# NAG Fortran Library Routine Document

# F12FFF

**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

F12FFF is a setup routine for F12FGF which may be used for finding some eigenvalues (and optionally the corresponding eigenvectors) of a standard or generalized eigenvalue problem defined by real, banded, symmetric matrices. The banded matrix must be stored using the LAPACK storage format for real banded **non**symmetric matrices.

## 2 Specification

```
SUBROUTINE F12FFF (N, NEV, NCV, ICOMM, LICOMM, COMM, LCOMM, IFAIL)
INTEGER             N, NEV, NCV, ICOMM(*), LICOMM, LCOMM, IFAIL
double precision    COMM(*)
```

## 3 Description

The pair of routines F12FFF and F12FGF together with the option setting routine F12FDF are designed to calculate some of the eigenvalues, $\lambda$, (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 symmetric.

F12FFF is a setup routine which must be called before the option setting routine F12FDF and the solver routine F12FGF. Internally, F12FGF makes calls to F12FBF and F12FCF; the routine documents for F12FBF and F12FCF 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 the user via the option setting routine F12FDF, and checks that the lengths of the communication arrays as passed by the user are of sufficient length. For details of the options available and how to set them see Section 10.2 of the document for F12FDF.

## 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 Parameters

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*: $0 < \mathrm{NEV} < \mathrm{N} - 1$.

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 $\mathrm{NCV} \geq 2 \times \mathrm{NEV} + 1$. 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*: $\mathrm{NEV} < \mathrm{NCV} \leq \mathrm{N}$.

4: ICOMM($*$) – INTEGER array *Communication Array*
5: LICOMM – INTEGER *Input*

*On entry*: the dimension of the array ICOMM as declared in the (sub)program from which F12FFF is called.

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

*Constraint*: $\mathrm{LICOMM} \geq 140$.

6: COMM($*$) – ***double precision*** array *Communication Array*
7: LCOMM – INTEGER *Input*

*On entry*: the dimension of the array COMM as declared in the (sub)program from which F12FFF is called.

If $\mathrm{LCOMM} = -1$, a workspace query is assumed and the routine only calculates the required dimension of COMM, which it returns in $\mathrm{COMM}(1)$.

*Constraint*: $\mathrm{LCOMM} \geq 3 \times \mathrm{N} + \mathrm{NCV} \times \mathrm{NCV} + 8 \times \mathrm{NCV} + 60$.

8: IFAIL – INTEGER *Input/Output*

*On entry*: IFAIL must be set to 0, $-1$ or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

*On exit*: $\mathrm{IFAIL} = 0$ unless the routine detects an error (see Section 6).

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, for users not familiar with this parameter the recommended value is 0. **When the value $-1$ or 1 is used it is essential to test the value of IFAIL on exit.**

## 6 Error Indicators and Warnings

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

Errors or warnings detected by the routine:

$\mathrm{IFAIL} = 1$

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

IFAIL = 2

On entry, NEV $\leq 0$.

IFAIL = 3

On entry, NCV $\leq$ NEV or NCV $>$ N.

IFAIL = 4

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

IFAIL = 5

On entry, LCOMM $< 3 \times \mathrm{N} + \mathrm{NCV} \times \mathrm{NCV} + 8 \times \mathrm{NCV} + 60$ and LCOMM $\neq -1$.

## 7 Accuracy

Not applicable.

## 8 Further Comments

None.

## 9 Example

The use of F12FFF is illustrated by the example program of F12FGF.