NAG Library Function Document

nag_real_symm_banded_sparse_eigensystem_init (f12ffc)

1 Purpose

nag_real_symm_banded_sparse_eigensystem_init (f12ffc) is a setup function for
nag_real_symm_banded_sparse_eigensystem_sol (f12fgc) which can be used to find 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 nonsymmetric matrices.

2 Specification

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

void nag_real_symm_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_symm_banded_sparse_eigensystem_init (f12ffc) and
nag_real_symm_banded_sparse_eigensystem_sol (f12fgc) together with the option setting function
nag_real_symm_sparse_eigensystem_option (f12fdc) are designed to calculate some of the eigenvalues,
\lambda, (and optionally the corresponding eigenvectors, \xi) 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.

nag_real_symm_banded_sparse_eigensystem_init (f12ffc) is a setup function which must be called
before the option setting function nag_real_symm_sparse_eigensystem_option (f12fdc) and the solver
function nag_real_symm_banded_sparse_eigensystem_sol (f12fgc). Internally,
nag_real_symm_banded_sparse_eigensystem_sol (f12fgc) makes calls to
nag_real_symm_sparse_eigensystem_iter (f12fbc) and nag_real_symm_sparse_eigensystem_sol
(f12fcc); the function documents for nag_real_symm_sparse_eigensystem_iter (f12fbc) and
nag_real_symm_banded_sparse_eigensystem_sol (f12fcc) 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_symm_sparse_eigensystem_option (f12fdc), 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
nag_real_symm_sparse_eigensystem_option (f12fdc).

4 References

Analysis and Applications 23 551–562

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

Eigenvalue Problems with Implicitly Restarted Arnoldi Methods SIAM, Philadelphia
5 Arguments

1: \( n \) – Integer  
   \( \text{Input} \)
   
   \( \text{On entry:} \) the order of the matrix \( A \) (and the order of the matrix \( B \) for the generalized problem) that defines the eigenvalue problem.
   
   \( \text{Constraint:} \ n > 0. \)

2: \( \text{nev} \) – Integer  
   \( \text{Input} \)
   
   \( \text{On entry:} \) the number of eigenvalues to be computed.
   
   \( \text{Constraint:} \ 0 < \text{nev} < n - 1. \)

3: \( \text{ncv} \) – Integer  
   \( \text{Input} \)
   
   \( \text{On entry:} \) the number of Lanczos basis vectors to use during the computation.
   
   At present there is no \textit{a priori} analysis to guide the selection of \( \text{ncv} \) relative to \( \text{nev} \). 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 \( \text{ncv} \) while keeping \( \text{nev} \) 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.
   
   \( \text{Constraint:} \ \text{nev} < \text{ncv} \leq n. \)

4: \( \text{icom}[\max(1, \text{licom})] \) – Integer  
   \( \text{Communication Array} \)
   
   \( \text{On exit:} \) contains data to be communicated to \texttt{nag_real_symm_banded_sparse_eigensystem_sol} (\texttt{f12fgc}).

5: \( \text{licom} \) – Integer  
   \( \text{Input} \)
   
   \( \text{On entry:} \) the dimension of the array \( \text{icom} \).
   
   If \( \text{licom} = -1 \), a workspace query is assumed and the function only calculates the required dimensions of \( \text{icom} \) and \( \text{comm} \), which it returns in \( \text{icom}[0] \) and \( \text{comm}[0] \) respectively.
   
   \( \text{Constraint:} \ \text{licom} \geq 140 \) or \( \text{licom} = -1. \)

6: \( \text{comm}[\max(1, \text{licom})] \) – double  
   \( \text{Communication Array} \)
   
   \( \text{On exit:} \) contains data to be communicated to \texttt{nag_real_symm_banded_sparse_eigensystem_sol} (\texttt{f12fgc}).

7: \( \text{lcom} \) – Integer  
   \( \text{Input} \)
   
   \( \text{On entry:} \) the dimension of the array \( \text{comm} \).
   
   If \( \text{lcom} = -1 \), a workspace query is assumed and the function only calculates the dimensions of \( \text{icom} \) and \( \text{comm} \) required by \texttt{nag_real_symm_banded_sparse_eigensystem_sol} (\texttt{f12fgc}), which it returns in \( \text{icom}[0] \) and \( \text{comm}[0] \) respectively.
   
   \( \text{Constraint:} \ \text{lcom} \geq 60 \) or \( \text{lcom} = -1. \)

8: \( \text{fail} \) – NagError *  
   \( \text{Input/Output} \)
   
   The NAG error argument (see Section 3.6 in the Essential Introduction).
6 Error Indicators and Warnings

NE_ALLOC_FAIL
Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

NE_BAD_PARAM
On entry, argument ⟨value⟩ had an illegal value.

NE_INT
On entry, n = ⟨value⟩.
Constraint: n > 0.
On entry, nev = ⟨value⟩.
Constraint: nev > 0.

NE_INT_2
The length of the integer array com is too small lcom = ⟨value⟩, but must be at least ⟨value⟩.
The length of the integer array icom is too small licom = ⟨value⟩, but must be at least ⟨value⟩.

NE_INT_3
On entry, nev = ⟨value⟩, nev = ⟨value⟩ and n = ⟨value⟩.
Constraint: nev > nev + 1 and nev ≤ n.

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 contact NAG for assistance.
An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE
Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

7 Accuracy
Not applicable.

8 Parallelism and Performance
Not applicable.

9 Further Comments
None.

10 Example
The use of nag_real_symm_banded_sparse_eigensystem_init (f12ffc) is illustrated by the example program of nag_real_symm_banded_sparse_eigensystem_sol (f12fgc) (see Section 10 in nag_real_symm_banded_sparse_eigensystem_sol (f12fgc)).