F01CPFP

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

F01CPFP finds the element-wise absolute maximum or minimum of a group of m by n integer matrices over the Library Grid. If the matrix on the logical processor $\{p_r, p_c\}$ is denoted by $A^{\{r,c\}}$ and the (i,j)th element by $a_{i,j}^{\{r,c\}}$ then any of the following maximum or minimum operations are permitted:

$$w_{i,j} = \mathrm{op}_{r,c} \quad | \, a_{i,j}^{\{r,c\}} \, |, \qquad \text{for } r = 0,1,\ldots,m_p-1; \; c = 0,1,\ldots,n_p-1 \quad \text{(operating over all processors)}$$

$$w_{i,j}^{(r)} = \mathrm{op}_{c} \quad | \, a_{i,j}^{\{r,c\}} \, |, \qquad \text{for } r = 0,1,\ldots,m_p-1 \quad \text{(operating over rows of processors)}$$

$$w_{i,j}^{(c)} = \mathrm{op}_{r} \quad | \, a_{i,j}^{\{r,c\}} \, |, \qquad \text{for } c = 0,1,\ldots,n_p-1 \quad \text{(operating over columns of processors)}$$

with the following matrix notations:

$$W = [\mathrm{sign}\{a_{i,j}^{\{\alpha,\beta\}}\} \; w_{i,j}], \quad W_r = [\mathrm{sign}\{a_{i,j}^{\{\alpha,\beta\}}\} \; w_{i,j}^{(r)}] \quad \text{and} \quad W_c = [\mathrm{sign}\{a_{i,j}^{\{\alpha,\beta\}}\} \; w_{i,j}^{(c)}],$$

where 'op' denotes either 'max' or 'min' and $\{\alpha, \beta\}$ denotes the processor coordinates in which the maximum or the minimum of a particular element is found (the values of α and β depend on r and c respectively).

F01CPFP uses absolute value when it performs the max or min operation, however, the returned value is the signed value not the absolute value. For example, if F01CPFP performs an absolute value maximum on the numbers -5, 3, 1, -8, the result will be -8.

The form of the operation computed, and the choice of processors to which the results are returned, are controlled by the values passed to three of the global arguments.

2 Specification

```
      SUBROUTINE F01CPFP(ICNTXT, MXORMN, SCOPE, M, N, A, LDA, IR, IC,
     1                   LDIRC, IREF, IRDST, ICDST, IFAIL)
      INTEGER            ICNTXT, M, N, A(LDA,N), LDA, IR(LDIRC,N),
     1                   IC(LDIRC,N), LDIRC, IREF, IRDST, ICDST, IFAIL
      CHARACTER*1        MXORMN, SCOPE
```

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.

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: MXORMN, SCOPE, M, N, IREF, IRDST, ICDST, IFAIL

Global output arguments: IFAIL

The remaining arguments are local.

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: MXORMN — CHARACTER*1 *Global Input*

On entry: the type of operation to be performed:

if MXORMN = 'X', then the maximun operation is performed;
if MXORMN = 'N', then the minimun operation is performed.

Constraint: MXORMN = 'X' or 'N'.

3: SCOPE — CHARACTER*1 *Global Input*

On entry: specifies the scope of the operations to be computed:

if SCOPE = 'A', the element-wise sum of the matrices $A^{\{r,c\}}$ is computed;
if SCOPE = 'R', the element-wise row sum of the matrices $A^{\{r,c\}}$ is computed;
if SCOPE = 'C', the element-wise column sum of the matrices $A^{\{r,c\}}$ is computed.

Constraint: SCOPE = 'R', 'C' or 'A'.

4: M — INTEGER *Global Input*

On entry: m, the number of rows in the matrices $A^{\{r,c\}}$.

Constraint: M $\geq$ 1.

5: N — INTEGER *Global Input*

On entry: n, the number of columns in the matrices $A^{\{r,c\}}$.

Constraint: N $\geq$ 1.

6: A(LDA,N) — INTEGER array *Local Input/Local Output*

On entry: the matrix $A^{\{r,c\}}$ whose maximum or minimum elements to be found.

On exit: the array containing the matrix W, W_r or W_c (depending on the value of SCOPE). The processors on which W, W_r or W_c are stored depends on the arguments SCOPE, IRDST and ICDST (see below).

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 F01CPFP is called.

Constraint: LDA $\geq$ M.

8: IR(LDIRC,N) — INTEGER array *Local Output*
9: IC(LDIRC,N) — INTEGER array *Local Output*

Note: if IREF = -1, then the arrays IR and IC are not referenced.

On exit: IR(i,j) and IC(i,j) contain the row and the column coordinates respectively of the processor, $\{\alpha,\beta\}$, on which the maximum or the minimum element of $a_{i,j}\{r,c\}$ was located. If the calling processor is not selected to receive W, W_r or W_c, then these arrays will contain meaningless values.

10: LDIRC — INTEGER *Local Input*

On entry: the first dimension of the arrays IR and IC as declared in the (sub)program from which F01CPFP is called.

Constraint: LDIRC $\geq$ M.

11: IREF — INTEGER *Global Input*

On entry: indicates whether the row and the column coordinates of the processor, $\{\alpha, \beta\}$, on which the maximum or the minimum element of $a_{i,j}^{\{r,c\}}$ was located, is to be determined.

If IREF $= -1$, then $\{\alpha, \beta\}$ are not determined. Otherwise, these coordinates are returned in the arrays IR and IC.

12: IRDST — INTEGER *Global Input*

13: ICDST — INTEGER *Global Input*

On entry: indicates the processor row or column coordinates which receive the maximum or the minimum elements of $A^{\{r,c\}}$.

If SCOPE = 'A', then

if IRDST $= -1$ or ICDST $= -1$, then all processors store s on exit;

if $0 \leq$ IRDST $\leq m_p-1$ and $0 \leq$ ICDST $\leq n_p-1$, then processor {IRDST, ICDST} will store $w_{i,j}$ on exit.

If SCOPE = 'R', then

if $0 \leq$ ICDST $\leq np-1$, then for each $r = 0, 1, \ldots, m_p - 1$ processor $\{p_r$, ICDST$\}$ will store $w_{i,j}^{(r)}$ on exit;

if ICDST $= -1$, then for each $r = 0, 1, \ldots, m_p-1$ processors $\{p_r, p_c\}$ for $c = 0, 1, \ldots, n_p-1$ will store $w_{i,j}^{(r)}$ on exit.

If SCOPE = 'C', then

if $0 \leq$ IRDST $\leq m_p-1$, then for each $c = 0, 1, \ldots, n_p-1$ processor {IRDST, p_c} will store $w_{i,j}^{(c)}$ on exit;

if IRDST $= -1$, then for each $c = 0, 1, \ldots, n_p-1$ processors $\{p_r, p_c\}$ for $r = 0, 1, \ldots, m_p-1$ store $w_{i,j}^{(c)}$ on exit.

Constraints:

$-1 \leq$ IRDST $\leq m_p - 1$;

$-1 \leq$ ICDST $\leq n_p - 1$.

14: 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 had an invalid value. This error occurred either because a global argument did not have the same value on all the logical processors (see Section 3.2), or because its value was incorrect. An explanatory message distinguishes between these two cases.

6 Further Comments

This routine is based on the BLACS routines IGAMX2D and IGAMN2D (see Blackford, *et al.* [1]).

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

A number of example programs in Chapter F11 utilize F01CPFP. See for instance Section 8 of the document for F11BBFP.