F11BBFP

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 trying to use this routine. In particular, some of the notation and terminology used in this document was introduced in Section 2.2 of the F11 Chapter Introduction.

F11BBFP is the second in a suite of three routines for the iterative solution of a real general (unsymmetric) system of linear simultaneous equations $Ax = b$ using the Restarted Generalised Minimal Residual method (RGMRES) (Saad and Schultz [5], Barrett *et al.* [1], Dias da Cunha and Hopkins [2]). F11BBFP solves the system of equations after the first routine in the suite, F11BAFP, has been called to initialize the computation. F11BAFP **must** have been called prior to the initial entry to F11BBFP, otherwise an error condition will be raised. The third routine in the suite, F11BCFP, can be used to return additional information generated by the computation, after F11BBFP has completed its tasks. These three routines are suitable for the solution of large sparse systems of equations.

F11BBFP uses **reverse communication**, i.e., F11BBFP returns repeatedly to the calling program with the parameter IREVCM set to 1, −1, 2, 3 or 4 requiring the calling program to carry out specific tasks (see Section 4).

2 Specification

```
SUBROUTINE F11BBFP(ICNTXT, IREVCM, U, V, WORK, LWORK, IFAIL)
DOUBLE PRECISION   U(*), V(*), WORK(LWORK)
INTEGER            ICNTXT, IREVCM, LWORK, IFAIL
```

3 Data Distribution

3.1 Definitions

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

m_p	–	the number of rows in the logical grid of processors.
n_p	–	the number of columns in the logical grid of processors.
n	–	the order of the matrix A.
$n_l(i,j)$	–	the number of elements of the distributed vectors stored locally on the processor at location $\{i,j\}$ of the grid.
m	–	the size of the RGMRES basis used: every m iterations, the solution procedure will be restarted using the last iterate of the solution and residual vectors.

3.2 Global and Local Arguments

The input arguments IREVCM and IFAIL are global and so must have the same value on entry to the routine on each processor. The output arguments IREVCM and IFAIL are 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

The vectors in F11BBFP may be distributed in two different ways, as specified by the input parameters of F11BAFP.

First, vectors in F11BBFP may be distributed across all processors in the grid, with different processors holding different parts of the vector. In this case, the output vectors are distributed in the same way as the input vectors.

Second, on initial entry to F11BBFP, input vectors may be distributed along the first column or row of the logical grid of processors. Each processor in the first column or row then broadcasts the elements stored locally to all the other processors in the same row or column, respectively. Hence, if vectors are distributed by column or row, all processors in the same row or column, respectively, of the grid will hold copies of the same vector elements. Similarly, the output vectors are distributed in the same way as the input vectors after the initial broadcast.

4 Arguments

Note: this routine uses **reverse communication**. Its use involves an initial entry, intermediate exits and re-entries, and a final exit, as indicated by the **argument IREVCM**. Between intermediate exits and re-entries **all arguments other than IREVCM and V must remain unchanged**.

1: ICNTXT — INTEGER *Local Input*

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

On intermediate re-entry: ICNTXT is not referenced. The value supplied on initial entry is used.

2: IREVCM — INTEGER *Global Input/Global Output*

On initial entry: IREVCM = 0, otherwise an error condition will be raised.

On intermediate re-entry: IREVCM must either be unchanged from its previous exit value, or can have one of the following values.

5 Tidy termination: the computation will terminate at the end of the current iteration. Further reverse communication exits may occur depending on when the termination request is issued. F11BBFP will then generate the information accessible via F11BCFP and return with the termination code IREVCM = 4 and IFAIL = 4. Note that before calling F11BBFP with IREVCM = 5 the calling program must have performed the tasks required by the value of IREVCM returned by the previous call to F11BBFP, otherwise subsequently returned values may be invalid.

6 Immediate termination: F11BBFP will return immediately with termination code IREVCM = 4 and IFAIL = 8 and with any useful information available. This includes the iterate of the solution vector computed at the end of the last super-iteration performed.
Immediate termination may be useful, for example, when errors are detected during a matrix–vector multiplication or during the solution of the preconditioning equation.

Changing IREVCM to any other value between calls will result in an error.

On intermediate exit: IREVCM has the following meanings:

1 the calling program must compute the matrix–vector product $v = Au$, where u and v are stored in U and V, respectively;

−1 the calling program must compute the matrix–vector product $v = A^T u$, where u and v are stored in U and V, respectively. This value is returned only during the computation of $\|A\|_1$ or $\|A\|_\infty$ by Higham's method (Higham [4]);

2 the calling program must solve the preconditioning equation $Mv = u$, where u and v are stored in U and V, respectively;

3 monitoring step: the solution and residual at the current iteration are returned in the arrays U and V, respectively. No action by the calling program is required.

