

# NAG Library Routine Document

## F07PHF (DSPRFS)

**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

F07PHF (DSPRFS) returns error bounds for the solution of a real symmetric indefinite system of linear equations with multiple right-hand sides,  $AX = B$ , using packed storage. It improves the solution by iterative refinement, in order to reduce the backward error as much as possible.

### 2 Specification

```

SUBROUTINE F07PHF(UPLO, N, NRHS, AP, AFP, IPIV, B, LDB, X, LDX, FERR,
1          BERR, WORK, IWORK, INFO)

INTEGER          N, NRHS, IPIV(*), LDB, LDX, IWORK(N), INFO
double precision AP(*), AFP(*), B(LDB,*), X(LDX,*), FERR(NRHS),
1          BERR(NRHS), WORK(3*N)
CHARACTER*1      UPLO

```

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

### 3 Description

F07PHF (DSPRFS) returns the backward errors and estimated bounds on the forward errors for the solution of a real symmetric indefinite system of linear equations with multiple right-hand sides  $AX = B$ , using packed storage. The routine handles each right-hand side vector (stored as a column of the matrix  $B$ ) independently, so we describe the function of F07PHF (DSPRFS) in terms of a single right-hand side  $b$  and solution  $x$ .

Given a computed solution  $x$ , the routine computes the *component-wise backward error*  $\beta$ . This is the size of the smallest relative perturbation in each element of  $A$  and  $b$  such that  $x$  is the exact solution of a perturbed system

$$(\mathbf{A} + \delta\mathbf{A})x = b + \delta b$$

$$|\delta a_{ij}| \leq \beta |a_{ij}| \quad \text{and} \quad |\delta b_i| \leq \beta |b_i|.$$

Then the routine estimates a bound for the *component-wise forward error* in the computed solution, defined by:

$$\max_i |x_i - \hat{x}_i| / \max_i |x_i|$$

where  $\hat{x}$  is the true solution.

For details of the method, see the F07 Chapter Introduction.

## 4 References

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

## 5 Parameters

- 1:    UPLO – CHARACTER\*1 *Input*  
*On entry:* indicates whether the upper or lower triangular part of  $A$  is stored and how  $A$  is to be factorized.  
UPLO = 'U'  
The upper triangular part of  $A$  is stored and  $A$  is factorized as  $PUDU^T P^T$ , where  $U$  is upper triangular.  
UPLO = 'L'  
The lower triangular part of  $A$  is stored and  $A$  is factorized as  $PLDL^T P^T$ , where  $L$  is lower triangular.  
*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.  
*Constraint:*  $NRHS \geq 0$ .
- 4:    AP(\*) – **double precision** array *Input*  
**Note:** the dimension of the array AP must be at least  $\max(1, N \times (N + 1)/2)$ .  
*On entry:* the  $n$  by  $n$  original symmetric matrix  $A$  as supplied to F07PDF (DSPTRF).
- 5:    AFP(\*) – **double precision** array *Input*  
**Note:** the dimension of the array AFP must be at least  $\max(1, N \times (N + 1)/2)$ .  
*On entry:* the factorization of  $A$  stored in packed form, as returned by F07PDF (DSPTRF).
- 6:    IPIV(\*) – INTEGER array *Input*  
**Note:** the dimension of the array IPIV must be at least  $\max(1, N)$ .  
*On entry:* details of the interchanges and the block structure of  $D$ , as returned by F07PDF (DSPTRF).
- 7:    B(LDB,\*) – **double precision** array *Input*  
**Note:** the second dimension of the array B must be at least  $\max(1, NRHS)$ .  
*On entry:* the  $n$  by  $r$  right-hand side matrix  $B$ .
- 8:    LDB – INTEGER *Input*  
*On entry:* the first dimension of the array B as declared in the (sub)program from which F07PHF (DSPRFS) is called.  
*Constraint:*  $LDB \geq \max(1, N)$ .

- 9:  $X(LDX,*)$  – **double precision** array *Input/Output*  
**Note:** the second dimension of the array  $X$  must be at least  $\max(1, NRHS)$ .  
*On entry:* the  $n$  by  $r$  solution matrix  $X$ , as returned by F07PEF (DSPTRS).  
*On exit:* the improved solution matrix  $X$ .
- 10:  $LDX$  – INTEGER *Input*  
*On entry:* the first dimension of the array  $X$  as declared in the (sub)program from which F07PHF (DSPRFS) is called.  
*Constraint:*  $LDX \geq \max(1, N)$ .
- 11:  $FERR(NRHS)$  – **double precision** array *Output*  
*On exit:*  $FERR(j)$  contains an estimated error bound for the  $j$ th solution vector, that is, the  $j$ th column of  $X$ , for  $j = 1, 2, \dots, r$ .
- 12:  $BERR(NRHS)$  – **double precision** array *Output*  
*On exit:*  $BERR(j)$  contains the component-wise backward error bound  $\beta$  for the  $j$ th solution vector, that is, the  $j$ th column of  $X$ , for  $j = 1, 2, \dots, r$ .
- 13:  $WORK(3 \times N)$  – **double precision** array *Workspace*
- 14:  $IWORK(N)$  – INTEGER array *Workspace*
- 15:  $INFO$  – 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 parameter had an illegal value. An explanatory message is output, and execution of the program is terminated.

## 7 Accuracy

The bounds returned in  $FERR$  are not rigorous, because they are estimated, not computed exactly; but in practice they almost always overestimate the actual error.

