F01ZSFP

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

F01ZSFP generates a m by n real matrix A on a logical grid of processors in a cyclic two-dimensional block distribution.

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

2 Specification

```
SUBROUTINE F01ZSFP(ICNTXT, GMAT, M, N, NB, A, LDA, IFAIL)
DOUBLE PRECISION   A(LDA,*)
INTEGER            ICNTXT, M, N, NB, LDA, IFAIL
EXTERNAL           GMAT
```

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.
N_b	–	the blocking factor for the distribution of the rows and columns of the matrix.
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 (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 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, NB, IFAIL

Global output arguments: IFAIL

The remaining arguments are local.

4 Arguments

1: ICNTXT — INTEGER *Local Input*

On entry: the Library context, usually returned by a call to the Library Grid initialisation routine Z01AAFP.

Note: the value of ICNTXT **must not** be changed.

2: GMAT — SUBROUTINE, supplied by the user. *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)
DOUBLE PRECISION    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,*) — DOUBLE PRECISION 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 $(i_2 - i_1 + 1)$ rows and $(j_2 - j_1 + 1)$ columns.

6: LDAL — INTEGER *Local Input*

On entry: the size of the first dimension of the array AL as declared in the (sub)program from which F01ZSFP is called.

GMAT must be declared as EXTERNAL in the (sub)program from which F01ZSFP is called. Arguments denoted as *Input* must **not** be changed by this procedure.

3: M — INTEGER *Global Input*

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

Constraint: M ≥ 0.

4: N — INTEGER *Global Input*

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

Constraint: N ≥ 0.

5: NB — INTEGER *Global Input*

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

Constraint: NB ≥ 1.

6: A(LDA,*) — DOUBLE PRECISION array *Local Output*

Note: the 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 two-dimensional 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 F01ZSFP is called.

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

8: IFAIL — INTEGER *Global Input/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 entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this argument (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 (or −9999 if reduced error checking is enabled) 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).

5.1 Full Error Checking Mode Only

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 a cyclic two-dimensional block distribution.

6.2 Parallelism Detail

The routine generates the matrix on each logical processor independently.

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

8 Example

To solve the system of equations

$$Ax = b,$$

where the matrix A is given by

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

and the right-hand side is the vector

$$b = \begin{pmatrix} 4 \\ 7 \\ 9 \\ 10 \end{pmatrix}.$$

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 F01ZSFP 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 of a fourth order system, 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

```
*     F01ZSFP 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
      PARAMETER        (LDB=LDA)
*     .. Local Scalars ..
      INTEGER          ICNTXT, IFAIL, N, NCOLS, NROWS
      LOGICAL          ROOT
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,TDA), B(LDB,TDB), WORK(NMAX)
*     .. External Functions ..
      LOGICAL          Z01ACFP
      EXTERNAL         Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         F01ZSFP, F04FBFP, GMATA, X04BGFP, X04BHFP,
     +                 Z01AAFP, Z01ABFP
*     .. Executable Statements ..
      ROOT = Z01ACFP()
      IF (ROOT) WRITE (NOUT,*) 'F01ZSFP Example Program Results'
*
      NROWS = MG
      NCOLS = NG
      IFAIL = 0
*
      CALL Z01AAFP(ICNTXT,NROWS,NCOLS,IFAIL)
*
```

```
      OPEN (NIN,FILE='f01zsfpe.d')
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
*
*     Generate the matrix A
*
      IFAIL = 0
      CALL F01ZSFP(ICNTXT,GMATA,N,N,NB,A,LDA,IFAIL)
*
*     Read in the right hand side
*
      IFAIL = 0
      CALL X04BGFP(ICNTXT,NIN,N,1,NB,B,LDB,IFAIL)
*
*     Solve the system
*
      IFAIL = 0
      CALL F04FBFP(ICNTXT,N,NB,A,LDA,1,B,LDB,IFAIL)
*
*     Print the results
*
      IFAIL = 0
      CALL X04BHFP(ICNTXT,NOUT,N,1,NB,B,LDB,'(1X,F7.3)',WORK,IFAIL)
*
      IFAIL = 0
      CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
      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)
*
*     in the array AL.
*
*     .. Scalar Arguments ..
      INTEGER          I1, I2, J1, J2, LDAL
*     .. Array Arguments ..
      DOUBLE PRECISION AL(LDAL,*)
*     .. Local Scalars ..
      INTEGER          I, J, K, L
*     .. Intrinsic Functions ..
      INTRINSIC        MIN
*     .. Executable Statements ..
      L = 1
      DO 40 J = J1, J2
         K = 1
         DO 20 I = I1, I2
            AL(K,L) = MIN(I,J)
            K = K + 1
   20    CONTINUE
         L = L + 1
   40 CONTINUE
*
```

```
      RETURN
      END
```

8.2 Example Data

```
F01ZSFP Example Program Data
4      : Value of N
 4.0
 7.0
 9.0
10.0   : End of right hand side vector B
```

8.3 Example Results

```
 F01ZSFP Example Program Results
   1.000
   1.000
   1.000
   1.000
```