

C 常用算法程序集

(第二版)

(含盘)

徐士良 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书是针对工程上常用的行之有效的算法而编写的 C 语言函数程序集, 在第一版的基础上作了修改和扩充。书中包括了近几年出现的许多新算法。全书分为数值计算与非数值计算两部分。其中数值计算部分的内容包括: 线性代数方程组的求解、矩阵运算、矩阵特征值与特征向量的计算、非线性方程与方程组的求解、插值、数值积分、常微分方程(组)的求解、拟合与逼近、数据处理与回归分析、极值问题、数学变换与滤波、特殊函数、随机数的产生、多项式与连分式函数的计算、复数运算; 非数值计算部分的内容包括: 排序、查找、图形模式下读写屏幕象点、基本图形操作、汉字操作等。

书中所有的算法函数程序均在 IBM-PC 系列及其兼容机上调试通过, 并存放在一张 3 软盘上, 此软盘附在本书的封三上。

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, 也可作为高等院校的《数值分析》或《程序设计》等类似课程的参考书。

版权所有, 翻印必究。
本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

C 常用算法程序集/ 徐士良编著. —2 版. —北京: 清华大学出版社, 1996
ISBN 7-302-02290-9

. C... . 徐... . C 语言-算法设计 . TP312C

中国版本图书馆 CIP 数据核字(96)第 16182 号

出 版 者: 清华大学出版社(北京清华大学校内, 邮编 100084)
责任编辑: 范素珍
印 刷 者: 中国科学院印刷厂
发 行 者: 新华书店总店北京科技发行所
开 本: 787× 1092 1/ 16 印张: 30 字数: 745 千字
版 次: 1996 年 11 月第 2 版 1996 年 11 月第 1 次印刷
书 号: ISBN 7-302-02290-9/ TP · 1128
印 数: 0001—5000
定 价: 48. 00 元

前 言

C 语言已不仅被计算机软件工作者用来设计与开发系统软件,而且也为广大计算机应用工作者(计算工作者、科学研究人员及工程技术人员)所喜欢使用。本书是为方便用户而编写,内容包括数值计算与非数值计算两大部分。

自本书第一版问世以来,得到了广大读者的肯定,不少读者还提出了许多有益的建议,其中最迫切的是希望能方便地同时得到与本书配套的软盘。为了满足广大读者的要求,在清华大学出版社的支持下,推出了本书的第二版。第二版与第一版相比,主要有以下两个方面的不同:

首先,从内容上进行了扩充与调整,并修正了第一版中的某些不妥之处。在数值计算这部分中,除了在原有各章中增加了某些算法程序外,还增加了独立的三章——随机数产生、复数运算与多项式的运算。非数值计算这部分改动比较大,主要是增加了图形功能与汉字操作方面的算法函数。特别需要指出的是,在数值计算这部分中,包括了工程上常用的行之有效的算法程序,有些是近年来出现的新算法,它们基本可以满足各学科领域中经常遇到的实际计算问题的需要。

其次,本书所提供的所有算法函数程序均存放在与书配套的一张 3 软盘上,在本书的正文中将不再出现而只提供文件名,读者使用时可直接从软盘上读出。为了便于读者理解算法函数程序的功能及调用方式,书中仍提供了例题程序的清单以便于阅读。

本书的宗旨是实用。所有算法函数程序都经过了实际计算的检验,适用于所有 IBM PC 系列及其兼容机。在提供的配套 3 软盘中,包括了所有的算法函数程序,例题程序以及大部分例题程序的运行结果。各章的程序存放在相应的子目录中,子目录名分别为 CHX,其中 X 为本书的章号。所有算法函数程序的文件名均为函数名.C,函数名统一由 5 个字符组成。例题程序的文件名为函数名 0.C,当有两个以上的例题程序时,其文件名依次为函数名 0.C、函数名 1.C 等。存放相应例题程序运行结果的文件名依次为函数名 0.DAT、函数名 1.DAT 等。

