NAG Library Routine Document

F07MAF (DSYSV)

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

F07MAF (DSYSV) computes the solution to a real system of linear equations

$$AX = B,$$

where A is an n by n symmetric matrix and X and B are n by r matrices.

2 Specification

```
SUBROUTINE F07MAF (UPLO, N, NRHS, A, LDA, IPIV, B, LDB, WORK, LWORK,       &
                   INFO)
INTEGER            N, NRHS, LDA, IPIV(*), LDB, LWORK, INFO
REAL (KIND=nag_wp) A(LDA,*), B(LDB,*), WORK(max(1,LWORK))
CHARACTER(1)       UPLO
```

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

3 Description

F07MAF (DSYSV) uses the diagonal pivoting method to factor A as $A = UDU^{\mathrm{T}}$ if UPLO = 'U' or $A = LDL^{\mathrm{T}}$ if UPLO = 'L', where U (or L) is a product of permutation and unit upper (lower) triangular matrices, and D is symmetric and block diagonal with 1 by 1 and 2 by 2 diagonal blocks. The factored form of A is then used to solve the system of equations $AX = B$.

Note that, in general, different permutations (pivot sequences) and diagonal block structures are obtained for UPLO = 'U' or 'L'

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia http://www.netlib.org/lapack/lug

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Arguments

1: UPLO – CHARACTER(1) *Input*

On entry: if UPLO = 'U', the upper triangle of A is stored.

If UPLO = 'L', the lower triangle of A is stored.

Constraint: UPLO = 'U' or 'L'.

2: N – INTEGER *Input*

On entry: n, the number of linear equations, i.e., the order of the matrix A.

Constraint: N $\geq$ 0.

3: NRHS – INTEGER *Input*

On entry: r, the number of right-hand sides, i.e., the number of columns of the matrix B.

Constraint: $\text{NRHS} \geq 0$.

4: $\text{A}(\text{LDA},*)$ – REAL (KIND=nag_wp) array *Input/Output*

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

On entry: the n by n symmetric matrix A.

If UPLO = 'U', the upper triangular part of A must be stored and the elements of the array below the diagonal are not referenced.

If UPLO = 'L', the lower triangular part of A must be stored and the elements of the array above the diagonal are not referenced.

On exit: if INFO = 0, the block diagonal matrix D and the multipliers used to obtain the factor U or L from the factorization $A = UDU^{\mathrm{T}}$ or $A = LDL^{\mathrm{T}}$ as computed by F07MDF (DSYTRF).

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F07MAF (DSYSV) is called.

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

6: $\text{IPIV}(*)$ – INTEGER array *Output*

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

On exit: details of the interchanges and the block structure of D. More precisely,

if $\text{IPIV}(i) = k > 0$, d_{ii} is a 1 by 1 pivot block and the ith row and column of A were interchanged with the kth row and column;

if UPLO = 'U' and $\text{IPIV}(i-1) = \text{IPIV}(i) = -l < 0$, $\begin{pmatrix} d_{i-1,i-1} & \bar{d}_{i,i-1} \\ \bar{d}_{i,i-1} & d_{ii} \end{pmatrix}$ is a 2 by 2 pivot block and the $(i-1)$th row and column of A were interchanged with the lth row and column;

if UPLO = 'L' and $\text{IPIV}(i) = \text{IPIV}(i+1) = -m < 0$, $\begin{pmatrix} d_{ii} & d_{i+1,i} \\ d_{i+1,i} & d_{i+1,i+1} \end{pmatrix}$ is a 2 by 2 pivot block and the $(i+1)$th row and column of A were interchanged with the mth row and column.

7: $\text{B}(\text{LDB},*)$ – REAL (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array B must be at least $\max(1,\text{NRHS})$.

On entry: the n by r right-hand side matrix B.

On exit: if INFO = 0, the n by r solution matrix X.

8: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07MAF (DSYSV) is called.

Constraint: $\text{LDB} \geq \max(1,\text{N})$.

9: $\text{WORK}(\max(1,\text{LWORK}))$ – REAL (KIND=nag_wp) array *Workspace*

On exit: if INFO = 0, WORK(1) returns the optimal LWORK.

10: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F07MAF (DSYSV) is called.

$\text{LWORK} \geq 1$, and for best performance $\text{LWORK} \geq \max(1, \text{N} \times nb)$, where nb is the optimal block size for F07MDF (DSYTRF).

