NAG Library Routine Document

F07FQF (ZCPOSV)

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

F07FQF (ZCPOSV) uses the Cholesky factorization

$$A = U^{\mathrm{H}}U \qquad \text{or} \qquad A = LL^{\mathrm{H}}$$

to compute the solution to a complex system of linear equations

$$AX = B,$$

where A is an n by n Hermitian positive definite matrix and X and B are n by r matrices.

2 Specification

```
SUBROUTINE F07FQF (UPLO, N, NRHS, A, LDA, B, LDB, X, LDX, WORK, SWORK,          &
                   RWORK, ITER, INFO)

INTEGER                N, NRHS, LDA, LDB, LDX, ITER, INFO
REAL (KIND=nag_wp)     RWORK(N)
COMPLEX (KIND=nag_wp) A(LDA,*), B(LDB,*), X(LDX,*), WORK(N,NRHS)
COMPLEX (KIND=nag_rp) SWORK(N*(N+NRHS))
CHARACTER(1)           UPLO
```

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

3 Description

F07FQF (ZCPOSV) first attempts to factorize the matrix in reduced precision and use this factorization within an iterative refinement procedure to produce a solution with full precision normwise backward error quality (see below). If the approach fails the method switches to a full precision factorization and solve.

The iterative refinement can be more efficient than the corresponding direct full precision algorithm. Since the strategy implemented by F07FQF (ZCPOSV) must perform iterative refinement on each right-hand side, any efficiency gains will reduce as the number of right-hand sides increases. Conversely, as the matrix size increases the cost of these iterative refinements become less significant relative to the cost of factorization. Thus, any efficiency gains will be greatest for a very small number of right-hand sides and for large matrix sizes. The cut-off values for the number of right-hand sides and matrix size, for which the iterative refinement strategy performs better, depends on the relative performance of the reduced and full precision factorization and back-substitution. F07FQF (ZCPOSV) always attempts the iterative refinement strategy first; you are advised to compare the performance of F07FQF (ZCPOSV) with that of its full precision counterpart F07FNF (ZPOSV) to determine whether this strategy is worthwhile for your particular problem dimensions.

The iterative refinement process is stopped if ITER > 30 where ITER is the number of iterations carried out thus far. The process is also stopped if for all right-hand sides we have

$$\|resid\| < \sqrt{\mathrm{N}}\|x\|\|A\|\epsilon,$$

where $\|resid\|$ is the ∞-norm of the residual, $\|x\|$ is the ∞-norm of the solution, $\|A\|$ is the ∞-norm of the matrix A and ϵ is the ***machine precision*** returned by X02AJF.

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: UPLO – CHARACTER(1) *Input*

On entry: specifies whether the upper or lower triangular part of A is stored.

UPLO = 'U'
The upper triangular part of A is stored.

UPLO = 'L'
The lower triangular part 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 ≥ 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 ≥ 0.

4: A(LDA,$*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

On entry: the n by n Hermitian positive definite matrix A.

If UPLO = 'U', the upper triangular part of A must be stored and the elements of the array below the diagonal are not referenced.

If UPLO = 'L', the lower triangular part of A must be stored and the elements of the array above the diagonal are not referenced.

On exit: if iterative refinement has been successfully used (INFO $= 0$ and ITER ≥ 0, see description below), then A is unchanged. If full precision factorization has been used (INFO $= 0$ and ITER < 0, see description below), then the array A contains the factor U or L from the Cholesky factorization $A = U^{\mathrm{H}}U$ or $A = LL^{\mathrm{H}}$.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F07FQF (ZCPOSV) is called.

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

6: 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 right-hand side matrix B.

7: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07FQF (ZCPOSV) is called.

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

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

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

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

9: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which F07FQF (ZCPOSV) is called.

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

10: WORK(N,NRHS) – COMPLEX (KIND=nag_wp) array *Workspace*

11: SWORK$(\mathrm{N} \times (\mathrm{N} + \mathrm{NRHS}))$ – COMPLEX (KIND=nag_rp) array *Workspace*

Note: this array is utilized in the reduced precision computation, consequently its type nag_rp reflects this usage.

