

# F11DGFP

## 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

F11DGFP applies the block Jacobi preconditioner for a real sparse matrix (see Section 2.7 of the F11 Chapter Introduction).

On each processor, F11DGFP solves the systems of linear equations

$$M_k x_k = y_k, \quad \text{or} \quad (M_k)^T x_k = y_k,$$

for the diagonal blocks  $M_k$ ,  $k = 1, 2, \dots, n_{\text{LB}}$ , of an  $n$  by  $n$  real sparse block diagonal matrix  $M$ .

$M$  must be stored in coordinate storage (CS) format distributed in cyclic row block form (see Sections 2.4.1 and 2.5 of the F11 Chapter Introduction).

A call to F11DGFP must always be preceded by a call to F11ZBFP to set up auxiliary information about the matrix  $A$  in the array IAINFO, and by a call to F11DFFP, to generate the preconditioner.

It is recommended that the user should read Section 6, before proceeding to use this routine for the first time.

### 2 Specification

```

SUBROUTINE F11DGFP(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 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 ( $m_l = \text{IAINFO}(3)$ , see IAINFO).
- $m_l^o$  – the overall number of rows (and columns) in the diagonal blocks  $A_k$ ,  $k = 1, 2, \dots, n_{\text{LB}}$ , assigned to the calling processor ( $m_l^o = \text{IAINFO}(4)$ , see IAINFO).
- $n_{\text{LB}}$  – the number of row blocks assigned to the calling processor ( $n_{\text{LB}} = \text{IAINFO}(8)$ , 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  $M$  must be distributed in cyclic row block form.

When  $M$  is distributed in cyclic row block form, blocks of  $M_b$  contiguous rows of the matrix  $M$  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 distribution of  $M$  is identical to the distribution of  $A$ , and  $M$  is automatically generated in the appropriate distributed form by F11DFFP.

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

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

A number of Library routines can be used to generate or distribute real sparse matrices in cyclic row block form:

Real sparse matrix generation: F01YAFP or F01YBFP

Real sparse matrix distribution: F01XAFP

Other Library routines can use block Jacobi preconditioners:

Black Box routines: F11DHFP for the solution of real general systems, F11JHFP for the solution of real symmetric systems

Basic routines: the suite comprising F11BAFP, F11BBFP and F11BCFP, for real general systems; the suite comprising F11GAFP, F11GBFP and F11GCFP, for real symmetric systems.

### 3.5 Requisites

The sparse matrix  $A$  must have been preprocessed to set up the auxiliary information vector IAINFO by F11ZBFP. The preconditioner must have been generated by calling F11DFFP.

## 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 whether or not the matrices  $M_k$ ,  $k = 1, 2, \dots, n_{LB}$ , are transposed:

if TRANS = 'N', then  $M_k x_k = y_k$  is solved;  
if TRANS = 'T', then  $(M_k)^T x_k = y_k$  is solved.

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

3: N — INTEGER *Global Input*

*On entry:*  $n$ , the order of the matrix  $M$ . It must contain the same value as the parameter N used in a prior call to F11ZBFP in which the array IAINFO was initialised.

*Constraint:*  $N \geq 1$ .

- 4:** NNZC — INTEGER *Local Input*  
*On entry:* the number of non-zero elements in the matrices  $C_k$ ,  $k = 1, 2, \dots, n_{LB}$ , assigned to the calling processor as returned by a prior call to F11DFFP (see Section 6).  
*Constraint:*  $NNZC \geq 0$ .
- 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 F11DFFP.
- 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 F11DFFP.
- 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  $\max(1, m_l^o)$ .  
*On entry:* the local row and column indices of pivot elements as returned by a prior call to F11DFFP.
- 10:** 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 =  $\pm 1$ , then CHECK is not referenced.  
*On entry:* specifies whether or not the validity of the arguments passed to F11DGFP should be checked:  
     if CHECK = 'C', checks are carried out on all arguments of F11DGFP;  
     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  $\max(200, IAINFO(2))$ .  
*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 F11ZBFP and additional information must be stored in IAINFO by a prior call to F11DFFP. The first IAINFO(2) elements of IAINFO must not be changed between successive calls to library routines involving the matrix  $A$ .
- 14:** WORK(\*) — DOUBLE PRECISION array *Workspace*  
**Note:** the dimension of the array WORK must be at least  $\max(1, 2 \times m_l^o)$ .

**15: 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 *i*th argument was invalid. This error occurred 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 prior calls to F11ZBFP and F11DFFP.

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 to F11ZBFP and F11DFFP, at least one of these arguments was changed before calling F11DGFPP.

## 6 Further Comments

### 6.1 Algorithmic Details

The diagonal blocks  $M_k$  are approximated by the incomplete  $LU$  (ILU) factorisations, generated by F11DFFP:

$$M_k = P_k L_k D_k U_k Q_k.$$

In the above equation,  $L_k$  and  $U_k$  are lower triangular and upper triangular respectively, both with unit diagonals, and  $D_k$  is a diagonal matrix;  $P_k$  and  $Q_k$  are permutation matrices.

$L_k$ ,  $U_k$  and  $D_k$  are supplied to F11DGFPP in the form of the matrix

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

where  $I_k$  is the identity matrix of appropriate order, stored in coordinate storage format. The matrices  $C_k$ ,  $P_k$  and  $Q_k$  are all generated by F11DFFP.

Each diagonal system  $M_k x_k = y_k$  is solved independently. The solution vector  $x$  of the overall preconditioning equation  $Mx = y$ , where  $M$  is the union of all the diagonal blocks, is then given by concatenating the partial solutions  $x_k$ .

## 6.2 Use of CHECK

When F11DGFP is used as a preconditioner, it is likely to be called several times to apply the same preconditioner. It is recommended that in these cases, that CHECK be set to ‘C’ on the first call F11DGFP, to ‘N’ on all subsequent calls.

## 6.3 Computational Costs

The time taken for a call to F11DGFP on each 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 Section 8 of the document for F11BAFP.

---