F01ZMFP

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

F01ZMFP generates and distributes an m by n real matrix A on the Library Grid in row block form, as required by some of the routines in Chapter C06 and Chapter F04. A user-supplied subroutine is required to generate a block of the matrix A.

2 Specification

```
SUBROUTINE F01ZMFP(ICNTXT, GMAT, M, N, A, LDA, MX, IFAIL)
DOUBLE PRECISION   A(LDA,*)
INTEGER            ICNTXT, M, N, LDA, MX, 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	–	$m_p \times n_p$, the total number of processors in the Library Grid.
M_b	–	the maximum blocksize for the distribution of the rows of the matrix.
M_x	–	the number of rows of the matrix A stored locally on a logical processor, where $0 \leq M_x \leq M_b$.
$\lceil x \rceil$	–	the ceiling function of x, which gives the smallest integer which is not less than x.

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, IFAIL

Global output arguments: IFAIL

The remaining arguments are local.

3.3 Distribution Strategy

Rows of the matrix A are allocated to logical processors on the two-dimensional Library Grid row by row (i.e., in the row major ordering of the grid) starting from the {0,0} logical processor. Each logical processor that contains rows of the matrix contains $M_b = \lceil m/p \rceil$ rows, except the last processor that actually contains data, for which the number of rows held may be less than M_b. This processor will contain $\mathrm{mod}(m, M_b)$ rows if $\mathrm{mod}(m, M_b) \neq 0$, and will contain M_b rows otherwise. Some logical processors may not contain any rows of the matrix if m is not large relative to p, but if $m > (p-1)^2$ then all processors will contain at least one row of the matrix.

The number of logical processors that contain rows of the matrix A is given by $p_d = \lceil m/M_b \rceil$.

The following example illustrates a case where the last processor with data is not the last processor of the grid. Furthermore the number of rows on the last processor with data is not equal to the number of rows on other processors.

If $m_p = 2$, $n_p = 4$ then $p = m_p \times n_p = 8$. If $m = 41$ then $M_b = \lceil m/p \rceil = \lceil 5.125 \rceil = 6$ and $\mathrm{mod}(m, M_b) = 5$.

processor {0,0} $M_x = 6$ rows (1:6)	processor {0,1} $M_x = 6$ rows (7:12)	processor {0,2} $M_x = 6$ rows (13:18)	processor {0,3} $M_x = 6$ rows (19:24)
processor {1,0} $M_x = 6$ rows (25:30)	processor {1,1} $M_x = 6$ rows (31:36)	processor {1,2} $M_x = 5$ rows (37:41)	processor {1,3} $M_x = 0$

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 $A(i_1 : i_2, 1 : n)$ of the matrix to be distributed.

Its specification is:

```
SUBROUTINE          GMAT(I1, I2, N, AL, LDAL)
DOUBLE PRECISION    AL(LDAL,*)
INTEGER             I1, I2, N, 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: N — INTEGER *Global Input*

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

4: AL(LDAL,*) — DOUBLE PRECISION array *Local Output*

On exit: AL must contain the block $A(i_1 : i_2, 1 : n)$ of the matrix A in its first n columns and $(i_2 - i_1 + 1)$ rows.

5: LDAL — INTEGER *Local Input*

On entry: the size of the first dimension of the array LDAL as declared in the (sub)program from which F01ZMFP is called.

GMAT must be declared as EXTERNAL in the (sub)program from which F01ZMFP 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: A(LDA,*) — DOUBLE PRECISION array *Local Output*

On exit: the local part of the matrix A.

6: LDA — INTEGER *Local Input*

On entry: the size of the first dimension of the array A as declared in the (sub)program from which F01ZMFP is called.

Constraint: LDA $\geq$ max(1,M_x).

Note: the utility routine Z01CFFP can be used to obtain M_x.

7: MX — INTEGER *Local Output*

On exit: M_x, the number of rows of the matrix A held by the logical processor.

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 generate distributed data in the form required by routines in Chapters C06 and F04.

6.1 Algorithmic Detail

None.

6.2 Parallelism Detail

The routine generates the row blocks 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 generate the 7 by 6 matrix A given by

$$A = \begin{pmatrix} 2.0 & 2.0 & 3.0 & 4.0 & 5.0 & 6.0 \\ 2.0 & 3.0 & 3.0 & 4.0 & 5.0 & 6.0 \\ \hline 3.0 & 3.0 & 4.0 & 4.0 & 5.0 & 6.0 \\ 4.0 & 4.0 & 4.0 & 5.0 & 5.0 & 6.0 \\ \hline 5.0 & 5.0 & 5.0 & 5.0 & 6.0 & 6.0 \\ 6.0 & 6.0 & 6.0 & 6.0 & 6.0 & 7.0 \\ \hline 7.0 & 7.0 & 7.0 & 7.0 & 7.0 & 7.0 \end{pmatrix}$$

on a two-dimensional processor grid and to print the matrix on the root processor. Routine F01ZMFP is used to generate the matrix A on a 2 by 2 logical processor grid. The horizontal lines in the matrix indicate the row block of the matrix. Routine X04BFFP is used to output the matrix.

