# F01YXFP

# 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

F01YXFP generates and distributes an $n$ by $n$ real banded symmetric matrix $A$, stored in compact storage format, on a one-dimensional logical grid of processors in column block distribution.

This routine distributes banded matrices in the form required by a number of the routines in Chapter F07 (especially F07HDFP (PDPBTRF) and F07HEFP (PDPBTRS) which are used for factorization and solution of real banded linear systems). A user-supplied subroutine is required to generate a block of the matrix $A$ on each processor.

## 2 Specification

```
SUBROUTINE F01YXFP(GMAT, UPLO, N, BW, A, IDESCA, IFAIL)
DOUBLE PRECISION   A(*)
INTEGER            N, BW, IDESCA(*), IFAIL
CHARACTER*1        UPLO
EXTERNAL           GMAT
```

## 3 Usage

### 3.1 Definitions

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

| | | |
|---|---|---|
| $b_w$ | – | the half band width of the symmetric banded matrix; |
| $p$ | – | $m_p \times n_p$, the total number of processors in the Library Grid. |
| $m_p$ | – | the number of rows in the Library Grid, for this routine $m_p = 1$ or $m_p = p$; |
| $n_p$ | – | the number of columns in the Library Grid, for this routine $n_p = 1$ or $n_p = p$. |
| $M_b^X$ | – | the blocking factor for the distribution of the rows of a matrix $X$. |
| $N_b^X$ | – | the blocking factor for the distribution of the columns of a matrix $X$. |

### 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: | UPLO, N, BW, some elements of IDESCA (see Section 4 for a description of IDESCA), IFAIL |
| Global output arguments: | IFAIL |

The remaining arguments are local.

### 3.3 Distribution Strategy

The matrix $A$ will be distributed over a one-dimensional array of processors, assuming a column block distribution and stored in an array A. The symmetric banded matrix, $A$ is stored in compact storage format (see the F07 Chapter Introduction). **It is important that** $p \times N_b^A \geq n$ and $N_b^A \geq 2 \times b_w$. This means that no processor may access more than one block of the matrix. The column block distribution is described in more detail in the F07 Chapter Introduction.

# 4 Arguments

**1:** GMAT — SUBROUTINE, supplied by the user. *External Procedure*

GMAT must return the block (represented below in bold) of the matrix $A$ in the array AL:

$$A = \begin{pmatrix}
a_{1,1} & \cdots & a_{1,b_w+1} & & & & & & & & \\
\vdots & \ddots & \ddots & \ddots & & & & & & & \\
a_{b_w+1,1} & \ddots & \ddots & \ddots & \mathbf{a_{k_1,j_1}} & & & & & & \\
 & \ddots & \ddots & \ddots & \boldsymbol{\vdots} & \boldsymbol{\ddots} & \mathbf{a_{k_2,j_2}} & & & & \\
 & & \ddots & \ddots & \mathbf{a_{j_1,j_1}} & \boldsymbol{\ddots} & \boldsymbol{\vdots} & \ddots & & & \\
 & & & \ddots & \vdots & \boldsymbol{\ddots} & \mathbf{a_{j_2,j_2}} & \ddots & \ddots & & \\
 & & & & a_{l_1,j_1} & \ddots & \vdots & \ddots & \ddots & \ddots & \\
 & & & & & \ddots & a_{l_2,j_2} & \ddots & \ddots & \ddots & a_{n-b_w,n} \\
 & & & & & & & \ddots & \ddots & \ddots & \vdots \\
 & & & & & & & & a_{n,n-b_w} & \cdots & a_{n,n}
\end{pmatrix},$$

