NAG Library Routine Document

F07GAF (DPPSV)

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

F07GAF (DPPSV) computes the solution to a real system of linear equations

$$AX = B,$$

where A is an n by n symmetric positive definite matrix stored in packed format and X and B are n by r matrices.

2 Specification

```
SUBROUTINE F07GAF (UPLO, N, NRHS, AP, B, LDB, INFO)

INTEGER            N, NRHS, LDB, INFO
REAL (KIND=nag_wp) AP(*), B(LDB,*)
CHARACTER(1)       UPLO
```

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

3 Description

F07GAF (DPPSV) uses the Cholesky decomposition to factor A as $A = U^{\mathrm{T}}U$ if UPLO = 'U' or $A = LL^{\mathrm{T}}$ if UPLO = 'L', where U is an upper triangular matrix and L is a lower triangular matrix. The factored form of A is then used to solve the system of equations $AX = B$.

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

5 Parameters

1: UPLO – CHARACTER(1) *Input*

On entry: if UPLO = 'U', the upper triangle of A is stored.

If UPLO = 'L', the lower triangle of A is stored.

Constraint: UPLO = 'U' or 'L'.

2: N – INTEGER *Input*

On entry: n, the number of linear equations, i.e., 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: AP($*$) – REAL (KIND=nag_wp) array *Input/Output*

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

On entry: the n by n symmetric matrix A, packed by columns.

More precisely,

if UPLO = 'U', the upper triangle of A must be stored with element A_{ij} in $\mathrm{AP}(i+j(j-1)/2)$ for $i \leq j$;

if UPLO = 'L', the lower triangle of A must be stored with element A_{ij} in $\mathrm{AP}(i+(2n-j)(j-1)/2)$ for $i \geq j$.

On exit: if INFO = 0, the factor U or L from the Cholesky factorization $A = U^{\mathrm{T}}U$ or $A = LL^{\mathrm{T}}$, in the same storage format as A.

5: B(LDB,$*$) – REAL (KIND=nag_wp) array *Input/Output*

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.

On exit: if INFO = 0, the n by r solution matrix X.

6: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07GAF (DPPSV) is called.

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

7: 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 $= i$, 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.

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

$$\|E\|_1 = O(\epsilon)\|A\|_1$$

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

$$\frac{\|\hat{x}-x\|_1}{\|x\|_1} \leq \kappa(A)\frac{\|E\|_1}{\|A\|_1},$$

where $\kappa(A) = \left\|A^{-1}\right\|_1 \|A\|_1$, 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.

F07GBF (DPPSVX) is a comprehensive LAPACK driver that returns forward and backward error bounds and an estimate of the condition number. Alternatively, F04BEF solves $Ax = b$ and returns a forward error bound and condition estimate. F04BEF calls F07GAF (DPPSV) to solve the equations.

8 Further Comments

The total number of floating point operations is approximately $\frac{1}{3}n^3 + 2n^2r$, where r is the number of right-hand sides.

The complex analogue of this routine is F07GNF (ZPPSV).

9 Example

This example solves the equations

$$Ax = b,$$

where A is the symmetric positive definite matrix

$$A = \begin{pmatrix} 4.16 & -3.12 & 0.56 & -0.10 \\ -3.12 & 5.03 & -0.83 & 1.18 \\ 0.56 & -0.83 & 0.76 & 0.34 \\ -0.10 & 1.18 & 0.34 & 1.18 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 8.70 \\ -13.35 \\ 1.89 \\ -4.14 \end{pmatrix}.$$

Details of the Cholesky factorization of A are also output.

9.1 Program Text

```
    Program f07gafe

!     F07GAF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dppsv, nag_wp, x04ccf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
      Character (1), Parameter         :: uplo = 'U'
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, j, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: ap(:), b(:)
!     .. Executable Statements ..
      Write (nout,*) 'F07GAF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n

      Allocate (ap((n*(n+1))/2),b(n))

!     Read the upper or lower triangular part of the matrix A from
!     data file

      If (uplo=='U') Then
        Read (nin,*)((ap(i+(j*(j-1))/2),j=i,n),i=1,n)
      Else If (uplo=='L') Then
        Read (nin,*)((ap(i+((2*n-j)*(j-1))/2),j=1,i),i=1,n)
      End If

!     Read b from data file

      Read (nin,*) b(1:n)
```

```
!     Solve the equations Ax = b for x
!     The NAG name equivalent of dppsv is f07gaf
      Call dppsv(uplo,n,1,ap,b,n,info)

      If (info==0) Then

!       Print solution

        Write (nout,*) 'Solution'
        Write (nout,99999) b(1:n)

!       Print details of factorization

        Write (nout,*)
        Flush (nout)

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04ccf(uplo,'Non-unit diagonal',n,ap,'Cholesky factor',ifail)

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

99999 Format ((3X,7F11.4))
99998 Format (1X,A,I3,A)
    End Program f07gafe
```

9.2 Program Data

```
F07GAF Example Program Data
  4                          :Value of N
  4.16  -3.12   0.56  -0.10
         5.03  -0.83   1.18
                0.76   0.34
                       1.18 :End of matrix A
  8.70 -13.35   1.89  -4.14 :End of vector b
```

9.3 Program Results

```
 F07GAF Example Program Results

 Solution
        1.0000    -1.0000     2.0000    -3.0000

 Cholesky factor
             1          2          3          4
 1      2.0396    -1.5297     0.2746    -0.0490
 2                 1.6401    -0.2500     0.6737
 3                            0.7887     0.6617
 4                                       0.5347
```