F11XPFP

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.

Note: you should read the the F11 Chapter Introduction before using this routine.

1 Description

F11XPFP computes either the matrix-vector product $y = Ax$, the transposed matrix-vector product $y = A^T x$ or the conjugate transposed matrix-vector product $y = A^H x$, where A is an n by n complex sparse matrix represented in coordinate storage format, distributed in cyclic row block form. The routine F11ZPFP must be called prior to F11XPFP to set up auxiliary information in the array IAINFO.

2 Specification

```
 SUBROUTINE F11XPFP(ICNTXT, TRANS, N, NNZ, A, IROW, ICOL, CHECK, X,
1                   Y, IAINFO, WORK, IFAIL)
 INTEGER            ICNTXT, N, NNZ, IROW(*), ICOL(*), IAINFO(*),
1                   IFAIL
 COMPLEX*16         A(*), X(*), Y(*), WORK(*)
 CHARACTER*1        TRANS, CHECK
```

3 Usage

3.1 Definitions

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

M_b	–	the blocking factor used in the cyclic row block distribution.
m_l	–	the number of rows of the matrix assigned to the calling processor (m_l = IAINFO(3), see IAINFO).
n_{int}^i	–	the number of internal interface indices (see Section 2.6.1 of the F11 Chapter Introduction) for the calling processor (n_{int}^i = IAINFO(6), see IAINFO).
n_{int}^e	–	the number of external interface indices (see Section 2.6.1 of the F11 Chapter Introduction) for the calling processor (n_{int}^e = IAINFO(7), see IAINFO).

3.2 Global and Local Arguments

Global input arguments: TRANS, N, CHECK, IFAIL

Global output arguments: IFAIL

The remaining arguments are local.

3.3 Distribution Strategy

The matrix A must be distributed in cyclic row block form.

When A is distributed in cyclic row block form, blocks of M_b contiguous rows of the matrix A are stored in coordinate storage format on the Library Grid cyclically row by row (i.e., in the row major ordering of the grid) starting from the {0,0} logical processor.

The vectors x and y are distributed conformally to the sparse matrix A, i.e., x must be distributed across the Library Grid in the same way as each of the columns of the matrix A.

These data distributions are described in more detail in Section 2.5 of the F11 Chapter Introduction.

This routine assumes that the data has already been correctly distributed, and if this is not the case will fail to produce correct results.

3.4 Related Routines

Some Library routines can be used to generate or distribute complex sparse matrices in cyclic row block form, or to generate or distribute complex vectors conformally to a given sparse matrix.

Complex sparse matrix generation:	F01YPFP or F01YQFP
Complex sparse matrix distribution:	F01XPFP
Complex vector generation:	F01YTFP
Complex vector scatter:	F01XTFP

3.5 Requisites

The complex sparse matrix A must have been preprocessed to set up the auxiliary information vector IAINFO by F11ZPFP.

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: TRANS — CHARACTER*1 *Global Input*

On entry: specifies which matrix vector product is computed:

if TRANS = 'N', then $y = Ax$ is computed;
if TRANS = 'T', then $y = A^T x$ is computed;
if TRANS = 'C', then $y = A^H x$ is computed.

Constraint: TRANS = 'N', 'T' or 'C'.

3: N — INTEGER *Global Input*

On entry: n, the order of the matrix A. It must contain the same value as the parameter N used in a prior call to F11ZPFP in which the array IAINFO was initialised.

Constraint: N ≥ 1.

4: NNZ — INTEGER *Local Input*

On entry: the number of non-zero entries in the matrix A stored on the calling processor. It must contain the same value as the parameter NNZ returned from a prior call to F11ZPFP in which the array IAINFO was initialised.

Constraint: NNZ > 0.

5: A($*$) — COMPLEX*16 array *Local Input*

Note: the dimension of the array A must be at least max(1,NNZ).

On entry: the non-zero entries in the blocks of the matrix A assigned to the calling processor. The local non-zero entries must have been reordered by a prior call to F11YNFP or F11ZPFP.

6: IROW($*$) — INTEGER array *Local Input*

7: ICOL($*$) — INTEGER array *Local Input*

Note: the dimension of the arrays IROW and ICOL must be at least max(1,NNZ).

On entry: the local row and column indices of the non-zero entries supplied in the array A. The contents of the arrays IROW and ICOL **must not** be changed between successive calls to library routines involving the matrix A.

8: CHECK — CHARACTER*1 *Global Input*

Note: when the error checking mode was set to reduced error checking by a prior call to Z02EAFP with LEVEL = ±1, then CHECK is not referenced.

On entry: specifies whether or not the validity of the arguments passed to F11XPFP should be checked:

if CHECK = 'C', checks are carried on all arguments of F11XPFP;
if CHECK = 'N', none of these checks are carried out.

See also Section 6.2.

Constraint: CHECK = 'C' or 'N'.

9: X(∗) — COMPLEX*16 array *Local Input*

Note: the dimension of the array X must be at least max(1,m_l).

On entry: the local part of the vector x.

10: Y(∗) — COMPLEX*16 array *Local Output*

Note: the dimension of the array Y must be at least max(1,m_l).

On exit: the local part of the vector y.

11: IAINFO(∗) — INTEGER array *Local Input*

Note: the dimension of the array IAINFO must be at least max(200,IAINFO(2)).

On entry: the first IAINFO(2) elements of IAINFO contain auxiliary information about the matrix A. The array IAINFO must be initialised by a prior call to F11ZPFP. The first IAINFO(2) elements of IAINFO must not be changed between successive calls to library routines involving the matrix A. See Section 3.3 of the F11 Chapter Introduction.

Note: on exit from F11ZPFP the element IAINFO(3) contains m_l, the number of rows of the matrix assigned to the calling processor, and the elements IAINFO(6) and IAINFO(7) contain n_{int}^i and n_{int}^e, the number of internal and external interface indices (see Section 2.6.1 of the F11 Chapter Introduction) for the calling processor, respectively.

12: WORK(∗) — COMPLEX*16 array *Workspace*

Note: the dimension of the array WORK must be at least max(1,n_{int}^i, n_{int}^e).

13: 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.

IFAIL = 1

IAINFO was not set up by a prior call to F11ZPFP.

IFAIL = 2

On entry, the data stored in the arguments N, NNZ, IROW, ICOL and IAINFO is inconsistent. This indicates that, after the array IAINFO was set up by a call to F11ZPFP at least one of these arguments was changed between successive calls to library routines.

6 Further Comments

6.1 Computational Costs

The number of arithmetic operations performed by F11XPFP on each logical processor is approximately 2*NNZ. The number of communication operations depends on the sparsity pattern of the matrix A and the particular row block distribution used.

6.2 Use of CHECK

It is expected that in most applications F11XPFP will be called many times with the same matrix A. In the interests of both reliability and efficiency you are recommended to set CHECK to 'C' for the first of such calls, and to 'N' for all subsequent calls.

7 References

[1] Saad Y (1996) *Iterative Methods for Sparse Linear Systems* PWS Publishing Company, Boston, MA

8 Example

See Section 8 of the document for F11BRFP.