NAG Toolbox

nag_sparseig_real_band_solve (f12ag)

1 Purpose

nag_sparseig_real_band_solve (f12ag) is the main solver function in a suite of functions consisting of nag_sparseig_real_option (f12ad), nag_sparseig_real_band_init (f12af) and nag_sparseig_real_band_solve (f12ag). It must be called following an initial call to nag_sparseig_real_band_init (f12af) and following any calls to nag_sparseig_real_option (f12ad).

nag_sparseig_real_band_solve (f12ag) returns approximations to selected 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 storage format for real banded nonsymmetric matrices.

2 Syntax

```
[nconv, dr, di, z, resid, v, comm, icomm, ifail] = nag_sparseig_real_band_solve
(kl, ku, ab, mb, sigmar, sigmai, resid, comm, icomm)
[nconv, dr, di, z, resid, v, comm, icomm, ifail] = f12ag(kl, ku, ab, mb, sigmar,
sigmai, resid, comm, icomm)
```

3 Description

The suite of functions is 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.

Following a call to the initialization function nag_sparseig_real_band_init (f12af), nag_sparseig_real_band_solve (f12ag) returns the converged approximations to eigenvalues and (optionally) the corresponding approximate eigenvectors and/or an orthonormal basis for the associated approximate invariant subspace. The eigenvalues (and eigenvectors) are selected from those of a standard or generalized eigenvalue problem defined by real banded nonsymmetric matrices. There is negligible additional computational cost to obtain eigenvectors; an orthonormal basis is always computed, but there is an additional storage cost if both are requested.

The banded matrices A and B must be stored using the LAPACK column ordered storage format for banded nonsymmetric matrices; please refer to Section 3.2.2 in the F07 Chapter Introduction for details on this storage format.

nag_sparseig_real_band_solve (f12ag) is based on the banded driver functions **dnbdr1** to **dnbdr6** from the ARPACK package, which uses the Implicitly Restarted Arnoldi iteration method. The method is described in Lehoucq and Sorensen (1996) and Lehoucq (2001) while its use within the ARPACK software is described in great detail in Lehoucq *et al.* (1998). An evaluation of software for computing eigenvalues of sparse nonsymmetric matrices is provided in Lehoucq and Scott (1996). This suite of functions offers the same functionality as the ARPACK banded driver software for real nonsymmetric problems, but the interface design is quite different in order to make the option setting clearer and to combine the different drivers into a general purpose function.

nag_sparseig_real_band_solve (f12ag), is a general purpose function that must be called following initialization by nag_sparseig_real_band_init (f12af). nag_sparseig_real_band_solve (f12ag) uses options, set either by default or explicitly by calling nag_sparseig_real_option (f12ad), to return the converged approximations to selected eigenvalues and (optionally):

– the corresponding approximate eigenvectors;

– an orthonormal basis for the associated approximate invariant subspace;

– both.

Please note that for **Generalized** problems, the **Shifted Inverse Imaginary** and **Shifted Inverse Real** inverse modes are only appropriate if either A or B is symmetric semidefinite. Otherwise, if A or B is non-singular, the **Standard** problem can be solved using the matrix $B^{-1}A$ (say).

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

5.1 Compulsory Input Parameters

1: **kl** – INTEGER

The number of subdiagonals of the matrices A and B.

Constraint: **kl** ≥ 0.

2: **ku** – INTEGER

The number of superdiagonals of the matrices A and B.

Constraint: **ku** ≥ 0.

3: **ab**$(ldab,:)$ – REAL (KIND=nag_wp) array

The first dimension of the array **ab** must be at least $2 \times \mathbf{kl} + \mathbf{ku} + 1$.

The second dimension of the array **ab** must be at least $\max(1, \mathbf{n})$.

Must contain the matrix A in LAPACK banded storage format for nonsymmetric matrices (see Section 3.2.4 in the F07 Chapter Introduction).

4: **mb**$(ldmb,:)$ – REAL (KIND=nag_wp) array

The first dimension of the array **mb** must be at least $2 \times \mathbf{kl} + \mathbf{ku} + 1$.

The second dimension of the array **mb** must be at least $\max(1, \mathbf{n})$.

Must contain the matrix B in LAPACK banded storage format for nonsymmetric matrices (see Section 3.2.4 in the F07 Chapter Introduction).

5: **sigmar** – REAL (KIND=nag_wp)

If one of the **Shifted Inverse Real** modes (see nag_sparseig_real_option (f12ad)) have been selected then **sigmar** must contain the real part of the shift used; otherwise **sigmar** is not referenced. Section 4.3.4 in the F12 Chapter Introduction describes the use of shift and inverse transformations.

