# NAG Library Routine Document

# F07JPF (ZPTSVX)

**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

F07JPF (ZPTSVX) uses the factorization

$$A = LDL^{\mathrm{H}}$$

to compute 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. Error bounds on the solution and a condition estimate are also provided.

## 2 Specification

```
SUBROUTINE F07JPF (FACT, N, NRHS, D, E, DF, EF, B, LDB, X, LDX, RCOND,          &
                   FERR, BERR, WORK, RWORK, INFO)
INTEGER               N, NRHS, LDB, LDX, INFO
REAL (KIND=nag_wp)    D(*), DF(*), RCOND, FERR(NRHS), BERR(NRHS), RWORK(N)
COMPLEX (KIND=nag_wp) E(*), EF(*), B(LDB,*), X(LDX,*), WORK(N)
CHARACTER(1)          FACT
```

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

## 3 Description

F07JPF (ZPTSVX) performs the following steps:

1. If FACT = 'N', the matrix $A$ is factorized as $A = LDL^{\mathrm{H}}$, where $L$ is a unit lower bidiagonal matrix and $D$ is diagonal. The factorization can also be regarded as having the form $A = U^{\mathrm{H}}DU$.
2. If the leading $i$ by $i$ principal minor is not positive definite, then the routine returns with INFO $= i$. Otherwise, the factored form of $A$ is used to estimate the condition number of the matrix $A$. If the reciprocal of the condition number is less than ***machine precision***, INFO $=$ N $+ 1$ is returned as a warning, but the routine still goes on to solve for $X$ and compute error bounds as described below.
3. The system of equations is solved for $X$ using the factored form of $A$.
4. Iterative refinement is applied to improve the computed solution matrix and to calculate error bounds and backward error estimates for it.

## 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

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

## 5 Parameters

1: FACT – CHARACTER(1) *Input*

*On entry*: specifies whether or not the factorized form of the matrix $A$ has been supplied.

FACT = 'F'
DF and EF contain the factorized form of the matrix $A$. DF and EF will not be modified.

FACT = 'N'
The matrix $A$ will be copied to DF and EF and factorized.

*Constraint*: FACT = 'F' or 'N'.

2: N – INTEGER *Input*

*On entry*: $n$, the order of the matrix $A$.

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

3: NRHS – INTEGER *Input*

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

*Constraint*: $\mathrm{NRHS} \geq 0$.

4: D($*$) – REAL (KIND=nag_wp) array *Input*

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

*On entry*: the $n$ diagonal elements of the tridiagonal matrix $A$.

5: E($*$) – COMPLEX (KIND=nag_wp) array *Input*

**Note**: the dimension of the array E must be at least $\max(1, \mathrm{N} - 1)$.

*On entry*: the $(n - 1)$ subdiagonal elements of the tridiagonal matrix $A$.

6: DF($*$) – REAL (KIND=nag_wp) array *Input/Output*

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

*On entry*: if FACT = 'F', DF must contain the $n$ diagonal elements of the diagonal matrix $D$ from the $LDL^{\mathrm{H}}$ factorization of $A$.

*On exit*: if FACT = 'N', DF contains the $n$ diagonal elements of the diagonal matrix $D$ from the $LDL^{\mathrm{H}}$ factorization of $A$.

7: EF($*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

**Note**: the dimension of the array EF must be at least $\max(1, \mathrm{N} - 1)$.

*On entry*: if FACT = 'F', EF must contain the $(n - 1)$ subdiagonal elements of the unit bidiagonal factor $L$ from the $LDL^{\mathrm{H}}$ factorization of $A$.

*On exit*: if FACT = 'N', EF contains the $(n - 1)$ subdiagonal elements of the unit bidiagonal factor $L$ from the $LDL^{\mathrm{H}}$ factorization of $A$.

8: B(LDB,$*$) – COMPLEX (KIND=nag_wp) array *Input*

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

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

9: LDB – INTEGER *Input*

*On entry*: the first dimension of the array B as declared in the (sub)program from which F07JPF (ZPTSVX) is called.

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

10: X(LDX,$*$) – COMPLEX (KIND=nag_wp) array *Output*

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

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

11: LDX – INTEGER *Input*

*On entry*: the first dimension of the array X as declared in the (sub)program from which F07JPF (ZPTSVX) is called.

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

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

*On exit*: the reciprocal condition number of the matrix $A$. If RCOND is less than the ***machine precision*** (in particular, if $\text{RCOND} = 0.0$), the matrix is singular to working precision. This condition is indicated by a return code of $\text{INFO} = \text{N} + 1$.

13: FERR(NRHS) – REAL (KIND=nag_wp) array *Output*

*On exit*: the forward error bound for each solution vector $\hat{x}_j$ (the $j$th column of the solution matrix $X$). If $x_j$ is the true solution corresponding to $\hat{x}_j$, $\text{FERR}(j)$ is an estimated upper bound for the magnitude of the largest element in $(\hat{x}_j - x_j)$ divided by the magnitude of the largest element in $\hat{x}_j$.

14: BERR(NRHS) – REAL (KIND=nag_wp) array *Output*

*On exit*: the component-wise relative backward error of each 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).