if UPLO = 'U'; and

$$A = \begin{pmatrix}
a_{1,1} & \cdots & a_{1,b_w+1} & & & & & & & & \\
\vdots & \ddots & \ddots & \ddots & & & & & & & \\
a_{b_w+1,1} & \ddots & \ddots & \ddots & a_{k_1,j_1} & & & & & & \\
 & \ddots & \ddots & \ddots & \vdots & \ddots & a_{k_2,j_2} & & & & \\
 & & \ddots & \ddots & \mathbf{a_{j_1,j_1}} & \ddots & \vdots & \ddots & & & \\
 & & & \ddots & \boldsymbol{\vdots} & \boldsymbol{\ddots} & \mathbf{a_{j_2,j_2}} & \ddots & \ddots & & \\
 & & & & \mathbf{a_{l_1,j_1}} & \boldsymbol{\ddots} & \boldsymbol{\vdots} & \ddots & \ddots & \ddots & \\
 & & & & & \boldsymbol{\ddots} & \mathbf{a_{l_2,j_2}} & \ddots & \ddots & \ddots & a_{n-b_w,n} \\
 & & & & & & & \ddots & \ddots & \ddots & \vdots \\
 & & & & & & & & a_{n,n-b_w} & \cdots & a_{n,n}
\end{pmatrix},$$

if UPLO = 'L'; where $k_1, k_2, l_1$ and $l_2$ are such that;

$k_i = \max(1, j_i - b_w)$ for $i = 1, 2$ and;
$l_i = \min(n{:}j_i + b_w)$ for $i = 1, 2$.

More precisely,

AL$(k,j) = a_{ij}$

where $i, j, k$ are such that:

$j_1 \le j \le j_2$;
$1 \le k \le b_w + 1$;
$\max(1, j - b_w) \le i \le j$ if UPLO = 'L';
$j \le i \le \min(n, j + b_w)$ if UPLO = 'U';
and $i$ is the $k^{th}$ index where the corresponding term $a_{ij}$ lies (see the F07 Chapter Introduction).

Its specification is:

```
SUBROUTINE          GMAT(J1, J2, BW,UPLO, AL, LDAL)
DOUBLE PRECISION    AL(LDAL,*)
INTEGER             J1, J2, BW, LDAL
```

**1:** J1 — INTEGER *Local Input*

*On entry:* $j_1$, the first column of the block of $A$ to be generated.

**2:** J2 — INTEGER *Local Input*

*On entry:* $j_2$, the last column of the block of $A$ to be generated.

**3:** BW — INTEGER *Global Input*

*On entry:* $b_w$, the half band width of the matrix $A$ to be generated.

**4:** UPLO — CHARACTER*1 *Global Input*

*On entry:* indicates whether the upper or the lower triangular part of the matrix $A$ is required.

If UPLO = 'U', then a block of the upper traingular part of $A$ is required;
if UPLO = 'L', then a block of the lower traingular part of $A$ is required.

**5:** AL(LDAL,*) — DOUBLE PRECISION array *Local Output*

*On exit:* AL must contain the upper or lower triangular (depending on the value of UPLO) part of columns $j_1$ to $j_2$ of the band matrix $A$.

**6:** LDAL — INTEGER *Local Input*

*On entry:* the size of the first dimension of the array AL as declared in the (sub)program from which F01YXFP is called.

GMAT must be declared as EXTERNAL in the (sub)program from which F01YXFP is called. Arguments denoted as *Input* must **not** be changed by this procedure.

**2:** UPLO — CHARACTER*1 *Global Input*

*On entry:* indicates whether the upper or lower triangular part of $A$ is stored:

if UPLO = 'U', then the upper triangular part of $A$ is stored.
if UPLO = 'L', then the lower triangular part of $A$ is stored.

*Constraint:* UPLO = 'U' or 'L'.

**3:** N — INTEGER *Global Input*

*On entry:* $n$, the order of the matrix $A$.

*Constraint:* N $\geq 0$.

**4:** BW — INTEGER *Global Input*

*On entry:* $b_w$, the half band width of matrix $A$.

*Constraints:* $0 \leq$ BW $\leq$ N $-$ 1.

**5:** A($*$) — DOUBLE PRECISION array *Local Output*

**Note:** the array A is formally defined as a vector. However, you may find it more convenient to consider A as a two-dimensional array of dimension $(\ell_A, \gamma)$ where $\ell_A$ = IDESCA(9) if IDESCA(1) = 1, or $\ell_A$ = IDESCA(6) if IDESCA(1) = 501, and $\gamma \geq N_b^A$.

