F01YQFP

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

F01YQFP generates an n by n complex sparse matrix A, in coordinate storage format and distributed in cyclic row block fashion (see Section 2.5 of the F11 Chapter Introduction).

Depending on the value of the input parameter WHAT, F01YQFP generates either (i) both the numerical values and the row and column coordinates of the non-zero entries of the matrix A or (ii) only the numerical values of the non-zero entries. The latter option should be used if the matrix A has the same pattern of non-zero entries as a previously generated matrix.

This routine generates matrices in the form required by a number of routines in Chapter F11.

A user-supplied subroutine is required to generate a row block of the matrix A.

2 Specification

```
     SUBROUTINE F01YQFP(ICNTXT, GMAT, WHAT, N, MB, NNZ, A, LA, IROW,
    1                   ICOL, IFAIL)
     COMPLEX*16         A(LA)
     CHARACTER*1        WHAT
     INTEGER            ICNTXT, N, MB, NNZ, LA, IROW(LA), ICOL(LA),
    1                   IFAIL
     EXTERNAL           GMAT
```

3 Usage

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 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: WHAT, N, MB, IFAIL

Global output arguments: IFAIL

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

Its specification is:

```
SUBROUTINE         GMAT(WHAT, I1, I2, N, NNZL, AL, LAL, IROWL, ICOLL)
COMPLEX*16         AL(LAL)
INTEGER            I1, I2, N, NNZL, LAL, IROWL(LAL), ICOLL(LAL)
CHARACTER*1        WHAT
```

1: WHAT — CHARACTER*1 *Global Input*

On entry: specifies which parts of the coordinate storage representation of A are to be generated:

if WHAT = 'C', both the numerical values and the row and column coordinates of the non-zero entries of A are to be generated;

if WHAT = 'N', only the numerical values of the non-zero entries of A are to be generated; in this case IROWL and ICOLL must not be modified by GMAT.

2: I1 — INTEGER *Local Input*

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

3: I2 — INTEGER *Local Input*

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

4: N — INTEGER *Global Input*

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

5: NNZL — INTEGER *Local Output*

On exit: the number of non-zero entries in the generated block $A(i_1 : i_2, 1 : n)$. NNZL must satisfy $0 \leq$ NNZL $\leq$ LAL.

6: AL(LAL) — COMPLEX*16 array *Local Output*

On exit: the first NNZL elements of AL must contain the non-zero entries in the block $A(i_1 : i_2, 1 : n)$.

7: LAL — INTEGER *Local Input*

On entry: the dimension of the arrays AL, IROWL and ICOLL.

8: IROWL(LAL) — INTEGER array *Local Output*
9: ICOLL(LAL) — INTEGER array *Local Output*

On exit: if WHAT = 'C', then the first NNZL elements of IROWL and ICOLL must contain the row and column indices, respectively, of the non-zero entries in the block $A(i_1 : i_2, 1 : n)$ returned in AL. The elements of IROWL and ICOLL must satisfy $i_1 \leq$ IROWL$(l) \leq i_2$ and $1 \leq$ ICOLL$(l) \leq n$, for $l = 1, 2, \ldots,$NNZL.
If WHAT = 'N', then IROWL and ICOLL must not be modified by GMAT.

GMAT must be declared as EXTERNAL in the (sub)program from which F01YQFP is called. Arguments denoted as *Input* must **not** be changed by this procedure.

3: WHAT — CHARACTER*1 *Global Input*

On entry: specifies which parts of the coordinate storage representation of A are to be generated:

if WHAT = 'C', both the numerical values and the row and column coordinates of the non-zero entries of A are to be generated;

if WHAT = 'N', only the numerical values of the non-zero entries of A are to be generated. In this case IROW and ICOLL are not referenced by F01YQFP.

Constraint: WHAT= 'C' or 'N'.

4: N — INTEGER *Global Input*

On entry: n, the order of the matrix A.

Constraint: N ≥ 1.

5: MB — INTEGER *Global Input*

On entry: M_b, the blocking factor, used to distribute the rows of the matrix A.

Constraint: MB ≥ 1.

6: NNZ — INTEGER *Local Output*

On exit: the number of non-zero entries in the blocks of the matrix A stored locally.

7: A(LA) — COMPLEX*16 array *Local Output*

On exit: if IFAIL = 0, the first NNZ elements of A contain the non-zero entries of the row blocks of the matrix A stored locally.

8: LA — INTEGER *Local Input*

On entry: the dimension of the arrays A, IROW and ICOL as declared in the (sub)program from which F01YQFP is called.

Constraint: LA $\geq$ max(1, NNZ). That is, LA must be large enough to store the non-zero entries of the row blocks of the matrix A assigned locally.

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

10: ICOL(LA) — INTEGER array *Local Output*

On exit: if WHAT = 'C' and IFAIL = 0, then the first NNZ elements of IROW and ICOL return the row and column indices, respectively, of the non-zero entries in the blocks of the matrix A returned in A.
If WHAT = 'N', then IROW and ICOL are not referenced by F01YQFP.

11: 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.

5.2 Any Error Checking Mode

IFAIL = 1

LA is too small on at least one processor: there is not enough space to store the non-zero entries of the row blocks of the matrix A assigned to it.

IFAIL = 2

On exit from GMAT, NNZL < 0.

IFAIL = 3

On exit from GMAT one or more of the elements of IROWL and/or ICOLL were out of range. Namely, IROWL(i) $< i_1$, IROWL(i) $> i_2$, ICOLL(i) < 1 or ICOLL(i) $>$ N for some $i = 1, 2, \ldots$,NNZL. This error condition can occur only if WHAT = 'C'.

6 Further Comments

6.1 Algorithmic Detail

This routine successively calls the user-supplied generation routine for each row block to be stored locally.

6.2 Parallelism Detail

This routine generates the coordinate storage representation of the row blocks of the matrix A independently on each processor.

7 References

None.

8 Example

See Section 8 of the document for F11YNFP.