限于水平,书中难免有错误及不妥之处,恳请读者批评指正。

徐士良

1996 年 5 月

目 录

第一篇 数值计算

第 1 章 线性代数方程组的求解	2	2.8 求矩阵秩的全选主元高斯消去法	48
1.1 全选主元高斯消去法	2	2.9 对称正定矩阵的乔里斯基分解与 行列式的求值	49
1.2 全选主元高斯-约当消去法	3	2.10 矩阵的三角分解	51
1.3 复系数方程组的全选主元高斯消 去法	5	2.11 一般实矩阵的 QR 分解	53
1.4 复系数方程组的全选主元高斯- 约当消去法	7	2.12 一般实矩阵的奇异值分解	55
1.5 求解三对角线方程组的追赶法	9	2.13 求广义逆的奇异值分解法	60
1.6 一般带型方程组的求解	11	第 3 章 矩阵特征值与特征向量的 计算	64
1.7 求解对称方程组的分解法	15	3.1 约化对称矩阵为对称三对角阵的 豪斯荷尔德变换法	64
1.8 求解对称正定方程组的平方根法	17	3.2 实对称三对角阵的全部特征值与 特征向量的计算	67
1.9 求解大型稀疏方程组的全选主元 高斯-约当消去法	19	3.3 约化一般实矩阵为赫申伯格矩阵 的初等相似变换法	69
1.10 求解托伯利兹方程组的列文逊 方法	20	3.4 求赫申伯格矩阵全部特征值的 QR 方法	70
1.11 高斯-赛德尔迭代法	24	3.5 求实对称矩阵特征值与特征向量 的雅可比法	74
1.12 求解对称正定方程组的共轭梯 度法	25	3.6 求实对称矩阵特征值与特征向量 的雅可比过关法	77
1.13 求解线性最小二乘问题的豪斯荷 尔德变换法	27	第 4 章 非线性方程与方程组的求解	79
1.14 求解线性最小二乘问题的广义 逆法	29	4.1 求非线性方程实根的对分法	79
1.15 病态方程组的求解	31	4.2 求非线性方程一个实根的牛顿法	80
第 2 章 矩阵运算	34	4.3 求非线性方程一个实根的埃特金 迭代法	82
2.1 实矩阵相乘	34	4.4 求非线性方程一个实根的连分式 解法	83
2.2 复矩阵相乘	35	4.5 求实系数代数方程全部根的 QR 方法	85
2.3 实矩阵求逆的全选主元高斯- 约当法	37	4.6 求实系数代数方程全部根的牛顿- 下山法	87
2.4 复矩阵求逆的全选主元高斯- 约当法	39	4.7 求复系数代数方程全部根的牛顿- 下山法	89
2.5 对称正定矩阵的求逆	42		
2.6 托伯利兹矩阵求逆的特兰持方法	44		
2.7 求行列式值的全选主元高斯消去 法	47		

4.8	求非线性方程组一组实根的梯度法	90	6.10	切比雪夫求积法	156
4.9	求非线性方程组一组实根的拟牛顿法	93	6.11	计算一维积分的蒙特卡洛法	157
4.10	求非线性方程组最小二乘解的广义逆法	96	6.12	变步长辛卜生二重积分法	158
4.11	求非线性方程一个实根的蒙特卡洛法	100	6.13	计算多重积分的高斯方法	161
4.12	求实函数或复函数方程一个复根的蒙特卡洛法	102	6.14	计算二重积分的连分式法	163
4.13	求非线性方程组一组实根的蒙特卡洛法	104	6.15	计算多重积分的蒙特卡洛法	165
第5章	插值	107	第7章	常微分方程(组)的求解	167
5.1	一元全区间不等距插值	107	7.1	全区间积分的定步长欧拉方法	167
5.2	一元全区间等距插值	108	7.2	积分一步的变步长欧拉方法	169
5.3	一元三点不等距插值	110	7.3	全区间积分的定步长维梯方法	172
5.4	一元三点等距插值	111	7.4	全区间积分的定步长龙格-库塔法	174
5.5	连分式不等距插值	112	7.5	积分一步的变步长龙格-库塔法	177
5.6	连分式等距插值	114	7.6	积分一步的变步长基尔方法	179
5.7	埃尔米特不等距插值	115	7.7	全区间积分的变步长基尔方法	182
5.8	埃尔米特等距插值	117	7.8	全区间积分的变步长默森方法	184
5.9	埃特金不等距逐步插值	118	7.9	积分一步的连分式法	187
5.10	埃特金等距逐步插值	120	7.10	全区间积分的连分式法	190
5.11	光滑不等距插值	121	7.11	全区间积分的双边法	192
5.12	光滑等距插值	123	7.12	全区间积分的阿当姆斯预报-校正法	195
5.13	第一种边界条件的三次样条函数插值、微商与积分	125	7.13	全区间积分的哈明方法	197
5.14	第二种边界条件的三次样条函数插值、微商与积分	128	7.14	积分一步的特雷纳方法	200
5.15	第三种边界条件的三次样条函数插值、微商与积分	131	7.15	全区间积分的特雷纳方法	204
5.16	二元三点插值	135	7.16	积分刚性方程组的吉尔方法	206
5.17	二元全区间插值	136	7.17	二阶微分方程边值问题的数值解法	213
第6章	数值积分	139	第8章	拟合与逼近	217
6.1	变步长梯形求积法	139	8.1	最小二乘曲线拟合	217
6.2	变步长辛卜生求积法	140	8.2	切比雪夫曲线拟合	220
6.3	自适应梯形求积法	142	8.3	最佳一致逼近的里米兹方法	222
6.4	龙贝格求积法	143	8.4	矩形域的最小二乘曲面拟合	225
6.5	计算一维积分的连分式法	145	第9章	数据处理与回归分析	229
6.6	高振荡函数求积法	147	9.1	随机样本分析	229
6.7	勒让德-高斯求积法	150	9.2	一元线性回归分析	231
6.8	拉盖尔-高斯求积法	152	9.3	多元线性回归分析	233
6.9	埃尔米特-高斯求积法	154	9.4	逐步回归分析	236
			9.5	半对数数据相关	245
			9.6	对数数据相关	247
			第10章	极值问题	250
			10.1	一维极值连分式法	250
			10.2	n 维极值连分式法	252

