

NAG Library Routine Document

F07GFF (DPPEQU)

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

F07GFF (DPPEQU) computes a diagonal scaling matrix S intended to equilibrate a real n by n symmetric positive definite matrix A , stored in packed format, and reduce its condition number.

2 Specification

```
SUBROUTINE F07GFF (UPLO, N, AP, S, SCOND, AMAX, INFO)
INTEGER          N, INFO
REAL (KIND=nag_wp) AP(*), S(N), SCOND, AMAX
CHARACTER(1)    UPLO
```

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

3 Description

F07GFF (DPPEQU) 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 AP, 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: AP(*) – REAL (KIND=nag_wp) array Input

Note: the dimension of the array AP must be at least $\max(1, N \times (N + 1)/2)$.

On entry: the n by n symmetric matrix A , packed by columns.

More precisely,

if UPLO = 'U', the upper triangle of A must be stored with element A_{ij} in $AP(i + j(j - 1)/2)$ for $i \leq j$;

if UPLO = 'L', the lower triangle of A must be stored with element A_{ij} in $AP(i + (2n - j)(j - 1)/2)$ for $i \geq j$.

Only the elements of AP corresponding to the diagonal elements A are referenced.

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 $SCOND \geq 0.1$ 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 complex analogue of this routine is F07GTF (ZPPEQU).

10 Example

This example equilibrates the symmetric positive definite matrix A given by

$$A = \begin{pmatrix} 4.16 & -3.12 \times 10^5 & 0.56 & -0.10 \\ -3.12 \times 10^5 & 5.03 \times 10^{10} & -0.83 \times 10^5 & 1.18 \times 10^5 \\ 0.56 & -0.83 \times 10^5 & 0.76 & 0.34 \\ -0.10 & 1.18 \times 10^5 & 0.34 & 1.18 \end{pmatrix}.$$

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

10.1 Program Text

```

Program f07gffe

!      F07GFF Example Program Text

!      Mark 25 Release. NAG Copyright 2014.

!      .. Use Statements ..
Use nag_library, Only: dppequ, dscal, f06fcf, nag_wp, x02ajf, x02amf,      &
                        x02bhf, x04ccf
!      .. 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, ifail, info, j, jinc, jj, n
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable     :: ap(:), s(:)
!      .. Intrinsic Procedures ..
Intrinsic                            :: real
!      .. Executable Statements ..
Write (nout,*) 'F07GFF Example Program Results'
Write (nout,*)
Flush (nout)
!      Skip heading in data file
Read (nin,*)
Read (nin,*) n

Allocate (ap((n*(n+1))/2),s(n))

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

If (uplo=='U') Then
  Read (nin,*)((ap(i+(j*(j-1))/2),j=i,n),i=1,n)
Else If (uplo=='L') Then
  Read (nin,*)((ap(i+((2*n-j)*(j-1))/2),j=1,i),i=1,n)
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
Call x04ccf(uplo,'Non-unit diagonal',n,ap,'Matrix A',ifail)

Write (nout,*)

!      Compute diagonal scaling factors
!      The NAG name equivalent of dppequ is f07gff
Call dppequ(uplo,n,ap,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

      If (uplo=='U') Then
!      The NAG name equivalent of dscal is f06edf
        jj = 1
        Do j = 1, n
          Call dscal(j,s(j),ap(jj),1)
          Call f06fcf(j,s,1,ap(jj),1)
          jj = jj + j
        End Do
      Else If (uplo=='L') Then
        jj = 1
        jinc = n
        Do j = 1, n
          Call dscal(jinc,s(j),ap(jj),1)
          Call f06fcf(jinc,s(j),1,ap(jj),1)
          jj = jj + jinc
          jinc = jinc - 1
        End Do
      End If

!      Print the scaled matrix

      ifail = 0
      Call x04ccf(uplo,'Non-unit diagonal',n,ap,'Scaled matrix',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 f07gffe

```

10.2 Program Data

F07GFF Example Program Data

```

4                                     :Value of N
4.16D+00  -3.12D+05   0.56D+00  -0.10D+00
           5.03D+10  -0.83D+05   1.18D+05
                               0.76D+00   0.34D+00
                                       1.18D+00 :End of matrix A

```

10.3 Program Results

F07GFF Example Program Results

Matrix A

	1	2	3	4
1	4.1600E+00	-3.1200E+05	5.6000E-01	-1.0000E-01
2		5.0300E+10	-8.3000E+04	1.1800E+05
3			7.6000E-01	3.4000E-01

4 1.1800E+00

SCOND = 3.9E-06, AMAX = 5.0E+10

Diagonal scaling factors

4.9E-01 4.5E-06 1.1E+00 9.2E-01

Scaled matrix

	1	2	3	4
1	1.0000	-0.6821	0.3149	-0.0451
2		1.0000	-0.4245	0.4843
3			1.0000	0.3590
4				1.0000
