

FORTRAN 77

科学计算子程序汇编

谭浩强 龙耀庭 主编

清华大学出版社

内 容 简 介

本书是为科学计算的用户而编写的。共提供了 100 多个最经常使用的、最基本的子程序。内容包括：实矩阵的运算、复矩阵的运算、多项式的计算、插值法、微分法、数值积分、方程求解、方程组求解、微分方程求解、函数计算、概率统计等。

本书适合 FORTRAN 初学者及程序设计人员。

(京)新登字 158 号

FORTRAN 77 科学计算子程序汇编

谭浩强 龙耀庭 主编

☆

清华大学出版社出版

北京 清华园

人民交通出版社印刷厂印刷

新华书店总店科技发行所发行

☆

开本：787×1092 1/16 印张：14.75 字数：356 千字

1993 年 7 月第 1 版 1993 年 7 月第 1 次印刷

印数：0001—6000

ISBN 7-302-01165-6/TP·431

定价：10.50 元

前 言

在使用 FORTRAN 语言编写程序时,常常需要调用函数和子程序以实现各种功能。随 FORTRAN 编译系统提供给用户使用的标准函数只能用来完成一些简单的功能(例如用 SQRT 函数可以计算平方根)。在完成一些较复杂的功能时,用户需自己选择算法,并设计出子程序。对用户来说,这无疑是一个沉重的负担,尤其对许多初学者,将会遇到许多困难。有一些计算机厂家提供“子程序库”,把常用的若干子程序以库的形式存放在磁盘或磁带上出售给用户使用。但一般不附源程序,用户可以使用它,但难于理解它,如果想对它作某些修改就更困难了。

许多 FORTRAN 用户不仅希望能使用这些子程序,还希望能分析它们,理解这些子程序的算法、思路、编程方法和技巧,以便在需要时可以对它们进行修改,并且可以参照它们举一反三,编写出所需要的其它子程序。本书就是根据这一要求而编写的。

FORTRAN 的主要应用领域是科学计算(即数值计算),因此,本书是为科学计算的用户而编写的。本书提供了一百多个最经常使用的、最基本的子程序。内容包括:实矩阵的运算、复矩阵的运算、多项式的计算、插值法、微分法、数值积分、方程求解、方程组求解、微分方程求解、函数计算、概率、数理统计与随机过程、实用程序与绘图等 14 个方面。

本书的出发点有两个方面:一是使用户能方便地选用所需的子程序,减少编程中的困难;二是向 FORTRAN 的初学者提供如何编写子程序的学习参考材料。据我们了解,许多初学者在学习 FORTRAN 语言之后,编写一些实用程序时往往遇到一些困难,感到无从下手,对编写的程序考虑不周,程序难以付诸实用。也就是说,还缺乏编程的经验。教材中所介绍的往往是一些较简单的例子,程序也较短。因此,本书可以作为 FORTRAN 初学者与 FORTRAN 程序设计人员之间的一个桥梁,它可以帮助初学者提高程序设计水平。

本书每一节中包含科学计算的一个问题。每一问题中包括以下几部分:

1. 本子程序的用途。
2. 在使用该子程序时需要调用的本书中其它子程序的有关说明。
3. 子程序中用到的形式参数的顺序、含义和作用。
4. 子程序所依据的数学方法。
5. 列出该子程序的源程序。
6. 给出调用该子程序的“测试程序”。它的作用是用来验证子程序是否正确,以及告诉读者正确调用该子程序的方法。读者如需在自己的程序中调用该子程序,可以参照“测试程序”中的方法。
7. 用“测试程序”调用该子程序时,运行所得的结果,读者可以通过对运行结果的分析更好地理解该子程序。

本书中的所有子程序和测试程序都已在 PC 机上运行通过。

本书可以作为 FORTRAN 用户使用子程序库时参考,也可作为学习“FORTRAN 语

言程序设计”课程的参考书。本书中介绍的子程序可以作为补充例题在课堂上讲授。

应当说明,由于篇幅所限,本书不可能提供用户所需要的全部子程序,而只能提供一些典型的子程序。为了使本书能对学习 FORTRAN 语言的不同层次的读者有所帮助,因此选择了程度深浅不同的子程序,读者可以由浅入深地进行学习。考虑到篇幅,每个子程序只介绍其数学方法,而没有详细介绍算法和画出流程图,读者可以自己理解和消化。此外,本书提供的程序并不是实现该算法唯一的程序,甚至不一定是最佳的程序,读者完全可能编写出更好的程序。

