

NAG Library Routine Document

F07ATF (ZGEEQU)

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

1 Purpose

F07ATF (ZGEEQU) computes real diagonal scaling matrices D_R and D_C intended to equilibrate a complex m by n matrix A and reduce its condition number.

2 Specification

```
SUBROUTINE F07ATF(M, N, A, LDA, R, C, ROWCND, COLCND, AMAX, INFO)
```

```
INTEGER M, N, LDA, INFO
```

```
double precision R(M), C(N), ROWCND, COLCND, AMAX
```

```
complex*16 A(LDA,*)
```

The routine may be called by its LAPACK name *zgeequ*.

3 Description

F07ATF (ZGEEQU) computes the diagonal scaling matrices. The diagonal scaling matrices are chosen to try to make the elements of largest absolute value in each row and column of the matrix B given by

$$B = D_R A D_C$$

have absolute value 1. The diagonal elements of D_R and D_C are restricted to lie in the safe range $(\delta, 1/\delta)$, where δ is the value returned by routine X02AMF. Use of these scaling factors is not guaranteed to reduce the condition number of A but works well in practice.

4 References

None.

5 Parameters

- 1: M – INTEGER *Input*
On entry: m , the number of rows of the matrix A .
Constraint: $M \geq 0$.
- 2: N – INTEGER *Input*
On entry: n , the number of columns of the matrix A .
Constraint: $N \geq 0$.
- 3: A(LDA,*) – *complex*16* array *Input*
Note: the second dimension of the array A must be at least $\max(1, N)$.
On entry: the matrix A whose scaling factors are to be computed.

- 4: LDA – INTEGER *Input*
On entry: the first dimension of the array A as declared in the (sub)program from which F07ATF (ZGEEQU) is called.
Constraint: $LDA \geq \max(1, M)$.
- 5: R(M) – **double precision** array *Output*
On exit: if INFO = 0 or M, R contains the row scale factors, the diagonal elements of D_R . The elements of R will be positive.
- 6: C(N) – **double precision** array *Output*
On exit: if INFO = 0, C contains the column scale factors, the diagonal elements of D_C . The elements of C will be positive.
- 7: ROWCND – **double precision** *Output*
On exit: if INFO = 0 or INFO > M, ROWCND contains the ratio of the smallest value of $R(i)$ to the largest value of $R(i)$. If $ROWCND \geq 0.1$ and AMAX is neither too large nor too small, it is not worth scaling by D_R .
- 8: COLCND – **double precision** *Output*
On exit: if INFO = 0, COLCND contains the ratio of the smallest value of $C(i)$ to the largest value of $C(i)$.
 If $COLCND \geq 0.1$, it is not worth scaling by D_C .
- 9: AMAX – **double precision** *Output*
On exit: $\max |a_{ij}|$. If AMAX is very close to overflow or underflow, the matrix A should be scaled.
- 10: INFO – INTEGER *Output*
On exit: INFO = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

If $INFO = -i$, the i th argument had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0 and $INFO \leq M$

If $INFO = i$, the i th row of A is exactly zero.

INFO > M

If $INFO = i$, the $(i - M)$ th column of A is exactly zero.

7 Accuracy

The computed scale factors will be close to the exact scale factors.

8 Further Comments

The real analogue of this routine is F07AFF (DGEEQU).

9 Example

This example equilibrates the general matrix A given by

$$A = \begin{pmatrix} -1.34 + 2.55i & (0.28 + 3.17i) \times 10^{10} & -6.39 - 2.20i \\ -1.70 - 1.41i & (3.31 - 0.15i) \times 10^{10} & -0.15 + 1.34i \\ (2.41 + 0.39i) \times 10^{-10} & -0.56 + 1.47i & (-0.83 - 0.69i) \times 10^{-10} \end{pmatrix}.$$

Details of the scaling factors, and the scaled matrix are output.

9.1 Program Text

```
*      F07ATF Example Program Text
*      Mark 21 Release. NAG Copyright 2004.
*      .. Parameters ..
INTEGER          NIN, NOUT
PARAMETER        (NIN=5,NOUT=6)
INTEGER          NMAX
PARAMETER        (NMAX=8)
INTEGER          LDA
PARAMETER        (LDA=NMAX)
*      .. Local Scalars ..
DOUBLE PRECISION AMAX, BIG, CJ, COLCND, ROWCND, SMALL
INTEGER          I, IFAIL, INFO, J, N
*      .. Local Arrays ..
COMPLEX *16      A(LDA,NMAX)
DOUBLE PRECISION C(NMAX), R(NMAX)
CHARACTER        CLABS(1), RLABS(1)
*      .. External Functions ..
DOUBLE PRECISION X02AJF, X02AMF
INTEGER          X02BHF
EXTERNAL         X02AJF, X02AMF, X02BHF
*      .. External Subroutines ..
EXTERNAL         X04DBF, ZGEEQU
*      .. Executable Statements ..
WRITE (NOUT,*) 'F07ATF Example Program Results'
WRITE (NOUT,*)
*      Skip heading in data file
READ (NIN,*)
READ (NIN,*) N
IF (N.LE.NMAX) THEN
*
*      Read the N by N matrix A from data file
*
READ (NIN,*) ((A(I,J),J=1,N),I=1,N)
*
*      Print the matrix A
*
IFAIL = 0
CALL X04DBF('General',' ',N,N,A,LDA,'Bracketed','1P,E10.2',
+          'Matrix A','Integer',RLABS,'Integer',CLABS,80,0,
+          IFAIL)
WRITE (NOUT,*)
*
*      Compute row and column scaling factors
*
CALL ZGEEQU(N,N,A,LDA,R,C,ROWCND,COLCND,AMAX,INFO)
*
IF (INFO.GT.0) THEN
  IF (INFO.LE.N) THEN
    WRITE (NOUT,99999) 'Row ', INFO, ' of A is exactly zero'
  ELSE
    WRITE (NOUT,99999) 'Column ', INFO - N,
+      ' of A is exactly zero'
  END IF
ELSE
*
*      Print ROWCND, COLCND, AMAX and the scale factors
*

