F07ASFP (PZGETRS)

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

F07ASFP (PZGETRS) solves an n by n complex system of linear equations with multiple right-hand sides, i.e., $A_sX = B_s$ or $A_s^TX = B_s$ or $A_S^HX = B_s$, where A_s is a submatrix of a larger m_A by n_A matrix A, i.e.,

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

and B_s is a (r right-hand sides) submatrix of a larger m_B by n_B matrix B, i.e.,

$$B_s(1:n, 1:r) \equiv B(i_B : i_B + n - 1, j_B : j_B + r - 1).$$

The matrix A_s must have been previously factorized by a call to F07ARFP (PZGETRF). F07ARFP (PZGETRF) performs an LU factorization and F07ASFP (PZGETRS) solves the system of equations by forward and backward substitution.

2 Specification

```
      SUBROUTINE F07ASFP(TRANS, N, NRHS, A, IA, JA, IDESCA, IPIV, B, IB,
     1                   JB, IDESCB, INFO)
      ENTRY      PZGETRS(TRANS, N, NRHS, A, IA, JA, IDESCA, IPIV, B, IB,
     1                   JB, IDESCB, INFO)
      COMPLEX*16         A(*), B(*)
      INTEGER            N, NRHS, IA, JA, IDESCA(*), IPIV(*), IB, JB,
     1                   IDESCB(*), INFO
      CHARACTER*1        TRANS
```

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.
numroc(α,b_ℓ,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 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:	TRANS, N, NRHS, IA, JA, IB, JB, IDESCA(1), IDESCA(3:8), IDESCB(1), IDESCB(3:8)
Global output arguments:	INFO

The remaining arguments are local.

3.3 Distribution Strategy

The array A must contain the *LU* factorization of the matrix A_s, previously factorized by F07ARFP (PZGETRF). The *LU* factors must be stored in a cyclic two-dimensional block distribution (described in the F07 Chapter Introduction), as returned by F07ARFP (PZGETRF). The right-hand sides of the equation, B_s are stored in the array B, in a cyclic two-dimensional block 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 F07ASFP (PZGETRS):

Complex matrix generation:	F01ZVFP
Complex matrix input:	X04BRFP
Complex matrix output:	X04BSFP
Complex matrix gather:	F01WGFP
Complex matrix scatter:	F01WUFP

4 Arguments

1: TRANS — CHARACTER*1 *Global Input*

On entry: indicates the form of the equations as follows:

if TRANS = 'N', then $A_sX = B_s$ is solved for X;
if TRANS = 'T', then $A_s^TX = B_s$ is solved for X;
if TRANS = 'C', then $A_s^HX = B_s$ is solved for X.

Constraint: TRANS = 'N', 'T' or 'C'.

2: N — INTEGER *Global Input*

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

Constraint: $0 \leq$ N $\leq$ min(IDESCA(3), IDESCA(4), IDESCB(3)).

3: NRHS — INTEGER *Global Input*

On entry: r, the number of right-hand sides.

Constraint: $0 \leq$ NRHS $\leq$ IDESCB(4).

4: A($*$) — COMPLEX*16 array *Local Input*

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 (IDESCA(9),γ), where
$\gamma \geq$ numroc(JA+N$-$1,IDESCA(6),p_c,IDESCA(8),n_p).

On entry: the local part of the *LU* factorization of the matrix A_s as returned by F07ARFP (PZGETRF).

5: IA — INTEGER *Global Input*

On entry: i_A, the row index of matrix A that identifies the first row of the *LU* factorization of A_s.

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

6: JA — INTEGER *Global Input*

On entry: j_A, the column index of matrix A that identifies the first column of the LU factorization of A_s.

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

7: 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) ≥ 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)).

8: IPIV($*$) — INTEGER array *Local Input*

Note: the dimension of the array IPIV must be at least β + IDESCA(5) where,
β = numroc(IDESCA(3),IDESCA(5),p_r,IDESCA(7),m_p).

On entry: the pivot indices, as returned by F07ARFP (PZGETRF).

9: B($*$) — COMPLEX*16 array *Local Input/Local Output*

Note: the array B is formally defined as a vector. However, you may find it more convenient to consider B as a two-dimensional array of dimension (IDESCB(9),γ), where
$\gamma \geq$ numroc(JB+NRHS$-$1,IDESCB(6),p_c,IDESCB(8),n_p).

On entry: the local part of the right-hand side matrix B which may contain parts of the n by r submatrix B_s.

On exit: the n by r solution matrix X distributed in the same cyclic two-dimensional block distribution.

10: IB — INTEGER *Global Input*

On entry: i_B, the row index of matrix B that identifies the first column of the submatrix B_s.