参加本书编写和程序调试工作的有:谭浩强、龙耀庭、杨思伟、薛淑斌、杨丽华、郭焕萍。在编写过程中参考了国内外一些有关书籍和材料。葛玉安副教授对本书所选用的数学方法进行了核对,提出了不少宝贵意见,谨此表示感谢。

有不妥之处,敬请批评指正。

编 者

1991

目 录

第一章 实矩阵	1
1.1 矩阵传送(MAMOVE)	1
1.2 矩阵交换(MASWCH)	2
1.3 矩阵转置(MATTRS)	4
1.4 对矩阵中行和列的运算(MAELEM)	6
1.5 向量的内积(VEPROD)	13
1.6 向量的叉积(VCRSPR)	14
1.7 矩阵的迹(MATRCE)	14
1.8 矩阵相乘(MATMUL)	15
1.9 线性矩阵算术运算(MARITH)	17
1.10 特征值和特征向量(MATEIG)	21
第二章 复矩阵	25
2.1 复矩阵的传送(CMAMOV)	25
2.2 复矩阵的交换(CMASWC)	26
2.3 复矩阵的迹(TRSQMT)	28
2.4 复矩阵相乘(CHAMUL)	30
2.5 复矩阵的算术运算(CMAARI)	32
第三章 多项式	36
3.1 多项式的计算(PLYEVL)	36
3.2 复系数多项式的计算(CXPOEV)	37
3.3 多项式的加法(PLYADD)	38
3.4 多项式的乘法(PLYMUL)	40
3.5 复系数多项式的乘法(CXPLYM)	41
3.6 多项式的除法(PLYDIV)	43
3.7 变量的线性变化(LINCNG)	47
3.8 最大公因数(PLYCFC)	48
3.9 部分分式法(PARFRA)	51
第四章 插值法	54
4.1 差分法(NAORDR)	54
4.2 Aitken 法(NAAITK)	55
4.3 Lagrangian 法(NALAGR)	57
第五章 微分法与数值积分	59
5.1 多项式微分(DLYDIF)	59

• II •

5.2	多项式的积分(PLYINT)	60
5.3	梯形法(TPZOD)	61
5.4	1/3 Simpson 法(INTEGS)	62
5.5	3/8 Simpson 法(SIMPS)	63
5.6	Adams-Bashforth 法(ADMBAS)	65
第六章	方程的解法	67
6.1	二次方程的根(QUADRT)	67
6.2	二元二次方程组的根(SIMSEC)	68
6.3	三次方程的根(CUBIC)	72
6.4	四次方程的根(BIQUAD)	74
6.5	用 Newton 法求多项式的根(PLYIRT)	77
6.6	用 Bairstows 法求多项式的根(NABAIR)	78
6.7	用试位法求根(NAREGU)	81
6.8	用 Mullers 法求根(NAMULL)	83
第七章	方程组的解法	88
7.1	矩阵求逆、方程组和行列式(MATINV)	88
7.2	CROUT 方法(CROUT)	92
7.3	N 个未知数 N 个方程的求解(YENBOB)	96
7.4	N 个未知数 M 个方程的求解(RANK)	97
第八章	微分方程的解法	104
8.1	解一阶微分方程的 Runge-Kutta 法(DEFORK)	104
8.2	解一阶微分方程的 Adams 法(DEFOAD)	105
8.3	解一阶微分方程改进的 Adams 法(DEFOMA)	107
8.4	解一阶微分方程的 Hamming 法(DEFOHA)	109
8.5	解二阶微分方程的 Runge-Kutta 法(DESORK)	111
8.6	解二阶微分方程的 Adams 法(DESOAD)	114
8.7	解二阶微分方程改进的 Adams 法(DESOMA)	116
8.8	解二阶微分方程的 Hamming 法(DESOHA)	118
8.9	解三阶微分方程的 Runge-Kutta 法(DETORK)	121
8.10	解三阶微分方程的 Adams 法(DEFOAD)	124
8.11	解三阶微分方程的改进的 Adams 法(DEFOMA)	126
8.12	解三阶微分方程的 Hamming 法(DESOHA)	129
第九章	函数值的计算	133
9.1	复数的 N 次方根(NTRTCX)	133
9.2	复数的乘幂(CXPOWR)	134
9.3	复数的正弦(CXSINE)	135
9.4	复数指数(CXEXPN)	136
9.5	复数对数(以 e 为底)(CXLOG2)	137

