F07FRFP (PZPOTRF)

NAG Parallel Library Routine Document

Note: Before using this routine, please read the Users' Note for your implementation to check for implementation-dependent details. You are advised to enclose any calls to NAG Parallel Library routines between calls to Z01AAFP and Z01ABFP.

1 Description

F07FRFP (PZPOTRF) computes the Cholesky factorization of an n by n complex Hermitian positive-definite matrix A_s, where A_s is a submatrix of a larger m_A by n_A matrix A, i.e.,

$$A_s(1:n,1:n) \equiv A(i_A:i_A+n-1, j_A:j_A+n-1).$$

Note: if $i_A = j_A = 1$ and $n = m_A = n_A$, then $A_s = A$.

The factorization may be formed either as $A_s = U^H U$ or $A_s = LL^H$, where U is an upper triangular matrix and L is lower triangular.

2 Specification

```
SUBROUTINE F07FRFP(UPLO, N, A, IA, JA, IDESCA, INFO)
ENTRY      PZPOTRF(UPLO, N, A, IA, JA, IDESCA, INFO)
COMPLEX*16         A(*)
INTEGER            N, IA, JA, IDESCA(9), INFO
CHARACTER*1        UPLO
```

The ENTRY statement enables the routine to be called by its ScaLAPACK name.

3 Data Distribution

3.1 Definitions

The following definitions are used in describing the data distribution within this document:

m_p – the number of rows in the logical processor grid.

n_p – the number of columns in the logical processor grid.

p_r – the row grid coordinate of the calling processor.

p_c – the column grid coordinate of the calling processor.

M_b^X – the blocking factor for the distribution of the rows of a matrix X.

N_b^X – the blocking factor for the distribution of the columns of a matrix X.

numroc$(\alpha, b_\ell, q, s, k)$ – a function which gives the **num**ber of **r**ows **o**r **c**olumns of a distributed matrix owned by the processor with the row or column coordinate q (p_r or p_c), where α is the total number of rows or columns of the matrix, b_ℓ is the blocking factor used (M_b^X or N_b^X), s is the row or column coordinate of the processor that possesses the first row or column of the distributed matrix and k is either n_p or m_p. The Library provides the function Z01CAFP (NUMROC) for the evaluation of this function.

3.2 Global and Local Arguments

The input arguments UPLO, N, IA, JA and the array elements IDESCA(1) and IDESCA(3),...,IDESCA(8) are all global and so must have the same values on entry to the routine on each processor. The output argument INFO is global and so will have the same value on exit from the routine on each processor. The remaining arguments are local.

3.3 Distribution Strategy

The matrix A must be partitioned into M_b^A by N_b^A rectangular blocks (in this release $M_b^A = N_b^A$) and stored in an array A in a cyclic 2-d block distribution. This data distribution is described in more detail in the F07 Chapter Introduction. The resulting Cholesky factorization is stored in the same data distribution.

This routine assumes that the data has already been correctly distributed, and if this is not the case will fail to produce correct results. However, the Library provides some utility routines which assist you in distributing data correctly. Descriptions of these routines can be found in Chapters F01 and X04 of the NAG Parallel Library Manual.

4 Arguments

1: UPLO — CHARACTER*1 *Global Input*

On entry: indicates whether the upper or lower triangular part of A_s is stored and how A_s is factorized, as follows:

> if UPLO = 'U', then the upper triangular part of A_s is stored and A_s is factorized as $U^H U$, where U is upper triangular;
>
> if UPLO = 'L', then the lower triangular part of A_s is stored and A_s is factorized as LL^H, where L is lower triangular.

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

2: N — INTEGER *Global Input*

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

Constraint: $0 \leq \text{N} \leq \min(\text{IDESCA}(3)\ , \text{IDESCA}(4))$.

3: A($*$) — COMPLEX*16 array *Local Input/Local Output*

Note: the array A is formally defined as a vector. However, you may find it more convenient to consider A as a 2-d array of dimension (IDESCA(9),γ), where
$\gamma \geq$ numroc(JA + N − 1, IDESCA(6), p_c, IDESCA(8), n_p). See the Example Program.

On entry: the local part of the matrix A which may contain parts of the n by n submatrix A_s to be factorized.

> If UPLO = 'U', the upper triangle of A_s must be stored and the elements of the matrix below the diagonal are not referenced;
>
> if UPLO = 'L', the lower triangle of A_s must be stored and the elements of the matrix above the diagonal are not referenced.

On exit: the upper or lower triangle of A_s is overwritten by the Cholesky factor U or L, as specified by UPLO, distributed in the same cyclic 2-d block fashion.

