

F01YAFP

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

F01YAFP generates and distributes an n by n real sparse unsymmetric matrix A , represented in coordinate storage format, on a logical grid of processors in cyclic row block form (see Section 2.5 of the F11 Chapter Introduction).

This routine distributes matrices in the form required by a number of F11 routines (see the F01 Chapter Introduction). A user-supplied subroutine is required to generate a block of the matrix A .

2 Specification

```

SUBROUTINE F01YAFP(ICNTXT, GMAT, N, MB, NNZ, A, LA, IROW, ICOL,
1          IFAIL)
DOUBLE PRECISION A(LA)
INTEGER          ICNTXT, N, MB, NNZ, LA, IROW(LA), ICOL(LA),
1          IFAIL
EXTERNAL        GMAT

```

3 Data Distribution

3.1 Definitions

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

M_b – the blocking factor for the distribution of the rows of the matrix.

3.2 Global and Local Arguments

The input arguments N , MB 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.

3.3 Distribution Strategy

Blocks of M_b contiguous rows of the matrix A are stored in coordinate storage format on a logical grid of processors cyclically row by row (i.e., in the row major ordering of the grid) starting from the $\{0,0\}$ logical processor. This data distribution is described in more detail in Section 2.5 of the F11 Chapter Introduction.

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 non-zero elements in the block $A(i_1 : i_2, 1 : n)$ of the matrix to be distributed in coordinate storage format.

Its specification is:

SUBROUTINE	GMAT(I1, I2, N, NNZL, AL, LAL, IROWL, ICOLL)	
DOUBLE PRECISION	AL(LAL)	
INTEGER	I1, I2, N, NNZL, LAL, IROWL(LAL), ICOLL(LAL)	
1:	I1 — INTEGER	<i>Local Input</i>
	<i>On entry:</i> i_1 , the first row of the block of A to be generated.	
2:	I2 — INTEGER	<i>Local Input</i>
	<i>On entry:</i> i_2 , the last row of the block of A to be generated.	
3:	N — INTEGER	<i>Global Input</i>
	<i>On entry:</i> n , the number of columns of the matrix A to be generated.	
4:	NNZL — INTEGER	<i>Local Output</i>
	<i>On exit:</i> the number of non-zero elements in the block $A(i_1 : i_2, 1 : n)$. NNZL must satisfy $NNZL \geq 0$.	
5:	AL(LAL) — DOUBLE PRECISION array	<i>Local Output</i>
	<i>On exit:</i> if $NNZL \leq LAL$, the first NNZL entries of AL must contain the non-zero elements in the block $A(i_1 : i_2, 1 : n)$.	
6:	LAL — INTEGER	<i>Local Input</i>
	<i>On entry:</i> the dimension of the arrays AL, IROWL and ICOLL.	
7:	IROWL(LAL) — INTEGER array	<i>Local Output</i>
8:	ICOLL(LAL) — INTEGER array	<i>Local Output</i>
	<i>On exit:</i> if $NNZL \leq LAL$, the first NNZL entries of IROWL and ICOLL must contain the row and column indices of the non-zero elements in the block $A(i_1 : i_2, 1 : n)$ supplied in AL. IROWL and ICOLL must satisfy $i_1 \leq IROWL(l) \leq i_2$ and $1 \leq ICOLL(l) \leq n$ for $l = 1, 2, \dots, NNZL$.	

GMAT must be declared as EXTERNAL in the (sub)program from which F01YAFF is called. Arguments denoted as *Input* must **not** be changed by this procedure.

- 3:** N — INTEGER *Global Input*
On entry: the order of the matrix A , n .
Constraint: $N \geq 1$.
- 4:** MB — INTEGER *Global Input*
On entry: the blocking factor, M_b , used to distribute the rows of the matrix A .
Constraint: $MB \geq 1$.
- 5:** NNZ — INTEGER *Local Output*
On exit: the number of non-zero elements in the blocks of the matrix A assigned to the calling processor.
- 6:** A(LA) — DOUBLE PRECISION array *Local Output*
On exit: if $NNZ \leq LA$, the first NNZ entries of A contain the non-zero elements in the blocks of the matrix A assigned to the calling processor.
- 7:** LA — INTEGER *Local Input*
On entry: the dimension of the arrays A, IROW and ICOL as declared in the (sub)program from which F01YAFF is called.
Constraint: $LA \geq 1$.

- 8:** IROW(LA) — INTEGER array *Local Output*
9: ICOL(LA) — INTEGER array *Local Output*

On exit: if $\text{NNZ} \leq \text{LA}$, the first NNZ entries of IROW and ICOL contain the row and column indices of the non-zero elements of A supplied in A.

- 10:** 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 i th argument had an invalid value. For global arguments, this may also be caused by an argument not having the same value on all logical processors. An explanatory message distinguishes between these two cases.

IFAIL = 1

LA is too small on at least one processor, resulting in insufficient space to store the local non-zero elements of A .

IFAIL = 2

On exit from GMAT the argument NNZL violated the constraint $\text{NNZL} \geq 0$.

IFAIL = 3

On exit from GMAT the argument IROWL or ICOLL violated the constraints $i_1 \leq \text{IROWL}(l) \leq i_2$ and $1 \leq \text{ICOLL}(l) \leq n$ for $l = 1, 2, \dots, \text{NNZL}$.

6 Further Comments

This routine may be used to distribute the data in the form required by a number of the routines in Chapter F11 (see the F01 Chapter Introduction).

6.1 Algorithmic Detail

The routine generates the coordinate storage representation of the matrix on a logical processor by cyclic row block.

6.2 Parallelism Detail

The routine generates the coordinate storage representation of the row blocks of the matrix on each logical processor independently.

7 References

None.

8 Example

See the Example Program for F11BAFP.