If $\text{LWORK} = -1$, a workspace query is assumed; the routine only calculates the optimal size of the WORK array, returns this value as the first entry of the WORK array, and no error message related to LWORK is issued.

11: INFO – INTEGER *Output*

On exit: $\text{INFO} = 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

$\text{INFO} < 0$

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

$\text{INFO} > 0$

Element $\langle value \rangle$ of the diagonal is exactly zero. The factorization has been completed, but the block diagonal matrix D is exactly singular, so the solution could not be computed.

7 Accuracy

The computed solution for a single right-hand side, $\hat{x}$, satisfies an equation of the form

$$(A + E)\hat{x} = b,$$

where

$$\|E\|_1 = O(\epsilon)\|A\|_1$$

and ϵ is the ***machine precision***. An approximate error bound for the computed solution is given by

$$\frac{\|\hat{x} - x\|_1}{\|x\|_1} \leq \kappa(A)\frac{\|E\|_1}{\|A\|_1},$$

where $\kappa(A) = \left\|A^{-1}\right\|_1 \|A\|_1$, the condition number of A with respect to the solution of the linear equations. See Section 4.4 of Anderson *et al.* (1999) for further details.

F07MBF (DSYSVX) is a comprehensive LAPACK driver that returns forward and backward error bounds and an estimate of the condition number. Alternatively, F04BHF solves $Ax = b$ and returns a forward error bound and condition estimate. F04BHF calls F07MAF (DSYSV) to solve the equations.

8 Parallelism and Performance

F07MAF (DSYSV) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The total number of floating-point operations is approximately $\frac{1}{3}n^3 + 2n^2r$, where r is the number of right-hand sides.

The complex analogues of F07MAF (DSYSV) are F07MNF (ZHESV) for Hermitian matrices, and F07NNF (ZSYSV) for symmetric matrices.

10 Example

This example solves the equations

$$Ax = b,$$

where A is the symmetric matrix

$$A = \begin{pmatrix} -1.81 & 2.06 & 0.63 & -1.15 \\ 2.06 & 1.15 & 1.87 & 4.20 \\ 0.63 & 1.87 & -0.21 & 3.87 \\ -1.15 & 4.20 & 3.87 & 2.07 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 0.96 \\ 6.07 \\ 8.38 \\ 9.50 \end{pmatrix}.$$

Details of the factorization of A are also output.

10.1 Program Text

```
    Program f07mafe

!     F07MAF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dsysv, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                :: nb = 64, nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                           :: i, ifail, info, lda, lwork, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable   :: a(:,:), b(:), work(:)
      Integer, Allocatable              :: ipiv(:)
!     .. Executable Statements ..
      Write (nout,*) 'F07MAF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      lwork = nb*n
      Allocate (a(lda,n),b(n),work(lwork),ipiv(n))

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

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

!     Read b from data file

      Read (nin,*) b(1:n)

!     Solve the equations Ax = b for x
!     The NAG name equivalent of dsysv is f07maf
      Call dsysv('Upper',n,1,a,lda,ipiv,b,n,work,lwork,info)

      If (info==0) Then

!       Print solution
```

```
        Write (nout,*) 'Solution'
        Write (nout,99999) b(1:n)

!       Print details of factorization

        Write (nout,*)
        Flush (nout)

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04caf('Upper','Non-unit diagonal',n,n,a,lda,                     &
          'Details of the factorization',ifail)

!       Print pivot indices

        Write (nout,*)
        Write (nout,*) 'Pivot indices'
        Write (nout,99998) ipiv(1:n)

      Else
        Write (nout,99997) 'The diagonal block ', info, ' of D is zero'
      End If

99999 Format ((3X,7F11.4))
99998 Format (1X,7I11)
99997 Format (1X,A,I3,A)
    End Program f07mafe
```

10.2 Program Data

```
F07MAF Example Program Data
  4                         :Value of N
 -1.81   2.06   0.63  -1.15
         1.15   1.87   4.20
               -0.21   3.87
                       2.07 :End of matrix A
  0.96   6.07   8.38   9.50 :End of vector b
```

10.3 Program Results

```
 F07MAF Example Program Results

 Solution
      -5.0000    -2.0000     1.0000     4.0000

 Details of the factorization
             1          2          3          4
 1      0.4074     0.3031    -0.5960     0.6537
 2                -2.5907     0.8115     0.2230
 3                            1.1500     4.2000
 4                                       2.0700

 Pivot indices
           1          2         -2         -2
```