计、文字处理等领域，而且越来越为广大程序员所喜爱。它的影响已达到了如此的地步：即能否使用C语言来编程是区分真正的计算机专业人员和一般计算机用户的分水岭。就各种编程语言来看，C语言的编程工具的发展是最快的。Microsoft公司和Borland公司不断地更新它们的C语言编程工具。这对广大程序员具有极大的吸引力。目前，C语言是唯一可以向具有统治地位的科学计算语言FORTRAN挑战的语言，C语言在很多方面都要强于FORTRAN语言。

由于C语言流行时间不长，比起流行了几十年的FORTRAN和BASIC语言来，有关C语言算法的书实在太少，尤其是在数值计算方面，这使C语言程序员颇感不便。因此特编此书，以供广大读者参考之用。

书中共给出了162个算法，其中每个算法中，都给出了算法概要、完整的C语言算法函数以及其应用示例。程序都是用Microsoft C6.0编写的，并都经过上机调试通过。这些程序绝大多数不需要修改就可适用于Turbo C。

目 录

第一章 导 论

1.1 C 语言的数学库函数

下面给出了Microsoft C 6.0 的数学库函数，它们的大多数都是采用double 数据类型，其中以l 结尾的函数表示支持80 位数据类型。有关这些函数的详细情况可参考其使用手册。

$acos, acosl$	计算反余弦
$asin, asinl$	计算反正弦
$atan, atanl$	计算反正切
$atan2, atan2l$	计算反正切
$cabs, cabsl$	计算复数的绝对值
$ceil, ceill$	求出上限整数
$cos, cosl$	计算余弦
$cosh, coshl$	计算双曲余弦
$div, ldiv$	整数相除，返回商和余数
$exp, expl$	计算指数函数
$fabs, fabsl$	求绝对值
$floor, floorl$	求小于等于变量的最大整数
$fmod, fmodl$	求浮点余数
$frexp, frexpl$	计算一指数值
$hypot, hypotl$	计算直角三角形的斜边
$j0, j0l$	计算0 阶第一类Bessel 函数
$j1, j1l$	计算1 阶第一类Bessel 函数
jn, jnl	计算n 阶第一类Bessel 函数
$ldexp, ldexpl$	计算变量与2^{exp} 的积
$log, logl$	计算自然对数
$log10, log10l$	计算以10 为底的对数
max, min	返回两个数值中较大或较小的值
$modf, modfl$	把变量分解为整数部分和小数部分
$pow, powl$	计算给定值的方幂
$sin, sinl$	计算正弦
$sinh, sinhl$	计算双曲正弦
$sqrt, sqrtl$	求平方根
$y0, y0l$	计算0 阶第二类Bessel 函数
$y1, y1l$	计算1 阶第二类Bessel 函数
yn, ynl	计算n 阶第二类Bessel 函数

1.2 动态数组

在编制一个通用的子程序时，若其中涉及到数组问题，最好采用动态数组，其优点是显而易见的。若采用传统的静态数组，我们不得不把数组设置得大些，以满足各种不同的需要。例如，在处理矩阵问题时，通常要用二维数组来存放矩阵元素，若在程序中设置了一个 50×50 的二维数组，则当所要处理的矩阵只是2×2 阶矩阵时，就造成了内存资源的浪费，而遇到51×51 阶矩阵时，程序又无能为力。解决这一问题的根本办法就是采用动态数组。

对于一维动态数组，C 语言中提供了一个内存分配函数calloc, 可以用它来为一维数组动态分配内存空间，calloc 为一维数组分配空间时，会把个元素初始化为0。当此数组不再需要时，可以用free 函数来释放所分配的空间。例如，我们要为一n 阶一维double 型数组x[n] 分配空间，则可表示如下:

```
double *x;

x=(double *)calloc(n, sizeof(double));
if( x == NULL )
   return( FALSE );
      .
      .
      .
free( x );
```

在进行动态数组分配时，一定要检查分配是否成功。当calloc 函数返回NULL 时，则表示分配失败。

对于二维动态数组，C 语言中并没有给出相应的函数，下面这段程序可以实现二维(r×c) 阶动态数组的分配和释放:

```
double **TwoArrayAlloc(int r, int c)
{
   double *x, **y;
   int    n;

   x=(double *)calloc(r*c, sizeof(double));
   y=(double **)calloc(r, sizeof(double *));
   for(n=0; n<=r-1; ++n)
      y[n]=&x[c*n];
   return( y );
}

void TwoArrayFree(double **x)
{
```

```
    free(x[0]);
    free(x);
}
```

这两段程序在以后的各章节中经常要用到。

1.3 C 语言中的复数表示

在C 语言中，没有复数型数据类型，但我们可以用C 语言中的结构来定义一复数。在Microsoft C 6.0 中已给出了这样的结构

```
struct complex {
                double x;
                double y;
              }
```

我们在程序中也可自己来定义这样的结构。有了它，我们就可以如下定义一个复变量z:

```
struct complex z;
```

对其赋值可以有两种方法(如$z = 3 + i4$):

```
struct complex z={3, 4};
```

或

```
z.x=3;   z.y=4;
```

类似地，我们也可以定义一个复变量数组，如

```
struct complex z[5];
struct complex z[5][5];
```

动态复变量数组的分配方法如下:

```
struct complex *z;

z=(struct complex *)calloc(5, sizeof(struct complex));
```

1.4 使用双精度数据类型

在Microsoft C 6.0 中，所有涉及浮点运算的数学库函数都使用double 型数据，鉴于此，在我们的所有程序中也都使用了double 型数据。

在Microsoft C 6.0 中，若在程序中有类似下列的语句

```
float  x;
x=sin(1);
```

并且使用了/W3 或/W4 选择项进行编译，则会显示下列警告信息

```
warning C4136: conversion between different floating types
```

