

NAG Library Routine Document

F07JHF (DPTRFS)

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

F07JHF (DPTRFS) computes error bounds and refines 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, using the modified Cholesky factorization returned by F07JDF (DPTTRF) and an initial solution returned by F07JEF (DPTTRS). Iterative refinement is used to reduce the backward error as much as possible.

2 Specification

```
SUBROUTINE F07JHF(N, NRHS, D, E, DF, EF, B, LDB, X, LDX, FERR, BERR,
1                      WORK, INFO)
    INTEGER          N, NRHS, LDB, LDX, INFO
    double precision D(*), E(*), DF(*), EF(*), B(LDB,*), X(LDX,*),
1                      FERR(NRHS), BERR(NRHS), WORK(2*N)
```

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

3 Description

F07JHF (DPTRFS) should normally be preceded by calls to F07JDF (DPTTRF) and F07JEF (DPTTRS). F07JDF (DPTTRF) computes a modified Cholesky factorization of the matrix A as

$$A = LDL^T,$$

where L is a unit lower bidiagonal matrix and D is a diagonal matrix, with positive diagonal elements. F07JEF (DPTTRS) then utilizes the factorization to compute a solution, $\hat{X}$, to the required equations. Letting $\hat{x}$ denote a column of $\hat{X}$, F07JHF (DPTRFS) 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

$$A = U^T D U,$$

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: N – INTEGER *Input*
On entry: n , the order of the matrix A .
Constraint: $N \geq 0$.

- 2: NRHS – INTEGER *Input*
On entry: r , the number of right-hand sides, i.e., the number of columns of the matrix B .
Constraint: $\text{NRHS} \geq 0$.

- 3: 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 .

- 4: E(*) – **double precision** array *Input*
Note: the dimension of the array E must be at least $\max(1, N - 1)$.
On entry: must contain the $(n - 1)$ subdiagonal elements of the matrix A .

- 5: 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 .

- 6: EF(*) – **double precision** array *Input*
Note: the dimension of the array EF must be at least $\max(1, N)$.
On entry: must contain the $(n - 1)$ subdiagonal elements of the unit bidiagonal matrix L from the LDL^T factorization of A .

- 7: B(LDB,*) – **double precision** array *Input*
Note: the second dimension of the array B must be at least $\max(1, \text{NRHS})$.
On entry: the n by r matrix of right-hand sides B .

- 8: LDB – INTEGER *Input*
On entry: the first dimension of the array B as declared in the (sub)program from which F07JHF (DPTRFS) is called.
Constraint: $\text{LDB} \geq \max(1, N)$.

- 9: X(LDX,*) – **double precision** array *Input/Output*
Note: the second dimension of the array X must be at least $\max(1, \text{NRHS})$.
On entry: the n by r initial solution matrix X .
On exit: the n by r refined solution matrix X .

- 10: LDX – INTEGER Input
On entry: the first dimension of the array X as declared in the (sub)program from which F07JHF (DPTRFS) is called.
Constraint: $LDX \geq \max(1, N)$.
- 11: FERR(NRHS) – **double precision** array Output
On exit: estimate of the forward error bound for each computed solution vector, such that $\|\hat{x}_j - x_j\|_\infty / \|x_j\|_\infty \leq \text{FERR}(j)$, 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.
- 12: 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).
- 13: WORK($2 \times N$) – **double precision** array Workspace
- 14: 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 F07JGF (DPTCON) 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 complex analogue of this routine is F07JVF (ZPTRFS).

9 Example

This example solves the equations

$$AX = B,$$

where A is the symmetric positive-definite tridiagonal matrix

$$A = \begin{pmatrix} 4.0 & -2.0 & 0 & 0 & 0 \\ -2.0 & 10.0 & -6.0 & 0 & 0 \\ 0 & -6.0 & 29.0 & 15.0 & 0 \\ 0 & 0 & 15.0 & 25.0 & 8.0 \\ 0 & 0 & 0 & 8.0 & 5.0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 6.0 & 10.0 \\ 9.0 & 4.0 \\ 2.0 & 9.0 \\ 14.0 & 65.0 \\ 7.0 & 23.0 \end{pmatrix}.$$

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

9.1 Program Text

```
*      F07JHF 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 ..
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          DCOPY, DPTRFS, DPTTRF, DPTTRS, F06QFF, X04CAF
*      .. Executable Statements ..
WRITE (NOUT,*) 'F07JHF 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 DCOPY(N-1,E,1,EF,1)
CALL F06QFF('General',N,NRHS,B,LDB,X,LDX)
*
*      Factorize the copy of the tridiagonal matrix A
*
CALL DPTTRF(N,DF,EF,INFO)
*
IF (INFO.EQ.0) THEN
*
*      Solve the equations AX = B
*
CALL DPTTRS(N,NRHS,DF,EF,X,LDX,INFO)
*

```

```

*           Improve the solution and compute error estimates
*
      CALL DPTRFS(N,NRHS,D,E,DF,EF,B,LDB,X,LDX,FERR,BERR,WORK,
+             INFO)
*
*           Print the solution and the forward and backward error
*           estimates
*
      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)
      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

F07JHF 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 super-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

F07JHF 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	6.0E-17
---------	---------

Estimated forward error bounds (machine-dependent)

2.4E-14	4.9E-14
---------	---------