## 8 Further Comments

For each right-hand side, computation of the backward error involves a minimum of  $4n^2$  floating-point operations. Each step of iterative refinement involves an additional  $6n^2$  operations. At most five steps of iterative refinement are performed, but usually only 1 or 2 steps are required.

Estimating the forward error involves solving a number of systems of linear equations of the form  $Ax = b$ ; the number is usually 4 or 5 and never more than 11. Each solution involves approximately  $2n^2$  operations.

The complex analogues of this routine are F07PVF (ZHPRFS) for Hermitian matrices and F07QVF (ZSPRFS) for symmetric matrices.

## 9 Example

This example solves the system of equations  $AX = B$  using iterative refinement and to compute the forward and backward error bounds, where

$$A = \begin{pmatrix} 2.07 & 3.87 & 4.20 & -1.15 \\ 3.87 & -0.21 & 1.87 & 0.63 \\ 4.20 & 1.87 & 1.15 & 2.06 \\ -1.15 & 0.63 & 2.06 & -1.81 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} -9.50 & 27.85 \\ -8.38 & 9.90 \\ -6.07 & 19.25 \\ -0.96 & 3.93 \end{pmatrix}.$$

Here  $A$  is symmetric indefinite, stored in packed form, and must first be factorized by F07PDF (DSPTRF).

### 9.1 Program Text

```
*      F07PHF Example Program Text
*      Mark 15 Release. NAG Copyright 1991.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, NRHMAX, LDB, LDX
      PARAMETER        (NMAX=8,NRHMAX=NMAX,LDB=NMAX,LDX=NMAX)
*      .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, N, NRHS
      CHARACTER        UPLO
*      .. Local Arrays ..
      DOUBLE PRECISION AFP(NMAX*(NMAX+1)/2), AP(NMAX*(NMAX+1)/2),
+      B(LDB,NRHMAX), BERR(NRHMAX), FERR(NRHMAX),
+      WORK(3*NMAX), X(LDX,NMAX)
      INTEGER          IPIV(NMAX), IWORK(NMAX)
*      .. External Subroutines ..
      EXTERNAL         DSPRFS, DSPTRF, DSPTRS, F06QFF, X04CAF
*      .. Executable Statements ..
      WRITE (NOUT,*) 'F07PHF Example Program Results'
*      Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, NRHS
      IF (N.LE.NMAX .AND. NRHS.LE.NRHMAX) THEN
*
*      Read A and B from data file, and copy A to AFP and B to X
*
      READ (NIN,*) UPLO
      IF (UPLO.EQ.'U') THEN
        READ (NIN,*) ((AP(I+J*(J-1)/2),J=I,N),I=1,N)
      ELSE IF (UPLO.EQ.'L') THEN
        READ (NIN,*) ((AP(I+(2*N-J)*(J-1)/2),J=1,I),I=1,N)
      END IF
      READ (NIN,*) ((B(I,J),J=1,NRHS),I=1,N)
      DO 20 I = 1, N*(N+1)/2
        AFP(I) = AP(I)
20    CONTINUE
*
*      CALL F06QFF('General',N,NRHS,B,LDB,X,LDX)
*
*      Factorize A in the array AFP
*
      CALL DSPTRF(UPLO,N,AFP,IPIV,INFO)
*
      WRITE (NOUT,*)
      IF (INFO.EQ.0) THEN
*
*      Compute solution in the array X
*
      CALL DSPTRS(UPLO,N,NRHS,AFP,IPIV,X,LDX,INFO)
*
*      Improve solution, and compute backward errors and
*      estimated bounds on the forward errors
*
      CALL DSPRFS(UPLO,N,NRHS,AP,AFP,IPIV,B,LDB,X,LDX,FERR,BERR,
+      WORK,IWORK,INFO)
```

```

*
*      Print solution
*
*      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,*)
+      'Estimated forward error bounds (machine-dependent)'
*      WRITE (NOUT,99999) (FERR(J),J=1,NRHS)
*      ELSE
*      WRITE (NOUT,*) 'The factor D is singular'
*      END IF
*      END IF
*
99999 FORMAT ((3X,1P,7E11.1))
END

```

## 9.2 Program Data

F07PHF Example Program Data

|       |       |      |       |  |                       |
|-------|-------|------|-------|--|-----------------------|
| 4     | 2     |      |       |  | :Values of N and NRHS |
| 'L'   |       |      |       |  | :Value of UPLO        |
| 2.07  |       |      |       |  |                       |
| 3.87  | -0.21 |      |       |  |                       |
| 4.20  | 1.87  | 1.15 |       |  |                       |
| -1.15 | 0.63  | 2.06 | -1.81 |  | :End of matrix A      |
| -9.50 | 27.85 |      |       |  |                       |
| -8.38 | 9.90  |      |       |  |                       |
| -6.07 | 19.25 |      |       |  |                       |
| -0.96 | 3.93  |      |       |  | :End of matrix B      |

## 9.3 Program Results

F07PHF Example Program Results

Solution(s)

|   | 1       | 2      |
|---|---------|--------|
| 1 | -4.0000 | 1.0000 |
| 2 | -1.0000 | 4.0000 |
| 3 | 2.0000  | 3.0000 |
| 4 | 5.0000  | 2.0000 |

Backward errors (machine-dependent)

|         |         |
|---------|---------|
| 9.9E-17 | 8.3E-17 |
|---------|---------|

Estimated forward error bounds (machine-dependent)

|         |         |
|---------|---------|
| 2.4E-14 | 3.2E-14 |
|---------|---------|