F01ZXFP

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

F01ZXFP distributes an m by n complex matrix A on a logical grid of processors in a cyclic 2-d block distribution.

This routine distributes matrices in the form required by a number of the F04 Black Box routines. A user-supplied subroutine is required to generate a block of the matrix A.

2 Specification

```
SUBROUTINE F01ZXFP(ICNTXT, GMAT, M, N, NB, A, LDA, IFAIL)
COMPLEX*16         A(LDA, *)
INTEGER            ICNTXT, M, N, NB, LDA, IFAIL
EXTERNAL           GMAT
```

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.
N_b	–	the blocking factor for the distribution of the rows and 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, N_b, 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 input arguments M, N, NB and IFAIL are global and so must have the same value on entry to the routine on each processor. The output argument IFAIL is global and so will return the same value on exit from the routine on each processor. The external procedure GMAT is global. The remaining arguments are local.

4 Arguments

1: ICNTXT — INTEGER *Local Input*

On entry: the BLACS context used by the communication mechanism, usually returned by a call to Z01AAFP.

2: GMAT — SUBROUTINE, supplied by the user. *Global External Procedure*

GMAT must return the block $A(i_1 : i_2, j_1 : j_2)$ of the matrix to be distributed, in the array AL.

Its specification is:

```
SUBROUTINE          GMAT(I1, I2, J1, J2, AL, LDAL)
COMPLEX*16          AL(LDAL,*)
INTEGER             I1, I2, J1, J2, LDAL
```

1: I1 — INTEGER *Local Input*
On entry: i_1, the first row of the block of A to be generated.

2: I2 — INTEGER *Local Input*
On entry: i_2, the last row of the block of A to be generated.

3: J1 — INTEGER *Local Input*
On entry: j_1, the first column of the block of A to be generated.

4: J2 — INTEGER *Local Input*
On entry: j_2, the last column of the block of A to be generated.

5: AL(LDAL,∗) — COMPLEX*16 array *Local Output*
On exit: AL must contain the block $A(i_1 : i_2, j_1 : j_2)$ of the matrix A in its first (I2−I1+1) rows and (J2−J1+1) columns.

6: LDAL — INTEGER *Local Input*
On entry: the first dimension of the array AL.

GMAT must be declared as EXTERNAL in the (sub)program from which F01ZXFP is called. Arguments denoted as *Input* must **not** be changed by this procedure.

3: M — INTEGER *Global Input*

On entry: the number of rows of the matrix A, m.

Constraint: M ≥ 0.

4: N — INTEGER *Global Input*

On entry: the number of columns of the matrix A, n.

Constraint: N ≥ 0.

5: NB — INTEGER *Global Input*

On entry: the blocking factor for distributing the matrix A, N_b.

Constraint: NB ≥ 1.

6: A(LDA,∗) — COMPLEX*16 array *Local Output*

Note: the size of the second dimension of the array A must be at least max(1,numroc(N,NB,p_c,0,n_p))..**Note:** the second dimension of the array A must be at least max(1,numroc(N,NB,p_c,0,n_p)).

On exit: the local part of the matrix A, distributed in a cyclic 2-d block fashion.

7: LDA — INTEGER *Local Input*

On entry: the size of the first dimension of the array A as declared in the (sub)program from which F01ZXFP is called.

On entry: the first dimension of the array A as declared in the (sub)program from which F01ZXFP is called.

Constraint: LDA $\geq$ max(1,numroc(M,NB,p_r,0,m_p))

8: IFAIL — INTEGER *Global Input/Global Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in the Essential Introduction) the recommended values are:

IFAIL = 0, if multigridding is **not** employed;
IFAIL = −1, if multigridding is employed.

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

5 Errors and Warnings

If on entry IFAIL = 0 or −1, 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).

Errors detected by the routine:

IFAIL = −2000

The routine has been called with an invalid value of ICNTXT on one or more processors.

IFAIL = −1000

The logical processor grid and library mechanism (Library Grid) have not been correctly defined, see Z01AAFP.

IFAIL = $-i$

On entry, the ith argument was invalid. This error occured 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. An explanatory message distinguishes between these two cases.

6 Further Comments

This routine may be used to distribute the data in the form required by a number of the Black Box routines in Chapter F04. This routine may also be used to distribute a vector.

6.1 Algorithmic Detail

The routine generates the matrix on a logical processor by block column.

6.2 Parallelism Detail

The routine generates the matrix on each logical processor independently.