4: IA — INTEGER *Global Input*

On entry: the row index of matrix A, i_A, that identifies the first row of the submatrix A_s to be factorized.

Constraints: $1 \leq \text{IA} \leq \text{IDESCA}(3) - \text{N} + 1$ and $\text{mod}(\text{IA} - 1, \text{IDESCA}(5)) = 0$.

5: JA — INTEGER *Global Input*

On entry: the column index of matrix A, j_A, that identifies the first column of the submatrix A_s to be factorized.

Constraints: $1 \leq \text{JA} \leq \text{IDESCA}(4) - \text{N} + 1$ and $\text{mod}(\text{JA} - 1, \text{IDESCA}(6)) = 0$.

6: IDESCA(9) — INTEGER array *Local Input*

Distribution: the array elements IDESCA(1) and IDESCA(3),...,IDESCA(8) must be global to the processor grid and the array elements IDESCA(2) and IDESCA(9) are local to each processor.

On entry: the description array for the matrix A. This array must contain details of the distribution of the matrix A and the logical processor grid.

IDESCA(1), the descriptor type. For this routine, which uses a cyclic 2-d block distribution, IDESCA(1) = 1;
IDESCA(2), the BLACS context (ICNTXT) for the processor grid, usually returned by Z01AAFP;
IDESCA(3), the number of rows, m_A, of the matrix A;
IDESCA(4), the number of columns, n_A, of the matrix A;
IDESCA(5), the blocking factor, M_b^A, used to distribute the rows of the matrix A;
IDESCA(6), the blocking factor, N_b^A, used to distribute the columns of the matrix A;
IDESCA(7), the processor row index over which the first row of the matrix A is distributed;
IDESCA(8), the processor column index over which the first column of the matrix A is distributed;
IDESCA(9), the leading dimension of the conceptual 2-d array A.

Constraints:

IDESCA(1) $=1$;
IDESCA(3) ≥ 0; IDESCA(4) ≥ 0;
IDESCA(5) $=$ IDESCA(6); IDESCA(5) ≥ 1; IDESCA(6) ≥ 1;
$0 \leq$ IDESCA(7) $\leq m_p - 1$; $0 \leq$ IDESCA(8) $\leq n_p - 1$;
IDESCA(9) $\geq$ max(1,numroc(IDESCA(3),IDESCA(5),p_r,IDESCA(7),m_p)).

7: INFO — INTEGER *Global Output*

On exit: INFO = 0 unless the routine detects an error (see Section 5).

5 Errors and Warnings

If INFO $\neq 0$ an explanatory message is output and control returned to the calling program.

INFO < 0

On entry, one of the arguments was invalid:

if the kth argument is a scalar INFO $= -k$;
if the kth argument is an array and its jth element is invalid, INFO $= -(100 \times k + j)$.

This error occurred either because a global argument did not have the same value on all logical processors, or because its value on one or more processors was incorrect.

INFO > 0

If INFO $= i$, the leading minor of order i is not positive-definite and the factorization could not be completed. Hence A_s itself is not positive-definite. This may indicate an error in forming the matrix A_s.

6 Further Comments

The total number of floating-point operations is approximately $\frac{4}{3}n^3$. A call to this routine may be followed by a call to the routine F07FSFP (PZPOTRS) to solve $A_sX = B_s$.

6.1 Algorithmic Detail

The algorithm used by this routine is described in Chapter 3 of [1].

6.2 Parallelism Detail

The Level 3 BLAS operations used in this routine are carried out in parallel.

6.3 Accuracy

If UPLO = 'U', the computed factor U is the exact factor of a perturbed matrix $A + E$, where

$$|E| \leq c(n)\epsilon|U^H| \cdot |U|,$$

$c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***. If UPLO = 'L', a similar statement holds for the computed factor L. It follows that $|e_{ij}| \leq c(n)\epsilon\sqrt{a_{ii}a_{jj}}$.

7 References

[1] Anderson E, Bai Z, Bischof C, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A, Ostrouchov S and Sorensen D (1995) *LAPACK Users' Guide* (2nd Edition) SIAM, Philadelphia

[2] Golub G H and Van Loan C F (1989) *Matrix Computations* Johns Hopkins University Press (2nd Edition), Baltimore

8 Example

To compute the Cholesky factorization of the matrix A, where

$$A = \begin{pmatrix} 3.23 + 0.00i & 1.51 - 1.92i & 1.90 + 0.84i & 0.42 + 2.50i \\ 1.51 + 1.92i & 3.58 + 0.00i & -0.23 + 1.11i & -1.18 + 1.37i \\ 1.90 - 0.84i & -0.23 - 1.11i & 4.09 + 0.00i & 2.33 - 0.14i \\ 0.42 - 2.50i & -1.18 - 1.37i & 2.33 + 0.14i & 4.29 + 0.00i \end{pmatrix}.$$

The example uses a 2 by 2 logical processor grid and a block size of 2.

