NAG C Library Function Document

nag_real_banded_sparse_eigensystem_init (f12afc)

1 Purpose

nag_real_banded_sparse_eigensystem_init (f12afc) is a setup function for nag_real_banded_sparse_eigensystem_sol (f12agc) which may be used for finding some eigenvalues (and optionally the corresponding eigenvectors) of a standard or generalized eigenvalue problem defined by real, banded, non-symmetric matrices. The banded matrix must be stored using the LAPACK column ordered storage format for real banded non-symmetric matrices (see Section 3.3.2 in the f07 Chapter Introduction).

2 Specification

```
#include <nag.h>
#include <nagf12.h>
```

```
void nag_real_banded_sparse_eigensystem_init (Integer n, Integer nev,
     Integer ncv, Integer icomm[], Integer licomm, double comm[],
     Integer lcomm, NagError *fail)
```

3 Description

The pair of functions nag_real_banded_sparse_eigensystem_init (f12afc) and nag_real_banded_sparse_eigensystem_sol (f12agc) together with the option setting function nag_real_sparse_eigensystem_option (f12adc) 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 non-symmetric.

nag_real_banded_sparse_eigensystem_init (f12afc) is a setup function which must be called before the option setting function nag_real_sparse_eigensystem_option (f12adc) and the solver function nag_real_banded_sparse_eigensystem_sol (f12agc). Internally, nag_real_banded_sparse_eigensystem_sol (f12agc) makes calls to nag_real_sparse_eigensystem_iter (f12abc) and nag_real_sparse_eigensystem_sol (f12acc); the function documents for nag_real_sparse_eigensystem_iter (f12abc) and nag_real_sparse_eigensystem_sol (f12acc) should be consulted for details of the algorithm used.

This setup function initializes the communication arrays, sets (to their default values) all options that can be set by you via the option setting function nag_real_sparse_eigensystem_option (f12adc), 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 10.2 of the document for nag_real_sparse_eigensystem_option (f12adc).

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: $\mathbf{n} > 0$.

2: **nev** – Integer *Input*

On entry: the number of eigenvalues to be computed.

Constraint: $0 < \mathbf{nev} < \mathbf{n} - 1$.

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 $\mathbf{ncv} \geq 2 \times \mathbf{nev} + 1$. If many problems of the same type are to be solved, you should experiment with varying **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: $\mathbf{nev} + 1 < \mathbf{ncv} \leq \mathbf{n}$.

4: **icomm**[*dim*] – Integer *Communication Array*

Note: the dimension, *dim*, of the array **icomm** must be at least $\max(1, \mathbf{licomm})$.

On exit: contains data to be communicated to nag_real_banded_sparse_eigensystem_sol (f12agc).

5: **licomm** – Integer *Input*

On entry: the dimension of the array **icomm** as declared in the function from which nag_real_banded_sparse_eigensystem_init (f12afc) is called.

If $\mathbf{licomm} = -1$, a workspace query is assumed and the function only calculates the required dimensions of **icomm** and **comm**, which it returns in $\mathbf{icomm}(1)$ and $\mathbf{comm}(1)$ respectively.

Constraint: $\mathbf{licomm} \geq 140$ or $\mathbf{licomm} = -1$.

6: **comm**[*dim*] – double *Communication Array*

Note: the dimension, *dim*, of the array **comm** must be at least $\max(1, \mathbf{lcomm})$.

On exit: contains data to be communicated to nag_real_banded_sparse_eigensystem_sol (f12agc).

7: **lcomm** – Integer *Input*

On entry: the dimension of the array **comm** as declared in the function from which nag_real_banded_sparse_eigensystem_init (f12afc) is called.

If $\mathbf{lcomm} = -1$, a workspace query is assumed and the function only calculates the required dimensions of **icomm** and **comm**, which it returns in $\mathbf{icomm}(1)$ and $\mathbf{comm}(1)$ respectively.

Constraint: $\mathbf{lcomm} \geq 3 \times \mathbf{n} + 3 \times \mathbf{ncv} \times \mathbf{ncv} + 6 \times \mathbf{ncv} + 60$ or $\mathbf{lcomm} = -1$.

8: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 2.6 of the Essential Introduction).

6 Error Indicators and Warnings

NE_BAD_PARAM

On entry, argument $\langle value\rangle$ had an illegal value.

NE_INT

On entry, **n** ≤ 0: **n** $= \langle value\rangle$.

On entry, **nev** ≤ 0: **nev** $= \langle value\rangle$.

NE_INT_2

The length of the integer array **comm** is too small **lcomm** $= \langle value\rangle$, but must be at least $\langle value\rangle$.

The length of the integer array **icomm** is too small **licomm** $= \langle value\rangle$, but must be at least $\langle value\rangle$.

NE_INT_3

On entry, **ncv** $\leq$ **nev** $+ 1$ or **ncv** $>$ **n** where: **ncv** $= \langle value\rangle$, **nev** $= \langle value\rangle$, **n** $= \langle value\rangle$.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please consult NAG for assistance.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

The use of nag_real_banded_sparse_eigensystem_init (f12afc) is illustrated by the example program of nag_real_banded_sparse_eigensystem_sol (f12agc) (see Section 9 of the document for nag_real_banded_sparse_eigensystem_sol (f12agc)).