这表示在等号的两边使用了不同的浮点类型数据，sin(1) 的返回值为double 型，而x 为float 型，当把sin(1) 的值赋与x 时，C 会自动作类型转换，首先要把sin(1) 的值变成float 型，然后再赋给x, 这样有可能会丢失数据的有效位数。

1.5 堆栈溢出问题

在一些程序中，由于静态数据量太大，超过了Microsoft C 6.0 编译器的缺省栈设置值2048 字节，则在程序执行时会出现如下的信息

```
stack overflow
```

在这种情况下，你可以在连接程序时，用Link 的/ST 选择项来改变栈设置值，如

```
link /ST:4096
```

但是，最大值不能超过64KB。

第二章 复数运算

2.1 复数的模

2.1.1 功能

本子程序用于求复数$z = x + iy$ 的模$|z|$。

2.1.2 方法简介

$$|z| = \sqrt{x^2 + y^2} = \begin{cases} |x|\sqrt{1 + \left(\dfrac{y}{x}\right)^2} & |x| > |y| \\ |y|\sqrt{1 + \left(\dfrac{x}{y}\right)^2} & |x| \le |y| \end{cases}$$

后一等式是为了防止上溢或下溢。

2.1.3 子程序及其使用说明

参数表

z　结构型变量，输入参数，存放复数z。

子程序

```
double CABS(struct complex z)
{
   z.x = fabs( z.x );
   z.y = fabs( z.y );
   if( z.x == 0 )
      return( z.y );
   else if( z.y == 0 )
      return( z.x );
   else
   {
      if( z.x > z.y )
         return( z.x * sqrt(1+(z.y/z.x)*(z.y/z.x)) );
      else
         return( z.y * sqrt(1+(z.x/z.y)*(z.x/z.y)) );
   }
}
```

2.1.4 例题

求复数$z = 3 + 4i$ 的模。

程序:

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

double CABS(struct complex z);

void main()
```

```
{
   struct complex z = {3, 4};

   printf("%.10e\n", CABS(z));
}
```

计算结果: $|z| = 5$.

2.2 复数除法

2.2.1 功能

本子程序用于求复数z_1 除以z_2 的商z。

2.2.2 方法简介

设$z_1 = a + ib, z_2 = c + id, z = x + iy$, 则

$$z = \frac{z_1}{z_2} = \frac{(ac+bd)+i(bc-ad)}{c^2+d^2} = \begin{cases} \dfrac{\left(a+b\dfrac{d}{c}\right)+i\left(b-a\dfrac{d}{c}\right)}{c+d\dfrac{d}{c}} & |c| \geq |d| \\[2ex] \dfrac{\left(a\dfrac{c}{d}+b\right)+i\left(b\dfrac{c}{d}-a\right)}{c\dfrac{c}{d}+d} & |c| < |d| \end{cases}$$

后一公式是为了防止在a, b, c, d 很大时或很小时用前一公式计算可能出现的上溢或下溢。

2.2.3 子程序及其使用说明

参数表

- z_1　结构型变量, 输入参数, 存放被除数。
- z_2　结构型变量, 输入参数, 存放除数。
- z　结构型变量, 输出参数, 存放商。

子程序

```
CDIV(struct complex z1, struct complex z2, struct complex *z)
{
   double e, f;

   if( fabs( z2.x ) >= fabs( z2.y ) )
   {
      e = z2.y / z2.x;
      f = z2.x + e * z2.y;
      z->x = ( z1.x + z1.y * e ) / f;
      z->y = ( z1.y - z1.x * e ) / f;
   }
   else
   {
      e = z2.x / z2.y;
      f = z2.y + e * z2.x;
```

```
        z->x = ( z1.x * e + z1.y ) / f;
        z->y = ( z1.y * e - z1.x ) / f;
    }
}
```

2.2.4 例题

求$z_1 = 10^{10} + i2 \times 10^{10}$ 除以$z_2 = 3 \times 10^{16} + i3 \times 10^{15}$ 的商。

程序:

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

int CDIV(struct complex, struct complex, struct complex *);

void main()
{
    struct complex z1 = {10e+10, 2*10e+10},
                   z2 = {3*10e+16, 3*10e+15},
                   z;

    CDIV(z1, z2, &z);
    printf("%.10e+i%.10e\n", z.x, z.y);
}
```

计算结果: $z = 3.9603960396e - 007 + i6.2706270627e - 007$

2.3 复数的根

2.3.1 功能

本子程序用于求复数的全部n 次根$z = x + iy = z_1^{1/n} = (a + ib)^{1/n}$。

2.3.2 方法简介

$$\begin{aligned} z = (a+ib)^{1/n} &= [r(\cos\phi + i\sin\phi)]^{1/n} \\ &= r^{1/n}\left(\cos\frac{\phi+2k\pi}{n} + i\sin\frac{\phi+2k\pi}{n}\right) \\ &= r^{1/n}(\cos(\theta+k\theta_1) + i\sin(\theta+k\theta_1)) \qquad k = 0, 1, \cdots, n-1. \end{aligned}$$

其中

$$\theta = \begin{cases} 0 & a>0, b=0 \\ \pi/n & a<0, b=0 \\ \pi/(2n) & a=0, b>0 \\ -\pi/(2n) & a=0, b<0 \\ \arctan(b/a)/n & a>0, b\neq 0 \\ (\arctan(b/a)+\pi)/n & a<0, b\neq 0 \end{cases} \qquad \theta_1 = \frac{2\pi}{n}$$