```

```

      WRITE (NOUT,99998) 'ROWCND = ', ROWCND, ', COLCND = ',
+      COLCND, ', AMAX = ', AMAX
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Row scale factors'
      WRITE (NOUT,99997) (R(I),I=1,N)
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Column scale factors'
      WRITE (NOUT,99997) (C(I),I=1,N)
      WRITE (NOUT,*)
*
*      Compute values close to underflow and overflow
*
      SMALL = X02AMF()/(X02AJF()*X02BHF())
      BIG = 1.0D0/SMALL
      IF ((ROWCND.GE.0.1D0) .AND. (AMAX.GE.SMALL)
+      .AND. (AMAX.LE.BIG)) THEN
        IF (COLCND.LT.0.1D0) THEN
*
*          Just column scale A
*
          DO 40 J = 1, N
            CJ = C(J)
            DO 20 I = 1, N
              A(I,J) = A(I,J)*CJ
20          CONTINUE
140        CONTINUE
*
*          Print the column scaled matrix
*
          IFAIL = 0
          CALL X04DBF('General',' ',N,N,A,LDA,'Bracketed',' ',
+          'Scaled matrix','Integer',RLABS,'Integer',
+          CLABS,80,0,IFAIL)
*
          END IF
        ELSE IF (COLCND.GE.0.1D0) THEN
*
*          Just row scale A
*
          DO 80 J = 1, N
            DO 60 I = 1, N
              A(I,J) = R(I)*A(I,J)
60          CONTINUE
160        CONTINUE
*
*          Print the row scaled matrix
*
          IFAIL = 0
          CALL X04DBF('General',' ',N,N,A,LDA,'Bracketed',' ',
+          'Scaled matrix','Integer',RLABS,'Integer',
+          CLABS,80,0,IFAIL)
*
        ELSE
*
*          Row and column scale A
*
          DO 120 J = 1, N
            CJ = C(J)
            DO 100 I = 1, N
              A(I,J) = R(I)*A(I,J)*CJ
100          CONTINUE
120        CONTINUE
*
*          Print the row and column scaled matrix
*
          IFAIL = 0
          CALL X04DBF('General',' ',N,N,A,LDA,'Bracketed',' ',
+          'Scaled matrix','Integer',RLABS,'Integer',
+          CLABS,80,0,IFAIL)
*
          END IF

```

```

        END IF
    ELSE
        WRITE (NOUT,*) 'NMAX too small'
    END IF
*
99999 FORMAT (1X,A,I4,A)
99998 FORMAT (1X,3(A,1P,E7.1))
99997 FORMAT ((1X,1P,7E11.1))
    END

```

9.2 Program Data

F07ATF Example Program Data

```

3                                     :Value of N

(-1.34D+00, 2.55D+00) ( 0.28D+10, 3.17D+10) (-6.39D+00,-2.20D+00)
(-1.70D+00,-1.41D+00) ( 3.31D+10,-0.15D+10) (-0.15D+00, 1.34D+00)
( 2.41D-10, 0.39D-10) (-0.56D+00, 1.47D+00) (-0.83D-10,-0.69D-10)

                                     :End of matrix A

```

9.3 Program Results

F07ATF Example Program Results

Matrix A

```

               1               2               3
1 ( -1.34E+00, 2.55E+00) ( 2.80E+09, 3.17E+10) ( -6.39E+00, -2.20E+00)
2 ( -1.70E+00, -1.41E+00) ( 3.31E+10, -1.50E+09) ( -1.50E-01, 1.34E+00)
3 ( 2.41E-10, 3.90E-11) ( -5.60E-01, 1.47E+00) ( -8.30E-11, -6.90E-11)

```

ROWCND = 5.9E-11, COLCND = 1.4E-10, AMAX = 3.5E+10

Row scale factors

```

2.9E-11    2.9E-11    4.9E-01

```

Column scale factors

```

7.3E+09    1.0E+00    4.0E+09

```

Scaled matrix

```

               1               2               3
1 ( -0.2816, 0.5359) ( 0.0812, 0.9188) ( -0.7439, -0.2561)
2 ( -0.3562, -0.2954) ( 0.9566, -0.0434) ( -0.0174, 0.1555)
3 ( 0.8607, 0.1393) ( -0.2759, 0.7241) ( -0.1642, -0.1365)

```
