F01ZPFP

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

F01ZPFP gathers a distributed real vector y of length n to each logical processor on a 2-d logical processor grid. On entry, it is assumed that an identical copy of the vector is distributed in cyclic 1-d block form across logical processors on each row of the grid.

The distributed vector is aligned with columns of a distributed matrix A and the parameters of this distribution are contained in a descriptor array IDESCA corresponding to A. This routine is primarily designed to be used following a call to F08FEFP, which reduces a real symmetric matrix A_s to tridiagonal form T, storing the diagonal and off-diagonal elements of T as distributed vectors d and e.

See the F08 Chapter Introduction for details of the matrix A and the associated principal submatrix A_s. The matrices A and A_s themselves are not arguments to the routine F01ZPFP. The description array IDESCA (typically, as previously used by F08FEFP) contains the necessary information to gather the distributed vector y, and hence it is an input argument to the routine F01ZPFP.

2 Specification

```
      SUBROUTINE F01ZPFP(N, IA, JA, IDESCA, BYCOL, BYALL, WORK, LWORK,
     1                   IFAIL)
      DOUBLE PRECISION   BYCOL(*), BYALL(*), WORK(LWORK)
      INTEGER            N, IA, JA, IDESCA(9), LWORK, IFAIL
```

3 Data Distribution

3.1 Definitions

The following definitions are used in describing the data distribution within this document:

m_p	–	the number of rows in the logical processor grid.
n_p	–	the number of columns in the logical processor grid.
p_r	–	the row grid coordinate of the calling processor.
p_c	–	the column grid coordinate of the calling processor.
M_b^A	–	the blocking factor for the distribution of the rows of a matrix A.
N_b^A	–	the blocking factor for the distribution of the columns of a matrix A.
s_r^A	–	the row coordinate of the processor that possesses the first row of a distributed matrix A.
s_c^A	–	the column coordinate of the processor that possesses the first column of a distributed matrix A.
numroc$(\hat{\ell}, L_b^A, p, s^A, \ell_p)$	–	a function which gives the **num**ber of **r**ows **o**r **c**olumns of a distributed matrix A owned by the processor with the row or column coordinate p (p_r or p_c), where $\hat{\ell}$ is the total number of rows or columns of the matrix, L_b^A is the blocking factor used (M_b^A or N_b^A), s^A is the row or column coordinate (s_r^A or s_c^A) of the processor that possesses the first row or column of the distributed matrix and ℓ_p is either m_p or n_p. The Library provides the utility function Z01CAFP (NUMROC) for the evaluation of numroc.

3.2 Global and Local Arguments

The input arguments N, IA, JA and the array elements IDESCA(1) and IDESCA(3),...,IDESCA(8) and IFAIL are all global are all global and so should have the same values on entry to the routine on each logical processor. The output arguments BYALL and IFAIL are global, they will have the same value on exit from the routine on each logical processor. The remaining arguments are local.

3.3 Distribution Strategy

The distributed matrices A and A_s must satisfy the following requirements:

$M_b^A = N_b^A > 0$;
$i_A = j_A = 1$;
$s_r^A = s_c^A = 0$.

Any further constraints, as well as the above, are stated in Section 4 under each argument.

4 Arguments

1: N — INTEGER *Global Input*

On entry: n, the length of the vector y.

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

2: IA — INTEGER *Global Input*

On entry: the row index of A, i_A, that identifies the first row of the submatrix A_s. Typically, this should be identical to the value of IA used in the previous call to F08FEFP.

Constraint: $\mathrm{IA} = 1$.

3: JA — INTEGER *Global Input*

On entry: the column index of A, j_A, that identifies the first column of the submatrix A_s. Typically, this should be identical to the value of JA used in the previous call to F08FEFP.

Constraint: $\mathrm{JA} = 1$.

4: IDESCA(9) — INTEGER array *Local Input*

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 2-d block distribution, IDESCA(1) = 1.
IDESCA(2), the BLACS context (ICNTXT) for the processor grid, usually returned by 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 2-d array A.

Constraints:

IDESCA(1) = 1;
IDESCA(3) ≥ 0; IDESCA(4) ≥ 0;
IDESCA(5) ≥ 1; IDESCA(6) ≥ 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)).

5: BYCOL(∗) — DOUBLE PRECISION array *Local Input*

Note: the dimension of the array BYCOL must be at least numroc(IA−1+N,IDESCA(5),p_r,IDESCA(7),m_p).

On entry: the local parts of the vector y which has been distributed across each logical processor row.

6: BYALL(∗) — DOUBLE PRECISION array *Global Output*

Note: the dimension of the array BYALL must be at least N.

On exit: the vector y.

7: WORK(LWORK) — DOUBLE PRECISION array *Local Workspace/Local Output*

On exit: WORK(1) contains the minimum value required for LWORK.

8: LWORK — INTEGER *Local Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F01ZPFP is called.

Constraint: LWORK $\geq$ numroc(N, IDESCA(6), 0, 0, n_p).

9: IFAIL — INTEGER *Global Input/Global Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (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 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).

Errors detected by the routine:

IFAIL = −2000

The routine has been called with an invalid value of ICNTXT (stored in IDESCA(2)) 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 and if this argument was invalid then, INFO $= -k$;
if the kth argument is an array and its jth element was invalid then, INFO $= -(100 \times k + j)$.

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

6.1 Algorithmic Detail

This routine is based on the ScaLAPACK routine PDLARED1D.

6.2 Parallelism Detail

None.

6.3 Accuracy

Not applicable.

7 References

None.

8 Example

See the example for the routine F08FEFP.