

NAG Library Routine Document

F07JVF (ZPTRFS)

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

F07JVF (ZPTRFS) computes error bounds and refines the solution to a complex system of linear equations $AX = B$, where A is an n by n Hermitian positive-definite tridiagonal matrix and X and B are n by r matrices, using the modified Cholesky factorization returned by F07JRF (ZPTTRF) and an initial solution returned by F07JSF (ZPTTRS). Iterative refinement is used to reduce the backward error as much as possible.

2 Specification

```

SUBROUTINE F07JVF(UPLO, N, NRHS, D, E, DF, EF, B, LDB, X, LDX, FERR,
1              BERR, WORK, RWORK, INFO)
    INTEGER          N, NRHS, LDB, LDX, INFO
    double precision D(*), DF(*), FERR(NRHS), BERR(NRHS), RWORK(N)
    complex*16       E(*), EF(*), B(LDB,*), X(LDX,*), WORK(N)
    CHARACTER*1      UPLO

```

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

3 Description

F07JVF (ZPTRFS) should normally be preceded by calls to F07JRF (ZPTTRF) and F07JSF (ZPTTRS). F07JRF (ZPTTRF) computes a modified Cholesky factorization of the matrix A as

$$A = LDL^H,$$

where L is a unit lower bidiagonal matrix and D is a diagonal matrix, with positive diagonal elements. F07JSF (ZPTTRS) then utilizes the factorization to compute a solution, $\hat{X}$, to the required equations. Letting $\hat{x}$ denote a column of $\hat{X}$, F07JVF (ZPTRFS) computes a *component-wise backward error*, β , the smallest relative perturbation in each element of A and b such that $\hat{x}$ is the exact solution of a perturbed system

$$(A + E)\hat{x} = b + f, \quad \text{with} \quad |e_{ij}| \leq \beta |a_{ij}|, \quad \text{and} \quad |f_j| \leq \beta |b_j|.$$

The routine also estimates a bound for the *component-wise forward error* in the computed solution defined by $\max |x_i - \hat{x}_i| / \max |\hat{x}_i|$, where x is the corresponding column of the exact solution, X .

Note that the modified Cholesky factorization of A can also be expressed as

$$A = U^H D U,$$

where U is unit upper bidiagonal.

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>

5 Parameters

- 1: UPLO – CHARACTER*1 *Input*
On entry: specifies the form of the factorization as follows:
UPLO = 'U'
 $A = U^H D U$.
UPLO = 'L'
 $A = L D L^H$.
Constraint: UPLO = 'U' or 'L'.
- 2: N – INTEGER *Input*
On entry: n , 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: $NRHS \geq 0$.
- 4: D(*) – **double precision** array *Input*
Note: the dimension of the array D must be at least $\max(1, N)$.
On entry: must contain the n diagonal elements of the matrix of A .
- 5: E(*) – **complex*16** array *Input*
Note: the dimension of the array E must be at least $\max(1, N - 1)$.
On entry: if UPLO = 'U', E must contain the $(n - 1)$ superdiagonal elements of the matrix A .
If UPLO = 'L', E must contain the $(n - 1)$ subdiagonal elements of the matrix A .
- 6: DF(*) – **double precision** array *Input*
Note: the dimension of the array DF must be at least $\max(1, N)$.
On entry: must contain the n diagonal elements of the diagonal matrix D from the LDL^T factorization of A .
- 7: EF(*) – **complex*16** array *Input*
Note: the dimension of the array EF must be at least $\max(1, N - 1)$.
On entry: if UPLO = 'U', EF must contain the $(n - 1)$ superdiagonal elements of the unit upper bidiagonal matrix U from the $U^H D U$ factorization of A .
If UPLO = 'L', EF must contain the $(n - 1)$ subdiagonal elements of the unit lower bidiagonal matrix L from the LDL^H factorization of A .
- 8: B(LDB,*) – **complex*16** array *Input*
Note: the second dimension of the array B must be at least $\max(1, NRHS)$.
On entry: the n by r matrix of right-hand sides B .

- 9: LDB – INTEGER *Input*
On entry: the first dimension of the array B as declared in the (sub)program from which F07JVF (ZPTRFS) is called.
Constraint: $LDB \geq \max(1, N)$.
- 10: X(LDX,*) – **complex*16** array *Input/Output*
Note: the second dimension of the array X must be at least $\max(1, NRHS)$.
On entry: the n by r initial solution matrix X .
On exit: the n by r refined solution matrix X .
- 11: LDX – INTEGER *Input*
On entry: the first dimension of the array X as declared in the (sub)program from which F07JVF (ZPTRFS) is called.
Constraint: $LDX \geq \max(1, N)$.
- 12: FERR(NRHS) – **double precision** array *Output*
On exit: estimate of the forward error bound for each computed solution vector, such that $\|\hat{x}_j - x_j\|_\infty / \|x_j\|_\infty \leq FERR(j)$, where $\hat{x}_j$ is the j th column of the computed solution returned in the array X and x_j is the corresponding column of the exact solution X . The estimate is almost always a slight overestimate of the true error.
- 13: BERR(NRHS) – **double precision** array *Output*
On exit: estimate of the component-wise relative backward error of each computed solution vector $\hat{x}_j$ (i.e., the smallest relative change in any element of A or B that makes $\hat{x}_j$ an exact solution).
- 14: WORK(N) – **complex*16** array *Workspace*
- 15: RWORK(N) – **double precision** array *Workspace*
- 16: INFO – INTEGER *Output*
On exit: INFO = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

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

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

$$\|E\|_\infty = O(\epsilon)\|A\|_\infty$$

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

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

where $\kappa(A) = \|A^{-1}\|_{\infty} \|A\|_{\infty}$, 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.