10.3	不等式约束线性规划问题	254			机数	317
10.4	求 n 维极值的单形调优法	258	13.2	0 到 1 之间均匀分布的随机数		
10.5	求约束条件下 n 维极值的复形调			序列	318	
	优法	261	13.3	任意区间内均匀分布的一个随机		
第 11 章	数学变换与滤波	266		整数	319	
11.1	傅里叶级数逼近	266	13.4	任意区间内均匀分布的随机整数		
11.2	快速傅里叶变换	268		序列	320	
11.3	快速沃什变换	273	13.5	任意均值与方差的一个正态分布		
11.4	五点三次平滑	275		随机数	321	
11.5	离散随机线性系统的卡尔曼		13.6	任意均值与方差的正态分布随机		
	滤波	277		数序列	322	
11.6	$\alpha\beta\gamma$ 滤波	281	第 14 章	多项式与连分式函数的		
第 12 章	特殊函数	285		计算	324	
12.1	伽马函数	285	14.1	一维多项式求值	324	
12.2	不完全伽马函数	287	14.2	一维多项式多组求值	325	
12.3	误差函数	288	14.3	二维多项式求值	327	
12.4	第一类整数阶贝塞耳函数	290	14.4	复系数多项式求值	329	
12.5	第二类整数阶贝塞耳函数	293	14.5	多项式相乘	330	
12.6	变型第一类整数阶贝塞耳函数	296	14.6	多项式相除	332	
12.7	变型第二类整数阶贝塞耳函数	299	14.7	复系数多项式相乘	333	
12.8	不完全贝塔函数	301	14.8	复系数多项式相除	335	
12.9	正态分布函数	304	14.9	函数连分式的计算	336	
12.10	t -分布函数	305	第 15 章	复数运算	338	
12.11	χ^2 分布函数	307	15.1	复数乘法	338	
12.12	F -分布函数	308	15.2	复数除法	339	
12.13	正弦积分	309	15.3	复数乘幂	340	
12.14	余弦积分	311	15.4	复数的 N 次方根	341	
12.15	指数积分	312	15.5	复数指数	342	
12.16	第一类椭圆积分	313	15.6	复数对数	343	
12.17	第二类椭圆积分	315	15.7	复数正弦	344	
第 13 章	随机数的产生	317	15.8	复数余弦	345	
13.1	0 到 1 之间均匀分布的一个随		15.9	复数作图	346	

第二篇 非数值计算

第 16 章	排序	350	16.7	字符快速排序	358
16.1	整数冒泡排序	350	16.8	字符串快速排序	359
16.2	实数冒泡排序	351	16.9	整数希尔排序	360
16.3	字符冒泡排序	353	16.10	实数希尔排序	362
16.4	字符串冒泡排序	354	16.11	字符希尔排序	364
16.5	整数快速排序	355	16.12	字符串希尔排序	365
16.6	实数快速排序	357	16.13	整数堆排序	366

