NAG Library Routine Document

F07FTF (ZPOEQU)

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

F07FTF (ZPOEQU) computes a diagonal scaling matrix S intended to equilibrate a complex n by n Hermitian positive definite matrix A and reduce its condition number.

2 Specification

```
SUBROUTINE F07FTF (N, A, LDA, S, SCOND, AMAX, INFO)
INTEGER                N, LDA, INFO
REAL (KIND=nag_wp)     S(N), SCOND, AMAX
COMPLEX (KIND=nag_wp)  A(LDA,*)
```

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

3 Description

F07FTF (ZPOEQU) 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: N – INTEGER *Input*

On entry: n, the order of the matrix A.

Constraint: $\mathrm{N} \geq 0$.

2: $\mathrm{A}(\mathrm{LDA},*)$ – COMPLEX (KIND=nag_wp) array *Input*

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

On entry: the matrix A whose scaling factors are to be computed. Only the diagonal elements of the array A are referenced.

3: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F07FTF (ZPOEQU) is called.

Constraint: $\mathrm{LDA} \geq \max(1,\mathrm{N})$.

4: S(N) – REAL (KIND=nag_wp) array *Output*

On exit: if INFO $= 0$, S contains the diagonal elements of the scaling matrix S.

5: 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 ≥ 0.1 and AMAX is neither too large nor too small, it is not worth scaling by S.

6: 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.

7: 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

Not applicable.

9 Further Comments

The real analogue of this routine is F07FFF (DPOEQU).

10 Example

This example equilibrates the Hermitian positive definite matrix A given by

$$A = \begin{pmatrix} 3.23 & 1.51 - 1.92i & (1.90 + 0.84i) \times 10^5 & 0.42 + 2.50i \\ 1.51 + 1.92i & 3.58 & (-0.23 + 1.11i) \times 10^5 & -1.18 + 1.37i \\ (1.90 - 0.84i) \times 10^5 & (-0.23 - 1.11i) \times 10^5 & 4.09 \times 10^{10} & (2.33 - 0.14i) \times 10^5 \\ 0.42 - 2.50i & -1.18 - 1.37i & (2.33 + 0.14i) \times 10^5 & 4.29 \end{pmatrix}.$$

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

10.1 Program Text

```
    Program f07ftfe

!     F07FTF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
```

```
      Use nag_library, Only: f06kcf, nag_wp, x02ajf, x02amf, x02bhf, x04dbf,   &
                             zdscal, zpoequ
!     .. 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
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: amax, big, scond, small
      Integer                          :: i, ifail, info, j, lda, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:)
      Real (Kind=nag_wp), Allocatable  :: s(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: real
!     .. Executable Statements ..
      Write (nout,*) 'F07FTF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      Allocate (a(lda,n),s(n))

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

      Read (nin,*)(a(i,i:n),i=1,n)

!     Print the matrix A

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call x04dbf('Upper','Non-unit',n,n,a,lda,'Bracketed','1P,E10.2', &
        'Matrix A','Integer',rlabs,'Integer',clabs,80,0,ifail)

      Write (nout,*)

!     Compute diagonal scaling factors

!     The NAG name equivalent of zpoequ is f07ftf
      Call zpoequ(n,a,lda,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 =', scond, ', 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
!         The NAG name equivalent of zdscal is f06jdf
          Do j = 1, n
            Call zdscal(j,s(j),a(1,j),1)
            Call f06kcf(j,s,1,a(1,j),1)
          End Do
```

```
!        Print the scaled matrix

         ifail = 0
         Call x04dbf('Upper','Non-unit',n,n,a,lda,'Bracketed','F8.4', &
           'Scaled matrix','Integer',rlabs,'Integer',clabs,80,0,ifail)

       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 f07ftfe
```

10.2 Program Data

```
F07FTF Example Program Data
   4                                                             :Value of N
 ( 3.23, 0.00) ( 1.51,-1.92) ( 1.90D+05, 0.84D+05) ( 0.42D+00, 2.50D+00)
               ( 3.58, 0.00) (-0.23D+05, 1.11D+05) (-1.18D+00, 1.37D+00)
                             ( 4.09D+10, 0.00D+00) ( 2.33D+05,-0.14D+05)
                                                   ( 4.29D+00, 0.00D+00)
                                                                 :End of matrix A
```

10.3 Program Results

```
 F07FTF Example Program Results

 Matrix A
                        1                        2                        3
 1  (   3.23E+00,  0.00E+00) (   1.51E+00, -1.92E+00) (   1.90E+05,  8.40E+04)
 2                           (   3.58E+00,  0.00E+00) (  -2.30E+04,  1.11E+05)
 3                                                    (   4.09E+10,  0.00E+00)
 4

                        4
 1  (   4.20E-01,  2.50E+00)
 2  (  -1.18E+00,  1.37E+00)
 3  (   2.33E+05, -1.40E+04)
 4  (   4.29E+00,  0.00E+00)

 SCOND = 8.9E-06, AMAX = 4.1E+10

 Diagonal scaling factors
     5.6E-01    5.3E-01    4.9E-06    4.8E-01

 Scaled matrix
                    1                    2                    3
 1  (   1.0000,  0.0000) (   0.4441, -0.5646) (   0.5227,  0.2311)
 2                       (   1.0000,  0.0000) (  -0.0601,  0.2901)
 3                                            (   1.0000,  0.0000)
 4

                    4
 1  (   0.1128,  0.6716)
 2  (  -0.3011,  0.3496)
 3  (   0.5562, -0.0334)
 4  (   1.0000,  0.0000)
```