*On exit:* the local part of the matrix $A$ (stored in compact storage format) which will contain the upper part if UPLO = 'U' or the lower part if UPLO = 'L'. See the F07 Chapter Introduction for further details.

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

**Note:** the dimension of the array IDESCA must be at least 9 when IDESCA(1) = 1, and at least 6 when IDESCA(1) = 501.

*Distribution:* if IDESCA(1) = 1, the array elements IDESCA(3:8), must be global to each processor on the Library Grid. If IDESCA(1) = 501, then only the array elements IDESCA(3:5) must be global. In either case IDESCA(2) is local to each processor.

*On entry:* the array descriptor 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.

If IDESCA(1) = 501, then $p = 1 \times n_p$ or $p = m_p \times 1$, and the remaining elements of IDESCA must be set as follows:

IDESCA(2), the Library context, usually returned by a call to the Library Grid initialisation routine Z01AAFP;

IDESCA(3), the size $n$, of the matrix $A$;

IDESCA(4), the blocking factor, $N_b^A$, used to distribute the columns of the matrix $A$;

IDESCA(5), the processor index over which the first column of the matrix $A$ is distributed;

IDESCA(6), the leading dimension of the conceptual two-dimensional array A.

IDESCA(7:9) are not referenced.

If IDESCA(1) = 1, then $p = 1 \times n_p$ and the remaining elements of IDESCA must be set as follows:

IDESCA(2), the Library context, usually returned by a call to the Library Grid initialisation routine Z01AAFP;

IDESCA(3), the number of rows, $n$, of the matrix $A$;

IDESCA(4), the number of columns, $n$, of the matrix $A$;

IDESCA(5), is not referenced;

IDESCA(6), the blocking factor, $N_b^A$, used to distribute the columns of the matrix $A$;

IDESCA(7), the processor index over which the first column of the matrix $A$ is distributed;

IDESCA(8), is not referenced;

IDESCA(9), the leading dimension of the conceptual two-dimensional array A.

*Suggested value:* IDESCA(1) = 501 and $p = 1 \times n_p$, if IDESCB(1) = 1, IDESCB(5) = IDESCB(8) = 1.

*Constraints:*

IDESCA(1) = 1 or 501;

if IDESCA(1) = 1, then $p = 1 \times n_p$;

if IDESCA(1) = 501; then $p = m_p \times 1$ or $p = 1 \times n_p$;

if IDESCA(1) = 501, then

IDESCA(3) = N;

IDESCA(4) $\geq$ 2$\times$BW and $p\times$IDESCA(4) $\geq$ N;

IDESCA(5) $\geq$ 0;

IDESCA(6) $\geq$ BW + 1;

if IDESCA(1) = 1, then

IDESCA(3) = IDESCA(4) = N;

IDESCA(6) $\geq$ 2$\times$BW and $p\times$IDESCA(6) $\geq$ N;

IDESCA(7) $\geq$ 0;

IDESCA(9) $\geq$ BW + 1.

**7:** 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

INFO = $-2000$

The routine has been called with a value of ICNTXT which was not returned by a call to Z01AAFP on one or more processors.

INFO = $-1000$

The utility routine Z01AAFP has not been called to define the logical processor grid and initialise the internal variables used by the Library.

IFAIL = $-i$

On entry, one of the arguments was invalid:

if the $k$th argument is a scalar IFAIL = $-k$;
if the $k$th argument is an array and its $j$th element is invalid, IFAIL = $-(100 \times k + j)$.

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

The routine generates a real banded symmetric matrix and stores it in a one-dimensional array distributed by block-columns on a one-dimensional logical grid of processors.

## 6.2 Parallelism Detail

The routine generates the matrix on each logical processor grid independently.

# 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

See Section 8 of the document for F07HDFP (PDPBTRF).