

# NAG Library Routine Document

## F07HTF (ZPBEQU)

**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

F07HTF (ZPBEQU) computes a diagonal scaling matrix  $S$  intended to equilibrate a complex  $n$  by  $n$  Hermitian positive-definite band matrix  $A$ , with bandwidth  $(2k_d + 1)$ , and reduce its condition number.

### 2 Specification

```
SUBROUTINE F07HTF(UPLO, N, KD, AB, LDAB, S, SCOND, AMAX, INFO)
INTEGER          N, KD, LDAB, INFO
double precision S(N), SCOND, AMAX
complex*16      AB(LDAB,*)
CHARACTER*1      UPLO
```

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

### 3 Description

F07HTF (ZPBEQU) computes a diagonal scaling matrix  $S$  chosen so that

$$s_j = 1/\sqrt{a_{jj}}.$$

This means that the matrix  $B$  given by

$$B = SAS,$$

has diagonal elements equal to unity. This in turn means that the condition number of  $B$ ,  $\kappa_2(B)$ , is within a factor  $n$  of the matrix of smallest possible condition number over all possible choices of diagonal scalings (see Corollary 7.6 of Higham (2002)).

### 4 References

Higham N J (2002) *Accuracy and Stability of Numerical Algorithms* (2nd Edition) SIAM, Philadelphia

### 5 Parameters

- 1: UPLO – CHARACTER\*1 *Input*  
*On entry:* indicates whether the upper or lower triangular part of  $A$  is stored in the array AB, as follows:  
UPLO = 'U'  
The upper triangle of  $A$  is stored.  
UPLO = 'L'  
The lower triangle of  $A$  is stored.  
*Constraint:* UPLO = 'U' or 'L'.
- 2: N – INTEGER *Input*  
*On entry:*  $n$ , the order of the matrix  $A$ .  
*Constraint:*  $N \geq 0$ .

- 3: KD – INTEGER *Input*  
*On entry:*  $k_d$ , the number of superdiagonals of the matrix  $A$  if UPLO = 'U', or the number of subdiagonals if UPLO = 'L'.  
*Constraint:*  $KD \geq 0$ .
- 4: AB(LDAB,\*) – **complex\*16** array *Input*  
**Note:** the second dimension of the array AB must be at least  $\max(1, N)$ .  
*On entry:* the upper or lower triangle of the Hermitian positive-definite band matrix  $A$  whose scaling factors are to be computed.  
The matrix is stored in rows 1 to  $k_d + 1$ , more precisely,  
if UPLO = 'U', the elements of the upper triangle of  $A$  within the band must be stored with element  $A_{ij}$  in  $AB(k_d + 1 + i - j, j)$  for  $\max(1, j - k_d) \leq i \leq j$ ;  
if UPLO = 'L', the elements of the lower triangle of  $A$  within the band must be stored with element  $A_{ij}$  in  $AB(1 + i - j, j)$  for  $j \leq i \leq \min(n, j + k_d)$ .  
Only the elements of the array AB corresponding to the diagonal elements of  $A$  are referenced. (Row  $(k_d + 1)$  of AB when UPLO = 'U', row 1 of AB when UPLO = 'L'.)
- 5: LDAB – INTEGER *Input*  
*On entry:* the first dimension of the array AB as declared in the (sub)program from which F07HTF (ZPBEQU) is called.  
*Constraint:*  $LDAB \geq KD + 1$ .
- 6: S(N) – **double precision** array *Output*  
*On exit:* if INFO = 0 on exit, S contains the diagonal elements of the scaling matrix  $S$ .
- 7: SCOND – **double precision** *Output*  
*On exit:* if INFO = 0 on exit, SCOND contains the ratio of the smallest value of  $S(i)$  to the largest value of  $S(i)$ . If  $SCOND \geq 0.1$  and AMAX is neither too large nor too small, it is not worth scaling by  $S$ .
- 8: AMAX – **double precision** *Output*  
*On exit:*  $\max |a_{ij}|$ . If AMAX is very close to overflow or underflow, the matrix  $A$  should be scaled.
- 9: 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

If INFO =  $i$ , the  $i$ th diagonal element of  $A$  is not positive (and hence  $A$  cannot be positive-definite).

## 7 Accuracy

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

## 8 Further Comments

The real analogue of this routine is F07HFF (DPBEQU).

## 9 Example

This example equilibrates the Hermitian positive-definite matrix  $A$  given by

$$A = \begin{pmatrix} 9.39 & 1.08 - 1.73i & 0 & 0 \\ 1.08 + 1.73i & 1.69 & (-0.04 + 0.29i) \times 10^{10} & 0 \\ 0 & (-0.04 - 0.29i) \times 10^{10} & 2.65 \times 10^{20} & (-0.33 + 2.24i) \times 10^{10} \\ 0 & 0 & (-0.33 - 2.24i) \times 10^{10} & 2.17 \end{pmatrix}.$$