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

13: ITER – INTEGER *Output*

On exit: information on the progress of the interative refinement process.

ITER < 0

Iterative refinement has failed for one of the reasons given below, full precision factorization has been performed instead.

-1 The routine fell back to full precision for implementation- or machine-specific reasons.

-2 Narrowing the precision induced an overflow, the routine fell back to full precision.

-3 An intermediate reduced precision factorization failed.

-31 The maximum permitted number of iterations was exceeded.

ITER > 0

Iterative refinement has been sucessfully used. ITER returns the number of iterations.

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

INFO > 0 and INFO $\leq$ N

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

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

if UPLO = 'U', $|E| \le c(n)\epsilon|U^{\mathrm{H}}||U|$;

if UPLO = 'L', $|E| \le c(n)\epsilon|L||L^{\mathrm{H}}|$,

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

An approximate error bound for the computed solution is given by

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

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

8 Further Comments

The real analogue of this routine is F07FCF (DSPOSV).

9 Example

This example solves the equations

$$AX=B,$$

where A is the Hermitian positive definite matrix

$$A=\begin{pmatrix} 3.23 & 1.51-1.92i & 1.90+0.84i & 0.42+2.50i \\ 1.51+1.92i & 3.58 & -0.23+1.11i & -1.18+1.37i \\ 1.90-0.84i & -0.23-1.11i & 4.09 & 2.33-0.14i \\ 0.42-2.50i & -1.18-1.37i & 2.33+0.14i & 4.29 \end{pmatrix}$$

and

$$B=\begin{pmatrix} 3.93-\ 6.14i \\ 6.17+\ 9.42i \\ -7.17-21.83i \\ 1.99-14.38i \end{pmatrix}.$$

9.1 Program Text

```
    Program f07fqfe

!     F07FQF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_rp, nag_wp, zcposv
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, info, iter, lda, ldb, ldx, n, r
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), b(:,:), work(:,:), x(:,:)
      Complex (Kind=nag_rp), Allocatable :: swork(:)
      Real (Kind=nag_wp), Allocatable  :: rwork(:)
!     .. Executable Statements ..
      Write (nout,*) 'F07FQF Example Program Results'
```

```
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, r
      lda = n
      ldb = n
      ldx = n
      Allocate (a(lda,n),b(ldb,r),work(n,r),x(ldx,r),swork(n*(n+r)),rwork(n))

!     Read A and B from data file

      Read (nin,*)(a(i,i:n),i=1,n)
      Read (nin,*)(b(i,1:r),i=1,n)

!     Solve the equations Ax = b for x
!     The NAG name equivalent of zcposv is f07fqf
      Call zcposv('U',n,r,a,lda,b,ldb,x,ldx,work,swork,rwork,iter,info)

      If (info==0) Then

!       Print solution

        Write (nout,*) 'Solution'
        Write (nout,99999)(x(i,1:r),i=1,n)

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

99999 Format ((3X,4(' (',F7.4,',',F7.4,')':)))
99998 Format (1X,A,I3,A)
    End Program f07fqfe
```

9.2 Program Data

```
F07FQF Example Program Data
   4          1                                                :Values of N, R
 ( 3.23,  0.00) ( 1.51, -1.92) ( 1.90,  0.84) ( 0.42,  2.50)
                ( 3.58,  0.00) (-0.23,  1.11) (-1.18,  1.37)
                               ( 4.09,  0.00) ( 2.33, -0.14)
                                              ( 4.29,  0.00) :End of matrix A
 ( 3.93, -6.14) ( 6.17,  9.42) (-7.17,-21.83) ( 1.99,-14.38) :End of vector b
```

9.3 Program Results

```
 F07FQF Example Program Results

 Solution
    ( 1.0000,-1.0000) (-0.0000, 3.0000) (-4.0000,-5.0000) ( 2.0000, 1.0000)
```