NAG Toolbox for MATLAB

f12af

1 Purpose

f12af is a setup function for f12ag which may be used for finding some 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 column ordered storage format for real banded nonsymmetric matrices (see Section 3.2.2 in the F07 Chapter Introduction).

2 Syntax

```
[icomm, comm, ifail] = f12af(n, nev, ncv)
```

3 Description

The pair of functions f12af and f12ag together with the option setting function f12ad 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 nonsymmetric.

f12af is a setup function which must be called before the option setting function f12ad and the solver function f12ag. Internally, f12ag makes calls to f12ab and f12ac; the function documents for f12ab and f12ac 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 f12ad, 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 f12ad.

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: **n – int32 scalar**

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 – int32 scalar**

The number of eigenvalues to be computed.

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

3: **ncv – int32 scalar**

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: $\mathbf{nev} + 1 < \mathbf{ncv} \leq \mathbf{n}$.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

licomm, lcomm

5.4 Output Parameters

1: **icomm**($*$) **– int32 array**

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

Contains data to be communicated to f12ag.

2: **comm**($*$) **– double array**

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

Contains data to be communicated to f12ag.

3: **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:

$\mathbf{ifail} = 1$

On entry, $\mathbf{n} \leq 0$.

$\mathbf{ifail} = 2$

On entry, $\mathbf{nev} \leq 0$.

$\mathbf{ifail} = 3$

On entry, $\mathbf{ncv} < \mathbf{nev} + 2$ or $\mathbf{ncv} > \mathbf{n}$.

$\mathbf{ifail} = 4$

On entry, $\mathbf{licomm} < 140$ and $\mathbf{licomm} \neq -1$.

$\mathbf{ifail} = 5$

On entry, $\mathbf{lcomm} < 3 \times \mathbf{n} + 3 \times \mathbf{ncv} \times \mathbf{ncv} + 6 \times \mathbf{ncv} + 60$ and $\mathbf{lcomm} \neq -1$.

7 Accuracy

Not applicable.

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
```