NAG Toolbox for MATLAB

f12fd

1 Purpose

f12fd is an option setting function in a suite of functions consisting of f12fa, f12fb, f12fc, f12fd and f12fe, and may be used to supply individual optional parameters to f12fb and f12fc. The initialization function f12fa **must** have been called prior to calling f12fd.

2 Syntax

```
[icomm, comm, ifail] = f12fd(str, icomm, comm)
```

3 Description

f12fd may be used to supply values for optional parameters to f12fb and f12fc. It is only necessary to call f12fd for those parameters whose values are to be different from their default values. One call to f12fd sets one parameter value.

Each optional parameter is defined by a single character string consisting of one or more items. The items associated with a given option must be separated by spaces, or equals signs [=]. Alphabetic characters may be upper or lower case. The string

```
'Pointers = Yes'
```

is an example of a string used to set an optional parameter. For each option the string contains one or more of the following items:

– a mandatory keyword;

– a phrase that qualifies the keyword;

– a number that specifies an integer or double value. Such numbers may be up to 16 contiguous characters in Fortran's I, F, E or D format.

f12fd does not have an equivalent function from the ARPACK package which passes options by directly setting values to scalar parameters or to specific elements of array arguments. f12fd is intended to make the passing of options more transparent and follows the same principle as the single option setting functions in Chapter E04.

The setup function f12fa must be called prior to the first call to f12fd and all calls to f12fd must precede the first call to f12fb, the reverse communication iterative solver.

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 10.

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: **str – string**

A single valid option string (as described in Section 3 and Section 10).

2: **icomm**$(*)$ **– int32 array**

Note: the dimension of the array **icomm** must be at least $\max(1, \mathbf{licomm})$ (see f12fa).

On initial entry: must remain unchanged following a call to the setup function f12fa.

3: **comm**$(*)$ **– double array**

Note: the dimension of the array **comm** must be at least 60.

On initial entry: must remain unchanged following a call to the setup function f12fa.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **icomm**$(*)$ **– int32 array**

Note: the dimension of the array **icomm** must be at least $\max(1, \mathbf{licomm})$ (see f12fa).

Contains data on the current options set.

2: **comm**$(*)$ **– double array**

Note: the dimension of the array **comm** must be at least 60.

Contains data on the current options set.

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:

ifail $= 1$

The string passed in **str** contains an ambiguous keyword.

ifail $= 2$

The string passed in **str** contains a keyword that could not be recognized.

ifail $= 3$

The string passed in **str** contains a second keyword that could not be recognized.

