NAG Library Routine Document

F07KRF (ZPSTRF)

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

F07KRF (ZPSTRF) computes the Cholesky factorization with complete pivoting of a complex Hermitian positive semidefinite matrix.

2 Specification

```
SUBROUTINE F07KRF (UPLO, N, A, LDA, PIV, RANK, TOL, WORK, INFO)
INTEGER               N, LDA, PIV(N), RANK, INFO
REAL (KIND=nag_wp)    TOL, WORK(2*N)
COMPLEX (KIND=nag_wp) A(LDA,*)
CHARACTER(1)          UPLO
```

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

3 Description

F07KRF (ZPSTRF) forms the Cholesky factorization of a complex Hermitian positive semidefinite matrix A either as $P^{\mathrm{T}}AP = U^{\mathrm{H}}U$ if UPLO = 'U' or $P^{\mathrm{T}}AP = LL^{\mathrm{H}}$ if UPLO = 'L', where P is a permutation matrix, U is an upper triangular matrix and L is lower triangular.

This algorithm does not attempt to check that A is positive semidefinite.

4 References

Higham N J (2002) *Accuracy and Stability of Numerical Algorithms* (2nd Edition) SIAM, Philadelphia

Lucas C (2004) LAPACK-style codes for Level 2 and 3 pivoted Cholesky factorizations *LAPACK Working Note No. 161. Technical Report CS-04-522* Department of Computer Science, University of Tennessee, 107 Ayres Hall, Knoxville, TN 37996-1301, USA http://www.netlib.org/lapack/lawnspdf/lawn161.pdf

5 Arguments

1: UPLO – CHARACTER(1) *Input*

On entry: specifies 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 $U^{\mathrm{H}}U$, where U is upper triangular.

UPLO = 'L'
: The lower triangular part of A is stored and A is factorized as LL^{H}, where L is lower triangular.

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

2: N – INTEGER *Input*

On entry: n, the order of the matrix A.

Constraint: $\mathrm{N} \geq 0$.

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

Note: the second dimension of the array A must be at least max(1, N).

On entry: the n by n Hermitian positive semidefinite 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 UPLO = 'U', the first RANK rows of the upper triangle of A are overwritten with the nonzero elements of the Cholesky factor U, and the remaining rows of the triangle are destroyed.

If UPLO = 'L', the first RANK columns of the lower triangle of A are overwritten with the nonzero elements of the Cholesky factor L, and the remaining columns of the triangle are destroyed.

4: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F07KRF (ZPSTRF) is called.

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

5: PIV(N) – INTEGER array *Output*

On exit: PIV is such that the nonzero entries of P are $P(\mathrm{PIV}(k), k) = 1$, for $k = 1, 2, \ldots, n$.

6: RANK – INTEGER *Output*

On exit: the computed rank of A given by the number of steps the algorithm completed.

7: TOL – REAL (KIND=nag_wp) *Input*

On entry: user defined tolerance. If TOL < 0, then $n \times \max_{k=1,n} |A_{kk}| \times$ ***machine precision*** will be used. The algorithm terminates at the rth step if the $(r + 1)$th step pivot $<$ TOL.

8: WORK(2 ∗ N) – REAL (KIND=nag_wp) array *Workspace*

9: INFO – INTEGER *Output*

On exit: INFO = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

INFO < 0

If INFO $= -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO = 1

The matrix A is not positive definite. It is either positive semidefinite with computed rank as returned in RANK and less than n, or it may be indefinite, see Section 9.

7 Accuracy

If UPLO = 'L' and RANK = r, the computed Cholesky factor L and permutation matrix P satisfy the following upper bound

$$\frac{\|A - PLL^{\mathrm{H}}P^{\mathrm{T}}\|_2}{\|A\|_2} \leq 2rc(r)\epsilon\big(\|W\|_2 + 1\big)^2 + O\big(\epsilon^2\big),$$

where

$$W = L_{11}^{-1}L_{12}, \quad L = \begin{pmatrix} L_{11} & 0 \\ L_{12} & 0 \end{pmatrix}, \quad L_{11} \in \mathbb{C}^{r\times r},$$

$c(r)$ is a modest linear function of r, ϵ is ***machine precision***, and

$$\|W\|_2 \leq \sqrt{\frac{1}{3}(n-r)(4^r - 1)}.$$

So there is no guarantee of stability of the algorithm for large n and r, although $\|W\|_2$ is generally small in practice.

8 Parallelism and Performance

F07KRF (ZPSTRF) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The total number of real floating-point operations is approximately $4nr^2 - 8/3r^3$, where r is the computed rank of A.