6: **sigmai** – REAL (KIND=nag_wp)

If one of the **Shifted Inverse Real** modes (see nag_sparseig_real_option (f12ad)) have been selected then **sigmai** must contain the imaginary part of the shift used; otherwise **sigmai** is not

referenced. Section 4.3.4 in the F12 Chapter Introduction describes the use of shift and inverse transformations.

7: **resid**(:) – REAL (KIND=nag_wp) array

The dimension of the array **resid** must be at least **n** (see nag_sparseig_real_band_init (f12af))

Need not be set unless the option **Initial Residual** has been set in a prior call to nag_sparseig_real_option (f12ad) in which case **resid** must contain an initial residual vector.

8: **comm**(:) – REAL (KIND=nag_wp) array

Must remain unchanged from the prior call to nag_sparseig_real_option (f12ad) and nag_sparseig_real_band_init (f12af).

9: **icomm**(:) – INTEGER array

Must remain unchanged from the prior call to nag_sparseig_real_option (f12ad) and nag_sparseig_real_band_init (f12af).

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **nconv** – INTEGER

The number of converged eigenvalues.

2: **dr**(:) – REAL (KIND=nag_wp) array

The dimension of the array **dr** will be $\mathbf{nev}+1$ (see nag_sparseig_real_band_init (f12af))

The first **nconv** locations of the array **dr** contain the real parts of the converged approximate eigenvalues. The number of eigenvalues returned may be one more than the number requested by **nev** since complex values occur as conjugate pairs and the second in the pair can be returned in position $\mathbf{nev}+1$ of the array.

3: **di**(:) – REAL (KIND=nag_wp) array

The dimension of the array **di** will be $\mathbf{nev}+1$ (see nag_sparseig_real_band_init (f12af))

The first **nconv** locations of the array **di** contain the imaginary parts of the converged approximate eigenvalues. The number of eigenvalues returned may be one more than the number requested by **nev** since complex values occur as conjugate pairs and the second in the pair can be returned in position $\mathbf{nev}+1$ of the array.

4: $\mathbf{z}(\mathbf{n} \times (\mathbf{nev}+\mathbf{1}))$ – REAL (KIND=nag_wp) array

The second dimension of the array **z** will be $\mathbf{n} \times (\mathbf{nev}+1)$.

If the default option **Vectors** = Ritz has been selected then **z** contains the final set of eigenvectors corresponding to the eigenvalues held in **dr** and **di**. The complex eigenvector associated with the eigenvalue with positive imaginary part is stored in two consecutive columns. The first column holds the real part of the eigenvector and the second column holds the imaginary part. The eigenvector associated with the eigenvalue with negative imaginary part is simply the complex conjugate of the eigenvector associated with the positive imaginary part.

5: **resid**(:) – REAL (KIND=nag_wp) array

The dimension of the array **resid** will be **n** (see nag_sparseig_real_band_init (f12af))

Contains the final residual vector.

6: **v**(**n** × **ncv**) – REAL (KIND=nag_wp) array

If the option **Vectors** (see nag_sparseig_real_option (f12ad)) has been set to Schur or Ritz then the first **nconv** × n elements of **v** will contain approximate Schur vectors that span the desired invariant subspace.

The ith Schur vector is stored in the ith column of **v**.

7: **comm**(:) – REAL (KIND=nag_wp) array

Contains no useful information.

8: **icomm**(:) – INTEGER array

Contains no useful information.

9: **ifail** – INTEGER

ifail = 0 unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, **kl** < 0.

ifail = 2

On entry, **ku** < 0.

ifail = 3

On entry, $ldab < 2 \times \mathbf{kl} + \mathbf{ku} + 1$.

ifail = 4

On entry, the option **Shifted Inverse Imaginary** was selected, and **sigmai** = zero, but **sigmai** must be nonzero for this computational mode.

ifail = 5

Iteration Limit < 0.

ifail = 6

The options **Generalized** and **Regular** are incompatible.

ifail = 7

The **Initial Residual** was selected but the starting vector held in **resid** is zero.

ifail = 8

Either the initialization function nag_sparseig_real_band_init (f12af) has not been called prior to the first call of this function or a communication array has become corrupted.

ifail = 9

On entry, $ldz < \mathbf{n}$ or $ldz < 1$ when no vectors are required.

ifail = 10

On entry, the option **Vectors** = Select was selected, but this is not yet implemented.

ifail $= 11$

The number of eigenvalues found to sufficient accuracy is zero.