Constraints: $1 \leq$ IB $\leq$ IDESCB(3) $-$ N + 1 and mod(IB$-$1,IDESCB(5)) = 0.
The IAth row of the array A and the IBth row of the array B must be located on the same row of the processor grid, i.e.,

$$\text{mod(IDESCA(7)} + (\text{IA} - 1)/\text{IDESCA(5)},\ n_p) = \text{mod(IDESCB(7)} + (\text{IB} - 1)/\text{IDESCB(5)},\ n_p)$$

11: JB — INTEGER *Global Input*

On entry: the column index of matrix B, j_B, that identifies the first column of the submatrix B_s.

Constraint: $1 \leq$ JB $\leq$ IDESCB(4) $-$ NRHS $+ 1$.

12: IDESCB($*$) — INTEGER array *Local Input*

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

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

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

IDESCB(1), the descriptor type. For this routine, which uses a cyclic two-dimensional block distribution, IDESCB(1) = 1;

IDESCB(2), the Library context, usually returned by a call to the Library Grid initialisation routine Z01AAFP;

IDESCB(3), the number of rows, m_B, of the matrix B;

IDESCB(4), the number of columns, n_B, of the matrix B;

IDESCB(5), the blocking factor, M_b^B, used to distribute the rows of the matrix B;

IDESCB(6), the blocking factor, N_b^B, used to distribute the columns of the matrix B;

IDESCB(7), the processor row index over which the first row of the matrix B is distributed;

IDESCB(8), the processor column index over which the first column of the matrix B is distributed;

IDESCB(9), the leading dimension of the conceptual two-dimensional array B.

Constraints:

IDESCB(1) = 1;

IDESCB(3) ≥ 0; IDESCB(4) ≥ 0;

IDESCB(2) = IDESCA(2);

IDESCB(5) = IDESCB(6); IDESCB(5) ≥ 1; IDESCB(6) ≥ 1;

$0 \leq$ IDESCB(7) $\leq m_p - 1$; $0 \leq$ IDESCB(8) $\leq n_p - 1$;

IDESCB(9) $\geq$ max(1,numroc(IDESCB(3),IDESCB(5),p_r,IDESCB(7),m_p)).

13: 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 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.

6 Further Comments

The total number of floating-point operations is approximately $8n^2r$.

6.1 Algorithmic Detail

Forward and backward substitution are used.

6.2 Parallelism Detail

Forward and backward substitution are applied to each block-column of the right-hand sides in parallel.

6.3 Accuracy

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

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

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

If x is the true solution, then the computed solution $\hat{x}$ satisfies a forward error bound of the form

$$\frac{\|x-\hat{x}\|_\infty}{\|x\|_\infty} \leq \epsilon c(n)\kappa(A),$$

where $\kappa(A)$ is the condition number of A. See the F07 Chapter Introduction.

7 References

[1] 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, Philadelpia, PA 19104-2688, USA. URL: http://www.netlib.org/scalapack/slug/scalapack_slug.html

[2] Golub G H and van Loan C F (1996) *Matrix Computations* Johns Hopkins University Press (3rd Edition), Baltimore

8 Example

To solve the system of equations $AX=B$, where

$$A = \begin{pmatrix} -1.34+2.55i & 0.28+3.17i & -6.39-2.20i & 0.72-0.92i \\ -0.17-1.41i & 3.31-0.15i & -0.15+1.34i & 1.29+1.38i \\ -3.29-2.39i & -1.91+4.42i & -0.14-1.35i & 1.72+1.35i \\ 2.41+0.39i & -0.56+1.47i & -0.83-0.69i & -1.96+0.67i \end{pmatrix} \quad \text{and}$$

$$B = \begin{pmatrix} 26.26+51.78i & 31.32-6.70i \\ 6.43-8.68i & 15.86-1.42i \\ -5.75+25.31i & -2.15+30.19i \\ 1.16+2.57i & -2.56+7.55i \end{pmatrix}.$$

A is first factorized by F07ARFP (PZGETRF).

The example uses a 2 by 2 logical processor grid and a block size of 2 for both A and B.

