F11DBFP

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

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

F11DBFP solves a system of linear equations

$$Mx = y, \quad \text{or} \quad M^T x = y,$$

for an n by n real sparse block diagonal matrix M, represented in coordinate storage format, distributed on a logical grid of processors in cyclic row block form. On each logical processor, the local diagonal blocks $M_k = P_k L_k D_k U_k Q_k$, $k = 1, 2, \ldots, n_{\mathrm{LB}}$, of M correspond to incomplete LU decompositions of the local diagonal blocks A_k, $k = 1, 2, \ldots, n_{\mathrm{LB}}$, of a general real sparse matrix A (see Sections 2.6 and 2.7 of the F11 Chapter Introduction), distributed identically to M, as generated by F11DAFP.

In each of the above n_{LB} decompositions, L_k is a lower triangular sparse matrix with unit diagonal elements, D_k is a diagonal matrix, U_k is an upper triangular sparse matrix with unit diagonal elements and P_k and Q_k are permutation matrices. L_k, D_k and U_k are supplied to F11DBFP through the matrix

$$C_k = L_k + D_k^{-1} + U_k - 2I_k,$$

stored in coordinate storage format, as returned by F11DAFP. The permutation matrices P_k and Q_k are returned from F11DAFP via the arrays IPIVP and IPIVQ. I_k denotes the identity matrix.

It is envisaged that a common use of F11DBFP will be to carry out the preconditioning step required in the application of F11BBFP to sparse linear systems. F11DBFP is used for this purpose by the black-box routine F11DCFP.

2 Specification

```
      SUBROUTINE F11DBFP(ICNTXT, TRANS, N, NNZC, C, IROWC, ICOLC, IPIVP,
     1                   IPIVQ, CHECK, Y, X, IAINFO, WORK, IFAIL)
      INTEGER            ICNTXT, N, NNZC, IROWC(*), ICOLC(*), IPIVP(*),
     1                   IPIVQ(*), IAINFO(*), IFAIL
      DOUBLE PRECISION   C(*), Y(*), X(*), WORK(*)
      CHARACTER*1        TRANS, CHECK
```

3 Data Distribution

3.1 Definitions

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

- M_b – the blocking factor for the distribution of the rows of the matrix.
- m_l – the number of rows of the matrix assigned to the calling processor (= IAINFO(3), see IAINFO).
- n_{LB} – the number of row blocks assigned to the calling processor (= IAINFO(8), see IAINFO).

3.2 Global and Local Arguments

The input arguments TRANS, N, CHECK and IFAIL are global and so must have the same value on entry to the routine on each processor. The output argument IFAIL is global and so will have the same value on exit from the routine on each processor. The remaining arguments are local.

3.3 Distribution Strategy

Blocks of M_b contiguous rows of the matrix A are stored in coordinate storage format on a logical grid of processors 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 matrix A, i.e., x and y are distributed across the logical processor grid in the same way as each of the columns of the matrix A. The pivot vectors IPIVP and IPIVQ are distributed in the same way.

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. However, the Library provides utility routines, F01YAFP and F01YEFP, which assists you in distributing the data correctly. Descriptions of these routines can be found in Chapter F01 of the NAG Parallel Library

4 Arguments

1: ICNTXT — INTEGER *Local Input*

On entry: the BLACS context used by the communication mechanism, usually returned by a call to Z01AAFP.

2: TRANS — CHARACTER*1 *Global Input*

On entry: specifies whether or not the matrix M is transposed:

if TRANS = 'N', then $Mx = y$ is solved;
if TRANS = 'T', then $M^T x = y$ is solved.

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

3: N — INTEGER *Global Input*

On entry: the order of the matrix M, n. It must contain the same value as the parameter N used in a prior call of F11ZAFP in which the array IAINFO was initialised.

Constraint: N ≥ 1.

4: NNZC — INTEGER *Local Input*

On entry: the number of non-zero elements in the matrices C_k, $k = 1, 2, \ldots, n_{\mathrm{LB}}$, assigned to the calling processor as returned by a prior call of F11DAFP.

Constraint: NNZC ≥ 0.

5: C($*$) — DOUBLE PRECISION array *Local Input*

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

On entry: the values returned in the array C by a prior call to F11DAFP.

6: IROWC($*$) — INTEGER array *Local Input*
7: ICOLC($*$) — INTEGER array *Local Input*

Note: the dimension of the arrays IROWC and ICOLC must be at least max(1,NNZC).

On entry: the local row and column indices of the non-zero elements supplied in C as returned by a prior call to F11DAFP.

8: IPIVP($*$) — INTEGER array *Local Input*
9: IPIVQ($*$) — INTEGER array *Local Input*

Note: the dimension of the arrays IPIVP and IPIVQ must be at least $\max(1, m_l)$.

On entry: the local row and column indices of pivot elements as returned by a prior call to F11DAFP.

10: CHECK — CHARACTER*1 *Global Input*

On entry: specifies whether or not the validity of the arguments passed to F11DBFP should be checked:

if CHECK = 'C', checks are carried on all arguments of F11DBFP;
if CHECK = 'N', none of these checks are carried out.

See also Section 6.2.

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

11: Y(∗) — DOUBLE PRECISION array *Local Input*

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

On entry: the local part of the vector y.

12: X(∗) — DOUBLE PRECISION array *Local Output*

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

On exit: the local part of the vector x.

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

Note: the dimension of the array IAINFO must be at least max(2,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 of F11ZAFP, and additional information must be stored in IAINFO by a prior call of F11DAFP. The first IAINFO(2) elements of IAINFO must not be changed between successive calls to library routines involving the matrix A.
Note: On exit from F11ZAFP, the elements IAINFO(3) and IAINFO(8) contain m_l and n_{LB}, the number of rows and row blocks of the matrix assigned to the calling processor, respectively.

14: WORK(∗) — DOUBLE PRECISION array *Workspace*

Note: in this Release WORK is not referenced.

15: 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 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. For global arguments, this may also be caused by an argument not having the same value on all logical processors. An explanatory message distinguishes between these two cases.

IFAIL = 1

IAINFO was not set up by prior calls of F11ZAFP and F11DAFP.

IFAIL = 2

On entry, the data stored in the arguments N, NNZC, IROWC, ICOLC, IPIVP, IPIVQ and IAINFO is inconsistent. This indicates that, after the array IAINFO was set up by calls of F11ZAFP and F11DAFP, at least one of these arguments was changed between successive calls to library routines.

6 Further Comments

6.1 Parallelism Detail

The routine performs all operations on each logical processor independently.

6.2 Use of CHECK

It is expected that a common use of F11DBFP will be to compute the preconditioning step required in the application of F11BBFP to sparse linear systems. In this situation F11DBFP is likely to 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.

6.3 Computational costs

The time taken for a call to F11DBFP on each logical processor is approximately proportional to the value of NNZC.

7 References

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

8 Example

See the Example Program for F11BAFP.