16.14	实数堆排序	367	18.9	TVGA 图形模式(5FH, 1024* 768, 16 色)	420
16.15	字符堆排序	369	18.10	TVGA 图形模式(62H, 1024* 768, 256 色)	421
16.16	字符串堆排序	370	第 19 章	基本图形操作	424
16.17	关键字成员为整数的结构排序 ...	371	19.1	直线	424
16.18	关键字成员为实数的结构排序 ...	374	19.2	线段构成的图形	425
16.19	关键字成员为字符的结构排序 ...	376	19.3	虚线	426
16.20	关键字成员为字符串的结构排序	378	19.4	单点划线	428
16.21	磁盘文件排序	380	19.5	双点划线	430
16.22	拓扑分类	383	19.6	坐标轴	431
第 17 章	查找	385	19.7	矩形及其填充	432
17.1	关键字成员为整型、实型及字符的结构体数组的顺序查找	385	19.8	矩形域图形的清除	433
17.2	关键字成员为字符串的结构体数组的顺序查找	387	19.9	矩形域图形的复制	434
17.3	磁盘随机文本文件的顺序查找	389	19.10	矩形域图形的平移	436
17.4	整型有序表的对分查找	391	19.11	圆形域图形的复制	437
17.5	实型有序表的对分查找	393	19.12	圆形域图形的平移	438
17.6	字符串有序表的对分查找	395	19.13	由中心、半轴(半径)以及起终点夹角画椭圆(圆) 弧	439
17.7	整型、实型及字符型关键字成员有序的结构体数组的对分查找	396	19.14	由中心、半轴(半径)以及起终点夹角画扇形	441
17.8	字符串关键字成员有序的结构体数组的对分查找	399	19.15	由中心与半轴(半径)画椭圆(圆)	442
17.9	按关键字有序的磁盘随机文本文件的对分查找	402	19.16	由中心、半轴(半径)以及起终点夹角画扇形填充	443
17.10	磁盘顺序文本文件的字符串匹配	405	19.17	由中心与半轴(半径)画椭圆(圆) 填充	444
第 18 章	图形模式下读写屏幕象点	407	19.18	抛物线	445
18.1	设置显示模式	407	19.19	双曲线	447
18.2	CGA 图形模式(04H, 320* 200, 4 色)	407	19.20	三次多项式曲线	449
18.3	EGA 图形模式(10H, 640* 350, 16 色)	409	19.21	一般函数曲线	451
18.4	VGA 图形模式(12H, 640* 480, 16 色)	413	19.22	矩形域三维图形透视图	452
18.5	VGA 图形模式(13H, 320* 200, 256 色)	415	19.23	矩形域三维图形平行投影	454
18.6	TVGA 图形模式(5BH, 800* 600, 16 色)	416	19.24	圆形域三维图形平行投影	456
18.7	TVGA 图形模式(5DH, 640* 480, 256 色)	417	第 20 章	汉字操作	458
18.8	TVGA 图形模式(5EH, 800* 600, 256 色)	419	20.1	小汉字库的建立	458
			20.2	ASCII 码字符图形库的建立	461
			20.3	一个汉字的显示	462
			20.4	一行汉字的显示	463
			20.5	ASCII 码字符串按图形显示	464
			20.6	汉字菜单的显示	466
			20.7	汉字菜单的选择	467
			参考文献		470

第一篇 数值计算

第 1 章 线性代数方程组的求解

1.1 全选主元高斯消去法

一、功能

用全选主元高斯(Gauss)消去法求解 n 阶线性代数方程组 $AX = B$ 。其中

$$A = \begin{pmatrix} a_{00} & a_{01} & \dots & a_{0,n-1} \\ a_{10} & a_{11} & \dots & a_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \dots & a_{n-1,n-1} \end{pmatrix}, X = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix}, B = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix}$$

二、方法说明

为保证数值计算的稳定性, 本函数采用全选主元。

全选主元高斯消去法分两步进行。

第一步 消去过程

对于 k 从 0 到 $n-2$ 作以下三步:

(1) 从系数矩阵 A 的第 k 行、第 k 列开始的右下角子阵中选取绝对值最大的元素, 并通过行交换与列交换将它交换到主元素的位置上。

(2) 归一化

$$\begin{aligned} a_{kj} &= a_{kj} / a_{kk}, \quad j = k+1, \dots, n-1 \\ b_k &= b_k / a_{kk} \end{aligned}$$

(3) 消去

$$\begin{aligned} a_{ij} &= a_{ij} - a_{ik}a_{kj}, \quad i, j = k+1, \dots, n-1 \\ b_i &= b_i - a_{ik}b_k, \quad i = k+1, \dots, n-1 \end{aligned}$$

