F01WVFP

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

F01WVFP distributes an m by n complex matrix B available in its natural form on the $\{0,0\}$ logical processor to the processors on the Library Grid. The distributed form of the matrix, denoted by A, conforms to the cyclic two-dimensional block distribution as required by routines in Chapter F04. On exit, the $(1,1)$ element of the matrix A will be located on the $\{0,0\}$ logical processor.

2 Specification

```
      SUBROUTINE F01WVFP(ICNTXT, M, N, A, LDA, MB, NB, B, LDB, WORK,
     1                   LWORK, IFAIL)
      COMPLEX*16         A(LDA,*), B(LDB,*), WORK(*)
      INTEGER            ICNTXT, M, N, MB, LDA, NB, LDB, LWORK, 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.
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, MB, NB, IFAIL

Global output arguments: IFAIL

The remaining arguments are local.

3.3 Distribution Strategy

On exit, the distributed matrix A is in the cyclic two-dimensional block distribution on the processors on the Library Grid. Each rectangular block is M_b^A by N_b^A an on each local array A on each logical processor contains the relevant blocks of the matrix. This data distribution is described in more detail 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: M — INTEGER *Global Input*

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

Constraint: M ≥ 0.

3: N — INTEGER *Global Input*

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

Constraint: N ≥ 0.

4: 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 relevant blocks of the distributed matrix A in the cyclic two-dimensional block distribution format.

5: LDA — INTEGER *Local Input*

On entry: the size of the first dimension of the array A as declared in the (sub)program from which F01WVFP is called.

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

6: MB — INTEGER *Global Input*

On entry: M_b^A, the blocking factor for distributing the rows of the matrix A.

Constraint: MB ≥ 1.

7: NB — INTEGER *Global Input*

On entry: N_b^A, the blocking factor for distributing the columns of the matrix B.

Constraint: NB ≥ 1.

8: B(LDB,*) — COMPLEX*16 array *Local Input*

Note: the size of the second dimension of the array B must be at least max(1,N) on the logical processor $\{0, 0\}$ and at least 1 elsewhere. B is only referenced on logical processor $\{0, 0\}$.

On entry: the matrix B in its natural form.

9: LDB — INTEGER *Local Input*

On entry: the size of the first dimension of the array B as declared in the (sub)program from which F01WVFP is called.

Constraints:

LDB $\geq$ max(1,M) on logical processor $\{0, 0\}$;
LDB ≥ 1 elsewhere.

10: WORK(∗) — COMPLEX*16 array *Local Workspace/Global Output*

Note: the dimension of the array WORK must be at least max(3,LWORK). WORK is used as a workspace only by the logical processor $\{0, 0\}$.

On exit: WORK(i) = l_i, $i = 1, 2, 3$. See LWORK for the definitions of l_i. If one of the following conditions are satisfied

IFAIL = 0 on exit or
IFAIL = −10 on exit and LWORK ≥ 4 on entry

then l_i, $i = 1, 2, 3$ give the workspace requirements.

11: LWORK — INTEGER *Local Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F01WVFP is called. The minimum requirement for LWORK is max(4,min(l_1,l_2)), but the higher value max(4,l_3) is recommended for higher efficiency where

$$\begin{array}{lcl} l_1 & = & \max\limits_{i=0,\ldots,m_p-1} \alpha(i) \\ \alpha(i) & = & \mathrm{numroc}(\mathrm{M},\mathrm{MB},i,0,m_p) \\ l_2 & = & \max\limits_{j=0,\ldots,n_p-1} \beta(j) \\ \beta(j) & = & \mathrm{numroc}(\mathrm{N},\mathrm{NB},j,0,n_p) \\ l_3 & = & \max\limits_{i=0,\ldots,m_p-1} \; \max\limits_{j=0,\ldots,n_p-1} \alpha(i)\beta(j) \end{array}$$

Note: if LWORK = −1, then a workspace query for LWORK is assumed; the routine only calculates the required minimum sizes of the array WORK as defined by l_1, l_2 and l_3. These values are returned in the real parts of the first three elements of the array WORK.

Constraint: LWORK ≥ max[4,min(l_1,l_2)] or LWORK = −1.

12: 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 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.

6 Further Comments

For distributing matrices from any arbitrary logical processor, the routine F01WUFP should be used instead of F01WVFP. The routine F01WUFP can also position the $(1, 1)$ element of the matrix A on any arbitrary logical processor on the Library Grid.

6.1 Algorithmic Detail

The performance of the algorithm depends upon the size of LWORK. The critical values of LWORK are l_i, $i = 1, 2, 3$. See LWORK for the definitions of l_i. For higher efficiency, LWORK should be set to max(l_3,4) (or greater). However, this routine will work with a workspace size of max(4,min(l_1,l_2)). Note that $l_3 \geq$ max(l_1,l_2).

6.2 Parallelism Detail

The logical processor $\{0, 0\}$ sequentially distributes the relevant parts of B to other processors.

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, Philadelphia, PA 19104-2688, USA. URL: http://www.netlib.org/scalapack/slug/scalapack_slug.html

8 Example

The example program illustrates the distribution of a matrix A.

8.1 Example Text

