F01ZHFP

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

F01ZHFP generates and distributes an l by m by n real array $A(i,j,k)$ on the Library Grid in i-block form, i.e. each processor which holds data holds a number of jk-planes of the array. This routine distributes arrays in the form required by routines in Chapter C06. A user-supplied subroutine is required to generate a block of the matrix A.

2 Specification

```
SUBROUTINE F01ZHFP(ICNTXT, GMAT, L, M, N, A, LDA1, LDA2, LX, IFAIL)
DOUBLE PRECISION   A(LDA1,LDA2,*)
INTEGER            ICNTXT, L, M, N, LDA1, LDA2, LX, 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.
L_b	–	the maximum blocking size for the distribution of the indices i (jk-planes) of the array A.
L_x	–	the number of indices i (jk-planes) of the array A stored locally on a logical processor, where $0 \leq L_x \leq L_b$.
$\lceil x \rceil$	–	the ceiling function of x, which gives the smallest integer greater than or equal to 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: L, M, N, IFAIL

Global output arguments: IFAIL

The remaining arguments are local.

3.3 Distribution Strategy

Slices of jk-planes of the three-dimensional array 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 slices of the array contains $L_b = \lceil l/p \rceil$ jk-planes, except the last processor that actually contains data, for which the number of jk-planes held may be less than L_b. This processor will contain $\mathrm{mod}(l, L_b)$ jk-planes if $\mathrm{mod}(l, L_b) \neq 0$, and will contain L_b jk-planes otherwise. Some logical processors may not contain any rows of the three-dimensional array if l is not large relative to p, but if $l > (p-1)^2$ then all processors will contain at least one jk-plane of the three-dimensional array.

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

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

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

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 $A(i_1 : i_2, 1 : m, 1 : n)$ of the array to be distributed, in the array AL. That is, GMAT must return jk-planes i_1 to i_2 of A.

Its specification is:

