X04BSFP

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

X04BSFP outputs an m by n complex matrix A_s stored in a cyclic two-dimensional block distribution on a logical grid of processors to an external file (in its natural, non-distributed form); A_s is a submatrix of a larger m_A by n_A matrix A, i.e.,

$$A_s(1:m,1:n) \equiv A(i_A:i_A+m-1, j_A:j_A+n-1).$$

Note: if $i_A = j_A = 1$, $m = m_A$ and $n = n_A$, then $A_s = A$.

This routine outputs matrices stored in the form required by some routines in Chapter F07 and Chapter F08.

2 Specification

```
    SUBROUTINE X04BSFP(NOUT, M, N, A, IA, JA, IDESCA, FORMAT, WORK,
   1                   IFAIL)
    COMPLEX*16         A(*), WORK(*)
    INTEGER            NOUT, M, N, IA, JA, IDESCA(*), IFAIL
    CHARACTER*(*)      FORMAT
```

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_r	–	the row grid coordinate of the calling processor.
p_c	–	the column grid coordinate of the calling processor.
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.
numroc(α,b_ℓ,q,s,k)	–	a function which gives the **num**ber of **r**ows **o**r **c**olumns of a distributed matrix owned by the processor with the row or column coordinate q (p_r or p_c), where α is the total number of rows or columns of the matrix, b_ℓ is the blocking factor used (M_b^X or N_b^X), s is the row or column coordinate of the processor that possesses the first row or column of the distributed matrix and k is either m_p or n_p. The Library provides the function Z01CAFP (NUMROC) for the evaluation of this function.

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, IA, JA, IFAIL, IDESCA(1), IDESCA(3:8)

Global output arguments: IFAIL

Note: NOUT and FORMAT are only referenced on the root (or {0,0}) processor since it is only the root (or {0,0} processor) which performs output.

3.3 Distribution Strategy

The array A must contain the local portion of the matrix A. The matrix A should be partitioned into M_b^A by N_b^A rectangular blocks, distributed in a cyclic two-dimensional block fashion. This data distribution is described in more detail in the Essential Introduction of the NAG Parallel Library and in the F07 Chapter Introduction and the F08 Chapter Introduction.

This routine assumes that the data has already been correctly distributed, and if this is not the case will fail to produce correct results.

4 Arguments

1: NOUT — INTEGER *Local Input*

On entry: the unit number to which the output will be directed.

2: M — INTEGER *Global Input*

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

Constraint: $0 \leq$ M $\leq$ IDESCA(3).

3: N — INTEGER *Global Input*

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

Constraint: $0 \leq$ N $\leq$ IDESCA(4).

4: A($*$) — COMPLEX*16 array *Local Input*

Note: the array A is formally defined as a vector. However, you may find it more convenient to consider A as a two-dimensional array of dimension (IDESCA(9),γ), where $\gamma \geq$ numroc(JA+N$-$1,IDESCA(6),p_c,IDESCA(8),n_p). (See Section 8.)

On entry: the local part of the matrix A, distributed in a cyclic two-dimensional block fashion.

5: IA — INTEGER *Global Input*

On entry: i_A, the row index of A that identifies the first row of the submatrix A_s.

Constraint: $1 \leq$ IA $\leq$ IDESCA(3) $-$ M + 1.

6: JA — INTEGER *Global Input*

On entry: j_A, the column index of A that identifies the first column of the submatrix A_s.

Constraint: $1 \leq$ JA $\leq$ IDESCA(4) $-$ N + 1.

7: IDESCA($*$) — INTEGER array *Local Input*

Note: the dimension of the array IDESCA must be at least 9.

Distribution: the array elements IDESCA(1) and IDESCA(3),...,IDESCA(8) must be global to the processor grid and the elements IDESCA(2) and IDESCA(9) are local to each processor.

On entry: the description array for the matrix A. This array must contain details of the distribution of the matrix A and the logical processor grid.

IDESCA(1), the descriptor type. For this routine, which uses a cyclic two-dimensional block distribution, IDESCA(1) = 1;
IDESCA(2), the Library context, usually returned by a call to the Library Grid initialisation routine Z01AAFP;
IDESCA(3), the number of rows, m_A, of the matrix A;
IDESCA(4), the number of columns, n_A, of the matrix A;
IDESCA(5), the blocking factor, M_b^A, used to distribute the rows of the matrix A;
IDESCA(6), the blocking factor, N_b^A, used to distribute the columns of the matrix A;

IDESCA(7), the processor row index over which the first row of the matrix A is distributed;
IDESCA(8), the processor column index over which the first column of the matrix A is distributed;
IDESCA(9), the leading dimension of the conceptual two-dimensional array A.

Constraints:

IDESCA(1) = 1;
IDESCA(3) $\geq$ 0; IDESCA(4) $\geq$ 0;
IDESCA(5) $\geq$ 1; IDESCA(6) $\geq$ 1;
$0 \leq$ IDESCA(7) $\leq m_p - 1$; $0 \leq$ IDESCA(8) $\leq n_p - 1$;
IDESCA(9) $\geq$ max(1,numroc(IDESCA(3),IDESCA(5),p_r,IDESCA(7),m_p)).

8: FORMAT — CHARACTER*(*) *Local Input*

On entry: the format which will be used for output of the elements of A_s.

Constraint: any legal Fortran format for the output of floating-point numbers.

Note: for reasons of compatibility with the equivalent real routine and of portability, FORMAT must now contain a valid Fortran format, not just a field descriptor for a floating point number.

9: WORK($*$) — COMPLEX*16 array *Local Workspace*

Note: the dimension of the array WORK must be at least max(1,N).

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

IFAIL = -1000

The logical processor grid and library mechanism (Library Grid) have not been correctly defined, see Z01AAFP.

IFAIL < 0

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 occurred 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

An error has occurred in writing to unit NOUT. The file may have been opened for reading only or the user may have run out of disk space.

6 Further Comments

Output is performed by the root (or {0,0}) processor if the root processor is not available. All other processors communicate their local portion of the matrix to the root (or {0,0}) processor.

7 References

None.

8 Example

See Section 8 of the document for X04BRFP.