On final exit: IREVCM = 4: F11BBFP has completed its tasks. The value of IFAIL determines whether the iteration has been successfully completed, errors have been detected or the calling program has requested termination.

Constraints: on initial entry, IREVCM = 0; on re-entry, either IREVCM must either remain unchanged or be reset to 5 or 6.

3: U (∗) — DOUBLE PRECISION array *Local Input/Local Output*

Note: the dimension of the array U must be at least $n_l(i,j)$, the number of vector elements stored locally, where $\{i,j\}$ denotes the processor location in the grid. The distribution strategies used in F11BBFP are described in Section 3.3.

On initial entry: an initial estimate, x_0.

On intermediate re-entry: U must remain unchanged.

On intermediate exit: the returned value of IREVCM determines the contents of U in the following way:

IREVCM $= -1$, 1, 2 U holds the vector u on which the operations specified by IREVCM are to be carried out;

IREVCM $= 3$ U holds the current iterate of the solution vector.

On final exit: U holds the last iterate of the solution of the system of equations.

4: V (∗) — DOUBLE PRECISION array *Local Input/Local Output*

Note: the dimension of the array V must be at least $n_l(i,j)$, the number of vector elements stored locally, where $\{i,j\}$ denotes the processor location in the grid. The distribution strategies used in F11BBFP are described in Section 3.3.

On initial entry: the right-hand side, b.

On intermediate re-entry: the returned value of IREVCM determines the contents of V in the following way:

IREVCM $= -1$, 1, 2 V must store the vector v, the result of the operation specified by the value of IREVCM returned by the previous call to F11BBFP;

IREVCM $= 3$ V must remain unchanged.

On intermediate exit: if IRECVM $= 3$, V holds the current iterate of the residual vector, otherwise it does not contain any useful information.

On final exit: V contains the last iterate of the residual vector of the system of equations. The value of IFAIL indicates the success or failure of the solution process. Note that if an immediate termination request was issued, V does not contain any useful information; in this case V$(1:n_l(i,j)) = 0.0$ is returned.

5: WORK(LWORK) — DOUBLE PRECISION array *Local Input/Local Output*

Note: The distribution strategies used in F11BBFP are described in Section 3.3.

On initial entry: if user-supplied weights are used in the computation of the vector norms in the termination criterion (see Section 2 of the F11 Chapter Introduction), these must be stored in WORK$(1:n_l(i,j))$, where $n_l(i,j)$ is the number of vector elements stored locally.

On intermediate re-entry: WORK must remain unchanged.

On final exit: if weights are used, WORK$(1:n_l(i,j))$ remains unchanged from the values supplied on initial entry.

6: LWORK — INTEGER *Local Input*

On initial entry: the size of the array WORK as declared in the (sub)program from which F11BBFP was called. The amount of workspace required is as follows:

$$\text{LWORK} = (m+3)\,(n_l(i,j)+1) + m\,(m+4) + p + q,$$

where $p = 5$ if the l_2-norm is used in the termination criterion, or the l_1-norm is used and $\|A\|_1$ has been supplied (see Section 4 of the document for F11BAFP);

$p = N_p + 4$ if the l_1-norm is used in the termination criterion and $\|A\|_1$ is estimated by F11BBFP;

$p = 2\,N_p + 3$ if the l_∞-norm is used in the termination criterion;

$q = n_l(i,j)$ if user-specified weights are used;

$q = 0$ otherwise.

$N_p = m_p n_p$ if vectors are distributed across all processors in the grid, $N_p = m_p$ if vectors are distributed along the columns of the logical grid of processors, or $N_p = n_p$ if they are distributed along its rows.

On intermediate re-entry: LWORK is not referenced. The value supplied on initial entry is used.

Constraint: LWORK $\geq$ LWREQ, where LWREQ is returned by F11BAFP.

7: IFAIL — INTEGER *Global Input/Global Output*

On initial 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.

IFAIL is stored internally by F11BBFP.

On intermediate re-entry: IFAIL is not referenced. The value supplied to F11BBFP on initial entry is used.

On final 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 illegal value(s) on one or more processors. For global arguments, this may also be caused by the ith argument not having the same value on **all** processors (see also Section 3.2).

IFAIL = 1

F11BBFP has been called again after returning the termination code IREVCM = 4. No further computation has been carried out.

