NAG Fortran Library Routine Document

F07JBF (DPTSVX)

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

F07JBF (DPTSVX) uses the factorization

$$A = LDL^T$$

to compute the solution to a real system of linear equations

$$AX = B,$$

where A is an n by n symmetric 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 F07JBF (FACT, N, NRHS, D, E, DF, EF, B, LDB, X, LDX, RCOND,
     1                   FERR, BERR, WORK, INFO)
      INTEGER            N, NRHS, LDB, LDX, INFO
      double precision   D(*), E(*), DF(*), EF(*), B(LDB,*), X(LDX,*), RCOND,
     1                   FERR(*), BERR(*), WORK(*)
      CHARACTER*1        FACT
```

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

3 Description

The following steps are performed:

1. If FACT $=$ 'N', the matrix A is factorized as $A = LDL^T$, where L is a unit lower bidiagonal matrix and D is diagonal. The factorization can also be regarded as having the form $A = U^T DU$.
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 URL: 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 factored form of A has been supplied on entry:

if FACT = 'F' on entry, DF and EF contain the factored form of A. D, E, DF and EF will not be modified;
if FACT = 'N', the matrix A will be copied to DF and EF and factored.

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($*$) – ***double precision*** 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($*$) – ***double precision*** array *Input*

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

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

6: DF($*$) – ***double precision*** array *Input/Output*

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

On entry: if FACT = 'F', DF contains the n diagonal elements of the diagonal matrix D from the LDL^T factorization of A.

On exit: if FACT = 'N', DF contains the n diagonal elements of the diagonal matrix D from the LDL^T factorization of A.

7: EF($*$) – ***double precision*** 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 contains the $(n-1)$ sub-diagonal elements of the unit bidiagonal factor L from the LDL^T factorization of A.

On exit: if FACT = 'N', EF contains the $(n-1)$ sub-diagonal elements of the unit bidiagonal factor L from the LDL^T factorization of A.

8: B(LDB,$*$) – ***double precision*** 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 F07JBF (DPTSVX) is called.

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

10: X(LDX,$*$) – ***double precision*** array *Output*

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

On exit: if INFO $= 0$ of INFO $= \mathrm{N} + 1$, 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 F07JBF (DPTSVX) is called.

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

12: RCOND – ***double precision*** *Output*

On exit: the reciprocal condition number of the matrix A. If RCOND is less than the ***machine precision*** (in particular, if RCOND $= 0$), the matrix is singular to working precision. This condition is indicated by a return code of INFO > 0.

13: FERR($*$) – ***double precision*** array *Output*

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

On exit: the forward error bound for each solution vector X(j) (the jth column of the solution matrix X). If x_j is the true solution corresponding to X(j), FERR(j) is an estimated upper bound for the magnitude of the largest element in (X$(j) - x_j$) divided by the magnitude of the largest element in X(j).

14: BERR($*$) – ***double precision*** array *Output*

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

On exit: the componentwise relative backward error of each solution vector X(j) (i.e., the smallest relative change in any element of A or B that makes X(j) an exact solution).

15: WORK($*$) – ***double precision*** array *Workspace*

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

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 ith argument had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0

If INFO $= i$ and $i \leq \mathrm{N}$, 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. RCOND $= 0$ is returned.

If INFO $= i$ and $i = \mathrm{N} + 1$, U 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

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

$c(n)$ is a modest linear function of n, and ϵ 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 $\operatorname{cond}(A,\hat{x},b) = \left\||A^{-1}|(|A||\hat{x}|+|b|)\right\|_\infty/\|\hat{x}\|_\infty \leq \operatorname{cond}(A) = \left\||A^{-1}||A|\right\|_\infty \leq \kappa_\infty(A)$. If $\hat{x}$ is the jth 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 $n \times r$, 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 complex analogue of this routine is F07JPF (ZPTSVX).

9 Example

To solve the equations

$$Ax=b,$$

where A is the symmetric positive-definite tridiagonal matrix

$$A = \begin{pmatrix} 4.0 & -2.0 & 0 & 0 & 0 \\ -2.0 & 10.0 & -6.0 & 0 & 0 \\ 0 & -6.0 & 29.0 & 15.0 & 0 \\ 0 & 0 & 15.0 & 25.0 & 8.0 \\ 0 & 0 & 0 & 8.0 & 5.0 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 6.0 & 10.0 \\ 9.0 & 4.0 \\ 2.0 & 9.0 \\ 14.0 & 65.0 \\ 7.0 & 23.0 \end{pmatrix}.$$

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

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F07JBF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX
      PARAMETER        (NMAX=8)
      INTEGER          LDB, LDX, NRHSMX
      PARAMETER        (LDB=NMAX,LDX=NMAX,NRHSMX=NMAX)
*     .. Local Scalars ..
      DOUBLE PRECISION RCOND
      INTEGER          I, IFAIL, INFO, J, N, NRHS
*     .. Local Arrays ..
      DOUBLE PRECISION B(LDB,NRHSMX), BERR(NRHSMX), D(NMAX), DF(NMAX),
     +                 E(NMAX-1), EF(NMAX-1), FERR(NRHSMX),
     +                 WORK(2*NMAX), X(LDX,NRHSMX)
*     .. External Subroutines ..
      EXTERNAL         DPTSVX, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F07JBF 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 and
*        the right hand side b from data file
*
         READ (NIN,*) (D(I),I=1,N)
         READ (NIN,*) (E(I),I=1,N-1)
         READ (NIN,*) ((B(I,J),J=1,NRHS),I=1,N)
*
*        Solve the equations AX = B for X
*
         CALL DPTSVX('Not factored',N,NRHS,D,E,DF,EF,B,LDB,X,LDX,RCOND,
     +               FERR,BERR,WORK,INFO)
*
         IF ((INFO.EQ.0) .OR. (INFO.EQ.N+1)) THEN
*
*           Print solution, error bounds and condition number
*
            IFAIL = 0
            CALL X04CAF('General',' ',N,NRHS,X,LDX,'Solution(s)',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)
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Estimate of reciprocal condition number'
            WRITE (NOUT,99999) RCOND
*
            IF (INFO.EQ.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
```

```
      ELSE
         WRITE (NOUT,*) 'NMAX and/or NRHSMX too small'
      END IF
      STOP
*
99999 FORMAT (1X,1P,7E11.1)
99998 FORMAT (1X,A,I3,A)
      END
```

9.2 Program Data

```
F07JBF Example Program Data
  5     2                         :Values of N and NRHS
  4.0  10.0  29.0  25.0   5.0 :End of diagonal D
 -2.0  -6.0  15.0   8.0       :End of sub-diagonal E
  6.0  10.0
  9.0   4.0
  2.0   9.0
 14.0  65.0
  7.0  23.0                   :End of matrix B
```

9.3 Program Results

```
 F07JBF Example Program Results

 Solution(s)
             1          2
 1      2.5000     2.0000
 2      2.0000    -1.0000
 3      1.0000    -3.0000
 4     -1.0000     6.0000
 5      3.0000    -5.0000

 Backward errors (machine-dependent)
     0.0E+00    4.9E-17

 Estimated forward error bounds (machine-dependent)
     2.4E-14    4.7E-14

 Estimate of reciprocal condition number
     9.5E-03
```