This algorithm does not attempt to check that A is positive semidefinite, and in particular the rank detection criterion in the algorithm is based on A being positive semidefinite. If there is doubt over semidefiniteness then you should use the indefinite factorization F07MRF (ZHETRF). See Lucas (2004) for further information.

The real analogue of this routine is F07KDF (DPSTRF).

10 Example

This example computes the Cholesky factorization of the matrix A, where

$$A = \begin{pmatrix} 12.40 + 0.00i & 2.39 + 0.00i & 5.50 + 0.05i & 4.47 + 0.00i & 11.89 + 0.00i \\ 2.39 + 0.00i & 1.63 + 0.00i & 1.04 + 0.10i & 1.14 + 0.00i & 1.81 + 0.00i \\ 5.50 + 0.05i & 1.04 + 0.10i & 2.45 + 0.00i & 1.98 - 0.03i & 5.28 - 0.02i \\ 4.47 + 0.00i & 1.14 + 0.00i & 1.98 - 0.03i & 1.71 + 0.00i & 4.14 + 0.00i \\ 11.89 + 0.00i & 1.81 + 0.00i & 5.28 - 0.02i & 4.14 + 0.00i & 11.63 + 0.00i \end{pmatrix}.$$

10.1 Program Text

```
    Program f07krfe

!     F07KRF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, x04ebf, zpstrf
!     .. Implicit None Statement ..
```

```
      Implicit None
!     .. Parameters ..
      Complex (Kind=nag_wp), Parameter :: zero = (0.0E0_nag_wp,0.0E0_nag_wp)
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: tol
      Integer                          :: i, ifail, info, j, lda, n, rank
      Character (1)                    :: uplo
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:)
      Real (Kind=nag_wp), Allocatable  :: work(:)
      Integer, Allocatable             :: piv(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F07KRF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, uplo
      lda = n
      Allocate (a(lda,n),piv(n),work(2*n))

!     Read A from data file
      If (uplo=='U') Then
        Read (nin,*)(a(i,i:n),i=1,n)
      Else If (uplo=='L') Then
        Read (nin,*)(a(i,1:i),i=1,n)
      End If
      tol = -1.0_nag_wp

!     Factorize A
!     The NAG name equivalent of zpstrf is f07krf
      Call zpstrf(uplo,n,a,lda,piv,rank,tol,work,info)

!     Zero out columns rank+1 to n
      If (uplo=='U') Then
        Do j = rank + 1, n
          a(rank+1:j,j) = zero
        End Do
      Else If (uplo=='L') Then
        Do j = rank + 1, n
          a(j:n,j) = zero
        End Do
      End If

!     Print rank
      Write (nout,*)
      Write (nout,'(1X,A15,I3)') 'Computed rank: ', rank

!     Print factor
      Write (nout,*)
      Flush (nout)
      ifail = 0
      Call x04dbf(uplo,'Nonunit',n,n,a,lda,'Bracketed','F5.2','Factor',        &
        'Integer',rlabs,'Integer',clabs,80,0,ifail)

!     Print pivot indices
      Write (nout,*)
      Write (nout,*) 'PIV'
      Flush (nout)
      ifail = 0
      Call x04ebf('General','Non-unit',1,n,piv,1,'I14',' ','No',rlabs,'No',    &
        clabs,80,1,ifail)

    End Program f07krfe
```

10.2 Program Data

```
F07KRF Example Program Data
  5 'L'                                                                  : n, uplo
(12.40, 0.00)
( 2.39, 0.00) ( 1.63, 0.00)
( 5.50, 0.05) ( 1.04, 0.10) ( 2.45, 0.00)
( 4.47, 0.00) ( 1.14, 0.00) ( 1.98,-0.03) ( 1.71, 0.00)
(11.89, 0.00) ( 1.81, 0.00) ( 5.28,-0.02) ( 4.14, 0.00) (11.63, 0.00) : End of A
```

10.3 Program Results

```
 F07KRF Example Program Results

 Computed rank:   3

 Factor
                1             2             3             4             5
 1  ( 3.52, 0.00)
 2  ( 0.68, 0.00) ( 1.08, 0.00)
 3  ( 1.27, 0.00) ( 0.26, 0.00) ( 0.18, 0.00)
 4  ( 1.56, 0.01) (-0.02, 0.08) ( 0.01,-0.05) ( 0.00, 0.00)
 5  ( 3.38, 0.00) (-0.45, 0.00) (-0.17, 0.00) ( 0.00, 0.00) ( 0.00, 0.00)

 PIV
                1             2             4             3             5
```