第二步 回代过程

$$(1) \quad x_{n-1} = b_{n-1} / a_{n-1,n-1}$$

$$(2) \quad b_i = b_i - \sum_{j=i+1}^{n-1} a_{ij}x_j, \quad i = n-2, \dots, 1, 0$$

最后对解向量中的元素顺序进行调整。

三、函数语句

int agaus (a, b, n)

本函数返回一个整型标志值。若返回的标志值为 0, 则表示原方程组的系数矩阵奇异, 输出信息“ fail ”; 若返回的标志值不为 0, 则表示正常返回。

四、形参说明

- a——双精度实型二维数组, 体积为 $m \times n$ 。存放方程组的系数矩阵, 返回时将被破坏。
- b——双精度实型一维数组, 长度为 n 。存放方程组右端的常数向量; 返回方程组的解向量。
- n——整型变量。存放方程组的阶数。

五、函数程序(文件名: agaus.c)

六、例

求解下列 4 阶方程组

$$\begin{aligned} 0.2368x_0 + 0.2471x_1 + 0.2568x_2 + 1.2671x_3 &= 1.8471 \\ 0.1968x_0 + 0.2071x_1 + 1.2168x_2 + 0.2271x_3 &= 1.7471 \\ 0.1581x_0 + 1.1675x_1 + 0.1768x_2 + 0.1871x_3 &= 1.6471 \\ 1.1161x_0 + 0.1254x_1 + 0.1397x_2 + 0.1490x_3 &= 1.5471 \end{aligned}$$

主函数程序(文件名: agaus 0. c) 如下:

```
# include "stdio.h "
# include "agaus.c "
main()
{ int i;
  static double a[4][4]=
    { {0.2368, 0.2471, 0.2568, 1.2671},
      {0.1968, 0.2071, 1.2168, 0.2271},
      {0.1581, 1.1675, 0.1768, 0.1871},
      {1.1161, 0.1254, 0.1397, 0.1490} };
  static double b[4]= {1.8471, 1.7471, 1.6471, 1.5471};
  if ( agaus(a, b, 4) != 0)
    for (i= 0; i<= 3; i++ )
      printf ( "%d = %e\n ", i, b[i] );
}
```

运行结果为:

```
x( 0) = 1.04058e+ 00
x( 1) = 9.87051e- 01
x( 2) = 9.35040e- 01
x( 3) = 8.81282e- 01
```

1.2 全选主元高斯-约当消去法

一、功能

用全选主元高斯-约当(Gauss-Jordan)消去法同时求解系数矩阵相同而右端具有 m 组常数向量的 n 阶线性代数方程组 $AX = B$ 。其中

$$A = \begin{pmatrix} a_{00} & a_{01} & \dots & a_{0,n-1} \\ a_{10} & a_{11} & \dots & a_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \dots & a_{n-1,n-1} \end{pmatrix}, X = \begin{pmatrix} x_{00} & x_{01} & \dots & x_{0,m-1} \\ x_{10} & x_{11} & \dots & x_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-1,0} & x_{n-1,1} & \dots & x_{n-1,m-1} \end{pmatrix}$$

$$B = \begin{pmatrix} b_{00} & b_{01} & \dots & b_{0,m-1} \\ b_{10} & b_{11} & \dots & b_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n-1,0} & b_{n-1,1} & \dots & b_{n-1,m-1} \end{pmatrix}$$