8.1 Example Text

```
*     F01ZMFP Example Program Text
*     NAG Parallel Library Release 3. NAG Copyright 1999.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      INTEGER          M, N
      PARAMETER        (M=7,N=6)
      INTEGER          MG, NG
      PARAMETER        (MG=2,NG=2)
      INTEGER          LDA, TDA
      PARAMETER        (LDA=(M/(MG*NG)+1),TDA=N)
      CHARACTER*20     FORMT
      PARAMETER        (FORMT='F12.4')
*     .. Local Scalars ..
      INTEGER          ICNTXT, ICOFF, IFAIL, MP, MX, NP
      LOGICAL          ROOT
      CHARACTER        CNUMOP, TITOP
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,TDA), W(LDA,TDA)
*     .. External Functions ..
      LOGICAL          Z01ACFP
      EXTERNAL         Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         F01ZMFP, GMATA, X04BFFP, Z01AAFP, Z01ABFP
*     .. Executable Statements ..
      ROOT = Z01ACFP()
      IF (ROOT) THEN
         WRITE (NOUT,*) 'F01ZMFP Example Program Results'
         WRITE (NOUT,*)
      END IF
*
*     Define the 2D processor grid
*
```

```
      MP = MG
      NP = NG
      IFAIL = 0
*
      CALL Z01AAFP(ICNTXT,MP,NP,IFAIL)
*
      IFAIL = 0
*
*     Generate the matrix A
*
      CALL F01ZMFP(ICNTXT,GMATA,M,N,A,LDA,MX,IFAIL)
*
*     Print the matrix A
*
      IF (ROOT) THEN
         WRITE (NOUT,*) 'Generated Matrix'
         WRITE (NOUT,*)
         TITOP = 'Y'
         CNUMOP = 'L'
      END IF
      ICOFF = 0
      IFAIL = 0
*
      CALL X04BFFP(ICNTXT,NOUT,MX,N,A,LDA,FORMT,TITOP,CNUMOP,ICOFF,W,
     +             LDA,IFAIL)
*
*     Undefine the 2D grid
*
      CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
      STOP
      END
*
      SUBROUTINE GMATA(I1,I2,N,AL,LDAL)
*
*     GMATA generates the block A( I1: I2, 1: N ) of the matrix A such
*     that
*
*       a(i,j) = i + 1    if i=j
*       a(i,j) = max(i,j) otherwise
*
*     in the array AL.
*
*     .. Scalar Arguments ..
      INTEGER          I1, I2, LDAL, N
*     .. Array Arguments ..
      DOUBLE PRECISION AL(LDAL,*)
*     .. Local Scalars ..
      INTEGER          I, J, L
*     .. Intrinsic Functions ..
      INTRINSIC        MAX
*     .. Executable Statements ..
      DO 40 J = 1, N
         L = 1
         DO 20 I = I1, I2
            IF (I.NE.J) THEN
               AL(L,J) = MAX(I,J)
            ELSE
```

```
              AL(L,J) = I + 1
           END IF
           L = L + 1
   20   CONTINUE
   40 CONTINUE
*
*     End of GMATA.
*
      RETURN
      END
```

8.2 Example Data

None.

8.3 Example Results

```
 F01ZMFP Example Program Results

 Generated Matrix

   Array from logical processor    0,   0

             1            2            3            4            5            6
        2.0000       2.0000       3.0000       4.0000       5.0000       6.0000
        2.0000       3.0000       3.0000       4.0000       5.0000       6.0000

   Array from logical processor    0,   1

             1            2            3            4            5            6
        3.0000       3.0000       4.0000       4.0000       5.0000       6.0000
        4.0000       4.0000       4.0000       5.0000       5.0000       6.0000

   Array from logical processor    1,   0

             1            2            3            4            5            6
        5.0000       5.0000       5.0000       5.0000       6.0000       6.0000
        6.0000       6.0000       6.0000       6.0000       6.0000       7.0000

   Array from logical processor    1,   1

             1            2            3            4            5            6
        7.0000       7.0000       7.0000       7.0000       7.0000       7.0000
```