```
SUBROUTINE          GMAT(I1, I2, M, N, AL, LDAL1, LDAL2)
DOUBLE PRECISION    AL(LDAL1,LDAL2,*)
INTEGER             I1, I2, M, N, LDAL1, LDAL2
```

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: M — INTEGER *Global Input*

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

4: N — INTEGER *Global Input*

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

5: AL(LDAL1,LDAL2,*) — DOUBLE PRECISION *Local Output*

On exit: AL must contain the block $A(i_1 : i_2, 1 : m, 1 : n)$ of the array A in its first $(i_2 - i_1 + 1)$ jk-planes.

6: LDAL1 — INTEGER *Local Input*

On entry: the size of the first dimension of the array AL as declared in the (sub)program from which F01ZHFP is called.

7: LDAL2 — INTEGER *Local Input*

On entry: the second dimension of the array AL as declared in the (sub)program from which F01ZHFP is called.

GMAT must be declared as EXTERNAL in the (sub)program from which F01ZHFP is called. Arguments denoted as *Input* must **not** be changed by this procedure.

3: L — INTEGER *Global Input*

On entry: l, the first dimension of A.

Constraint: L ≥ 0.

4: M — INTEGER *Global Input*

On entry: m, the second dimension of A.

Constraint: M ≥ 0.

5: N — INTEGER *Global Input*

On entry: n, the third dimension of A.

Constraint: N ≥ 0.

6: A(LDA1,LDA2,*) — DOUBLE PRECISION *Local Output*

On exit: the local part of the array A.

7: LDA1 — INTEGER *Local Input*

On entry: the size of the first dimension of the array A as declared in the (sub)program from which F01ZHFP is called.

Constraint: LDA1 $\geq \max(1,L_x)$.

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

8: LDA2 — INTEGER *Local Input*

On entry: the second dimension of the array A as declared in the (sub)program from which F01ZHFP is called.

Constraint: LDA2 $\geq \max(1,\text{M})$.

9: LX — INTEGER *Local Output*

On exit: L_x, the number of jk-planes of the array A held by the logical processor.

10: 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 the routines in Chapter C06.

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 by 3 array A whose first plane is given by

$$A_{*,*,1} = \left(\begin{array}{cccccc} 2.0 & 2.0 & 3.0 & 4.0 & 5.0 & 6.0 \\ 2.0 & 3.0 & 3.0 & 4.0 & 5.0 & 6.0 \\ \hline 3.0 & 3.0 & 4.0 & 4.0 & 5.0 & 6.0 \\ 4.0 & 4.0 & 4.0 & 5.0 & 5.0 & 6.0 \\ \hline 5.0 & 5.0 & 5.0 & 5.0 & 6.0 & 6.0 \\ 6.0 & 6.0 & 6.0 & 6.0 & 6.0 & 7.0 \\ \hline 7.0 & 7.0 & 7.0 & 7.0 & 7.0 & 7.0 \end{array}\right),$$

second plane given by

$$A_{*,*,2} = \left(\begin{array}{cccccc} 3.0 & 3.0 & 4.0 & 5.0 & 6.0 & 7.0 \\ 3.0 & 4.0 & 4.0 & 5.0 & 6.0 & 7.0 \\ \hline 4.0 & 4.0 & 5.0 & 5.0 & 6.0 & 7.0 \\ 5.0 & 5.0 & 5.0 & 6.0 & 6.0 & 7.0 \\ \hline 6.0 & 6.0 & 6.0 & 6.0 & 7.0 & 7.0 \\ 7.0 & 7.0 & 7.0 & 7.0 & 7.0 & 8.0 \\ \hline 8.0 & 8.0 & 8.0 & 8.0 & 8.0 & 8.0 \end{array}\right)$$

and third plane given by

$$A_{*,*,3} = \left(\begin{array}{cccccc} 4.0 & 4.0 & 5.0 & 6.0 & 7.0 & 8.0 \\ 4.0 & 5.0 & 5.0 & 6.0 & 7.0 & 8.0 \\ \hline 5.0 & 5.0 & 6.0 & 6.0 & 7.0 & 8.0 \\ 6.0 & 6.0 & 6.0 & 7.0 & 7.0 & 8.0 \\ \hline 7.0 & 7.0 & 7.0 & 7.0 & 8.0 & 8.0 \\ 8.0 & 8.0 & 8.0 & 8.0 & 8.0 & 9.0 \\ \hline 9.0 & 9.0 & 9.0 & 9.0 & 9.0 & 9.0 \end{array}\right)$$

on a two-dimensional processor grid and to print the matrix on the root processor.

Routine F01ZHFP is used to generate the matrix A on a 2 by 2 logical processor grid. The horizontal lines in each array indicate the row blocks of the array. Routine X04BFFP is used to output the matrix.

8.1 Example Text

```
*     F01ZHFP Example Program Text
*     NAG Parallel Library Release 3. NAG Copyright 1999.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      INTEGER          L, M, N
      PARAMETER        (L=7,M=6,N=3)
      INTEGER          MG, NG
      PARAMETER        (MG=2,NG=2)
      INTEGER          LDA1, LDA2
      PARAMETER        (LDA1=(L/(MG*NG)+1),LDA2=M)
      CHARACTER*20     FORMT
      PARAMETER        (FORMT='F12.4')
*     .. Local Scalars ..
      INTEGER          ICNTXT, ICOFF, IFAIL, II, LX, MP, NP
      LOGICAL          ROOT
      CHARACTER        CNUMOP, TITOP
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA1,LDA2,N), W(LDA1,M)
*     .. External Functions ..
      LOGICAL          Z01ACFP
      EXTERNAL         Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         F01ZHFP, GMATA, X04BFFP, Z01AAFP, Z01ABFP
*     .. Executable Statements ..
*
      ROOT = Z01ACFP()
      IF (ROOT) THEN
         WRITE (NOUT,*) 'F01ZHFP Example Program Results'
         WRITE (NOUT,*)
      END IF
*
*     Define the Library Grid
*
      MP = MG
      NP = NG
      IFAIL = 0
*
      CALL Z01AAFP(ICNTXT,MP,NP,IFAIL)
*
      IFAIL = 0
*
*     Generate the matrix A
*
      CALL F01ZHFP(ICNTXT,GMATA,L,M,N,A,LDA1,LDA2,LX,IFAIL)
*
*     Print the matrix A
*
      IF (ROOT) THEN
         WRITE (NOUT,*) 'Generated Matrix'
         WRITE (NOUT,*)
         TITOP = 'Y'
         CNUMOP = 'L'
      END IF
*
*     Print the jk-planes of the matrix on each local processor
```