9.6	Gamma 函数 (GAMMA)	137
9.7	BesselJ 函数 (BESSJ)	139
9.8	Legendre 函数 (LGPOL)	141
第十章	数值分析	143
10.1	Chebyshev 多项式的产生 (NACPLY)	143
10.2	Fourier 分析 (FORIE)	144
第十一章	概率	148
11.1	均方差和 T- 比 (STMEAN)	148
11.2	几何均值和标准差 (STGEOM)	151
11.3	随机数发生器 (STRNUM)	152
11.4	正态随机数发生器 (NORNG)	153
11.5	Poisson 随机数发生器 (PORNG)	155
11.6	中值测试 (STMEDT)	156
11.7	指数趋势 (STEXTN)	158
11.8	正态 (概率) 分布 (NMDT)	160
11.9	二项 (概率) 分布 (STBNPB)	161
11.10	Poisson (概率) 分布 (PODTF)	162
11.11	Gamma (概率) 分布 (STGAMA)	163
11.12	指数 (概率) 分布 (STEXPD)	165
11.13	负二项 (概率) 分布 (STNGBI)	166
11.14	超几何 (概率) 分布 (STHYGD)	167
11.15	相关系数 (STCORR)	169
11.16	积矩相关矩阵 (STCRMT)	170
11.17	2 * 2 表 χ 平方校验 (STCHI2)	173
11.18	M * N 表 χ 平方校验 (STCHIS)	174
11.19	一维制表 (TAB1)	176
11.20	二维制表 (TAB2)	178
第十二章	数理统计与随机过程	183
12.1	线性回归 (STLNRG)	183
12.2	最小二乘回归 (STPLRG)	186
12.3	多重回归 (STMLRG)	188
12.4	一向分类 (方差分析) (STANV1)	190
12.5	随机化的完全区组 (方差分析) (STANVR)	193
12.6	拉丁平方 (方差分析) (STANVL)	196
12.7	希腊-拉丁方 (方差分析) (STANVG)	199
12.8	移动平均 (时序分析) (MOVEAV)	203
12.9	自协方差 (时序分析) (AUTOCO)	206
12.10	季节指数与循环移动 (时序分析) (CYCMOV)	207

12.11 平滑序列(时序分析) (SMOTH)	210
第十三章 实用程序和绘图	213
13.1 简单的线性排序 (SORT)	213
13.2 双线性排序 (SORT2)	214
13.3 极坐标-直角坐标转换 (CVPOLR)	215
13.4 简单的绘图 (GRAPHS)	216
13.5 直方图 (HIST)	219
附录 如何使用子程序库软盘	224
参考文献	226

第一章 实 矩 阵

1.1 矩阵传送(MAMOVE)

【用途】 将一个矩阵(或一个向量)传送到另一个矩阵(或向量)。

【子程序名和形参】 MAMOVE (LRA,LRB,NROW,NCOL,A,B)

形参名	用途	说 明
LRA	输入	在主程序中用 DIMENSION 语句定义的 A 数组作为二维数组时的行数
LRB	输入	在主程序中用 DIMENSION 语句定义的 B 数组作为二维数组时的行数
NROW	输入	矩阵 A 中实际的行数
NCOL	输入	矩阵 A 中实际的列数
A	输入	被传送的矩阵或向量
B	输出	传送后的矩阵或向量

【说明】 如果为一维数组,设 LRA=LRB=1。

本子程序可以用于对二维数组(矩阵)或一维数组(向量)的传送,如果要传送的是二维数组,则在调用子程序 MAMOVE 时应给出矩阵的行数(NROW)和列数(NCOL)。LRA 和 LRB 是矩阵 A 和 B 的行数。在 MAMOVE 子程序中,逐行将矩阵 A 的元素值传送给矩阵 B 的相应行。如果要传送的 A 是一维数组,则实参 LRA、LRB 和 NROW 均应定为 1。

【程序设计方法】

在程序中的输入是 A 数组:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

输出为:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad B = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