ifail $= 4$

The initialization function f12fa has not been called or a communication array has become corrupted.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

```
n = int32(100);
nx = int32(10);
nev = int32(4);
ncv = int32(10);

irevcm = int32(0);
resid = zeros(100,1);
v = zeros(100,20);
x = zeros(100,1);
mx = zeros(100,1);

sigma = 0;


% Initialisation Step
[icomm, comm, ifail] = f12fa(n, nev, ncv);

% Set Optional Parameters
[icomm, comm, ifail] = f12fd('SMALLEST MAGNITUDE', icomm, comm);

% Solve
while (irevcm ~= 5)
  [irevcm, resid, v, x, mx, nshift, comm, icomm, ifail] = ...
    f12fb(irevcm, resid, v, x, mx, comm, icomm);
  if (irevcm == 1 || irevcm == -1)
    x = f12f_av(nx, x);
  elseif (irevcm == 4)
    [niter, nconv, ritz, rzest] = f12fe(icomm, comm);
      fprintf('Iteration %d, No. converged = %d, norm of estimates =
%16.8g\n', niter, nconv, norm(rzest(1:nev),2));
  end
end

% Post-process to compute eigenvalues/vectors
[nconv, d, z, v, comm, icomm, ifail] = f12fc(sigma, resid, v, comm,
icomm)
```

```
Iteration 1, No. converged = 0, norm of estimates =        81.010211
Iteration 2, No. converged = 0, norm of estimates =        45.634095
Iteration 3, No. converged = 0, norm of estimates =        42.747772
Iteration 4, No. converged = 0, norm of estimates =        8.6106757
Iteration 5, No. converged = 0, norm of estimates =       0.71330195
Iteration 6, No. converged = 0, norm of estimates =       0.15050738
Iteration 7, No. converged = 0, norm of estimates =      0.015776765
Iteration 8, No. converged = 0, norm of estimates =     0.0038996544
Iteration 9, No. converged = 0, norm of estimates =     0.0004324447
Iteration 10, No. converged = 0, norm of estimates =    0.00011026365
Iteration 11, No. converged = 0, norm of estimates =    1.2358564e-05
Iteration 12, No. converged = 0, norm of estimates =    3.1712516e-06
Iteration 13, No. converged = 1, norm of estimates =    3.5633917e-07
Iteration 14, No. converged = 1, norm of estimates =    4.1370551e-08
Iteration 15, No. converged = 2, norm of estimates =    5.3820859e-09
Iteration 16, No. converged = 1, norm of estimates =     7.557502e-10
Iteration 17, No. converged = 1, norm of estimates =    4.0268792e-11
Iteration 18, No. converged = 2, norm of estimates =    1.4048495e-11
Iteration 19, No. converged = 2, norm of estimates =     1.774664e-10
```

```
Iteration 20, No. converged = 2, norm of estimates =    1.1754284e-09
Iteration 21, No. converged = 2, norm of estimates =     7.782271e-09
Iteration 22, No. converged = 2, norm of estimates =    5.2551985e-10
Iteration 23, No. converged = 2, norm of estimates =    1.8326785e-10
Iteration 24, No. converged = 2, norm of estimates =    7.8753868e-11
Iteration 25, No. converged = 2, norm of estimates =    2.2420618e-11
Iteration 26, No. converged = 2, norm of estimates =    8.5727233e-13
Iteration 27, No. converged = 2, norm of estimates =    3.4872081e-13
Iteration 28, No. converged = 2, norm of estimates =    2.6274361e-14
Iteration 29, No. converged = 2, norm of estimates =    2.1845559e-14
Iteration 30, No. converged = 3, norm of estimates =     8.670653e-16
Iteration 31, No. converged = 3, norm of estimates =    2.5548816e-16
Iteration 32, No. converged = 3, norm of estimates =    4.9338532e-18
nconv =
           4
d =
   19.6054
   48.2193
   48.2193
   76.8333
         0
         0
         0
         0
         0
         0
z =
     array elided
v =
     array elided
comm =
     array elided
icomm =
     array elided
ifail =
           0
```

10 Optional Parameters

Several optional parameters for the computational functions f12fb and f12fc define choices in the problem specification or the algorithm logic. In order to reduce the number of formal parameters of f12fb and f12fc these optional parameters have associated *default values* that are appropriate for most problems. Therefore, you need only specify those optional parameters whose values are to be different from their default values.

The remainder of this section can be skipped if you wish to use the default values for *all* optional parameters. A complete list of optional parameters and their default values is given in Section 10.1.

Optional parameters may be specified by calling f12fd prior to a call to f12fb, but after a call to f12fa. One call is necessary for each optional parameter.

All optional parameters not specified by you are set to their default values. Optional parameters specified by you are unaltered by f12fb and f12fc (unless they define invalid values) and so remain in effect for subsequent calls unless altered by you.

10.1 Optional Parameter Checklist and Default Values

The following list gives the valid options. For each option, we give the keyword, any essential optional qualifiers and the default value. A definition for each option can be found in Section 10.2. The minimum abbreviation of each keyword is underlined. The qualifier may be omitted. The letters i and r denote integer and double values required with certain options. The number ϵ is a generic notation for ***machine precision*** (see x02aj).

Optional Parameters	**Default Values**
Advisory	Default = the value returned by x04ab
Both Ends	See **Largest Magnitude**

Buckling	See **Regular**
Cayley	See **Regular**
Defaults	
Exact Shifts	Default
Generalized	See **Standard**
Initial Residual	See **Random Residual**
Iteration Limit	Default $= 300$
Largest Algebraic	See **Largest Magnitude**
Largest Magnitude	Default
List	See **Nolist**
Monitoring	Default $= -1$
Nolist	Default
Pointers	Default $=$ No
Print Level	Default $= 0$
Random Residual	Default
Regular	Default
Regular Inverse	See **Regular**
Shifted Inverse	See **Regular**
Smallest Algebraic	See **Largest Magnitude**
Smallest Magnitude	See **Largest Magnitude**
Standard	Default
Supplied Shifts	See **Exact Shifts**
Tolerance	Default $= \epsilon$
Vectors	Default $=$ Schur

10.2 Description of the Optional Parameters

Advisory i Default $=$ the value returned by x04ab

The output channel for advisory messages.