Note: the listing of the Example Program presented below does not give a full pathname for the data file being opened, but in general the user must give the full pathname in this and any other OPEN statement.

8.1 Example Text

```
*     F07FRFP Example Program Text
*     NAG Parallel Library Release 2. NAG Copyright 1996.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          DT
      PARAMETER        (DT=1)
      INTEGER          MB, NB
      PARAMETER        (MB=2,NB=MB)
      INTEGER          NMAX, IAROW, IACOL, LDA, LW
      PARAMETER        (NMAX=8,IAROW=0,IACOL=0,LDA=NMAX,LW=NMAX)
*     .. Local Scalars ..
      INTEGER          IA, ICNTXT, IFAIL, INFO, JA, MP, N, NP
      LOGICAL          ROOT
      CHARACTER        UPLO
      CHARACTER*80     FORMAT
*     .. Local Arrays ..
      COMPLEX*16       A(LDA,NMAX), WORK(LW)
      INTEGER          IDESCA(9)
*     .. External Functions ..
      LOGICAL          Z01ACFP
      EXTERNAL         Z01ACFP
```

```
*     .. External Subroutines ..
      EXTERNAL         F07FRFP, X04BRFP, X04BSFP, Z01AAFP, Z01ABFP
*     .. Executable Statements ..
      ROOT = Z01ACFP()
      IF (ROOT) WRITE (NOUT,*) 'F07FRFP Example Program Results'
*
      MP = 2
      NP = 2
      IFAIL = 0
*
      CALL Z01AAFP(ICNTXT,MP,NP,IFAIL)
*
      OPEN (NIN,FILE='f07frfpe.d')
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, UPLO, FORMAT

      IF (N.LE.NMAX) THEN
*
*         Set the array descriptor of A
*
         IDESCA(1) = DT
         IDESCA(2) = ICNTXT
         IDESCA(3) = N
         IDESCA(4) = N
         IDESCA(5) = MB
         IDESCA(6) = NB
         IDESCA(7) = IAROW
         IDESCA(8) = IACOL
         IDESCA(9) = LDA
         IA = 1
         JA = 1
*
*        Read A from the data file
*
         IFAIL = 0
         CALL X04BRFP(NIN,N,N,A,1,1,IDESCA,IFAIL)
*
*        Factorize the matrix
*
         CALL F07FRFP(UPLO,N,A,IA,JA,IDESCA,INFO)
*
         IF (INFO.EQ.0) THEN
*
*           Print factor
*
            IF (ROOT) THEN
               WRITE (NOUT,*)
               WRITE (NOUT,*) 'Factor '
               WRITE (NOUT,*)
            END IF
            IFAIL = 0
*
            CALL X04BSFP(NOUT,N,N,A,IA,JA,IDESCA,FORMAT,WORK,IFAIL)
*
         ELSE IF (INFO.GT.0) THEN
            IF (ROOT) THEN
               WRITE (NOUT,*)
               WRITE (NOUT,*) 'Matrix is not positive-definite'
            END IF
*
         END IF
*
      END IF
      CLOSE (NIN)
*
      IFAIL = 0
      CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
      STOP
      END
```

8.2 Example Data

```
F07FRFP Example Program Data
 4 'L' 'F7.4'                               :Value of N, UPLO and FORMAT
 (3.23, 0.00) ( 0.0 , 0.0 ) ( 0.0 , 0.0 ) ( 0.0 , 0.0 )
 (1.51, 1.92) ( 3.58, 0.00) ( 0.0 , 0.0 ) ( 0.0 , 0.0 )
 (1.90,-0.84) (-0.23,-1.11) ( 4.09, 0.00) ( 0.0 , 0.0 )
 (0.42,-2.50) (-1.18,-1.37) ( 2.33, 0.14) ( 4.29, 0.00)  :End of matrix A
```

8.3 Example Results

```
 F07FRFP Example Program Results

 Factor

 ( 1.7972, 0.0000) ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 0.0000, 0.0000)
 ( 0.8402, 1.0683) ( 1.3164, 0.0000) ( 0.0000, 0.0000) ( 0.0000, 0.0000)
 ( 1.0572,-0.4674) (-0.4702, 0.3131) ( 1.5604, 0.0000) ( 0.0000, 0.0000)
 ( 0.2337,-1.3910) ( 0.0834, 0.0368) ( 0.9360, 0.9900) ( 0.6603, 0.0000)
```