二、方法说明

全选主元高斯-约当法的步骤如下。

对于 k 从 0 到 $n-1$ 作如下变换：

(1) 全选主元

从系数矩阵 A 的第 k 行、第 k 列开始的右下角子阵中选取绝对值最大的元素，并通过行交换与列交换将它交换到主元素的位置上。

(2) 归一化

$$a_{kj} / a_{kk} = a_{kj}, \quad j = k+1, \dots, n-1$$

$$b_{kj} / a_{kk} = b_{kj}, \quad j = 0, 1, \dots, m-1$$

(3) 消去

$$a_{ij} = a_{ij} - a_{ik}a_{kj} = a_{ij}, \quad j = k+1, \dots, n-1$$

$$i = 0, 1, \dots, n-1, i \neq k$$

$$b_{ij} = a_{ik}b_{kj} = b_{ij}, \quad j = 0, 1, \dots, m-1$$

$$i = 0, 1, \dots, n-1, i \neq k$$

最后对 B 进行恢复，则 B 中的每一列即为每一个方程组的解。

三、函数语句

`int agjdn (a, b, n, m)`

本函数返回一个整型标志值。若返回的标志值为 0，则表示系数矩阵为奇异，输出信息“fail”；若返回的标志值不为 0，则表示正常返回。

四、形参说明

a ——双精度实型二维数组，体积为 $n \times n$ 。存放方程组的系数矩阵，在本函数中要被破坏。

b ——双精度实型二维数组，体积为 $n \times m$ 。存放方程组右端的 m 组常数向量（一列为一组）；返回时存放 m 组解向量（一列为一组）。

n ——整型变量。方程组的阶数。

m ——整型变量。方程组右端常数向量的个数。

五、函数程序(文件名: agj dn. c)

六、例

求解 4 阶方程组 $AX = B$, 其中

$$A = \begin{pmatrix} 1 & 3 & 2 & 13 \\ 7 & 2 & 1 & -2 \\ 9 & 15 & 3 & -2 \\ -2 & -2 & 11 & 5 \end{pmatrix}, B = \begin{pmatrix} 9 & 0 \\ 6 & 4 \\ 11 & 7 \\ -2 & -1 \end{pmatrix}$$

主函数程序(文件名: agjdn 0. c) 如下:

```
# include "stdio.h"
# include "agjdn.c"
main()
{ int i;
  static double a[4][4]= { {1.0, 3.0, 2.0, 13.0},
                           {7.0, 2.0, 1.0, - 2.0},
                           {9.0, 15.0, 3.0, - 2.0},
                           {- 2.0, - 2.0, 11.0, 5.0}};
  static double b[4][2]= { {9.0, 0.0}, {6.0, 4.0},
                           {11.0, 7.0}, {- 2.0, - 1.0}};
  if (agjdn(a, b, 4, 2)! = 0)
    for (i= 0; i<= 3; i++ )
      printf( "x( %d)= %13. 7e, %13. 7e\n", i, b[i][ 0], b[i][ 1]);
}
```

运行结果为:

```
x( 0)= 9. 807447e- 01, 4. 979313e- 01
x( 1)= 2. 679822e- 01, 1. 444940e- 01
x( 2)= - 2. 226289e- 01, 6. 285805e- 02
x( 3)= 5. 892743e- 01, - 8. 131763e- 02
```

1. 3 复系数方程组的全选主元高斯消去法

一、功能

用全选主元高斯(Gauss) 消去法求解 n 阶复系数线性代数方程组 $AX = B$ 。其中

$$\begin{aligned} A &= A_R + jA_I \\ B &= B_R + jB_I \\ X &= X_R + jX_I \end{aligned}$$

二、方法说明

同 1. 1 节的说明, 但所有运算为复数运算。

为了计算两个复数的乘积 $e+ jf = (a+ jb)(c+ jd)$, 本函数采用如下算法:

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

$$e = p - q, f = s - p - q$$

为了计算两个复数的商 $e + jf = (c + jd)/(a + jb)$, 本函数采用如下算法:

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

$$e = (p - q)/w, f = (s - p - q)/w$$

三、函数语句

int acgas(ar, ai, n, br, bi)

本函数返回一个整型标志值。若返回的标志值为 0, 则表示系数矩阵奇异, 输出信息“err
* * fail”; 若返回的标志值不为 0, 则表示正常返回。

四、形参说明

ar——双精度实型二维数组, 体积为 $m \times n$ 。存放复系数矩阵的实部, 在本函数中要被破坏。

ai——双精度实型二维数组, 体积为 $m \times n$ 。存放复系数矩阵的虚部, 在本函数中要被破坏。

n——整型变量。方程组的阶数。

br——双精度实型一维数组, 长度为 n。存放方程组右端复常数向量的实部; 返回时存放方程组解向量的实部。

bi——双精度实型一维数组, 长度为 n。存放方程组右端复常数向量的虚部; 返回时存放方程组解向量的虚部。

五、函数程序(文件名: acgas.c)

