F01YZFP

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

F01YZFP generates and distributes an $m \times n$ complex dense matrix B on a one-dimensional logical grid of processors in a row block distribution.

This routine distributes matrices in the form required by a number of the routines in Chapter F07 (especially F07HSFP (PZPBTRS) and F07JSFP (PZPTTRS) for solving banded and tridiagonal systems respectively). A user-supplied subroutine is required to generate a block of the matrix B on each processor.

2 Specification

```
SUBROUTINE F01YZFP(GMAT, M, N, B, IDESCB, IFAIL)
COMPLEX*16         B(*)
INTEGER            M, N, IDESCB(*), IFAIL
EXTERNAL           GMAT
```

3 Usage

3.1 Definitions

The following definitions are used in describing the data distribution within this document:

p	–	$m_p \times n_p$, the total number of processors in the Library Grid.
m_p	–	the number of rows in the Library Grid, for this routine $m_p = 1$ or $m_p = p$;
n_p	–	the number of columns in the Library Grid, for this routine $n_p = 1$ or $n_p = p$.
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.

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, some elements of IDESCB (see Section 4 for a description of IDESCB), IFAIL
Global output arguments:	IFAIL

The remaining arguments are local.

3.3 Distribution Strategy

The matrix B will be partitioned into M_b^B row blocks on a one-dimensional processor grid, and stored in the array B. This data distribution (**row block distribution**) is described in more detail in the F07 Chapter Introduction.

4 Arguments

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

GMAT must return the block $B(i_1 : i_2, 1 : n)$ of the matrix to be distributed, in the array BL (i.e., $B_{i,j}$ where i, j are such that $i_1 \leq i \leq i_2$ and $1 \leq j \leq n$).

Its specification is:

```
SUBROUTINE          GMAT(I1, I2, N, BL, LDBL)
COMPLEX*16          BL(LDBL,*)
INTEGER             I1, I2, N, LDBL
```

1: I1 — INTEGER *Local Input*

On entry: i_1, the first row of the block of B to be generated.

2: I2 — INTEGER *Local Input*

On entry: i_2, the last row of the block of B to be generated.

3: N — INTEGER *Local Input*

On entry: n, the number of columns of the matrix B to be generated, and in some routines it is the number of right-hand sides of a linear system.

4: BL(LDBL,*) — COMPLEX*16 array *Local Output*

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

5: LDBL — INTEGER *Local Input*

On entry: the size of the first dimension of the array BL as declared in the (sub)program from which F01YZFP is called.

GMAT must be declared as EXTERNAL in the (sub)program from which F01YZFP is called. Arguments denoted as *Input* must **not** be changed by this procedure.

2: M — INTEGER *Global Input*

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

Constraint: M ≥ 0.

3: N — INTEGER *Global Input*

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

Constraint: N ≥ 0.

4: B(∗) — COMPLEX*16 array *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 (LDB,γ) where LDB = IDESCB(9) if IDESCB(1) = 1, or LDB = IDESCB(6) if IDESCB(1) = 502; and $\gamma \geq$ N.

On exit: the local part of the matrix B stored in row block fashion.

5: IDESCB(∗) — INTEGER array *Local Input*

Note: the dimension of the array IDESCB must be at least 9 when IDESCB(1) = 1, and at least 6 when IDESCB(1) = 502.

Distribution: if IDESCB(1) = 1, the array elements IDESCB(3:8) must be global to each processor on the Library Grid. If IDESCB(1) = 502, then only the array elements IDESCB(3:5) must be global. In either cases IDESCB(2) is 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.
If IDESCB(1) = 502, then $p = 1 \times n_p$ or $p = m_p \times 1$, and the remaining elements of IDESCB must be set as follows:

IDESCB(2), the Library context, usually returned by a call to the Library Grid initialisation routine Z01AAFP;
IDESCB(3), the number of rows, m, of the matrix B;
IDESCB(4), the blocking factor, M_b^B, used to distribute the matrix B;
IDESCB(5), the processor index over which the first row of the matrix B is distributed;
IDESCB(6), the leading dimension of the conceptual two-dimensional array B;
IDESCB(7:9) are not referenced.

If IDESCB(1) = 1, then $p = m_p \times 1$ and the remaining elements of IDESCB must be set as follows:

IDESCB(2), the Library context, usually returned by a call to the Library Grid initialisation routine Z01AAFP;
IDESCB(3), the number of rows, m, of the matrix B;
IDESCB(4), the number of columns, n, of the matrix B;
IDESCB(5), is not referenced;
IDESCB(6), the blocking factor, M_b^B, used to distribute the rows of the matrix B;
IDESCB(7), the processor index over which the first row of the matrix B is distributed;
IDESCB(8), is not referenced;
IDESCB(9), the leading dimension of the conceptual two-dimensional array B.

Suggested value: IDESCB(1) = 502 and $p = 1 \times n_p$, if IDESCB(1) = 1, IDESCB(5) = IDESCB(8) = 1.

Constraints:

IDESCB(1) = 1 or 502;
if IDESCB(1) = 1, then $p = m_p \times 1$;
if IDESCB(1) = 502, then $p = m_p \times 1$ or $p = 1 \times n_p$;
if IDESCB(1) = 502, then
IDESCB(3) = M;
IDESCB(4) $\geq$ 2 and $p\times$IDESCB(4) $\geq$ M;
IDESCB(5) $\geq$ 0;
if IDESCB(1) = 1, then
IDESCB(3) = M;
IDESCB(4) = N;
IDESCB(6) $\geq$ 2 and $p\times$IDESCB(6) $\geq$ M;
IDESCB(7) $\geq$ 0.

