NAG Toolbox for MATLAB

f12ag

1 Purpose

f12ag is the main solver function in a suite of functions consisting of f12ad, f12af and f12ag, that must be called following an initial call to f12af and following any calls to f12ad.

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] = 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 f12af, 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.

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 to you and to combine the different drivers into a general purpose function.

f12ag, is a general purpose forward communication function that must be called following initialization by f12af. f12ag uses options, set either by default or explicitly by calling 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.

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 – int32 scalar**

the number of subdiagonals of the matrices A and B.

Constraint: $\mathbf{kl} \geq 0$.

2: **ku – int32 scalar**

the number of superdiagonals of the matrices A and B.

Constraint: $\mathbf{ku} \geq 0$.

3: **ab**(**ldab,**$*$) **– double 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 must be at least $\max(1, \mathbf{n})$

Must contain the matrix A in LAPACK banded storage format for nonsymmetric matrices (see Section 3.2.2 in the F07 Chapter Introduction).

4: **mb**(**ldmb,**$*$) **– double 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 must be at least $\max(1, \mathbf{n})$

Must contain the matrix B in LAPACK banded storage format for nonsymmetric matrices (see Section 3.2.2 in the F07 Chapter Introduction).

5: **sigmar – double scalar**

If one of the **Shifted** modes have been selected then **sigmar** contains the real part of the shift used; otherwise **sigmar** is not referenced.

6: **sigmai – double scalar**

If one of the **Shifted** modes have been selected then **sigmai** contains the imaginary part of the shift used; otherwise **sigmai** is not referenced.

7: **resid**$(*)$ **– double array**

Note: the dimension of the array **resid** must be at least **n** (see f12af).

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

8: **comm**$(*)$ **– double array**

Note: the dimension of the array **comm** must be at least $\max(1, \mathbf{lcomm})$ (see f12af).

On initial entry: must remain unchanged from the prior call to f12af and f12ad.

9: **icomm**$(*)$ **– int32 array**

Note: the dimension of the array **icomm** must be at least $\max(1, \mathbf{licomm})$ (see f12af).

On initial entry: must remain unchanged from the prior call to f12af and f12ad.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

ldab, ldmb, ldz, ldv

5.4 Output Parameters

1: **nconv – int32 scalar**

The number of converged eigenvalues.

2: **dr**$(*)$ **– double array**

Note: the dimension of the array **dr** must be at least **nev** (see f12af).

The first **nconv** locations of the array **dr** contain the real parts of the converged approximate eigenvalues.

3: **di**$(*)$ **– double array**

Note: the dimension of the array **di** must be at least **nev** (see f12af).

The first **nconv** locations of the array **di** contain the imaginary parts of the converged approximate eigenvalues.

4: **z**$(\mathbf{n} \times (\mathbf{nev} + \mathbf{1}))$ **– double array**

The first dimension of the array **z** must be at least

The second dimension of the array must be at least $\mathbf{n} \times (\mathbf{nev} + 1)$

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**$(*)$ **– double array**

Note: the dimension of the array **resid** must be at least **n** (see f12af).

Contains the final residual vector.

6: **v**$(\mathbf{n} \times \mathbf{ncv})$ **– double array**

If the option **Vectors** has been set to Schur or Ritz and **z** is not set to **v**, then the first $\mathbf{nconv} \times n$ elements of **v** will contain approximate Schur vectors that span the desired invariant subspace. The ith Schur vector is stored in locations $\mathbf{v}(i \times n + j)$, for $i = 0, 1, \ldots, \mathbf{nconv} - 1$ and $j = 0, 1, \ldots, n - 1$.

7: **comm**$(*)$ **– double array**

Note: the dimension of the array **comm** must be at least $\max(1, \mathbf{lcomm})$ (see f12af).

Contains no useful information.

8: **icomm**$(*)$ – **int32 array**

Note: the dimension of the array **icomm** must be at least $\max(1, \mathbf{licomm})$ (see f12af).

Contains no useful information.

9: **ifail** – **int32 scalar**

0 unless the function detects an error (see Section 6).

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 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** $< \max(1, \mathbf{n})$ 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 of the document for f12af for details of these parameters).

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 of the document for f12af for details of these parameters).

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.

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 x02aj.

8 Further Comments

None.

9 Example

```
nx = 10;
n = int32(nx*nx);
nev = int32(4);
ncv = int32(10);
kl = int32(10);
ku = int32(10);

% 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 = 1/11;
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;
sigmai = 0;
resid = zeros(100,1);


[icomm, comm, ifail] = f12af(n, nev, ncv);
[icomm, comm, ifail] = f12ad('REGULAR INVERSE', icomm, comm);
[icomm, comm, ifail] = f12ad('GENERALIZED', icomm, comm);
[nconv, dr, di, z, residOut, v, comm, icomm, ifail] = ...
    f12ag(kl, ku, ab, mb, sigmar, sigmai, resid, comm, icomm)
```

```
nconv =
           4
dr =
    3.5094
```

```
    3.5094
    3.3702
    3.3702
di =
    1.1197
   -1.1197
    1.1145
   -1.1145
z =
     array elided
residOut =
     array elided
v =
     array elided
comm =
     array elided
icomm =
     array elided
ifail =
           0
```