NAG Library Routine Document

F12AFF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F12AFF is a setup routine for F12AGF which may be used for finding some eigenvalues (and optionally the corresponding eigenvectors) of a standard or generalized eigenvalue problem defined by real, banded, nonsymmetric matrices. The banded matrix must be stored using the LAPACK column ordered storage format for real banded nonsymmetric matrices (see Section 3.3.4 in the F07 Chapter Introduction).

2 Specification

```
SUBROUTINE F12AFF (N, NEV, NCV, ICOMM, LICOMM, COMM, LCOMM, IFAIL)
INTEGER            N, NEV, NCV, ICOMM(max(1,LICOMM)), LICOMM, LCOMM,      &
                   IFAIL
REAL (KIND=nag_wp) COMM(max(1,LCOMM))
```

3 Description

The pair of routines F12AFF and F12AGF together with the option setting routine F12ADF are designed to calculate some of the eigenvalues, λ, (and optionally the corresponding eigenvectors, x) of a standard eigenvalue problem $Ax = \lambda x$, or of a generalized eigenvalue problem $Ax = \lambda Bx$ of order n, where n is large and the coefficient matrices A and B are banded real and nonsymmetric.

F12AFF is a setup routine which must be called before the option setting routine F12ADF and the solver routine F12AGF. Internally, F12AGF makes calls to F12ABF and F12ACF; the routine documents for F12ABF and F12ACF should be consulted for details of the algorithm used.

This setup routine initializes the communication arrays, sets (to their default values) all options that can be set by you via the option setting routine F12ADF, and checks that the lengths of the communication arrays as passed by you are of sufficient length. For details of the options available and how to set them, see Section 11.1 in F12ADF.

4 References

Lehoucq R B (2001) Implicitly restarted Arnoldi methods and subspace iteration *SIAM Journal on Matrix Analysis and Applications* **23** 551–562

Lehoucq R B and Scott J A (1996) An evaluation of software for computing eigenvalues of sparse nonsymmetric matrices *Preprint MCS-P547-1195* Argonne National Laboratory

Lehoucq R B and Sorensen D C (1996) Deflation techniques for an implicitly restarted Arnoldi iteration *SIAM Journal on Matrix Analysis and Applications* **17** 789–821

Lehoucq R B, Sorensen D C and Yang C (1998) *ARPACK Users' Guide: Solution of Large-scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods* SIAM, Philidelphia

5 Arguments

1: N – INTEGER *Input*

On entry: the order of the matrix A (and the order of the matrix B for the generalized problem) that defines the eigenvalue problem.

Constraint: $N > 0$.

2: NEV – INTEGER *Input*

On entry: the number of eigenvalues to be computed.

Constraint: $0 < \text{NEV} < \text{N} - 1$.

3: NCV – INTEGER *Input*

On entry: the number of Lanczos basis vectors to use during the computation.

At present there is no *a priori* analysis to guide the selection of NCV relative to NEV. However, it is recommended that $\text{NCV} \geq 2 \times \text{NEV} + 1$. If many problems of the same type are to be solved, you should experiment with increasing NCV while keeping NEV fixed for a given test problem. This will usually decrease the required number of matrix-vector operations but it also increases the work and storage required to maintain the orthogonal basis vectors. The optimal 'cross-over' with respect to CPU time is problem dependent and must be determined empirically.

Constraint: $\text{NEV} + 1 < \text{NCV} \leq \text{N}$.

4: ICOMM$(\max(1, \text{LICOMM}))$ – INTEGER array *Communication Array*

On exit: contains data to be communicated to F12AGF.

5: LICOMM – INTEGER *Input*

On entry: the dimension of the array ICOMM as declared in the (sub)program from which F12AFF is called.

If $\text{LICOMM} = -1$, a workspace query is assumed and the routine only calculates the required dimensions of ICOMM and COMM, which it returns in ICOMM(1) and COMM(1) respectively.

Constraint: $\text{LICOMM} \geq 140$ or $\text{LICOMM} = -1$.

6: COMM$(\max(1, \text{LCOMM}))$ – REAL (KIND=nag_wp) array *Communication Array*

On exit: contains data to be communicated to F12AGF.

7: LCOMM – INTEGER *Input*

On entry: the dimension of the array COMM as declared in the (sub)program from which F12AFF is called.

If $\text{LCOMM} = -1$, a workspace query is assumed and the routine only calculates the dimensions of ICOMM and COMM required by F12AGF, which it returns in ICOMM(1) and COMM(1) respectively.

Constraint: $\text{LCOMM} \geq 60$ or $\text{LCOMM} = -1$.

8: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: $\text{IFAIL} = 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, $\mathrm{N} \leq 0$.

IFAIL $= 2$

On entry, $\mathrm{NEV} \leq 0$.

IFAIL $= 3$

On entry, $\mathrm{NCV} < \mathrm{NEV} + 2$ or $\mathrm{NCV} > \mathrm{N}$.

IFAIL $= 4$

On entry, $\mathrm{LICOMM} < 140$ and $\mathrm{LICOMM} \neq -1$.

IFAIL $= 5$

On entry, $\mathrm{LCOMM} < 60$ and $\mathrm{LCOMM} \neq -1$.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

F12AFF is not threaded in any implementation.

9 Further Comments

None.

10 Example

The use of F12AFF is illustrated in Section 10 in F12AGF.