6: 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 a value of ICNTXT (stored in IDESCB(2)) which was not returned by a call to Z01AAFP on one or more processors.

IFAIL = −1000

The utility routine Z01AAFP has not been called to define the logical processor grid and initialise the internal variables used by the Library.

IFAIL = $-i$

On entry, one of the arguments was invalid:

if the kth argument is a scalar IFAIL = $-k$;
if the kth argument is an array and its jth element is invalid, IFAIL = $-(100 \times k + j)$.

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

6.1 Algorithmic Detail

The routine generates a complex matrix, on a one-dimensional grid of logical processor using a row 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

This example program generates a 12×2 matrix such that: $B = [b^{(1)}\ b^{(2)}]$, and

$$b_j^{(1)} = 2(1+2i) \times j \text{ and } b_j^{(2)} = (1+2i)j \quad \text{for } j = 1, \ldots, 12.$$

8.1 Example Text

```
*     F01YZFP Example Program Text
*     NAG Parallel Library Release 3. NAG Copyright 1999.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      INTEGER          NMAX, NB
      PARAMETER        (NMAX=15,NB=3)
```

```
      INTEGER          IAROW, NCOLMX
      PARAMETER        (IAROW=0,NCOLMX=5)
      INTEGER          NG
      PARAMETER        (NG=4)
      INTEGER          TDB, LDB
      PARAMETER        (TDB=NMAX/NG,LDB=TDB)
*     .. Local Scalars ..
      INTEGER          ICNTXT, IFAIL, MP, N, NCOL, NP
      LOGICAL          ROOT
      CHARACTER*80     FORMATB
*     .. Local Arrays ..
      COMPLEX*16       B(LDB,NCOLMX), WORKB(LDB,NCOLMX)
      INTEGER          IDESCB(9)
*     .. External Functions ..
      LOGICAL          Z01ACFP
      EXTERNAL         Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         F01YZFP, GMAT, X04BZFP, Z01AAFP, Z01ABFP
*     .. Executable Statements ..
      ROOT = Z01ACFP()
      IF (ROOT) THEN
         WRITE (NOUT,*) 'F01YZFP Example Program Results '
         WRITE (NOUT,*)
      END IF
*
*     Define the 1D processor grid.
*
      MP = 1
      NP = NG
      IFAIL = 0
*
      CALL Z01AAFP(ICNTXT,MP,NP,IFAIL)
*
      N = 12
      NCOL = 2
      FORMATB = '(4F12.4)'
*
      IF (N.LE.NMAX .AND. NCOL.LE.NCOLMX) THEN
*
*     Set the array descriptor of B
*
         IDESCB(1) = 502
         IDESCB(2) = ICNTXT
         IDESCB(3) = N
         IDESCB(4) = NB
         IDESCB(5) = IAROW
         IDESCB(6) = LDB
         IDESCB(7) = 0
*
*     Generate the matrix B
*
         CALL F01YZFP(GMAT,N,NCOL,B,IDESCB,IFAIL)
*
*     Print the matrix
*
         IF (ROOT) THEN
            WRITE (NOUT,*) ' The matrix B '
            WRITE (NOUT,*)
```

```
         END IF
*
         CALL X04BZFP(NOUT,N,NCOL,B,IDESCB,FORMATB,WORKB,IFAIL)
*
      END IF
*
      IFAIL = 0
      CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
      STOP
      END
*
      SUBROUTINE GMAT(I1,IL,NCOL,BL,LDBL)
*     .. Parameters ..
      COMPLEX*16      ONEP2I
      PARAMETER       (ONEP2I=(1.0D0,2.0D0))
*     .. Scalar Arguments ..
      INTEGER         I1, IL, LDBL, NCOL
*     .. Array Arguments ..
      COMPLEX*16      BL(LDBL,*)
*     .. Local Scalars ..
      INTEGER         I, K
*     .. Intrinsic Functions ..
      INTRINSIC       DBLE
*     .. Executable Statements ..
      K = 1
      DO 20 I = I1, IL
*
*        In this example NCOL = 2 so no need for an explicit loop
*
         BL(K,1) = DBLE(2*I)*ONEP2I
         BL(K,2) = DBLE(I)*ONEP2I
         K = K + 1
   20 CONTINUE
*
*     End of GMAT
*
      RETURN
      END
```

8.2 Example Data

None.

8.3 Example Results

```
F01YZFP Example Program Results

 The matrix B

     2.0000      4.0000      1.0000      2.0000
     4.0000      8.0000      2.0000      4.0000
     6.0000     12.0000      3.0000      6.0000
     8.0000     16.0000      4.0000      8.0000
    10.0000     20.0000      5.0000     10.0000
    12.0000     24.0000      6.0000     12.0000
    14.0000     28.0000      7.0000     14.0000
    16.0000     32.0000      8.0000     16.0000
```

```
18.0000     36.0000      9.0000     18.0000
20.0000     40.0000     10.0000     20.0000
22.0000     44.0000     11.0000     22.0000
24.0000     48.0000     12.0000     24.0000
```