Routine F07JUF (ZPTCON) can be used to compute the condition number of A .

8 Further Comments

The total number of floating-point operations required to solve the equations $AX = B$ is proportional to nr . At most five steps of iterative refinement are performed, but usually only one or two steps are required.

The real analogue of this routine is F07JHF (DPTRFS).

9 Example

This example solves the equations

$$AX = B,$$

where A is the Hermitian positive-definite tridiagonal matrix

$$A = \begin{pmatrix} 16.0 & 16.0 - 16.0i & 0 & 0 \\ 16.0 + 16.0i & 41.0 & 18.0 + 9.0i & 0 \\ 0 & 18.0 - 9.0i & 46.0 & 1.0 + 4.0i \\ 0 & 0 & 1.0 - 4.0i & 21.0 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 64.0 + 16.0i & -16.0 - 32.0i \\ 93.0 + 62.0i & 61.0 - 66.0i \\ 78.0 - 80.0i & 71.0 - 74.0i \\ 14.0 - 27.0i & 35.0 + 15.0i \end{pmatrix}.$$

Estimates for the backward errors and forward errors are also output.

9.1 Program Text

```
*      F07JVF Example Program Text
*      Mark 21 Release. NAG Copyright 2004.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, NRHSMX
      PARAMETER        (NMAX=50,NRHSMX=4)
      INTEGER          LDB, LDX
      PARAMETER        (LDB=NMAX,LDX=NMAX)
*      .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, N, NRHS
*      .. Local Arrays ..
      COMPLEX *16      B(LDB,NRHSMX), E(NMAX-1), EF(NMAX-1), WORK(NMAX),
+                    X(LDX,NRHSMX)
      DOUBLE PRECISION BERR(NRHSMX), D(NMAX), DF(NMAX), FERR(NRHSMX),
+                    RWORK(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*      .. External Subroutines ..
      EXTERNAL         DCOPY, F06TFF, X04DBF, ZCOPY, ZPTRFS, ZPTTRF,
+                    ZPTTRS
*      .. Executable Statements ..
      WRITE (NOUT,*) 'F07JVF Example Program Results'
      WRITE (NOUT,*)
*      Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, NRHS
      IF (N.LE.NMAX .AND. NRHS.LE.NRHSMX) THEN
*
*          Read the lower bidiagonal part of the tridiagonal matrix A from
```

```

*      data file
*
      READ (NIN,*) (D(I),I=1,N)
      READ (NIN,*) (E(I),I=1,N-1)
*
*      Read the right hand matrix B
*
      READ (NIN,*) ((B(I,J),J=1,NRHS),I=1,N)
*
*      Copy A into DF and EF, and copy B into X
*
      CALL DCOPY(N,D,1,DF,1)
      CALL ZCOPY(N-1,E,1,EF,1)
      CALL F06TFF('General',N,NRHS,B,LDB,X,LDX)
*
*      Factorize the copy of the tridiagonal matrix A
*
      CALL ZPTTRF(N,DF,EF,INFO)
*
      IF (INFO.EQ.0) THEN
*
*          Solve the equations AX = B
*
*          CALL ZPTTRS('Lower',N,NRHS,DF,EF,X,LDX,INFO)
*
*          Improve the solution and compute error estimates
*
*          CALL ZPTRFS('Lower',N,NRHS,D,E,DF,EF,B,LDB,X,LDX,FERR,BERR,
+              WORK,RWORK,INFO)
*
*          Print the solution and the forward and backward error
*          estimates
*
*          IFAIL = 0
*          CALL X04DBF('General',' ',N,NRHS,X,LDX,'Bracketed','F7.4',
+              'Solution(s)','Integer',RLABS,'Integer',CLABS,
+              80,0,IFAIL)
*
*          WRITE (NOUT,*)
*          WRITE (NOUT,*) 'Backward errors (machine-dependent)'
*          WRITE (NOUT,99999) (BERR(J),J=1,NRHS)
*          WRITE (NOUT,*)
*          WRITE (NOUT,*)
+          'Estimated forward error bounds (machine-dependent)'
*          WRITE (NOUT,99999) (FERR(J),J=1,NRHS)
*          ELSE
*          WRITE (NOUT,99998) 'The leading minor of order ', INFO,
+          ' is not positive definite'
*          END IF
*          ELSE
*          WRITE (NOUT,*) 'NMAX and/or NRHSMX too small'
*          END IF
*
99999 FORMAT ((3X,1P,7E11.1))
99998 FORMAT (1X,A,I3,A)
      END

```

9.2 Program Data

F07JVF Example Program Data

4	2		:Values of N and NRHS
16.0	41.0	46.0	21.0 :End of diagonal D
(16.0, 16.0)	(18.0, -9.0)	(1.0, -4.0)	:End of sub-diagonal E
(64.0, 16.0)	(-16.0,-32.0)		
(93.0, 62.0)	(61.0,-66.0)		
(78.0,-80.0)	(71.0,-74.0)		
(14.0,-27.0)	(35.0, 15.0)		:End of matrix B

9.3 Program Results

F07JVF Example Program Results

Solution(s)

	1	2
1	(2.0000, 1.0000)	(-3.0000,-2.0000)
2	(1.0000, 1.0000)	(1.0000, 1.0000)
3	(1.0000,-2.0000)	(1.0000,-2.0000)
4	(1.0000,-1.0000)	(2.0000, 1.0000)

Backward errors (machine-dependent)

0.0E+00 0.0E+00

Estimated forward error bounds (machine-dependent)

9.0E-12 6.1E-12