【子程序】

```

SUBROUTINE MAMOVE(LRA,LRB,NROW,NCOL,A,B)
C   THIS SUBROUTINE MOVES A MATRIX OR A VECTOR TO ANOTHER AREA.
    DIMENSION A(1), B(1)
    DO 20 I = 1,NROW
      IJA = I
      IJB = I
      DO 10 J = 1,NCOL

```

```

        B(IJB) = A(IJA)
        IJA = IJA + LRA
        IJB = IJB + LRB
10      CONTINUE
20      CONTINUE
        RETURN
        END

```

【测试程序】

```

        DIMENSION A(30),B(30)
        DATA A/1.,2.,3.,4.,5.,2.,4.,6.,8.,10.,3.,6.,9.,12.,15.,
&4.,8.,12.,16.,20.,5.,10.,15.,20.,25.,6.,12.,18.,24.,30./
C      ACTUAL FORM OF BASIC ARRAY:
C
C          1   2   3   4   5   6
C          2   4   6   8  10  12
C          3   6   9  12  15  18
C          4   8  12  16  20  24
C          5  10  15  20  25  30
C

```

```

        LRA=5
        LRB=5
        NROW=5
        NCOL=6
        NE=NROW * NCOL
        CALL MAMOVE(LRA,LRB,NROW,NCOL,A,B)
        CALL PRINT(B,NROW,NCOL)
        WRITE (*,*) ''
        END
        SUBROUTINE PRINT(AA,NR,NC)
        DIMENSION AA(1)
        NE=(NC * NR)-NC+1
        DO 20 J=1,NR
        WRITE (*,100) (AA(K),K=J,J+NE,NR)
100      FORMAT (1X,6(F7.1))
20      CONTINUE
        WRITE (*,*) ''
        RETURN
        END

```

【试运行结果】

1.0	2.0	3.0	4.0	5.0	6.0
2.0	4.0	6.0	8.0	10.0	12.0
3.0	6.0	9.0	12.0	15.0	18.0
4.0	8.0	12.0	16.0	20.0	24.0
5.0	10.0	15.0	20.0	25.0	30.0

1.2 矩阵交换(MASWCH)

【用途】 将矩阵或向量进行交换。

【子程序名和形参】 MASWCH (LRA, LRB,NROW,NCOL,A,B)

形参名	用途	说 明
LRA	输入	在主程序中用 DIMENSION 语句定义的数组 A 作为二维数组时的行数
LRB	输入	在主程序中用 DIMENSION 语句定义的数组 B 作为二维数组时的行数
NROW	输入	当前矩阵传递给子程序的实际行数
NCOL	输入	当前矩阵传递给子程序的实际列数
A,B	输入输出	进行交换的矩阵

【说明】 如果为一维数组, 设 $LRA=LRB=1$ 。

【程序设计方法】

输入:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{bmatrix}$$

输出:

$$A = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{bmatrix} \quad B = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

【子程序】

```

SUBROUTINE MASWCH(LRA,LRB,NROW,NCOL,A,B)
C   THIS SUBROUTINE INTERCHANGES TWO MATRICES OR VECTORS.
    DIMENSION A(1), B(1)
    DO 20 I = 1,NROW
      IJA = I
      IJB = I
      DO 10 J = 1,NCOL
        C = A(IJA)
        A(IJA) = B(IJB)
        B(IJB) = C
        IJA = IJA + LRA
        IJB = IJB + LRB
10     CONTINUE
20     CONTINUE
    RETURN
    END

```

【测试程序】

```

    DIMENSION A(24),B(24)
    DATA A/1.,2.,3.,4.,5.,6.,2.,4.,6.,8.,10.,12.,3.,6.,9.,
& 12.,15.,18.,4.,8.,12.,16.,20.,24./
    DATA B/1.,3.,5.,7.,9.,11.,0.,2.,4.,6.,8.,10.,
& -1.,1.,3.,5.,7.,9.,-2.,0.,2.,4.,6.,8./
C
C   ACTUAL FORM OF ARRAY:
C           A           B
C           1   2   3   4           1   0  -1  -1