```
*     F01WVFP Example Program Text
*     NAG Parallel Library Release 3. NAG Copyright 1999.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      INTEGER          M, N
      PARAMETER        (M=10,N=3)
      INTEGER          MB, NB
      PARAMETER        (MB=2,NB=MB)
      INTEGER          NA
      PARAMETER        (NA=25)
      INTEGER          LDA, TDA, LWORK, LDB, LDC
      PARAMETER        (LDA=NA,TDA=NA,LWORK=LDA,LDB=LDA,LDC=LDA)
*     .. Local Scalars ..
      INTEGER          I, I1, I2, I3, ICNTXT, IFAIL, J, MP, NP
      LOGICAL          ROOT
*     .. Local Arrays ..
      COMPLEX*16       A(LDA,TDA), B(LDB,TDA), C(LDC,TDA), WORK(LWORK)
*     .. External Functions ..
      LOGICAL          Z01ACFP
      EXTERNAL         Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         F01WHFP, F01WVFP, Z01AAFP, Z01ABFP
*     .. Intrinsic Functions ..
      INTRINSIC        CMPLX, DBLE, NINT
*     .. Executable Statements ..
      ROOT = Z01ACFP()
```

```
      IF (ROOT) THEN
         WRITE (NOUT,*) 'F01WVFP Example Program Results'
         WRITE (NOUT,*)
      END IF
*
      MP = 2
      NP = 2
      IFAIL = 0
      CALL Z01AAFP(ICNTXT,MP,NP,IFAIL)
*
*     Generate a matrix on the root processor
*
      IF (ROOT) THEN
         DO 20 J = 1, N
            DO 20 I = 1, M
               B(I,J) = CMPLX(DBLE(I),DBLE(J))
   20    CONTINUE
      END IF
*
*     Distribute the matrix from the root processor
*
      IFAIL = 0
      CALL F01WVFP(ICNTXT,M,N,A,LDA,MB,NB,B,LDB,WORK,LWORK,IFAIL)
*
*     Store the values of l(1), l(2) and l(3) in I1, I2 and I3
*
      I1 = NINT(DBLE(WORK(1)))
      I2 = NINT(DBLE(WORK(2)))
      I3 = NINT(DBLE(WORK(3)))
*
*     Gather the matrix back to the root processor as the matrix C, and
*     print the matrix
*
      IFAIL = 0
      CALL F01WHFP(ICNTXT,M,N,A,LDA,MB,NB,C,LDC,WORK,LWORK,IFAIL)
*
      IF (ROOT) THEN
         WRITE (NOUT,'(1X,"The matrix",/)')
         DO 40 I = 1, M
            WRITE (NOUT,'(1X,3( "(",F4.1,1X,",",F4.1,")",3X ))')
     +        (C(I,J),J=1,N)
   40    CONTINUE
         WRITE (NOUT,*)
      END IF
*
*     Print the values l(1), l(2) and l(3) that determine the
*     recommended dimension of WORK
*
      IF (ROOT) THEN
         WRITE (NOUT,
     +     '(1X,"The recommended workspace sizes for LWORK",/)')
         WRITE (NOUT,'(1X,"Real part of WORK(1) = ",I3)') I1
         WRITE (NOUT,'(1X,"Real part of WORK(2) = ",I3)') I2
         WRITE (NOUT,'(1X,"Real part of WORK(3) = ",I3)') I3
      END IF
*
      IFAIL = 0
      CALL Z01ABFP(ICNTXT,'N',IFAIL)
```

```
*
      STOP
      END
```

8.2 Example Data

None.

8.3 Example Results

```
F01WVFP Example Program Results

The matrix

( 1.0 , 1.0)   ( 1.0 , 2.0)   ( 1.0 , 3.0)
( 2.0 , 1.0)   ( 2.0 , 2.0)   ( 2.0 , 3.0)
( 3.0 , 1.0)   ( 3.0 , 2.0)   ( 3.0 , 3.0)
( 4.0 , 1.0)   ( 4.0 , 2.0)   ( 4.0 , 3.0)
( 5.0 , 1.0)   ( 5.0 , 2.0)   ( 5.0 , 3.0)
( 6.0 , 1.0)   ( 6.0 , 2.0)   ( 6.0 , 3.0)
( 7.0 , 1.0)   ( 7.0 , 2.0)   ( 7.0 , 3.0)
( 8.0 , 1.0)   ( 8.0 , 2.0)   ( 8.0 , 3.0)
( 9.0 , 1.0)   ( 9.0 , 2.0)   ( 9.0 , 3.0)
(10.0 , 1.0)   (10.0 , 2.0)   (10.0 , 3.0)

The recommended workspace sizes for LWORK

Real part of WORK(1) =   6
Real part of WORK(2) =   2
Real part of WORK(3) =  12
```