NAG Library Routine Document

F04DJF

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

F04DJF computes the solution to a complex system of linear equations $AX = B$, where A is an n by n complex symmetric matrix, stored in packed format and X and B are n by r matrices. An estimate of the condition number of A and an error bound for the computed solution are also returned.

2 Specification

```
SUBROUTINE F04DJF (UPLO, N, NRHS, AP, IPIV, B, LDB, RCOND, ERRBND,         &
                   IFAIL)

INTEGER                 N, NRHS, IPIV(N), LDB, IFAIL
REAL (KIND=nag_wp)      RCOND, ERRBND
COMPLEX (KIND=nag_wp) AP(*), B(LDB,*)
CHARACTER(1)            UPLO
```

3 Description

The diagonal pivoting method is used 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 $AX = B$.

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

Higham N J (2002) *Accuracy and Stability of Numerical Algorithms* (2nd Edition) SIAM, Philadelphia

5 Parameters

1: UPLO – CHARACTER(1) *Input*

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

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

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

2: N – INTEGER *Input*

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

Constraint: N ≥ 0.

3: NRHS – INTEGER *Input*

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

Constraint: NRHS ≥ 0.

4: AP($*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

On entry: the n by n symmetric matrix A, packed column-wise in a linear array. The jth column of the matrix A is stored in the array AP as follows:

More precisely,

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

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

On exit: if IFAIL ≥ 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 F07QRF (ZSPTRF), stored as a packed triangular matrix in the same storage format as A.

5: IPIV(N) – INTEGER array *Output*

On exit: if no constraints are violated, details of the interchanges and the block structure of D, as determined by F07QRF (ZSPTRF).

If $\mathrm{IPIV}(k) > 0$, then rows and columns k and $\mathrm{IPIV}(k)$ were interchanged, and d_{kk} is a 1 by 1 diagonal block;

if UPLO = 'U' and $\mathrm{IPIV}(k) = \mathrm{IPIV}(k-1) < 0$, then rows and columns $k-1$ and $-\mathrm{IPIV}(k)$ were interchanged and $d_{k-1:k,k-1:k}$ is a 2 by 2 diagonal block;

if UPLO = 'L' and $\mathrm{IPIV}(k) = \mathrm{IPIV}(k+1) < 0$, then rows and columns $k+1$ and $-\mathrm{IPIV}(k)$ were interchanged and $d_{k:k+1,k:k+1}$ is a 2 by 2 diagonal block.

6: B(LDB, $*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

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

On exit: if IFAIL $= 0$ or $\mathrm{N}+1$, the n by r solution matrix X.

7: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F04DJF is called.

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

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

On exit: if no constraints are violated, an estimate of the reciprocal of the condition number of the matrix A, computed as $\mathrm{RCOND} = 1/\left(\|A\|_1 \|A^{-1}\|_1\right)$.

9: ERRBND – REAL (KIND=nag_wp) *Output*

On exit: if IFAIL $= 0$ or $\mathrm{N}+1$, an estimate of the forward error bound for a computed solution $\hat{x}$, such that $\|\hat{x} - x\|_1 / \|x\|_1 \leq \mathrm{ERRBND}$, where $\hat{x}$ is a column of the computed solution returned in the array B and x is the corresponding column of the exact solution X. If RCOND is less than ***machine precision***, then ERRBND is returned as unity.

10: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL < 0 and IFAIL $\neq -999$

If IFAIL $= -i$, the ith argument had an illegal value.

IFAIL > 0 and IFAIL $\leq$ N

If IFAIL $= i$, d_{ii} is exactly zero. The factorization has been completed, but the block diagonal matrix D is exactly singular, so the solution could not be computed.

IFAIL $=$ N $+ 1$

RCOND is less than ***machine precision***, so that the matrix A is numerically singular. A solution to the equations $AX = B$ has nevertheless been computed.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

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. F04DJF uses the approximation $\|E\|_1 = \epsilon\|A\|_1$ to estimate ERRBND. See Section 4.4 of Anderson *et al.* (1999) for further details.

8 Parallelism and Performance

F04DJF is not threaded by NAG in any implementation.

F04DJF 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 packed storage scheme is illustrated by the following example when $n = 4$ and UPLO = 'U'. Two-dimensional storage of the symmetric matrix A:

$$\begin{array}{cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ & a_{22} & a_{23} & a_{24} \\ & & a_{33} & a_{34} \\ & & & a_{44} \end{array} \quad (a_{ij} = a_{ji})$$

Packed storage of the upper triangle of A:

$$\text{AP} = \begin{bmatrix} a_{11}, & a_{12}, & a_{22}, & a_{13}, & a_{23}, & a_{33}, & a_{14}, & a_{24}, & a_{34}, & a_{44} \end{bmatrix}$$

The total number of floating-point operations required to solve the equations $AX = B$ is proportional to $\left(\frac{1}{3}n^3 + 2n^2r\right)$. The condition number estimation typically requires between four and five solves and never more than eleven solves, following the factorization.

In practice the condition number estimator is very reliable, but it can underestimate the true condition number; see Section 15.3 of Higham (2002) for further details.