Defaults

This special keyword may be used to reset all optional parameters to their default values.

Exact Shifts Default
Supplied Shifts

During the Lanczos iterative process, shifts are applied internally as part of the implicit restarting scheme. The shift strategy used by default and selected by the **Exact Shifts** is strongly recommended over the alternative **Supplied Shifts** (see Lehoucq *et al.* 1998 for details of shift strategies).

If **Exact Shifts** are used then these are computed internally by the algorithm in the implicit restarting scheme.

If **Supplied Shifts** are used then, during the Lanczos iterative process, you must supply shifts through array arguments of f12fb; this option should only be used if you are an experienced user since this requires some algorithmic knowledge and because more operations are usually required than for the implicit shift scheme.

If **Supplied Shifts** are used then, during the Lanczos iterative process, you must supply shifts through array arguments of f12fb when f12fb returns with **irevcm** $= 3$; the real and imaginary parts of the shifts are returned in **x** and **mx** respectively (or in **comm** when the option **Pointers** $=$ Yes is set). This option should only be used if you are an experienced user since this requires some algorithmic knowledge and because more operations are usually required than for the implicit shift scheme. Details on the use of explicit shifts and further references on shift strategies are available in Lehoucq *et al.* 1998.

Iteration Limit i Default $= 300$

The limit on the number of Lanczos iterations that can be performed before f12fb exits. If not all requested eigenvalues have converged to within **Tolerance** and the number of Lanczos iterations has reached this limit then f12fb exits with an error; f12fc can still be called subsequently to return the number of converged eigenvalues, the converged eigenvalues and, if requested, the corresponding eigenvectors.

Largest Magnitude Default
Both Ends
Largest Algebraic
Smallest Algebraic
Smallest Magnitude

The Lanczos iterative method converges on a number of eigenvalues with given properties. The default is for f12fb to compute the eigenvalues of largest magnitude using **Largest Magnitude**. Alternatively, eigenvalues may be chosen which have **Largest Algebraic** part **Smallest Magnitude**, or **Smallest Algebraic** part; or eigenvalues which are from **Both Ends** of the algebraic spectrum.

Note that these options select the eigenvalue properties for eigenvalues of OP (and B for **Generalized** problems), the linear operator determined by the computational mode and problem type.

Nolist Default
List

Normally each optional parameter specification is not printed to the advisory channel as it is supplied. Optional parameter **List** may be used to enable printing and optional parameter **Nolist** may be used to suppress the printing.

Monitoring i Default $= -1$

If $i > 0$, monitoring information is output to channel number i during the solution of each problem; this may be the same as the **Advisory** channel number. The type of information produced is dependent on the value of **Print Level**, see the description of the optional parameter **Print Level** for details of the information produced. Please see x04ac to associate a file with a given channel number.

Pointers Default $=$ No

During the iterative process and reverse communication calls to f12fb, required data can be communicated to and from f12fb in one of two ways. When **Pointers** $=$ No is selected (the default) then the array arguments **x** and **mx** are used to supply you with required data and used to return computed values back to f12fb. For example, when **irevcm** $= 1$ f12fb returns the vector x in **x** and the matrix-vector product Bx in **mx** and expects the result of the linear operation OP(x) to be returned in **x**.

If **Pointers** $=$ Yes is selected then the data is passed through sections of the array argument **comm**. The section corresponding to **x** when **Pointers** $=$ No begins at a location given by the first element of **icomm**; similarly the section corresponding to **mx** begins at a location given by the second element of **icomm**. This option allows f12fb to perform fewer copy operations on each intermediate exit and entry, but can also lead to less elegant code in the calling program.

Print Level i Default $= 0$

This controls the amount of printing produced by f12fd as follows.

$= 0$ No output except error messages. If you want to suppress all output, set **Print Level** $= 0$.

≥ 0 The set of selected options.

$= 2$ Problem and timing statistics on final exit from f12fb.

≥ 5 A single line of summary output at each Lanczos iteration.

≥ 10 If **Monitoring** > 0, **Monitoring** is set, then at each iteration, the length and additional steps of the current Lanczos factorization and the number of converged Ritz values; during re-orthogonalisation, the norm of initial/restarted starting vector; on a final Lanczos iteration, the number of update iterations taken, the number of converged eigenvalues, the converged eigenvalues and their Ritz estimates.