```

```

C          2    4    6    8          3    2    1    0
C          3    6    9   12          5    4    3    2
C          4    8   12   16          7    6    5    4
C          5   10   15   20          9    8    7    6
C          6   12   18   24          11   10    9    8
C
C      LRA=6
C      LRB=6
C      NROW=6
C      NCOL=4
C
C      CALL MASWCH(LRA,LRB,NROW,NCOL,A,B)
C      CALL PRINT(A,NROW,NCOL)
C      CALL PRINT(B,NROW,NCOL)
C
C      WRITE (*,*)''
C      END
C
C      SUBROUTINE PRINT(AA,NR,NC)
C
C      DIMENSION AA(1)
C
C      NE=(NC*NR)-NR
C
C      DO 20 J=1,NR
C      WRITE (*,100) (AA(K),K=J,J+NE,NR)
100      FORMAT (1X,4(F7.1))
20      CONTINUE
C      WRITE (*,*)''
C      RETURN
C      END

```

【试运行结果】

```

1.0    .0   -1.0   -2.0
3.0    2.0    1.0    .0
5.0    4.0    3.0    2.0
7.0    6.0    5.0    4.0
9.0    8.0    7.0    6.0
11.0   10.0    9.0    8.0
1.0    2.0    3.0    4.0
2.0    4.0    6.0    8.0
3.0    6.0    9.0   12.0
4.0    8.0   12.0   16.0
5.0   10.0   15.0   20.0
6.0   12.0   18.0   24.0

```

1.3 矩阵转置(MATTRS)

【用途】 对一个 $n \times n$ 的矩阵进行转置。已转置的矩阵送回到主程序中代替原来的

矩阵。

【子程序名和形参】 MATTRS(N,LLR,X)

形参名	用途	说 明
N	输入	矩阵 X 的行数
LLR	输入	主程序中 DIMENSION 语句为矩阵 X 所保留的行数
X	输入	要被转置的矩阵
	输出	转置过的矩阵

【数学方法】

$$X = \begin{bmatrix} X_{11} & X_{12} & X_{13} & \cdots & X_{1n} \\ X_{21} & X_{22} & X_{23} & \cdots & X_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ X_{n1} & X_{n2} & X_{n3} & \cdots & X_{nn} \end{bmatrix} \quad X^T = \begin{bmatrix} X_{11} & X_{21} & \cdots & X_{n1} \\ X_{12} & X_{22} & \cdots & X_{n2} \\ X_{13} & X_{23} & \cdots & X_{n3} \\ \cdots & \cdots & \cdots & \cdots \\ X_{1n} & X_{2n} & \cdots & X_{nn} \end{bmatrix}$$

SUBROUTINE MATTRS(N,LLR,X)

C
C
C

THIS SUBROUTINE TRANSPOSES AN N BY N MATRIX.

```

DIMENSION X(1)
KLLR = 0
NM1 = N - 1
DO 20 I = 1,NM1
  IP1 = I + 1
  IJ = I + KLLR
  DO 10 J = IP1,N
    IJ = IJ + LLR
    JI = KLLR + J
    A = X(JI)
    X(JI) = X(IJ)
    X(IJ) = A
  10 CONTINUE
  KLLR = KLLR + LLR
  20 CONTINUE
  RETURN
END

```

10
20

【测试程序】

```

DIMENSION X(2,2)
DATA X/1.,1.,0.,0./

```

C
C
C
C
C
C

ORIGINAL FORM OF MATRIX A:

```

      1    0
      1    0

```

```

N=2
LLR=2

```

C

CALL MATTRS(N,LLR,X)

```

C      WRITE (*,200) X(1,1),X(1,2)
      WRITE (*,200) X(2,1),X(2,2)
200    FORMAT (1X,2F14.6)
10     CONTINUE
      WRITE (*,*)''
      END

```

【试运行结果】

```

1.000000      1.000000
.000000      .000000

```

1.4 对矩阵中行和列的运算(MAELEM)

【用途】 对矩阵的行或列进行以下基本运算：

1. 用一个常数去乘一行；
2. 对二行进行调换；
3. 一行与一个数相乘后加到另一行上去；
4. 用一个常数去乘一列；
5. 调换二列；
6. 一列与一个数相乘后加到另一列上去。

【子程序名和形参】 MAELEM(LRA,NROW,NCOL,A,I,J)

