NAG Library Routine Document

F07AFF (DGEEQU)

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

F07AFF (DGEEQU) computes diagonal scaling matrices D_R and D_C intended to equilibrate a real m by n matrix A and reduce its condition number.

2 Specification

```
SUBROUTINE F07AFF (M, N, A, LDA, R, C, ROWCND, COLCND, AMAX, INFO)
INTEGER            M, N, LDA, INFO
REAL (KIND=nag_wp) A(LDA,*), R(M), C(N), ROWCND, COLCND, AMAX
```

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

3 Description

F07AFF (DGEEQU) computes the diagonal scaling matrices. The diagonal scaling matrices are chosen to try to make the elements of largest absolute value in each row and column of the matrix B given by

$$B = D_R A D_C$$

have absolute value 1. The diagonal elements of D_R and D_C are restricted to lie in the safe range $(\delta, 1/\delta)$, where δ is the value returned by routine X02AMF. Use of these scaling factors is not guaranteed to reduce the condition number of A but works well in practice.

4 References

None.

5 Arguments

1: M – INTEGER *Input*

On entry: m, the number of rows of the matrix A.

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

2: N – INTEGER *Input*

On entry: n, the number of columns of the matrix A.

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

3: A(LDA, $*$) – REAL (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.

4: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F07AFF (DGEEQU) is called.

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

5: R(M) – REAL (KIND=nag_wp) array *Output*

On exit: if INFO $= 0$ or INFO $>$ M, R contains the row scale factors, the diagonal elements of D_R. The elements of R will be positive.

6: C(N) – REAL (KIND=nag_wp) array *Output*

On exit: if INFO $= 0$, C contains the column scale factors, the diagonal elements of D_C. The elements of C will be positive.

7: ROWCND – REAL (KIND=nag_wp) *Output*

On exit: if INFO $= 0$ or INFO $>$ M, ROWCND contains the ratio of the smallest value of R(i) to the largest value of R(i). If ROWCND ≥ 0.1 and AMAX is neither too large nor too small, it is not worth scaling by D_R.

8: COLCND – REAL (KIND=nag_wp) *Output*

On exit: if INFO $= 0$, COLCND contains the ratio of the smallest value of C(i) to the largest value of C(i).

If COLCND ≥ 0.1, it is not worth scaling by D_C.

9: AMAX – REAL (KIND=nag_wp) *Output*

On exit: $\max \left|a_{ij}\right|$. If AMAX is very close to overflow or underflow, the matrix A should be scaled.

10: 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 and INFO $\leq$ M

Row $\langle$*value*$\rangle$ of A is exactly zero.

INFO $>$ M

Column $\langle$*value*$\rangle$ of A is exactly zero.

7 Accuracy

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

8 Parallelism and Performance

F07AFF (DGEEQU) is not threaded in any implementation.

9 Further Comments

The complex analogue of this routine is F07ATF (ZGEEQU).

10 Example

This example equilibrates the general matrix A given by

$$A = \begin{pmatrix} 1.80 \times 10^{10} & 2.88 \times 10^{10} & 2.05 & -8.90 \times 10^{9} \\ 5.25 & -2.95 & -9.50 \times 10^{-9} & -3.80 \\ 1.58 & -2.69 & -2.90 \times 10^{-10} & -1.04 \\ -1.11 & -0.66 & -5.90 \times 10^{-11} & 0.80 \end{pmatrix}.$$

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

10.1 Program Text

```
    Program f07affe

!     F07AFF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dgeequ, 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, colcnd, rowcnd, small
      Integer                          :: i, ifail, info, j, lda, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), c(:), r(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: real
!     .. Executable Statements ..
      Write (nout,*) 'F07AFF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      Allocate (a(lda,n),c(n),r(n))

!     Read the N by N matrix A from data file

      Read (nin,*)(a(i,1: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('General',' ',n,n,a,lda,'Matrix A',ifail)
      Write (nout,*)

!     Compute row and column scaling factors

!     The NAG name equivalent of dgeequ is f07aff
      Call dgeequ(n,n,a,lda,r,c,rowcnd,colcnd,amax,info)

      If (info>0) Then
        If (info<=n) Then
          Write (nout,99999) 'Row ', info, ' of A is exactly zero'
        Else
          Write (nout,99999) 'Column ', info - n, ' of A is exactly zero'
        End If
      Else
```

```
!       Print ROWCND, COLCND, AMAX and the scale factors

        Write (nout,99998) 'ROWCND =', rowcnd, ', COLCND =', colcnd,           &
          ', AMAX =', amax
        Write (nout,*)
        Write (nout,*) 'Row scale factors'
        Write (nout,99997) r(1:n)
        Write (nout,*)
        Write (nout,*) 'Column scale factors'
        Write (nout,99997) c(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 ((rowcnd>=thresh) .And. (amax>=small) .And. (amax<=big)) Then
          If (colcnd<thresh) Then

!           Just column scale A
!           The NAG name equivalent of dscal is f06edf
            Do j = 1, n
              Call dscal(n,c(j),a(1,j),1)
            End Do

          End If
        Else If (colcnd>=thresh) Then

!         Just row scale A
          Do j = 1, n
            Call f06fcf(n,r,1,a(1,j),1)
          End Do

        Else

!         Row and column scale A
          Do j = 1, n
            Call dscal(n,c(j),a(1,j),1)
            Call f06fcf(n,r,1,a(1,j),1)
          End Do

        End If

!       Print the scaled matrix
        ifail = 0
        Call x04caf('General',' ',n,n,a,lda,'Scaled matrix',ifail)

      End If

99999 Format (1X,A,I4,A)
99998 Format (1X,3(A,1P,E8.1))
99997 Format ((1X,1P,7E11.2))
    End Program f07affe
```

10.2 Program Data

```
F07AFF Example Program Data

  4                                          :Value of N

  1.80D+10   2.88D+10   2.05D+00  -8.90D+09
  5.25D+00  -2.95D+00  -9.50D-09  -3.80D+00
  1.58D+00  -2.69D+00  -2.90D-10  -1.04D+00
 -1.11D+00  -6.60D-01  -5.90D-11   8.00D-01 :End of matrix A
```

10.3 Program Results

```
 F07AFF Example Program Results

 Matrix A
                1             2             3             4
 1    1.8000E+10   2.8800E+10   2.0500E+00  -8.9000E+09
 2    5.2500E+00  -2.9500E+00  -9.5000E-09  -3.8000E+00
 3    1.5800E+00  -2.6900E+00  -2.9000E-10  -1.0400E+00
 4   -1.1100E+00  -6.6000E-01  -5.9000E-11   8.0000E-01

 ROWCND = 3.9E-11, COLCND = 1.8E-09, AMAX = 2.9E+10

 Row scale factors
    3.47E-11   1.90E-01   3.72E-01   9.01E-01

 Column scale factors
    1.00E+00   1.00E+00   5.53E+08   1.38E+00

 Scaled matrix
              1          2          3          4
 1       0.6250     1.0000     0.0393    -0.4269
 2       1.0000    -0.5619    -1.0000    -1.0000
 3       0.5874    -1.0000    -0.0596    -0.5341
 4      -1.0000    -0.5946    -0.0294     0.9957
```