7 References

[1] Dongarra J J and Whaley R C (1995) A users' guide to the BLACS v1.0. *LAPACK Working Note 94 (Technical Report CS-95-281)* Department of Computer Science, University of Tennessee, 107 Ayres Hall, Knoxville, TN 37996-1301, USA.
URL: http://www.netlib.org/lapack/lawns/lawn94.ps

8 Example

To solve the equations

$$Ax = b,$$

where the matrix A is given by

$$a_{ij} = (\min(i, j), \min(i, j))$$

and the right-hand side is the vector

$$b = \begin{pmatrix} 4+ & 4i \\ 7+ & 7i \\ 9+ & 9i \\ 10+ & 10i \end{pmatrix}$$

The exact solution is the vector $(1, 1, 1, 1)^T$. The Black Box routine F04FBFP is called to solve the equations following the distribution of A by a call to F01ZXFP and a call to X04BGFP to read b from a data file. The routine X04BHFP is used to print the solution vector x. The example illustrates the solution for four equations, with a 2 by 2 logical grid of processors 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

```
*     F01ZXFP Example Program Text
*     NAG Parallel Library Release 2. NAG Copyright 1996.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX
      PARAMETER        (NMAX=100)
      INTEGER          MG, NG
      PARAMETER        (MG=2,NG=2)
      INTEGER          NB
      PARAMETER        (NB=2)
      INTEGER          LDA, TDA, TDB
      PARAMETER        (LDA=NMAX/MG+NB,TDA=NMAX/NG+NB,TDB=1)
      INTEGER          LDB, LDI
      PARAMETER        (LDB=LDA,LDI=LDA)
*     .. Local Scalars ..
      INTEGER          ICNTXT, IFAIL, N, NCOLS, NROWS
      LOGICAL          ROOT
*     .. Local Arrays ..
      COMPLEX*16       A(LDA,TDA), B(LDB,TDB), WORK(NMAX)
      INTEGER          IPIV(LDI)
      CHARACTER*4      FORMAT
*     .. External Functions ..
      LOGICAL          Z01ACFP
      EXTERNAL         Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         F01ZXFP, F04ECFP, GMATA, X04BVFP, X04BWFP,
     +                 Z01AAFP, Z01ABFP
*     .. Executable Statements ..
      ROOT = Z01ACFP()
      IF (ROOT) THEN
         WRITE (NOUT,*) 'F01ZXFP Example Program Results'
         WRITE (NOUT,*)
      END IF
*
      NROWS = MG
      NCOLS = NG
      IFAIL = 0
*
      CALL Z01AAFP(ICNTXT,NROWS,NCOLS,IFAIL)
*
      OPEN (NIN,FILE='f01zxfpe.d')
*     Skip heading in data file
```

```
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*     Generate the matrix A
*
         IFAIL = 0
         CALL F01ZXFP(ICNTXT,GMATA,N,N,NB,A,LDA,IFAIL)
*
*     Read in the right hand side
*
         IFAIL = 0
         CALL X04BVFP(ICNTXT,NIN,N,1,NB,B,LDB,IFAIL)
*
*     Solve the system
*
         IFAIL = 0
         CALL F04ECFP(ICNTXT,'No transpose',N,NB,A,LDA,1,B,LDB,IPIV,
     +                IFAIL)
*
*     Print the results
*
         IFAIL = 0
         FORMAT = 'F7.3'
         CALL X04BWFP(ICNTXT,NOUT,N,1,NB,B,LDB,FORMAT,WORK,IFAIL)
*
         IFAIL = 0
         CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
      END IF
      STOP
      END
*
      SUBROUTINE GMATA(I1,I2,J1,J2,AL,LDAL)
*     GMATA generates the block A(I1: I2, J1: J2) of the matrix A such
*     that
*
*        a(i,j) = (min(i,j), min(i,j))
*
*     in the array AL.
*
*     .. Scalar Arguments ..
      INTEGER          I1, I2, J1, J2, LDAL
*     .. Array Arguments ..
      COMPLEX*16       AL(LDAL,*)
*     .. Local Scalars ..
      INTEGER          I, J, K, L
*     .. Intrinsic Functions ..
      INTRINSIC        CMPLX, MIN
*     .. Executable Statements ..
      L = 1
      DO 40 J = J1, J2
         K = 1
         DO 20 I = I1, I2
            AL(K,L) = CMPLX(MIN(I,J),MIN(I,J))
            K = K + 1
   20    CONTINUE
         L = L + 1
```

```
   40 CONTINUE
*
      RETURN
      END
```

8.2 Example Data

```
F01ZXFP Example Program Data
4      : Value of N
 (4.0, 4.0)
 (7.0, 7.0)
 (9.0, 9.0)
(10.0, 10.0)   : End of right hand side vector B
```

8.3 Example Results

```
 F01ZXFP Example Program Results

 (  1.000,  0.000)
 (  1.000,  0.000)
 (  1.000,  0.000)
 (  1.000,  0.000)
```