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

```
*     F07ASFP Example Program Text
*     NAG Parallel Library Release 3 Revised. NAG Copyright 1999.
*     .. 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, LDB, NRHMAX, LW
      PARAMETER        (NMAX=8,IAROW=0,IACOL=0,LDA=NMAX,LDB=NMAX,
     +                 NRHMAX=2,LW=NMAX)
*     .. Local Scalars ..
      INTEGER          IA, IB, ICNTXT, IFAIL, INFO, JA, JB, MP, N, NP,
     +                 NRHS
      LOGICAL          ROOT
      CHARACTER        TRANS
      CHARACTER*80     FORMAT
*     .. Local Arrays ..
      COMPLEX*16       A(LDA,NMAX), B(LDB,NRHMAX), WORK(LW)
      INTEGER          IDESCA(9), IDESCB(9), IPIV(NMAX+MB)
*     .. External Functions ..
      LOGICAL          Z01ACFP
      EXTERNAL         Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         F07ARFP, F07ASFP, X04BRFP, X04BSFP, Z01AAFP,
     +                 Z01ABFP
*     .. Executable Statements ..
      ROOT = Z01ACFP()
      IF (ROOT) THEN
         WRITE (NOUT,*) 'F07ASFP Example Program Results'
         WRITE (NOUT,*)
      END IF
*
      MP = 2
      NP = 2
      IFAIL = 0
*
      CALL Z01AAFP(ICNTXT,MP,NP,IFAIL)
*
      OPEN (NIN,FILE='f07asfpe.d')
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, NRHS, 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 F07ARFP(N,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 X04BSFP(NOUT,N,N,A,IA,JA,IDESCA,FORMAT,WORK,IFAIL)
*
            TRANS = 'N'
*
*           Set the array descriptor of B
*
            IDESCB(1) = DT
            IDESCB(2) = IDESCA(2)
            IDESCB(3) = N
            IDESCB(4) = NRHS
            IDESCB(5) = MB
            IDESCB(6) = NB
            IDESCB(7) = IAROW
            IDESCB(8) = IACOL
            IDESCB(9) = LDB
            IB = 1
            JB = 1
*
*           Read B from data file
*
            IFAIL = 0
            CALL X04BRFP(NIN,N,NRHS,B,1,1,IDESCB,IFAIL)
*
            IF (IFAIL.EQ.0) THEN
*
*               Print RHS
*
               IF (ROOT) THEN
                  WRITE (NOUT,*)
                  WRITE (NOUT,*) 'RHS(s)'
```

```
                  WRITE (NOUT,*)
               END IF
               IFAIL = 0
*
               CALL X04BSFP(NOUT,N,NRHS,B,IB,JB,IDESCB,FORMAT,WORK,
     +                      IFAIL)
*
            ELSE
               IF (ROOT) WRITE (NOUT,*)
     +             'Problem with reading data from file, IFAIL =  ',
     +             IFAIL, ' from X04BSFP'
               GO TO 20
            END IF
*
            CALL F07ASFP(TRANS,N,NRHS,A,IA,JA,IDESCA,IPIV,B,IB,JB,
     +                   IDESCB,INFO)
            IF (INFO.EQ.0) THEN
*
*               Print solution(s)
*
               IF (ROOT) THEN
                  WRITE (NOUT,*)
                  WRITE (NOUT,*) 'Solution(s)'
                  WRITE (NOUT,*)
               END IF
               IFAIL = 0
*
               CALL X04BSFP(NOUT,N,NRHS,B,IB,JB,IDESCB,FORMAT,WORK,
     +                      IFAIL)
*
            ELSE
               IF (ROOT) WRITE (NOUT,*)
     +             'Unable to solve triangular system, IFAIL = ', IFAIL,
     +             ' from F07ASFP'
            END IF
         ELSE
            IF (ROOT) WRITE (NOUT,*)
     +          'Problem in factoring matrix, IFAIL = ', IFAIL,
     +          ' from F07ARFP'
         END IF
*
      END IF
*
   20 CLOSE (NIN)
*
      CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
      STOP
      END
```

8.2 Example Data

```
F07ASFP Example Program Data
 4  2  '(4(:,'' ('',F7.4,'','',F7.4,'')''))'              :Values of N, NRHS, FORMAT
 (-1.34, 2.55) ( 0.28, 3.17) (-6.39,-2.20) ( 0.72,-0.92)
 (-0.17,-1.41) ( 3.31,-0.15) (-0.15, 1.34) ( 1.29, 1.38)
 (-3.29,-2.39) (-1.91, 4.42) (-0.14,-1.35) ( 1.72, 1.35)
```

```
( 2.41, 0.39) (-0.56, 1.47) (-0.83,-0.69) (-1.96, 0.67)  :End of matrix A
(26.26, 51.78) (31.32, -6.70)
( 6.43, -8.68) (15.86, -1.42)
(-5.75, 25.31) (-2.15, 30.19)
( 1.16,  2.57) (-2.56,  7.55)                            :End of matrix B
```

8.3 Example Results

```
F07ASFP Example Program Results


Details of the factorization

(-3.2900,-2.3900) (-1.9100, 4.4200) (-0.1400,-1.3500) ( 1.7200, 1.3500)
( 0.2376, 0.2560) ( 4.8952,-0.7114) (-0.4623, 1.6966) ( 1.2269, 0.6190)
(-0.1020,-0.7010) (-0.6691, 0.3689) (-5.1414,-1.1300) ( 0.9983, 0.3850)
(-0.5359, 0.2707) (-0.2040, 0.8601) ( 0.0082, 0.1211) ( 0.1482,-0.1252)

RHS(s)

(26.2600,51.7800) (31.3200,-6.7000)
( 6.4300,-8.6800) (15.8600,-1.4200)
(-5.7500,25.3100) (-2.1500,30.1900)
( 1.1600, 2.5700) (-2.5600, 7.5500)

Solution(s)

( 1.0000, 1.0000) (-1.0000,-2.0000)
( 2.0000,-3.0000) ( 5.0000, 1.0000)
(-4.0000,-5.0000) (-3.0000, 4.0000)
( 0.0000, 6.0000) ( 2.0000,-3.0000)
```