形参名	用途	说 明
LRA	输入	主程序中 DIMENSION 语句为矩阵所定义的行数
NROW	输入	矩阵的实际行数
NCOL	输入	矩阵的实际列数
A	输入	进行运算之前的矩阵
	输出	进行运算之后的矩阵。
I	输入	当前运算的行
J	输入	当前运算的列
SCALAR	输入	与行或列相乘的系数
IFLAG	输入	选择标志： 1=行 I 与 SCALAR 相乘 2=行 I 与行 J 进行交换 3=SCALAR 与行 J 相乘后加到 I 行上去 4=用 SCALAR 去乘列 I 5=I 列与 J 列进行交换 6=SCALAR 与 J 列相乘后加到 I 列上去

【说明】 任何一个未用的变量名可以作为形参。

【规定】

1. 在主程序中对应形参 SCALAR 的实参为 S。
2. 如果所选择的运算项目不要求乘系数 SCALAR,则可以不设 S 值,或使 S=1。

3. 如果不要求进行列的运算,则不设J值。

【程序设计方法】

IFLAG=1

输入:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

输出:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{i1} * S & a_{i2} * S & \cdots & a_{in} * S \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad S \text{ 是一个常数}$$

IFLAG=2

输入:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i1} & a_{i2} & a_{i3} & \cdots & a_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{j1} & a_{j2} & a_{j3} & \cdots & a_{jn} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

输出:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{j1} & a_{j2} & a_{j3} & \cdots & a_{jn} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i1} & a_{i2} & a_{i3} & \cdots & a_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

IFLAG=3

输入:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

输出:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ S * a_{j1} + a_{i1} & S * a_{j2} + a_{i2} & \cdots & S * a_{jn} + a_{in} \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

IFLAG=4

输入:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1i} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2i} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mi} & \cdots & a_{mn} \end{bmatrix}$$

输出:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & S * a_{1i} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & S * a_{2i} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & S * a_{mi} & \cdots & a_{mn} \end{bmatrix}$$

IFLAG=5

输入:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1i} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2i} & \cdots & a_{2j} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mi} & \cdots & a_{mj} & \cdots & a_{mn} \end{bmatrix}$$

输出:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1i} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2i} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mj} & \cdots & a_{mi} & \cdots & a_{mn} \end{bmatrix}$$

IFLAG=6

输入:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1i} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2i} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mi} & \cdots & a_{mn} \end{bmatrix}$$

输出:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & S * a_{1j} + a_{1i} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & S * a_{2j} + a_{2i} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & S * a_{mj} + a_{mi} & \cdots & a_{mn} \end{bmatrix}$$

【子程序】

```
      SUBROUTINE MAELEM(LRA,NROW,NCOL,A,I,J,SCALAR,IFLAG)
C
C      THIS SUBROUTINE IS USED TO DO SOME MATRIX ELEMENTARY
C      OPERATIONS ON ROWS OR COLUMNS.
C
      DIMENSION A(1)
C      ROW OPERATION
      IF (IFLAG.EQ. 1) THEN
C      MULTIPLY A ROW BY A CONSTANT
      IR = I
      DO 10 II = 1,NCOL
        A(IR) = SCALAR * A(IR)
      IR = IR + LRA
10    CONTINUE
      END IF
      IF (IFLAG.EQ. 2) THEN
C      SWITCH TWO ROWS
      IR = I
      IS = J
      DO 20 II = 1,NCOL
        C = A(IR)
        A(IR) = A(IS)
        A(IS) = C
      IR = IR + LRA
      IS = IS + LRA
20    CONTINUE
      END IF
      IF (IFLAG.EQ. 3) THEN
C      ADD MULTIPLE OF ONE ROW TO ANOTHER
      IR = I
      IS = J
      DO 30 II = 1,NCOL
        A(IR) = A(IR) + SCALAR * A(IS)
      IR = IR + LRA
      IS = IS + LRA
30    CONTINUE
      END IF
      IF (IFLAG.EQ. 7) THEN
C      COLUMN OPERATION
      IR = LRA * (I - 1) + 1
      DO 40 II = 1,NROW
        A(IR) = A(IR) * SCALAR
      IR = IR + 1
40    CONTINUE
      END IF
      IF (IFLAG.EQ. 8) THEN
      IR = LRA * (I - 1) + 1
      IS = LRA * (J - 1) + 1
      DO 50 II = 1,NROW
        C = A(IR)
        A(IR) = A(IS)
```