```
*
      DO 20 II = 1, N
         ICOFF = 0
         IFAIL = 0
         IF (ROOT) THEN
            WRITE (NOUT,*) 'Plane', II,
     +        ' from each logical processor'
            WRITE (NOUT,*)
         END IF

         CALL X04BFFP(ICNTXT,NOUT,LX,M,A(1,1,II),LDA1,FORMT,TITOP,
     +                CNUMOP,ICOFF,W,LDA1,IFAIL)
*
   20 CONTINUE
*
*     Undefine the Library Grid
*
      CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
      STOP
      END
*
      SUBROUTINE GMATA(I1,I2,M,N,AL,LDAL1,LDAL2)
*
*     GMATA generates the block A( I1: I2, 1: M, 1:N) of the array A
*     such that
*
*       a(i,j,k) = i + 1 + (k-1)    if i=j
*       a(i,j,k) = max(i,j) + (k-1) otherwise
*
*     in the array AL.
*
*     .. Scalar Arguments ..
      INTEGER          I1, I2, LDAL1, LDAL2, M, N
*     .. Array Arguments ..
      DOUBLE PRECISION AL(LDAL1,LDAL2,*)
*     .. Local Scalars ..
      INTEGER          I, II, J, K
*     .. Intrinsic Functions ..
      INTRINSIC        MAX
*     .. Executable Statements ..
      DO 60 K = 1, N
         DO 40 J = 1, M
            II = 1
            DO 20 I = I1, I2
               IF (I.NE.J) THEN
                  AL(II,J,K) = MAX(I,J) + K - 1
               ELSE
                  AL(II,J,K) = I + 1 + K - 1
               END IF
               II = II + 1
   20       CONTINUE
   40    CONTINUE
   60 CONTINUE
*
*     End of GMATA.
*
      RETURN
```

```
      END
```

8.2 Example Data

None.

8.3 Example Results

```
 F01ZHFP Example Program Results

 Generated Matrix

 Plane           1 from each logical processor

   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

 Plane           2 from each logical processor

   Array from logical processor    0,   0

           1          2          3          4          5          6
      3.0000     3.0000     4.0000     5.0000     6.0000     7.0000
      3.0000     4.0000     4.0000     5.0000     6.0000     7.0000

   Array from logical processor    0,   1

           1          2          3          4          5          6
      4.0000     4.0000     5.0000     5.0000     6.0000     7.0000
      5.0000     5.0000     5.0000     6.0000     6.0000     7.0000

   Array from logical processor    1,   0

           1          2          3          4          5          6
      6.0000     6.0000     6.0000     6.0000     7.0000     7.0000
      7.0000     7.0000     7.0000     7.0000     7.0000     8.0000
```

```
  Array from logical processor    1,   1

          1          2          3          4          5          6
     8.0000     8.0000     8.0000     8.0000     8.0000     8.0000

Plane           3 from each logical processor

  Array from logical processor    0,   0

          1          2          3          4          5          6
     4.0000     4.0000     5.0000     6.0000     7.0000     8.0000
     4.0000     5.0000     5.0000     6.0000     7.0000     8.0000

  Array from logical processor    0,   1

          1          2          3          4          5          6
     5.0000     5.0000     6.0000     6.0000     7.0000     8.0000
     6.0000     6.0000     6.0000     7.0000     7.0000     8.0000

  Array from logical processor    1,   0

          1          2          3          4          5          6
     7.0000     7.0000     7.0000     7.0000     8.0000     8.0000
     8.0000     8.0000     8.0000     8.0000     8.0000     9.0000

  Array from logical processor    1,   1

          1          2          3          4          5          6
     9.0000     9.0000     9.0000     9.0000     9.0000     9.0000
```