IFAIL = 2

The required accuracy could not be obtained. However, F11BBFP has terminated with reasonable accuracy: an internal inexpensive termination criterion, based on quantities readily available during the iteration, was satisfied but the termination criterion required, which uses the explicitly computed residual $r = b - Ax$, could not be satisfied. A small number of iterations have been carried out after the internal criterion has been satisfied, but have been unable to improve on the accuracy.

IFAIL = 3

F11BAFP was either not called before calling F11BBFP or returned an error.

IFAIL = 4

The calling program requested a tidy termination before the solution had converged.

IFAIL = 5

The solution did not converge within the maximum number of iterations.

IFAIL = 8

The calling program requested an immediate termination.

6 Further Comments

6.1 Algorithmic Detail

If $\|A\|_1$ or $\|A\|_\infty$ are to be computed internally by Higham's method, the calling program must also provide the means to compute the matrix-vector product $v = A^T u$. This corresponds to a returned value IREVCM $= -1$.

In order to avoid the lack of stability of ordinary Gram–Schmidt orthogonalization (Golub and Van Loan [3]), the RGMRES algorithm employed performs the Gram–Schmidt orthogonalization twice (see also Section 6.2 below).

Further algorithmic details are considered in Section 2.2 of the F11 Chapter Introduction.

6.2 Parallelism Detail

Some parallelism details are considered in Section 2.8.1 of the F11 Chapter Introduction.

Global operations, excluding the matrix–vector products and solutions of preconditioning equations carried out in the calling program, can be viewed as **synchronization points**. Because of the nature of the algorithms, RGMRES with iterated Gram–Schmidt orthogonalization requires two such synchronization points per iteration. In the case when the vectors are distributed across all the processor grid, the global operations involve communication between all the participating processors in the grid. However, when the vectors are distributed across the column or row, the global operations involve communications with processors on the same column or row of the grid. In the latter case different columns or rows can then act entirely in parallel, independently of each other.

6.3 Accuracy

On completion, i.e., IREVCM = 4 on exit, the arrays U and V will return the distributed solution and residual vectors, x_k and $r_k = b - Ax_k$, respectively, at the kth iteration, the last iteration performed, unless an immediate termination was requested, in which case information about the residual vector is generally not available.

On successful termination, the termination criterion is satisfied within the user-specified tolerance, as described in Section 2.2 of the F11 Chapter Introduction and in the documentation of F11BAFP. In any case, the left- and right-hand side of the termination criterion selected can be returned by a call to F11BCFP.

6.4 Computational costs

The number of operations carried out by F11BBFP for each iteration is likely to be principally determined by the computation of the matrix–vector products $v = Au$ and by the solution of the preconditioning equation $Mv = u$ in the calling program. Each of these operations are carried out once every iteration.

The number of the remaining operations in F11BBFP for each iteration is approximately proportional to $\max_{i,j}(n_l(i,j))$. This is multiplied by a factor which depends linearly on the iteration count within the current super-iteration (see Section 2.2 of the F11 Chapter Introduction).

The computational costs for each iteration and scalability issues are more difficult to assess, as they may depend on the efficiency of the underlying communication mechanism. In principle, almost linear speed-up can be achieved for efficient communication mechanisms.

The number of iterations required to achieve a prescribed accuracy cannot be easily determined at the onset, as it can depend dramatically on the conditioning and spectrum of the preconditioned matrix of the coefficients $\bar{A} = M^{-1}A$.

Additional matrix–vector products are required for the estimation of $\|A\|_1$ or $\|A\|_\infty$, when this has not been supplied to F11BAFP and is required by the termination criterion employed.

7 References

[1] Barrett R, Berry M, Chan T F, Demmel J, Donato J, Dongarra J, Eijkhout V, Pozo R, Romine C and van der Vorst H (1994) *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* SIAM, Philadelphia

[2] Dias da Cunha R and Hopkins T (1994) PIM 1.1 — the parallel iterative method package for systems of linear equations user's guide — Fortran 77 version *Technical Report* Computing Laboratory, University of Kent at Canterbury, Kent CT2 7NZ, UK

[3] Golub G H and Van Loan C F (1989) *Matrix Computations* Johns Hopkins University Press (2nd Edition), Baltimore

[4] Higham N J (1988) FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation *ACM Trans. Math. Software* **14** 381–396

[5] Saad Y and Schultz M (1986) GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems *SIAM J. Sci. Statist. Comput.* **7** 856–869

8 Example

See the Example Program for F11BAFP.