

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
REAL (KIND=nag_wp) S(N), SCOND, AMAX
COMPLEX (KIND=nag_wp) 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 Arguments

- 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 (KIND=nag_wp) 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) – REAL (KIND=nag_wp) array *Output*
On exit: if INFO = 0, S contains the diagonal elements of the scaling matrix S .
- 7: SCOND – REAL (KIND=nag_wp) *Output*
On exit: if INFO = 0, SCOND contains the ratio of the smallest value of S to the largest value of S. If $SCOND \geq 0.1$ and AMAX is neither too large nor too small, it is not worth scaling by S .
- 8: AMAX – REAL (KIND=nag_wp) *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

INFO < 0

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

INFO > 0

The $\langle value \rangle$ 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 Parallelism and Performance

F07HTF (ZPBEQU) is not threaded in any implementation.

9 Further Comments

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

10 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.

10.1 Program Text

```

Program f07htfe

!      F07HTF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
!      Use nag_library, Only: f06kcf, nag_wp, x02ajf, x02amf, x02bhf, x04dff, &
!                               zdscal, zpbequ

!      .. Implicit None Statement ..
!      Implicit None

!      .. Parameters ..
!      Real (Kind=nag_wp), Parameter      :: one = 1.0_nag_wp
!      Real (Kind=nag_wp), Parameter      :: thresh = 0.1_nag_wp
!      Integer, Parameter                  :: nin = 5, nout = 6
!      Character (1), Parameter            :: uplo = 'U'

!      .. Local Scalars ..
!      Real (Kind=nag_wp)                  :: amax, big, scond, small
!      Integer                              :: i, i0, i1, ifail, ilen, info, j, kd, &
!                                             ldab, n

!      .. Local Arrays ..
!      Complex (Kind=nag_wp), Allocatable :: ab(:, :)
!      Real (Kind=nag_wp), Allocatable    :: s(:)
!      Character (1)                       :: clabs(1), rlabs(1)

!      .. Intrinsic Procedures ..
!      Intrinsic                            :: max, min, real

!      .. Executable Statements ..
!      Write (nout,*) 'F07HTF Example Program Results'
!      Write (nout,*)
!      Flush (nout)
!      Skip heading in data file
!      Read (nin,*)
!      Read (nin,*) n, kd
!      ldab = kd + 1
!      Allocate (ab(ldab,n),s(n))

!      Read the upper or lower triangular part of the band matrix A
!      from data file

!      If (uplo=='U') Then
!         Do i = 1, n
!            Read (nin,*)(ab(kd+1+i-j,j),j=i,min(n,i+kd))
!         End Do
!      Else If (uplo=='L') Then
!         Do i = 1, n
!            Read (nin,*)(ab(1+i-j,j),j=max(1,i-kd),i)

```

```

      End Do
    End If

!   Print the matrix A

!   ifail: behaviour on error exit
!   =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
  ifail = 0
  If (uplo=='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=='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

!   The NAG name equivalent of zpbequ is f07htf
  Call zpbequ(uplo,n,kd,ab,ldab,s,scond,amax,info)

  If (info>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 =', sconfd, ', AMAX =', amax
    Write (nout,*)
    Write (nout,*) 'Diagonal scaling factors'
    Write (nout,99997) s(1:n)
    Write (nout,*)
    Flush (nout)

!   Compute values close to underflow and overflow

    small = x02amf()/(x02ajf()*real(x02bhf(),kind=nag_wp))
    big = one/small
    If ((scond<thresh) .Or. (amax<small) .Or. (amax>big)) Then

!   Scale A

      If (uplo=='U') Then
!   The NAG name equivalent of zdscal is f06jdf
        Do j = 1, n
          i0 = max(1,j-kd)
          i1 = 1 + i0 - (j-kd)
          ilen = j - i0 + 1
          Call zdscal(ilen,s(j),ab(i1,j),1)
          Call f06kcf(ilen,s(i0),1,ab(i1,j),1)
        End Do

      Else If (uplo=='L') Then
        Do j = 1, n
          i1 = 1
          ilen = min(n,j+kd) - j + 1
          Call zdscal(ilen,s(j),ab(i1,j),1)
          Call f06kcf(ilen,s(j),1,ab(i1,j),1)
        End Do
      End If

!   Print the scaled matrix

    ifail = 0
    If (uplo=='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=='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

```

```

          'Integer',rlabs,'Integer',clabs,80,0,ifail)
      End If
    End If
  End If

99999 Format (1X,A,I4,A)
99998 Format (1X,2(A,1P,E8.1))
99997 Format ((1X,1P,7E11.1))
      End Program f07htfe

```

10.2 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)
                                (  2.64E+20, 0.00 ) ( -0.33E+10, 2.24E+10)
                                                (  2.17, 0.00 )
                                                    :End of matrix A

```

10.3 Program Results

F07HTF Example Program Results

Matrix A

```

          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          (  2.64E+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 = 2.6E+20

Diagonal scaling factors

```

  3.3E-01  7.7E-01  6.2E-11  6.8E-01

```

Scaled matrix

```

          1                                     2                                     3                                     4
1 (  1.0000, 0.0000) (  0.2711,-0.4343)
2          (  1.0000, 0.0000) (-0.0189, 0.1373)
3          (  1.0000, 0.0000) (-0.1379, 0.9359)
4          (  1.0000, 0.0000)

```
