

NAG Library Routine Document

F07FFF (DPOEQU)

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

F07FFF (DPOEQU) computes a diagonal scaling matrix S intended to equilibrate a real n by n symmetric positive definite matrix A and reduce its condition number.

2 Specification

```
SUBROUTINE F07FFF (N, A, LDA, S, SCOND, AMAX, INFO)
  INTEGER          N, LDA, INFO
  REAL (KIND=nag_wp) A(LDA,*), S(N), SCOND, AMAX
```

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

3 Description

F07FFF (DPOEQU) 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: N – INTEGER *Input*
On entry: n , the order of the matrix A .
Constraint: $N \geq 0$.
- 2: A(LDA,*) – REAL (KIND=nag_wp) array *Input*
Note: the second dimension of the array A must be at least $\max(1, N)$.
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 F07FFF (DPOEQU) is called.
Constraint: $LDA \geq \max(1, N)$.

- 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

F07FFF (DPOEQU) is not threaded in any implementation.

9 Further Comments

The complex analogue of this routine is F07FTF (ZPOEQU).

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 f07fffe
!
!   F07FFF Example Program Text
!
!   Mark 26 Release. NAG Copyright 2016.
!
!   .. Use Statements ..

```

```

Use nag_library, Only: dpoequ, dscal, f06fcf, nag_wp, x02ajf, x02amf,    &
                        x02bhf, x04caf
!
! .. 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 ..
Real (Kind=nag_wp), Allocatable    :: a(:,,:), s(:)
!
! .. Intrinsic Procedures ..
Intrinsic                           :: real
!
! .. Executable Statements ..
Write (nout,*) 'F07FFF 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 x04caf('Upper', 'Non-unit', n, n, a, lda, 'Matrix A', ifail)

Write (nout,*)

!
Compute diagonal scaling factors
!
The NAG name equivalent of dpoequ is f07fff
Call dpoequ(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 dscal is f06edf
    Do j = 1, n
      Call dscal(j,s(j),a(1,j),1)
      Call f06fcf(j,s,1,a(1,j),1)
    End Do

!
    Print the scaled matrix

    ifail = 0

```

```

      Call x04caf('Upper','Non-unit',n,n,a,lda,'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 f07fffe

```

10.2 Program Data

F07FFF Example Program Data

```

4
4.16D+00  -3.12D+05   0.56D+00  -0.10D+00   :Value of N
           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

F07FFF 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
