

# F07ADFP (PDGETRF)

## 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

F07ADFP (PDGETRF) computes the  $LU$  factorization of a real  $m$  by  $n$  matrix  $A_s$ , where  $A_s$  is a submatrix of a larger  $m_A$  by  $n_A$  matrix  $A$ , i.e.,

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

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

F07ADFP (PDGETRF) factorizes  $A_s$  as  $A_s = PLU$ , where  $P$  is a permutation matrix,  $L$  is lower triangular with unit diagonal elements (lower trapezoidal if  $m > n$ ) and  $U$  is upper triangular (upper trapezoidal if  $m < n$ ). Usually  $A$  is square ( $m = n$ ), and both  $L$  and  $U$  are triangular. The routine uses partial pivoting, with row interchanges.

### 2 Specification

```
SUBROUTINE F07ADFP(M, N, A, IA, JA, IDESCA, IPIV, INFO)
ENTRY      PDGETRF(M, N, A, IA, JA, IDESCA, IPIV, INFO)
DOUBLE PRECISION  A(*)
INTEGER           M, N, IA, JA, IDESCA(*), IPIV(*), INFO
```

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

### 3 Usage

#### 3.1 Definitions

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

$m_p$	–	the number of rows in the Library Grid.
$n_p$	–	the number of columns in the Library 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 $X$ .
$\text{numroc}(\alpha, b_\ell, q, s, k)$	–	a function which gives the <b>number of rows or columns</b> of a distributed matrix owned by the processor with the row or column coordinate $q$ ( $p_r$ or $p_c$ ), where $\alpha$ is the total number of rows or columns of the matrix, $b_\ell$ 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 $m_p$ or $n_p$ . The Library provides the function Z01CAFP (NUMROC) for the evaluation of this function.

#### 3.2 Global and Local Arguments

The following global **input** arguments must have the same value on entry to the routine on each processor and the global **output** arguments will have the same value on exit from the routine on each processor:

Global input arguments:      M, N, IA, JA, IDESCA(1), IDESCA(3:8)

Global output arguments:    INFO

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 two-dimensional block distribution. This data distribution is described in more detail in the the F07 Chapter Introduction. The resulting  $LU$  factorization is stored in the same data distribution.

### 3.4 Related Routines

The Library provides many support routines for the generation, scattering/gathering and input/output of matrices/vectors in cyclic two-dimensional block form. The following routines may be used in conjunction with F07ADFP (PDGETRF):

Real matrix generation:	F01ZQFP
Real matrix input:	X04BCFP
Real matrix output:	X04BDFP
Real matrix gather:	F01WAFP
Real matrix scatter:	F01WNFP

## 4 Arguments

- 1: M — INTEGER *Global Input*  
*On entry:*  $m$ , the number of rows of the submatrix  $A_s$ .  
*Constraint:*  $0 \leq M \leq \text{IDESCA}(3)$
- 2: N — INTEGER *Global Input*  
*On entry:*  $n$ , the number of columns of the submatrix  $A_s$ .  
*Constraint:*  $0 \leq N \leq \text{IDESCA}(4)$
- 3: A(\*) — DOUBLE PRECISION 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 two-dimensional array of dimension  $(\text{IDESCA}(9), \gamma)$ , where  $\gamma \geq \text{numroc}(\text{JA} + \text{N} - 1, \text{IDESCA}(6), p_c, \text{IDESCA}(8), n_p)$ .  
*On entry:* the local part of the  $m$  by  $n$  matrix  $A_s$  to be factorized.  
*On exit:*  $A_s$  is overwritten by the factors  $L$  and  $U$  distributed in the same cyclic two-dimensional block fashion; the unit diagonal elements of  $L$  are not stored.
- 4: IA — INTEGER *Global Input*  
*On entry:*  $i_A$ , the row index of matrix  $A$  that identifies the first row of the submatrix  $A_s$  to be factorized.  
*Constraints:*  $1 \leq \text{IA} \leq \text{IDESCA}(3) - M + 1$  and  $\text{mod}(\text{IA} - 1, \text{IDESCA}(5)) = 0$ .
- 5: JA — INTEGER *Global Input*  
*On entry:*  $j_A$ , the column index of matrix  $A$  that identifies the first column of the submatrix  $A_s$  to be factorized.  
*Constraints:*  $1 \leq \text{JA} \leq \text{IDESCA}(4) - N + 1$  and  $\text{mod}(\text{JA} - 1, \text{IDESCA}(6)) = 0$ .