六、例

求解 4 阶复系数方程组 $AX = B$ 。其中

$$A = AR + jAI, B = BR + jBI$$

$$AR = \begin{pmatrix} 1 & 3 & 2 & 13 \\ 7 & 2 & 1 & -2 \\ 9 & 15 & 3 & -2 \\ -2 & -2 & 11 & 5 \end{pmatrix}, AI = \begin{pmatrix} 3 & -2 & 1 & 6 \\ -2 & 7 & 5 & 8 \\ 9 & -3 & 15 & 1 \\ -2 & -2 & 7 & 6 \end{pmatrix}$$

$$BR = (2, 7, 3, 9)^T, BI = (1, 2, -2, 3)^T$$

主函数程序(文件名: acgas 0.c)如下:

```
# include "stdio.h"
# include "acgas.c"
main ()
{ int i;
  static double ar[4][4] = { {1.0, 3.0, 2.0, 13.0},
                              {7.0, 2.0, 1.0, -2.0},
                              {9.0, 15.0, 3.0, -2.0},
                              {-2.0, -2.0, 11.0, 5.0}};
  static double ai[4][4] = { {3.0, -2.0, 1.0, 6.0},
                              {-2.0, 7.0, 5.0, 8.0},
```

```

        {9.0, - 3.0, 15.0, 1.0},
        {- 2.0, - 2.0, 7.0, 6.0} };
static double br[4]= {2.0, 7.0, 3.0, 9.0};
static double bi[4]= {1.0, 2.0, - 2.0, 3.0};
if (acgas(ar, ai, 4, br, bi) != 0)
    for (i= 0; i<= 3; i++ )
        printf( "b( %d)= %13.7e + j %13.7e\n ", i, br[i], bi[i]);
}

```

运行结果为:

```

b(0)= 6.782330e- 02      + j      7.078231e- 02
b(1)= - 1.623413e- 01    + j      - 7.612936e- 01
b(2)= 5.985239e- 01      + j      - 4.371312e- 01
b(3)= 2.464562e- 01      + j      1.139959e- 01

```

1.4 复系数方程组的全选主元高斯-约当消去法

一、功能

用全选主元高斯-约当(Gauss-Jordan)消去法求解右端具有 m 组常数向量的 n 阶复系数线性代数方程组 $AX = B$ 。其中

$$\begin{aligned} A &= A_R + jA_I \\ B &= B_R + jB_I \\ X &= X_R + jX_I \end{aligned}$$

二、方法说明

同 1.2 节的说明,但所有运算均为复数运算。

为了计算两个复数的乘积 $e + jf = (a + jb)(c + jd)$, 本函数采用如下算法:

$$\begin{aligned} p &= ac, q = bd, s = (a + b)(c + d) \\ e &= p - q, f = s - p - q \end{aligned}$$

为了计算两个复数的商 $e + jf = (c + jd)/(a + jb)$, 本函数采用如下算法:

$$\begin{aligned} p &= ac, q = -bd, s = (a - b)(c + d), w = a^2 + b^2 \\ e &= (p - q)/w, f = (s - p - q)/w \end{aligned}$$

三、函数语句

int acjdn(ar, ai, br, bi, n, m)

本函数返回一个整型标志值。若返回的标志值为 0, 则表示系数矩阵奇异, 输出信息“err
* * fail”; 若返回的标志值不为 0, 则表示正常返回。

四、形参说明

ar——双精度实型二维数组, 体积为 $n \times n$ 。存放复系数矩阵的实部, 在本函数中要被破坏。

ai——双精度实型二维数组, 体积为 $n \times n$ 。存放复系数矩阵的虚部, 在本函数中要被破坏。

br——双精度实型二维数组, 体积为 $n \times m$ 。存放方程组右端 m 组复常数向量的实部; 返回时存放 m 组解向量的实部。

bi——双精度实型二维数组, 体积为 $n \times m$ 。存放方程组右端 m 组复常数向量的虚部; 返回时存放 m 组解向量的虚部。

n——整型变量, 方程组的阶数。

m——整型变量。方程组右端复常数向量的个数。

五、函数程序(文件名: acjdn.c)

六、例

求解 4 阶复系数方程组 $AX = B$ 。其中

$$A = AR + jAI, B = BR + jBI$$

$$AR = \begin{bmatrix} 1 & 3 & 2 & 13 \\ 7 & 2 & 1 & -2 \\ 9 & 15 & 3 & -2 \\ -2 & -2 & 11 & 5 \end{bmatrix}, AI = \begin{bmatrix} 3 & -2 & 1 & 6 \\ -2 & 7 & 5 & 8 \\ 9 & -3 & 15 & 1 \\ -2 & -2 & 7 & 6 \end{bmatrix}$$

$$BR = \begin{bmatrix} 2 & -2 \\ 7 & 3 \\ 3 & 2 \\ 9 & 1 \end{bmatrix}, BI = \begin{bmatrix} 1 & 3 \\ 2 & 7 \\ -2 & 9 \\ 3 & 2 \end{bmatrix}$$