Routine F04CJF is for complex Hermitian matrices, and the real analogue of F04DJF is F04BJF.

10 Example

This example solves the equations

$$AX = B,$$

where A is the symmetric indefinite matrix

$$A = \begin{pmatrix} -0.56 + 0.12i & -1.54 - 2.86i & 5.32 - 1.59i & 3.80 + 0.92i \\ -1.54 - 2.86i & -2.83 - 0.03i & -3.52 + 0.58i & -7.86 - 2.96i \\ 5.32 - 1.59i & -3.52 + 0.58i & 8.86 + 1.81i & 5.14 - 0.64i \\ 3.80 + 0.92i & -7.86 - 2.96i & 5.14 - 0.64i & -0.39 - 0.71i \end{pmatrix}$$

and

$$B = \begin{pmatrix} -6.43 + 19.24i & -4.59 - 35.53i \\ -0.49 - 1.47i & 6.95 + 20.49i \\ -48.18 + 66.00i & -12.08 - 27.02i \\ -55.64 + 41.22i & -19.09 - 35.97i \end{pmatrix}.$$

An estimate of the condition number of A and an approximate error bound for the computed solutions are also printed.

10.1 Program Text

```
    Program f04djfe

!     F04DJF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: f04djf, nag_wp, x04dbf, x04ddf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
      Character (1), Parameter         :: uplo = 'U'
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: errbnd, rcond
      Integer                          :: i, ierr, ifail, j, ldb, n, nrhs
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: ap(:), b(:,:)
      Integer, Allocatable             :: ipiv(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F04DJF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, nrhs
      ldb = n
      Allocate (ap((n*(n+1))/2),b(ldb,nrhs),ipiv(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

!     Read B from data file
      Read (nin,*)(b(i,1:nrhs),i=1,n)

!     Solve the equations AX = B for X

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 1
      Call f04djf(uplo,n,nrhs,ap,ipiv,b,ldb,rcond,errbnd,ifail)

      If (ifail==0) Then
!       Print solution, estimate of condition number and approximate
!       error bound

        ierr = 0
        Call x04dbf('General',' ',n,nrhs,b,ldb,'Bracketed',' ','Solution', &
          'Integer',rlabs,'Integer',clabs,80,0,ierr)

        Write (nout,*)
        Write (nout,*) 'Estimate of condition number'
        Write (nout,99999) 1.0E0_nag_wp/rcond
        Write (nout,*)
        Write (nout,*) 'Estimate of error bound for computed solutions'
        Write (nout,99999) errbnd
      Else If (ifail==n+1) Then
!       Matrix A is numerically singular.  Print estimate of
!       reciprocal of condition number and solution
        Write (nout,*)
        Write (nout,*) 'Estimate of reciprocal of condition number'
        Write (nout,99999) rcond
        Write (nout,*)
        Flush (nout)
```

```
        ierr = 0
        Call x04dbf('General',' ',n,nrhs,b,ldb,'Bracketed',' ','Solution', &
          'Integer',rlabs,'Integer',clabs,80,0,ierr)

      Else If (ifail>0 .And. ifail<=n) Then
!       The upper triangular matrix U is exactly singular.  Print
!       details of factorization
        Write (nout,*)
        Flush (nout)

        ierr = 0
        Call x04ddf(uplo,'Non-unit diagonal',n,ap,'Bracketed',' ', &
          'Details of factorization','Integer',rlabs,'Integer',clabs,80,0, &
          ierr)

!       Print pivot indices
        Write (nout,*)
        Write (nout,*) 'Pivot indices'
        Write (nout,99998) ipiv(1:n)
      Else
        Write (nout,99997) ifail
      End If

99999 Format (8X,1P,E9.1)
99998 Format ((1X,7I11))
99997 Format (1X,' ** F04DJF returned with IFAIL = ',I5)
    End Program f04djfe
```

10.2 Program Data

```
F04DJF Example Program Data

   4              2                                            : n, nrhs

 ( -0.56,  0.12) ( -1.54, -2.86) (  5.32, -1.59) (  3.80,  0.92)
                 ( -2.83 ,-0.03) ( -3.52,  0.58) ( -7.86, -2.96)
                                 (  8.86,  1.81) (  5.14, -0.64)
                                                 ( -0.39 ,-0.71) : matrix A

 ( -6.43, 19.24) ( -4.59,-35.53)
 ( -0.49, -1.47) (  6.95, 20.49)
 (-48.18, 66.00) (-12.08,-27.02)
 (-55.64, 41.22) (-19.09,-35.97)                                 : matrix B
```

10.3 Program Results

```
 F04DJF Example Program Results

 Solution
                           1                        2
 1  (    -4.0000,     3.0000) (    -1.0000,     1.0000)
 2  (     3.0000,    -2.0000) (     3.0000,     2.0000)
 3  (    -2.0000,     5.0000) (     1.0000,    -3.0000)
 4  (     1.0000,    -1.0000) (    -2.0000,    -1.0000)

 Estimate of condition number
         2.1E+01

 Estimate of error bound for computed solutions
         2.3E-15
```