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

**Note:** the dimension of the array IDESCA must be at least 9.

*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 two-dimensional block distribution, IDESCA(1) = 1;

IDESCA(2), the Library context, usually returned by a call to the Library Grid initialisation routine 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 two-dimensional array  $A$ .

*Constraints:*

IDESCA(1) = 1;

IDESCA(3)  $\geq$  0; IDESCA(4)  $\geq$  0;

IDESCA(5) = IDESCA(6); IDESCA(5)  $\geq$  1; IDESCA(6)  $\geq$  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:** IPIV(\*) — INTEGER array *Local Output*

**Note:** the dimension of the array IPIV must be at least  $\beta + \text{IDESCA}(5)$  where,  $\beta = \text{numroc}(\text{IDESCA}(3), \text{IDESCA}(5), p_r, \text{IDESCA}(7), m_p)$ .

*On exit:* the pivot indices. The global row IPIV( $k$ ) was interchanged with the local row  $k$ . This array is aligned with the distributed matrix  $A$ .

**8:** INFO — INTEGER *Global Output*

The NAG Parallel Library provides a mechanism, via the routine Z02EAFP, to reduce the amount of parameter validation performed by this routine. For a full description refer to the Z02 Chapter Introduction.

*On exit:* INFO = 0 (or -9999 if reduced error checking is enabled) unless the routine detects an error (see Section 5).

## 5 Errors and Warnings

If INFO  $\neq$  0 explanatory error messages are output from the root processor (or processor {0,0} when the root processor is not available) on the current error message unit (as defined by X04AAF).

INFO < 0

On entry, one of the arguments was invalid:

if the  $k$ th argument is a scalar INFO =  $-k$ ;

if the  $k$ th argument is an array and its  $j$ th 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 element  $u_{ii}$  of the upper triangular matrix  $U$  is exactly zero. The factorization has been completed but the factor  $U$  is exactly singular, and division by zero will occur if it is subsequently used to solve a system of linear equations or to compute  $A^{-1}$ .

## 6 Further Comments

The total number of floating-point operations is approximately  $\frac{2}{3}n^3$  if  $m = n$  (the usual case),  $\frac{1}{3}n^2(3m-n)$  if  $m > n$  and  $\frac{1}{3}m^2(3n-m)$  if  $m < n$ .

A call to this routine with  $m = n$  may be followed by a call to the routine F07AEFP (PDGETRS) to solve a system of equations  $A_s X = B_s$ .

### 6.1 Algorithmic Detail

The routine uses a block-partitioned  $LU$  factorization with partial pivoting. See Anderson *et al.* [1] for details of the method used by the routine.

### 6.2 Parallelism Detail

Each processor column performs an  $LU$  factorization with partial pivoting on successive column blocks of the matrix. Details of this factorization and pivoting are passed to all processors which perform the update of the remaining submatrix in parallel.

### 6.3 Accuracy

The computed factors  $L$  and  $U$  are the exact factors of a perturbed matrix  $A + E$ , where

$$|E| \leq c(\min(m, n))\epsilon P|L| \cdot |U|,$$

where  $c(n)$  is a modest linear function of  $n$ , and  $\epsilon$  is the *machine precision*.

