NAG Fortran Library Routine Document

F07BTF (ZGBEQU)

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

F07BTF (ZGBEQU) computes diagonal scaling matrices D_R and D_C intended to equilibrate a complex m by n band matrix A of band width $(k_l + k_u + 1)$, and reduce its condition number.

2 Specification

```
SUBROUTINE F07BTF (M, N, KL, KU, AB, LDAB, R, C, ROWCND, COLCND, AMAX,
1                  INFO)
INTEGER            M, N, KL, KU, LDAB, INFO
double precision   R(*), C(*), ROWCND, COLCND, AMAX
complex*16         AB(LDAB,*)
```

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

3 Description

F07BTF (ZGBEQU) 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 ≥ 0.

2: N – INTEGER *Input*

On entry: n, the number of columns of the matrix A.

Constraint: N ≥ 0.

3: KL – INTEGER *Input*

On entry: k_l, the number of subdiagonals of the matrix A.

Constraint: KL ≥ 0.

4: KU – INTEGER *Input*

On entry: k_u, the number of superdiagonals of the matrix A.

Constraint: KU ≥ 0.

5: AB(LDAB,$*$) – ***complex*16*** array *Input*

Note: the second dimension of the array AB must be at least $\max(1, \mathrm{N})$.

On entry: the band matrix A whose scaling factors are to be computed, stored in rows 1 to $k_l + k_u + 1$ of AB. The jth column of the matrix A is stored in the jth column of the array AB as follows:

$$\mathrm{AB}(k_u + 1 + i - j, j) = a_{ij}, \quad \text{for } \max(1, j - k_u) \leq i \leq \min(m, j + k_l).$$

6: LDAB – INTEGER *Input*

On entry: the first dimension of the array AB as declared in the (sub)program from which F07BTF (ZGBEQU) is called.

Constraint: $\mathrm{LDAB} \geq \mathrm{KL} + \mathrm{KU} + 1$.

7: R($*$) – ***double precision*** array *Output*

Note: the dimension of the array R must be at least $\max(1, \mathrm{M})$.

On exit: if INFO $= 0$ or INFO $>$ M, R contains the row scale factors, the diagonal elements of D_R. The elements of R will be positive.

8: C($*$) – ***double precision*** array *Output*

Note: the dimension of the array C must be at least $\max(1, \mathrm{N})$.

On exit: if INFO $= 0$, C contains the column scale factors, the diagonal elements of D_C. The elements of C will be positive.

9: 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 ≥ 0.1 and AMAX is neither too large nor too small, it is not worth scaling by D_R.

10: 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 ≥ 0.1, it is not worth scaling by D_C.

11: AMAX – ***double precision*** *Output*

On exit: $\max |a_{ij}|$. If AMAX is very close to overflow or underflow, the matrix A should be scaled.

12: 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 ith 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 ith row of A is exactly zero.

INFO $>$ M

If INFO $= i$, the $(i - \mathrm{M})$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 F07BFF (DGBEQU).

9 Example

To equilibrate the complex band matrix A given by

$$A = \begin{pmatrix} -1.65 + 2.26i & (-2.05 - 0.85i) \times 10^{-10} & 0.97 - 2.84i & 0 \\ 0.00 + 6.30i & (-1.48 - 1.75i) \times 10^{-10} & -3.99 + 4.01i & 0.59 - 0.48i \\ 0 & -0.77 + 2.83i & (-1.06 + 1.94i) \times 10^{10} & (3.33 - 1.04i) \times 10^{10} \\ 0 & 0 & 0.48 - 1.09i & -0.46 - 1.72i \end{pmatrix}.$$