15: WORK(N) – COMPLEX (KIND=nag_wp) array *Workspace*

16: RWORK(N) – REAL (KIND=nag_wp) array *Workspace*

17: INFO – INTEGER *Output*

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

# 6 Error Indicators and Warnings

Errors or warnings detected by the routine:

$\text{INFO} < 0$

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

$\text{INFO} > 0$ and $\text{INFO} \leq \text{N}$

If $\text{INFO} = i$ and $i \leq \text{N}$, the leading minor of order $i$ of $A$ is not positive definite, so the factorization could not be completed, and the solution has not been computed. $\text{RCOND} = 0.0$ is returned.

$\text{INFO} = \text{N} + 1$

The diagonal matrix $D$ is nonsingular, but RCOND is less than ***machine precision***, meaning that the matrix is singular to working precision. Nevertheless, the solution and error bounds are computed because there are a number of situations where the computed solution can be more accurate than the value of RCOND would suggest.

## 7 Accuracy

For each right-hand side vector $b$, the computed solution $\hat{x}$ is the exact solution of a perturbed system of equations $(A+E)\hat{x}=b$, where

$$|E| \leq c(n)\epsilon \left|R^{\mathrm{T}}\right||R|, \text{ where } R = D^{\frac{1}{2}}U,$$

$c(n)$ is a modest linear function of $n$, and $\epsilon$ is the ***machine precision***. See Section 10.1 of Higham (2002) for further details.

If $x$ is the true solution, then the computed solution $\hat{x}$ satisfies a forward error bound of the form

$$\frac{\|x-\hat{x}\|_\infty}{\|\hat{x}\|_\infty} \leq w_c \operatorname{cond}(A,\hat{x},b)$$

where $\operatorname{cond}(A,\hat{x},b) = \left\|\left|A^{-1}\right|(|A||\hat{x}|+|b|)\right\|_\infty / \|\hat{x}\|_\infty \leq \operatorname{cond}(A) = \left\|\left|A^{-1}\right||A|\right\|_\infty \leq \kappa_\infty(A)$. If $\hat{x}$ is the $j$th column of $X$, then $w_c$ is returned in $\mathrm{BERR}(j)$ and a bound on $\|x-\hat{x}\|_\infty/\|\hat{x}\|_\infty$ is returned in $\mathrm{FERR}(j)$. See Section 4.4 of Anderson *et al.* (1999) for further details.

## 8 Further Comments

The number of floating point operations required for the factorization, and for the estimation of the condition number of $A$ is proportional to $n$. The number of floating point operations required for the solution of the equations, and for the estimation of the forward and backward error is proportional to $nr$, where $r$ is the number of right-hand sides.

The condition estimation is based upon Equation (15.11) of Higham (2002). For further details of the error estimation, see Section 4.4 of Anderson *et al.* (1999).

The real analogue of this routine is F07JBF (DPTSVX).

## 9 Example

This example solves the equations

$$AX = B,$$

where $A$ is the Hermitian positive definite tridiagonal matrix

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

and

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

Error estimates for the solutions and an estimate of the reciprocal of the condition number of $A$ are also output.

## 9.1 Program Text

```
    Program f07jpfe

!     F07JPF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, zptsvx
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter              :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)              :: rcond
      Integer                         :: i, ifail, info, ldb, ldx, n, nrhs
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: b(:,:), e(:), ef(:), work(:), x(:,:)
      Real (Kind=nag_wp), Allocatable  :: berr(:), d(:), df(:), ferr(:),      &
                                         rwork(:)
      Character (1)                   :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F07JPF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, nrhs
      ldb = n
      ldx = n
      Allocate (b(ldb,nrhs),e(n-1),ef(n-1),work(n),x(ldx,nrhs),berr(nrhs), &
        d(n),df(n),ferr(nrhs),rwork(n))

!     Read the lower bidiagonal part of the tridiagonal matrix A and
!     the right hand side b from data file

      Read (nin,*) d(1:n)
      Read (nin,*) e(1:n-1)
      Read (nin,*)(b(i,1:nrhs),i=1,n)

!     Solve the equations AX = B for X
!     The NAG name equivalent of zptsvx is f07jpf
      Call zptsvx('Not factored',n,nrhs,d,e,df,ef,b,ldb,x,ldx,rcond,ferr,berr, &
        work,rwork,info)

      If ((info==0) .Or. (info==n+1)) Then

!       Print solution, error bounds and condition number

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        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(1:nrhs)
        Write (nout,*)
        Write (nout,*) 'Estimated forward error bounds (machine-dependent)'
        Write (nout,99999) ferr(1:nrhs)
        Write (nout,*)
        Write (nout,*) 'Estimate of reciprocal condition number'
        Write (nout,99999) rcond

        If (info==n+1) Then
          Write (nout,*)
          Write (nout,*) 'The matrix A is singular to working precision'
        End If
      Else
```

```
        Write (nout,99998) 'The leading minor of order ', info, &
          ' is not positive definite'
      End If

99999 Format (1X,1P,7E11.1)
99998 Format (1X,A,I3,A)
    End Program f07jpfe
```

## 9.2 Program Data

```
F07JPF 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

```
 F07JPF 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

 Estimate of reciprocal condition number
     1.1E-04
```