## 7 References

- [1] 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
- [2] Blackford L S, Choi J, Cleary A, D'Azevedo E, Demmel J, Dhillon I, Dongarra J, Hammarling S, Henry G, Petitet A, Stanley K, Walker D and Whaley R C (1997) *ScaLAPACK Users' Guide* SIAM 3600 University City Science Center, Philadelphia, PA 19104-2688, USA. URL: [http://www.netlib.org/scalapack/slug/scalapack\\_slug.html](http://www.netlib.org/scalapack/slug/scalapack_slug.html)
- [3] Golub G H and van Loan C F (1996) *Matrix Computations* Johns Hopkins University Press (3rd Edition), Baltimore

## 8 Example

To compute the  $LU$  factorization of the matrix  $A$ , where

$$A = \begin{pmatrix} 1.80 & 2.88 & 2.05 & -0.89 \\ 5.25 & -2.95 & -0.95 & -3.80 \\ 1.58 & -2.69 & -2.90 & -1.04 \\ -1.11 & -0.66 & -0.59 & 0.80 \end{pmatrix}.$$

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

```

*   F07ADFP 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          MMAX, NMAX, IAROW, IACOL, LDA, LW
PARAMETER       (MMAX=8,NMAX=8,IAROW=0,IACOL=0,LDA=MMAX,LW=NMAX)
*   .. Local Scalars ..
INTEGER          IA, ICNTXT, IFAIL, INFO, JA, M, MP, N, NP
LOGICAL          ROOT
CHARACTER*80     FORMAT
*   .. Local Arrays ..
DOUBLE PRECISION A(LDA,NMAX), WORK(LW)
INTEGER          IDESCA(9), IPIV(MMAX+MB)
*   .. External Functions ..
LOGICAL          Z01ACFP
EXTERNAL         Z01ACFP
*   .. External Subroutines ..
EXTERNAL         F07ADFP, X04BCFP, X04BDFP, Z01AAFP, Z01ABFP
*   .. Executable Statements ..
ROOT = Z01ACFP()
IF (ROOT) WRITE (NOUT,*) 'F07ADFP Example Program Results'

*
MP = 2
NP = 2
IFAIL = 0

*
CALL Z01AAFP(ICNTXT,MP,NP,IFAIL)

*
OPEN (NIN,FILE='f07adfpe.d')
*   Skip heading in data file
READ (NIN,*)
READ (NIN,*) M, N, FORMAT

*
IF (M.LE.MMAX .AND. N.LE.NMAX) THEN

*
*   Set the array descriptor of A
*
      IDESCA(1) = DT
      IDESCA(2) = ICNTXT
      IDESCA(3) = M
      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 X04BCFP(NIN,M,N,A,1,1,IDESCA,IFAIL)
*
*      Factorize the matrix
*
        CALL F07ADFP(M,N,A,IA,JA,IDESCA,IPIV,INFO)
*
        IF (INFO.EQ.0) THEN
*
*          Print factor
*
          IF (ROOT) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Details of the factorization'
            WRITE (NOUT,*)
          END IF
          IFAIL = 0
*
          CALL X04BDFP(NOUT,M,N,A,IA,JA,IDESCA,FORMAT,WORK,IFAIL)
*
        ELSE IF (INFO.GT.0) THEN
          IF (ROOT) WRITE (NOUT,*) 'Matrix is singular'
        END IF
*
      END IF
*
      CLOSE (NIN)
*
      IFAIL = 0
      CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
      STOP
      END

```

## 8.2 Example Data

```

F07ADFP Example Program Data
4 4 '(4F12.4)'           :Values of M, N and FORMAT
1.80  2.88  2.05 -0.89
5.25 -2.95 -0.95 -3.80
1.58 -2.69 -2.90 -1.04
-1.11 -0.66 -0.59  0.80   :End of matrix A

```

### 8.3 Example Results

F07ADFP Example Program Results

Details of the factorization

5.2500	-2.9500	-0.9500	-3.8000
0.3429	3.8914	2.3757	0.4129
0.3010	-0.4631	-1.5139	0.2948
-0.2114	-0.3299	0.0047	0.1314

---