主函数程序(文件名: acjdn 0.c) 如下:

```
# include "stdio.h"
# include "acjdn.c"
main ()
{ int i;
  static double ar[4][4]= { {1.0, 3.0, 2.0, 13.0},
                             {7.0, 2.0, 1.0, - 2.0},
                             {9.0, 15.0, 3.0, - 2.0},
                             {- 2.0, - 2.0, 11.0, 5.0}};
  static double ai[4][4]= { {3.0, - 2.0, 1.0, 6.0},
                             {- 2.0, 7.0, 5.0, 8.0},
                             {9.0, - 3.0, 15.0, 1.0},
                             {- 2.0, - 2.0, 7.0, 6.0}};
  static double br[4][2]= { {2.0, - 2.0}, {7.0, 3.0},
                             {3.0, 2.0}, {9.0, 1.0}};
  static double bi[4][2]= { {1.0, 3.0}, {2.0, 7.0},
                             {- 2.0, 9.0}, {3.0, 2.0}};
  if (acjdn(ar, ai, br, bi, 4, 2)! = 0)
    for (i= 0; i<= 3; i+ + )
      printf( "x(%d)= %13.7e+ j %13.7e, %13.7e + j %13.7e\n",
              i, br[i][0], bi[i][0], br[i][1], bi[i][1]);
}
```

运行结果为:

$$\begin{aligned} x(0) &= 6.782330e-02 + j \quad 7.078231e-02, \quad 2.511789e-01 + j \quad 5.810123e-01 \\ x(1) &= -1.623413e-01 + j \quad -7.612936e-01, \quad 4.024521e-01 + j \quad -1.436012e-01 \\ x(2) &= 5.985239e-01 + j \quad -4.371312e-01, \quad 3.356590e-01 + j \quad 1.034377e-01 \\ x(3) &= 2.464562e-01 + j \quad 1.139959e-01, \quad -5.755388e-02 + j \quad 2.079959e-01 \end{aligned}$$

1.5 求解三对角线方程组的追赶法

一、功能

用追赶法求解 n 阶三对角线方程组 $AX = D$ 。其中

$$A = \begin{pmatrix} a_{00} & a_{01} & & & \\ a_{10} & a_{11} & a_{12} & & \\ & a_{21} & a_{22} & a_{23} & \\ & & w & w & w \\ & 0 & & a_{n-2,n-3} & a_{n-2,n-2} & a_{n-2,n-1} \\ & & & & a_{n-1,n-2} & a_{n-1,n-1} \end{pmatrix}$$
$$X = (x_0, x_1, x_2, \dots, x_{n-2}, x_{n-1})^T$$
$$D = (d_0, d_1, d_2, \dots, d_{n-2}, d_{n-1})^T$$

二、方法说明

追赶法的本质是没有选主元的高斯(Gauss)消去法,只是在计算过程中考虑了三对角线矩阵的特点,对于绝大部分的零元素不再作处理。其步骤如下。

(1) 对于 $k=0, 1, \dots, n-2$ 作归一化及消元处理。即

$$\begin{aligned} a_{k,k+1}/a_{kk} &= a_{k,k+1} \\ d_k/a_{kk} &= d_k \\ a_{k+1,k+1} - a_{k+1,k} a_{k,k+1} &= a_{k+1,k+1} \\ d_{k+1} - a_{k+1,k} d_k &= d_{k+1} \end{aligned}$$

(2) 进行回代。即

$$\begin{aligned} d_{n-1}/a_{n-1,n-1} &= x_{n-1} \\ d_k - a_{k,k+1} x_{k+1} &= x_k, \quad k = n-2, \dots, 1, 0 \end{aligned}$$

在本函数中,为便于三对角线矩阵的输入,用一个一维数组 $B(0:3n-3)$ 以行为主存放三对角线矩阵 A 中的三条对角线上的非零元素。其中 n 为矩阵 A 的阶数。显然,三对角线矩阵 A 与一维数组 B 之间有如下关系:

$$A(i,j) = \begin{cases} B(2i+j), & i-1 \leq j \leq i+1 \\ 0, & \text{其它} \end{cases}$$

其中 $i, j=0, 1, \dots, n-1$ 。

根据以上关系,可以得到相应的计算过程为:

(1) 对于 $k=0, 1, \dots, n-2$ 作归一化及消元处理