ifail $= 12$

Could not build an Arnoldi factorization. Consider changing **ncv** or **nev** in the initialization function (see Section 5 in nag_sparseig_real_band_init (f12af) for details of these arguments).

ifail $= 13$

Unexpected error in internal call to compute eigenvalues and corresponding error bounds of the current upper Hessenberg matrix. Please contact NAG.

ifail $= 14$

Unexpected error during calculation of a real Schur form: there was a failure to compute all the converged eigenvalues. Please contact NAG.

ifail $= 15$

Unexpected error: the computed Schur form could not be reordered by an internal call. Please contact NAG.

ifail $= 16$

Unexpected error in internal call while calculating eigenvectors. Please contact NAG.

ifail $= 17$

Failure during internal factorization of real banded matrix. Please contact NAG.

ifail $= 18$

Failure during internal solution of real banded system. Please contact NAG.

ifail $= 19$

Failure during internal factorization of complex banded matrix. Please contact NAG.

ifail $= 20$

Failure during internal solution of complex banded system. Please contact NAG.

ifail $= 21$

The maximum number of iterations has been reached. Some Ritz values may have converged; **nconv** returns the number of converged values.

ifail $= 22$

No shifts could be applied during a cycle of the implicitly restarted Arnoldi iteration. One possibility is to increase the size of **ncv** relative to **nev** (see Section 5 in nag_sparseig_real_band_init (f12af) for details of these arguments).

ifail $= 23$

Overflow occurred during transformation of Ritz values to those of the original problem.

ifail $= 24$

The function was unable to dynamically allocate sufficient internal workspace. Please contact NAG.

ifail $= 25$

An unexpected error has occurred. Please contact NAG.

ifail $= -99$

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

ifail $= -399$

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

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

The relative accuracy of a Ritz value, λ, is considered acceptable if its Ritz estimate $\leq$ **Tolerance** $\times |\lambda|$. The default **Tolerance** used is the ***machine precision*** given by nag_machine_precision (x02aj).

8 Further Comments

None.

9 Example

This example constructs the matrices A and B using LAPACK band storage format and solves $Ax = \lambda Bx$ in shifted imaginary mode using the complex shift σ.

9.1 Program Text

```
function f12ag_example

fprintf('f12ag example results\n\n');

nx  = nag_int(10);
n   = nx*nx;
nev = nag_int(4);
ncv = nx;
kl  = nx;
ku  = nx;

% Construct ab and mb
ab = zeros(2*kl+ku+1,n);
mb = zeros(2*kl+ku+1,n);

% Main diagonal of A.
idiag = kl + ku + 1;
for j=1:n
  ab(idiag,j) = 4;
  mb(idiag,j) = 4;
end

% First subdiagonal and superdiagonal of A.
isup = kl + ku;
isub = kl + ku + 2;
rho = 100;
h = 1/double(nx+1);
for i=1:nx
  lo = (i-1)*nx;
  for j=lo+1:lo+nx-1
    ab(isub,j+1) = -1 + 0.5*h*rho;
    ab(isup,j)   = -1 - 0.5*h*rho;
  end
end
```

```
for j = 1:n - 1
  mb(isub,j+1) = 1;
  mb(isup,j) = 1;
end

% kl-th subdiagonal and ku-th super-diagonal.
isup = kl + 1;
isub = 2*kl + ku + 1;
for i = 1:nx - 1
  lo = (i-1)*nx;
  for j = lo + 1:lo + nx
    ab(isup,nx+j) = -1;
    ab(isub,j)    = -1;
  end
end

sigmar = 0.4;
sigmai = 0.6;
resid = zeros(100,1);

[icomm, comm, ifail] = f12af( ...
                              n, nev, ncv);
[icomm, comm, ifail] = f12ad( ...
                              'Shifted imaginary', icomm, comm);
[icomm, comm, ifail] = f12ad( ...
                              'Generalized', icomm, comm);
[nconv, dr, di, z, resid, v, comm, icomm, ifail] = ...
  f12ag( ...
         kl, ku, ab, mb, sigmar, sigmai, resid, comm, icomm);

fprintf('The %4d Ritz values closest to %8.2f %+8.2fi are:\n\n', ...
        nconv, sigmar, sigmai);
fprintf('%9.4f %+9.4fi\n', [dr(1:nconv) di(1:nconv)]');
```

9.2 Program Results

```
     f12ag example results

The    4 Ritz values closest to     0.40    +0.60i are:

   0.3610   +0.7223i
   0.3610   -0.7223i
   0.4598   -0.7199i
   0.4598   +0.7199i
```