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

9.1 Program Text

```
*     F07BTF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, KLMAX, KUMAX
      PARAMETER        (NMAX=8,KLMAX=4,KUMAX=4)
      INTEGER          LDAB
      PARAMETER        (LDAB=KLMAX+KUMAX+1)
*     .. Local Scalars ..
      DOUBLE PRECISION AMAX, BIG, CJ, COLCND, ROWCND, SMALL
      INTEGER          I, IFAIL, INFO, J, K, KL, KU, N
*     .. Local Arrays ..
      COMPLEX *16      AB(LDAB,NMAX)
      DOUBLE PRECISION C(NMAX), R(NMAX)
*     .. External Functions ..
      DOUBLE PRECISION X02AJF, X02AMF
      INTEGER          X02BHF
      EXTERNAL         X02AJF, X02AMF, X02BHF
*     .. External Subroutines ..
      EXTERNAL         X04DEF, ZGBEQU
*     .. Intrinsic Functions ..
      INTRINSIC        MAX, MIN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F07BTF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, KL, KU
      IF (N.LE.NMAX .AND. KL.LE.KLMAX .AND. KU.LE.KUMAX) THEN
*
*        Read the band matrix A from data file
*
         K = KU + 1
         READ (NIN,*) ((AB(K+I-J,J),J=MAX(I-KL,1),MIN(I+KU,N)),I=1,N)
*
*        Print the matrix A
*
         IFAIL = 0
         CALL X04DEF(N,N,KL,KU,AB,LDAB,'Matrix A',IFAIL)
         WRITE (NOUT,*)
```

```
*
*        Compute row and column scaling factors
*
         CALL ZGBEQU(N,N,KL,KU,AB,LDAB,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)
                     K = KU + 1 - J
                     DO 20 I = MAX(1,J-KU), MIN(N,J+KL)
                        AB(K+I,J) = AB(K+I,J)*CJ
   20                CONTINUE
   40             CONTINUE
*
*                 Print the column scaled matrix
*
                  IFAIL = 0
                  CALL X04DEF(N,N,KL,KU,AB,LDAB,'Scaled matrix',IFAIL)
*
               END IF
            ELSE IF (COLCND.GE.0.1D0) THEN
*
*              Just row scale A
*
               DO 80 J = 1, N
                  K = KU + 1 - J
                  DO 60 I = MAX(1,J-KU), MIN(N,J+KL)
                     AB(K+I,J) = R(I)*AB(K+I,J)
   60             CONTINUE
   80          CONTINUE
*
*              Print the row scaled matrix
*
               IFAIL = 0
               CALL X04DEF(N,N,KL,KU,AB,LDAB,'Scaled matrix',IFAIL)
*
            ELSE
*
*              Row and column scale A
*
               DO 120 J = 1, N
```

```
                  CJ = C(J)
                  K = KU + 1 - J
                  DO 100 I = MAX(1,J-KU), MIN(N,J+KL)
                     AB(K+I,J) = R(I)*AB(K+I,J)*CJ
  100             CONTINUE
  120          CONTINUE
*
*              Print the row and column scaled matrix
*
               IFAIL = 0
               CALL X04DEF(N,N,KL,KU,AB,LDAB,'Scaled matrix',IFAIL)
*
            END IF
         END IF
      ELSE
         WRITE (NOUT,*)
     +     'One or more of NMAX, KLMAX or KUMAX is too small'
      END IF
      STOP
*
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

```
F07BTF Example Program Data
  4  1  2                                                    :Values of N, KL and KU
 (-1.65, 2.26) (-2.05D-10,-8.50D-11) ( 9.70D-01,-2.84D+00)
 ( 0.00, 6.30) (-1.48D-10,-1.75D-10) (-3.99D+00, 4.01D+00) ( 0.59D+00,-0.48D+00)
               (-7.70D-01, 2.83D+00) (-1.06D+10, 1.94D+10) ( 3.33D+10,-1.04D+10)
                                     ( 4.48D+00,-1.09D+00) (-0.46D+00,-1.72D+00)
                                                           :End of matrix A
```

9.3 Program Results

```
 F07BTF Example Program Results

 Matrix A
                1            2            3            4
 1   -1.6500E+00  -2.0500E-10   9.7000E-01
      2.2600E+00  -8.5000E-11  -2.8400E+00

 2    0.0000E+00  -1.4800E-10  -3.9900E+00   5.9000E-01
      6.3000E+00  -1.7500E-10   4.0100E+00  -4.8000E-01

 3                -7.7000E-01  -1.0600E+10   3.3300E+10
                   2.8300E+00   1.9400E+10  -1.0400E+10

 4                              4.4800E+00  -4.6000E-01
                               -1.0900E+00  -1.7200E+00

 ROWCND = 8.9E-11, COLCND = 8.2E-11, AMAX = 4.4E+10

 Row scale factors
     2.6E-01    1.2E-01    2.3E-11    1.8E-01

 Column scale factors
     1.0E+00    1.2E+10    1.0E+00    1.0E+00

 Scaled matrix
          1       2       3       4
 1  -0.4220 -0.6364  0.2481
     0.5780 -0.2639 -0.7263

 2   0.0000 -0.2246 -0.4988  0.0737
     0.7875 -0.2655  0.5012 -0.0600

 3          -0.2139 -0.2426  0.7620
```

```
             0.7861  0.4439 -0.2380

 4                    0.8043 -0.0826
                     -0.1957 -0.3088
```