F01WBFP
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

F01WBFP gathers an m by n real distributed matrix A to the $\{0,0\}$ logical processor on a two-dimensional Library Grid and stores it in the natural (non-distributed) format. The gathered matrix is denoted by B.

This routine is useful for gathering full or partial solutions which have been computed using routines in Chapter F04. It is assumed that the matrix A has been distributed on a logical grid of processors in the cyclic two-dimensional block format and that the $(1,1)$ element of the matrix A is positioned on the $\{0,0\}$ logical processor.

2 Specification

```
      SUBROUTINE F01WBFP(ICNTXT, M, N, A, LDA, MB, NB, B, LDB, WORK,
     1                   LWORK, IFAIL)
      DOUBLE PRECISION   A(LDA,*), B(LDB,*), WORK(*)
      INTEGER            ICNTXT, M, N, MB, 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 entry, it is assumed that the matrix A has been partitioned into M_b^A by N_b^A rectangular blocks according to the cyclic two-dimensional block distribution. Each local array A on a 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 A.

Constraint: $0 \le$ M.

3: N — INTEGER *Global Input*

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

Constraint: $0 \le$ N.

4: A(LDA,*) — DOUBLE PRECISION array *Local Input*

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 entry: the relevant parts of the distributed matrix A.

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 F01WBFP is called.

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

6: MB — INTEGER *Global Input*

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

Constraint: MB ≥ 1.

7: NB — INTEGER *Global Input*

On entry: N_b^A, the blocking factor used to distribute the columns of the matrix A.

Constraint: NB ≥ 1.

8: B(LDB,*) — DOUBLE PRECISION array *Local Output*

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

On exit: a copy of the matrix A on logical processor $\{0,0\}$.

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 F01WBFP is called.

Constraints:

LDB $\ge$ max(1,m) on the processor $\{0,0\}$;
LDB ≥ 1 elsewhere.

10: WORK(∗) — DOUBLE PRECISION 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 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 F01WBFP 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

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

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 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.

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.

6 Further Comments

For gathering matrices to an arbitrary logical processor, the routine F01WAFP should be used instead of F01WBFP. The routine F01WAFP can handle matrices with the $(1,1)$ element located 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 $l_3 >$ max(l_1,l_2).

6.2 Parallelism Detail

The logical processor $\{0, 0\}$ sequentially gathers the relevant parts of A from 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, Philadelpia, PA 19104-2688, USA. URL: http://www.netlib.org/scalapack/slug/scalapack_slug.html

8 Example

The example program illustrates the gathering of a matrix A. This matrix is generated using the routine F01ZSFP.

8.1 Example Text

```
*     F01WBFP Example Program Text
*     NAG Parallel Library Release 3. NAG Copyright 1999.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      INTEGER          M, N
      PARAMETER        (M=10,N=5)
      INTEGER          MB, NB
      PARAMETER        (MB=3,NB=MB)
      INTEGER          NA
      PARAMETER        (NA=25)
      INTEGER          LDA, TDA, LWORK, LDB
      PARAMETER        (LDA=NA,TDA=NA,LWORK=LDA,LDB=LDA)
*     .. Local Scalars ..
      INTEGER          I, ICNTXT, IFAIL, J, MP, NP
      LOGICAL          ROOT
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,TDA), B(LDA,TDA), WORK(LWORK)
*     .. External Functions ..
      LOGICAL          Z01ACFP
```

```
      EXTERNAL         Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         F01WBFP, F01ZSFP, GMATA, Z01AAFP, Z01ABFP
*     .. Intrinsic Functions ..
      INTRINSIC        NINT
*     .. Executable Statements ..
      ROOT = Z01ACFP()
      IF (ROOT) THEN
         WRITE (NOUT,*) 'F01WBFP Example Program Results'
         WRITE (NOUT,*)
      END IF
*
      MP = 2
      NP = 2
      IFAIL = 0
      CALL Z01AAFP(ICNTXT,MP,NP,IFAIL)
*
*     Set up and generate an m by n matrix
*
      CALL F01ZSFP(ICNTXT,GMATA,M,N,NB,A,LDA,IFAIL)
*
*     Gather the matrix to processor (0,0) and print the matrix on the
*     root processor
*
      IFAIL = 0
      CALL F01WBFP(ICNTXT,M,N,A,LDA,MB,NB,B,LDB,WORK,LWORK,IFAIL)
      IF (ROOT) THEN
         WRITE (NOUT,'(1X,"The matrix",/)')
         DO 20 I = 1, M
            WRITE (NOUT,'(1X,5(F8.4,2X))') (B(I,J),J=1,N)
   20    CONTINUE
         WRITE (NOUT,*)
      END IF
*
*     Print the values l(1), l(2) and l(3) that determine the
*     recommended dimension of WORK(3)
*
      IF (ROOT) THEN
         WRITE (NOUT,
     +     '(1X,"The values of l(1), l(2) and l(3) are:",/)')
         WRITE (NOUT,'(1X,"WORK(1) = ",I3)') NINT(WORK(1))
         WRITE (NOUT,'(1X,"WORK(2) = ",I3)') NINT(WORK(2))
         WRITE (NOUT,'(1X,"WORK(3) = ",I3)') NINT(WORK(3))
      END IF

      IFAIL = 0
      CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
      STOP
      END
*
      SUBROUTINE GMATA(I1,I2,J1,J2,AL,LDAL)
*     GMATA generates the block A(I1: I2, J1: J2) of the matrix A such
*     that
*
*        a(i,j) = dble(i) + dble(j)/10000.0
*
*     in the array AL.
```

```
*
*     .. Scalar Arguments ..
      INTEGER          I1, I2, J1, J2, LDAL
*     .. Array Arguments ..
      DOUBLE PRECISION AL(LDAL,*)
*     .. Local Scalars ..
      INTEGER          I, J, K, L
*     .. Intrinsic Functions ..
      INTRINSIC        DBLE
*     .. Executable Statements ..
      L = 1
      DO 40 J = J1, J2
         K = 1
         DO 20 I = I1, I2
            AL(K,L) = DBLE(I) + DBLE(J)/10000.0D0
            K = K + 1
   20    CONTINUE
         L = L + 1
   40 CONTINUE
      RETURN
      END
```

8.2 Example Data

None.

8.3 Example Results

```
F01WBFP Example Program Results

The matrix

  1.0001    1.0002    1.0003    1.0004    1.0005
  2.0001    2.0002    2.0003    2.0004    2.0005
  3.0001    3.0002    3.0003    3.0004    3.0005
  4.0001    4.0002    4.0003    4.0004    4.0005
  5.0001    5.0002    5.0003    5.0004    5.0005
  6.0001    6.0002    6.0003    6.0004    6.0005
  7.0001    7.0002    7.0003    7.0004    7.0005
  8.0001    8.0002    8.0003    8.0004    8.0005
  9.0001    9.0002    9.0003    9.0004    9.0005
 10.0001   10.0002   10.0003   10.0004   10.0005

The values of l(1), l(2) and l(3) are:

WORK(1) =   6
WORK(2) =   3
WORK(3) =  18
```