X04BVFP

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

X04BVFP reads an m by n complex matrix A from an external file (stored in its natural, non-distributed form) into an array in a cyclic two-dimensional block distribution on a logical grid of processors.

This routine distributes matrices in the form required by some routines in Chapter F04.

2 Specification

```
SUBROUTINE X04BVFP(ICNTXT, NIN, M, N, NB, A, LDA, IFAIL)
COMPLEX*16         A(LDA,*)
INTEGER            ICNTXT, NIN, M, N, NB, LDA, IFAIL
```

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.
N_b	–	the blocking factor for the distribution of the rows and 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 (N_b), 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 n_p or m_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, NB, IFAIL

Global output arguments: IFAIL

Note: NIN may have a different value on each processor but it is likely to be the same on each processor – the only constraint is that it must be attached to the same file (or identical files) on each processor.

3.3 Distribution Strategy

The cyclic two-dimensional block data distribution is described in more detail in the Essential Introduction of the NAG Parallel Library and in the F04 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: NIN — INTEGER *Local Input*

On entry: the unit number on which the external file is to be read.

Constraint: $0 \leq \text{NIN} \leq 99$.

3: M — INTEGER *Global Input*

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

Constraint: $\text{M} \geq 0$.

4: N — INTEGER *Global Input*

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

Constraint: $\text{N} \geq 0$.

5: NB — INTEGER *Global Input*

On entry: N_b, the blocking factor for distributing the matrix A.

Constraint: $\text{NB} \geq 1$.

6: A(LDA,$*$) — COMPLEX*16 array *Local Output*

Note: the size of the second dimension of the array A must be at least max(1,numroc(N,NB,p_c,0,n_p)).

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

7: LDA — INTEGER *Local Input*

On entry: the size of the first dimension of the array A as declared in the (sub)program from which X04BVFP is called.

Constraint: LDA $\geq$ max(1,numroc(M,NB,p_r,0,m_p))

8: 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

The routine has reached the end of the external file unexpectedly. This is probably due to supplying values for M and N which are too large for the amount of data actually stored in the file.

IFAIL = 2

An error has occurred while reading the data, probably due to the routine trying to read an incorrect data type. This may occur if there are non-numerical characters accidentally mixed with the matrix data or if the current read position is not at the beginning of the matrix data when the routine is called.

6 Further Comments

All of the processors read from the data file concurrently. On exit from the routine all processors will have read data to the same point in the file. There is no way of checking that the data file is the same for all processors; if the file is not the same for all processors, the behaviour of the routine could be unpredictable.

The routine reads exactly N matrix elements for each row of A and assumes that the next row of A begins on the next line of the external file.

7 References

None.

8 Example

This example reads in a matrix A and the right-hand side B of a system of equations $AX = B$ from a data file, computes the solution of the system of equations X using routine F04ECFP and prints the solution to standard output. The example uses a 2 by 2 logical processor grid and a block size of 2.

Note: the listing of the Example Program presented below does not give a full pathname for the data file being opened, but in general the user must give the full pathname in this and any other OPEN statement.

8.1 Example Text

```
*     X04BVFP Example Program Text
*     NAG Parallel Library Release 3 Revised. NAG Copyright 1999.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NB
      PARAMETER        (NB=2)
      INTEGER          NMAX, LDA, LDB, NPMAX, MPMAX, NRHMAX
      PARAMETER        (NMAX=8,LDA=NMAX,LDB=NMAX,MPMAX=2,NPMAX=2,
     +                 NRHMAX=2)
*     .. Local Scalars ..
      INTEGER          ICNTXT, IFAIL, N, NCOLS, NRHS, NROWS
      LOGICAL          ROOT
      CHARACTER*80     FORMAT
*     .. Local Arrays ..
      COMPLEX*16       A(LDA,NMAX), B(LDB,NRHMAX), WORK(NMAX)
      INTEGER          IPIV(NMAX+NB)
*     .. External Functions ..
      LOGICAL          Z01ACFP
      EXTERNAL         Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         F04ECFP, X04BVFP, X04BWFP, Z01AAFP, Z01ABFP
*     .. Executable Statements ..
      ROOT = Z01ACFP()
      IF (ROOT) THEN
         WRITE (NOUT,*) 'X04BVFP Example Program Results'
         WRITE (NOUT,*)
      END IF
*
      NROWS =  NPMAX
      NCOLS =  MPMAX
      IFAIL = 0
*
*     Define Library Grid
*
      CALL Z01AAFP(ICNTXT,NROWS,NCOLS,IFAIL)
*
      OPEN (NIN,FILE='x04bvfpe.d')
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, NRHS, FORMAT
*
      IF (N.LE.NMAX .AND. NRHS.LE.NRHMAX) THEN
*
*        Read A from the data file
*
         IFAIL = 0
         CALL X04BVFP(ICNTXT,NIN,N,N,NB,A,LDA,IFAIL)
*
*        Read B from the data file
*
         IFAIL = 0
         CALL X04BVFP(ICNTXT,NIN,N,NRHS,NB,B,LDB,IFAIL)
*
*        Solve the system of equations
*
```

```
         IFAIL = 0
         CALL F04ECFP(ICNTXT,'No transpose',N,NB,A,LDA,NRHS,B,LDB,IPIV,
     +               IFAIL)
*
*        Print the solution
*
         IF (ROOT) THEN
            WRITE (NOUT,*) 'The solution'
            WRITE (NOUT,*)
         ENDIF
         IFAIL = 0
         CALL X04BWFP(ICNTXT,NOUT,N,NRHS,NB,B,LDB,FORMAT,WORK,IFAIL)
      END IF
*
      CLOSE (NIN)
*
      IFAIL = 0
      CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
      STOP
*
      END
```

8.2 Example Data

```
X04BVFP Example Program Data
 4 2 '(2(:,'' ('',F7.4,'','',F7.4,'')''))'          :Values of N, NRHS and FORMAT
(1.80, 1.80)   (2.88, 2.88)   (2.05, 2.05)   (-0.89, -0.89)
(5.25, 5.25)   (-2.95, -2.95) (-0.95, -0.95) (-3.80, -3.80)
(1.58, 1.58)   (-2.69, -2.69) (-2.90, -2.90) (-1.04, -1.04)
(-1.11, -1.11) (-0.66, -0.66) (-0.59, -0.59) (0.80, 0.80) :End of matrix A
(10.1500, 10.1500)   (42.4100, 42.4100)
(-18.7000, -18.7000) (-3.3500, -3.3500)
(-16.6600, -16.6600) (-28.7900, -28.7900)
(-1.0000, -1.0000)   (-13.0400, -13.0400)             :End of matrix B
```

8.3 Example Results

```
 X04BVFP Example Program Results

 The solution

 ( 1.0000, 0.0000) ( 8.0000, 0.0000)
 ( 2.0000, 0.0000) ( 7.0000, 0.0000)
 ( 3.0000, 0.0000) ( 6.0000, 0.0000)
 ( 4.0000, 0.0000) ( 5.0000, 0.0000)
```