≥ 20 Problem and timing statistics on final exit from f12fb. If **Monitoring** > 0, **Monitoring** is set, then at each iteration, the number of shifts being applied, the eigenvalues and estimates of the symmetric tridiagonal matrix H, the size of the Lanczos basis, the wanted Ritz values and associated Ritz estimates and the shifts applied; vector norms prior to and following re-orthogonalisation.

≥ 30 If **Monitoring** > 0, **Monitoring** is set, then on final iteration, the norm of the residual; when computing the Schur form, the eigenvalues and Ritz estimates both before and after sorting; for each iteration, the norm of residual for compressed factorization and the symmetric tridiagonal matrix H; during re-orthogonalisation, the initial/restarted starting vector; during the Lanczos iteration loop, a restart is flagged and the number of the residual requiring iterative refinement; while applying shifts, some indices.

≥ 40 If **Monitoring** > 0, **Monitoring** is set, then during the Lanczos iteration loop, the Lanczos vector number and norm of the current residual; while applying shifts, key measures of progress and the order of H; while computing eigenvalues of H, the last rows of the Schur and eigenvector matrices; when computing implicit shifts, the eigenvalues and Ritz estimates of H.

≥ 50 If **Monitoring** is set, then during Lanczos iteration loop: norms of key components and the active column of H, norms of residuals during iterative refinement, the final symmetric tridiagonal matrix H; while applying shifts: number of shifts, shift values, block indices, updated tridiagonal matrix H; while computing eigenvalues of H: the diagonals of H, the computed eigenvalues and Ritz estimates.

Note that setting **Print Level** ≥ 30 can result in very lengthy **Monitoring** output.

Random Residual Default
Initial Residual

To begin the Lanczos iterative process, f12fb requires an initial residual vector. By default f12fb provides its own random initial residual vector; this option can also be set using optional parameter **Random Residual**. Alternatively, you can supply an initial residual vector (perhaps from a previous computation) to f12fb through the array argument **resid**; this option can be set using optional parameter **Random Residual**.

Regular Default
Regular Inverse
Shifted Inverse
Buckling
Cayley

These options define the computational mode which in turn defines the form of operation $\mathrm{OP}(x)$ to be performed when f12fb returns with **irevcm** $= -1$ or 1 and the matrix-vector product Bx when f12fb returns with **irevcm** $= 2$.

Given a **Standard** eigenvalue problem in the form $Ax = \lambda x$ then the following modes are available with the appropriate operator $\mathrm{OP}(x)$.

Regular $\mathrm{OP} = A$
Shifted Inverse $\mathrm{OP} = (A - \sigma I)^{-1}$ where σ is real

Given a **Generalized** eigenvalue problem in the form $Ax = \lambda Bx$ then the following modes are available with the appropriate operator $\mathrm{OP}(x)$.

Regular Inverse $\mathrm{OP} = B^{-1}A$
Shifted Inverse $\mathrm{OP} = (A - \sigma B)^{-1}B$, where σ is real

Buckling	OP $= (B - \sigma A)^{-1}A$, where σ is real
Cayley	OP $= (A - \sigma B)^{-1}(A + \sigma B)$, where σ is real

Standard Default
Generalized

The problem to be solved is either a standard eigenvalue problem, $Ax = \lambda x$, or a generalized eigenvalue problem, $Ax = \lambda Bx$. The optional parameter **Standard** should be used when a standard eigenvalue problem is being solved and the optional parameter **Generalized** should be used when a generalized eigenvalue problem is being solved.

Tolerance r Default $= \epsilon$

An approximate eigenvalue has deemed to have converged when the corresponding Ritz estimate is within **Tolerance** relative to the magnitude of the eigenvalue.

Vectors Default = Schur

The function f12fc can optionally compute the Schur vectors and/or the eigenvectors corresponding to the converged eigenvalues. To turn off computation of any vectors the option **Vectors** = None should be set. To compute only the Schur vectors (at very little extra cost), the option **Vectors** = Schur should be set and these will be returned in the array argument **v** of f12fc. To compute the eigenvectors (Ritz vectors) corresponding to the eigenvalue estimates, the option **Vectors** = Ritz should be set and these will be returned in the array argument **z** of f12fc, if **z** is set equal to **v** (as in Section 9) then the Schur vectors in **v** are overwritten by the eigenvectors computed by f12fc.