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

### 9.1 Program Text

```
*      F07HTF Example Program Text
*      Mark 21 Release. NAG Copyright 2004.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          KDMAX, NMAX
      PARAMETER        (KDMAX=4,NMAX=8)
      INTEGER          LDAB
      PARAMETER        (LDAB=KDMAX+1)
      CHARACTER        UPLO
      PARAMETER        (UPLO='U')
*      .. Local Scalars ..
      DOUBLE PRECISION AMAX, BIG, SCOND, SJ, SMALL
      INTEGER          I, IFAIL, INFO, J, JJ, KD, N
*      .. Local Arrays ..
      COMPLEX *16      AB(LDAB,NMAX)
      DOUBLE PRECISION S(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*      .. External Functions ..
      DOUBLE PRECISION X02AJF, X02AMF
      INTEGER          X02BHF
      EXTERNAL         X02AJF, X02AMF, X02BHF
*      .. External Subroutines ..
      EXTERNAL         X04DFF, ZPBEQU
*      .. Intrinsic Functions ..
      INTRINSIC        MAX, MIN
*      .. Executable Statements ..
      WRITE (NOUT,*) 'F07HTF Example Program Results'
      WRITE (NOUT,*)
*      Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, KD
      IF (N.LE.NMAX .AND. KD.LE.KDMAX) THEN
*
*         Read the upper or lower triangular part of the band matrix A
*         from data file
*
         IF (UPLO.EQ.'U') THEN
            DO 20 I = 1, N
               READ (NIN,*) (AB(KD+1+I-J,J),J=I,MIN(N,I+KD))
20          CONTINUE
         ELSE IF (UPLO.EQ.'L') THEN
            DO 40 I = 1, N
               READ (NIN,*) (AB(1+I-J,J),J=MAX(1,I-KD),I)
40          CONTINUE
         END IF
*
*         Print the matrix A
*
         IFAIL = 0
         IF (UPLO.EQ.'U') THEN
```

```

      CALL X04DFF(N,N,0,KD,AB,LDAB,'Bracketed','1P,E10.2',
+           'Matrix A','Integer',RLABS,'Integer',CLABS,80,0,
+           IFAIL)
      ELSE IF (UPLO.EQ.'L') THEN
      CALL X04DFF(N,N,KD,0,AB,LDAB,'Bracketed','1P,E10.2',
+           'Matrix A','Integer',RLABS,'Integer',CLABS,80,0,
+           IFAIL)
      END IF
      WRITE (NOUT,*)

*
*      Compute diagonal scaling factors
*
      CALL ZPBEQU(UPLO,N,KD,AB,LDAB,S,SCOND,AMAX,INFO)

*
      IF (INFO.GT.0) THEN
      WRITE (NOUT,99999) 'Diagonal element', INFO,
+           ' of A is non positive'
      ELSE

*
*      Print SCOND, AMAX and the scale factors
*
      WRITE (NOUT,99998) 'SCOND = ', SCOND, ', AMAX = ', AMAX
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Diagonal scaling factors'
      WRITE (NOUT,99997) (S(I),I=1,N)
      WRITE (NOUT,*)

*
*      Compute values close to underflow and overflow
*
      SMALL = X02AMF()/(X02AJF()*X02BHF())
      BIG = 1.0D0/SMALL
      IF ((SCOND.LT.0.1D0) .OR. (AMAX.LT.SMALL) .OR. (AMAX.GT.BIG)
+          ) THEN

*
*      Scale A
*
      IF (UPLO.EQ.'U') THEN
      DO 80 J = 1, N
      SJ = S(J)
      JJ = KD + 1 - J
      DO 60 I = MAX(1,J-KD), J
      AB(I+JJ,J) = S(I)*AB(I+JJ,J)*SJ
60      CONTINUE
80      CONTINUE
      ELSE IF (UPLO.EQ.'L') THEN
      DO 120 J = 1, N
      SJ = S(J)
      JJ = 1 - J
      DO 100 I = J, MIN(N,J+KD)
      AB(I+JJ,J) = S(I)*AB(I+JJ,J)*SJ
100      CONTINUE
120      CONTINUE
      END IF

*
*      Print the scaled matrix
*
      IFAIL = 0
      IF (UPLO.EQ.'U') THEN
      CALL X04DFF(N,N,0,KD,AB,LDAB,'Bracketed','F7.4',
+           'Scaled matrix','Integer',RLABS,'Integer',
+           CLABS,80,0,IFAIL)
      ELSE IF (UPLO.EQ.'L') THEN
      CALL X04DFF(N,N,KD,0,AB,LDAB,'Bracketed','F7.4',
+           'Scaled matrix','Integer',RLABS,'Integer',
+           CLABS,80,0,IFAIL)
      END IF
      END IF
      END IF
      ELSE
      WRITE (NOUT,*) 'NMAX and/or KDMAX too small'
      END IF

```

```

*
99999 FORMAT (1X,A,I4,A)
99998 FORMAT (1X,A,1P,E7.1,A,1P,E8.2)
99997 FORMAT ((1X,1P,7E11.1))
      END

```

## 9.2 Program Data

F07HTF Example Program Data

```

4 1                                     :Values of N and KD
( 9.39, 0.00) ( 1.08,-1.73)
      ( 1.69, 0.00) ( -0.04E+10, 0.29E+10)
      ( 2.65E+20, 0.00 ) ( -0.33E+10, 2.24E+10)
      ( 2.17, 0.00 )
                                     :End of matrix A

```

## 9.3 Program Results

F07HTF Example Program Results

Matrix A

```

1                                     1                                     2                                     3
1 ( 9.39E+00, 0.00E+00) ( 1.08E+00, -1.73E+00)
2                                     ( 1.69E+00, 0.00E+00) ( -4.00E+08, 2.90E+09)
3                                     ( 2.65E+20, 0.00E+00)
4
                                     4
1
2
3 ( -3.30E+09, 2.24E+10)
4 ( 2.17E+00, 0.00E+00)

```

SCOND = 8.0E-11, AMAX = 2.65E+20

Diagonal scaling factors

```

3.3E-01 7.7E-01 6.1E-11 6.8E-01

```

Scaled matrix

```

1                                     1                                     2                                     3                                     4
1 ( 1.0000, 0.0000) ( 0.2711,-0.4343)
2                                     ( 1.0000, 0.0000) (-0.0189, 0.1370)
3                                     ( 1.0000, 0.0000) (-0.1376, 0.9341)
4                                     ( 1.0000, 0.0000)

```

---