NAG Fortran Library Routine Document

F12FCF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F12FCF is a post-processing routine in a suite of routines consisting of F12FCF, F12FAF, F12FBF, F12FDF and F12FEF, that must be called following a final exit from F12FBF.

2 Specification

```
SUBROUTINE F12FCF (NCONV, D, Z, LDZ, SIGMA, RESID, V, LDV, COMM, ICOMM,
1                  IFAIL)
INTEGER            NCONV, LDZ, LDV, ICOMM(*), IFAIL
double precision   D(*), Z(LDZ,*), SIGMA, RESID(*), V(LDV,*), COMM(*)
```

3 Description

The suite of routines 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 sparse, real and symmetric. The suite can also be used to find selected eigenvalues/eigenvectors of smaller scale dense, real and symmetric problems.

Following a call to F12FBF, F12FCF 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 symmetric matrices. There is negligible additional cost to obtain eigenvectors; an orthonormal basis is always computed, but there is an additional storage cost if both are requested.

F12FCF is based on the routine **dseupd** from the ARPACK package, which uses the Implicitly Restarted Lanczos 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 symmetric matrices is provided in Lehoucq and Scott (1996). This suite of routines offers the same functionality as the ARPACK software for real symmetric problems, but the interface design is quite different in order to make the option setting clearer to the user and to simplify some of the interfaces.

F12FCF, is a post-processing routine that must be called following a successful final exit from F12FBF. F12FCF uses data returned from F12FBF and options, set either by default or explicitly by calling F12FDF, 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

1: NCONV – INTEGER *Output*

On exit: the number of converged eigenvalues as found by F12FBF.

2: D(∗) – ***double precision*** array *Output*

Note: the dimension of the array D must be at least NEV.

On exit: the first NCONV locations of the array D contain the the converged approximate eigenvalues.

3: Z(LDZ,∗) – ***double precision*** array *Output*

Note: the second dimension of the array Z must be at least NEV if the default option **Vectors** = Ritz has been selected and at least 1 if the option **Vectors** = None or Schur has been selected.

On exit: if the default option **Vectors** = Ritz has been selected then Z contains the final set of eigenvectors corresponding to the eigenvalues held in D. The real eigenvector associated with an eigenvalue is stored in the corresponding column of Z.

4: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F12FCF is called.

Constraints:

if the default option **Vectors** = Ritz has been selected, LDZ $\geq$ N;
if the option **Vectors** = None or Schur has been selected, LDZ ≥ 1.

5: SIGMA – ***double precision*** *Input*

On entry: if one of the **Shifted** modes has been selected then SIGMA contains the real shift used; otherwise SIGMA is not referenced.

6: RESID(∗) – ***double precision*** array *Input*

Note: the dimension of the array RESID must be at least N.

On entry: RESID must not be modified following a call to F12FBF since it contains data required by F12FCF.

7: V(LDV,∗) – ***double precision*** array *Input/Output*

Note: the second dimension of the array V must be at least $\max(1, \mathrm{NCV})$.

On entry: the NCV columns of V contain the Lanczos basis vectors for OP as constructed by F12FBF.

On exit: if the option **Vectors** = Schur has been set, or the option **Vectors** = Ritz has been set and a separate array Z has been passed, then the first NCONV columns of V will contain approximate Schur vectors that span the desired invariant subspace.

8: LDV – INTEGER *Input*

On entry: the first dimension of the array V as declared in the (sub)program from which F12FCF is called.

Constraint: LDV $\geq n$.

9: COMM(∗) – ***double precision*** array *Communication Array*

COMM must remain unchanged from the prior call to F12FBF.

10: ICOMM(∗) – INTEGER array *Communication Array*

ICOMM must remain unchanged from the prior call to F12FBF.

11: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL $= 0$ unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, LDZ $<$ max$(1, \mathrm{N})$ or LDZ < 1 when no vectors are required.

IFAIL $= 2$

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

IFAIL $= 3$

The number of eigenvalues found to sufficient accuracy prior to calling F12FCF, as communicated through the parameter ICOMM, is zero.

IFAIL $= 4$

The number of converged eigenvalues as calculated by F12FBF differ from the value passed to it through the parameter ICOMM.

IFAIL $= 5$

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

IFAIL $= 6$

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

IFAIL $= 7$

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 X02AJF.

8 Further Comments

None.

9 Example

The example solves $Ax = \lambda Bx$ in regular mode, where A and B are obtained from the standard central difference discretization of the one-dimensional Laplacian operator $\frac{d^2u}{dx^2}$ on $[0, 1]$, with zero Dirichlet boundary conditions.

9.1 Program Text

```
*     F12FCF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          IMON, LICOMM, NIN, NOUT
      PARAMETER        (IMON=0,LICOMM=134,NIN=5,NOUT=6)
      INTEGER          MAXN, MAXNCV, LDV
      PARAMETER        (MAXN=256,MAXNCV=30,LDV=MAXN)
      INTEGER          LCOMM
      PARAMETER        (LCOMM=3*MAXN+MAXNCV*MAXNCV+8*MAXNCV+60)
      DOUBLE PRECISION FOUR, ONE, SIX, ZERO
      PARAMETER        (FOUR=4.0D+0,ONE=1.0D+0,SIX=6.0D+0,ZERO=0.0D+0)
*     .. Local Scalars ..
      DOUBLE PRECISION H, R1, R2, SIGMA
      INTEGER          IFAIL, INFO, IREVCM, J, N, NCONV, NCV, NEV,
     +                 NITER, NSHIFT
*     .. Local Arrays ..
      DOUBLE PRECISION AD(MAXN), ADL(MAXN), ADU(MAXN), ADU2(MAXN),
     +                 COMM(LCOMM), D(MAXNCV,2), MX(MAXN), RESID(MAXN),
     +                 V(LDV,MAXNCV), X(MAXN)
      INTEGER          ICOMM(LICOMM), IPIV(MAXN)
*     .. External Functions ..
      DOUBLE PRECISION DNRM2
      EXTERNAL         DNRM2
*     .. External Subroutines ..
      EXTERNAL         AV, DCOPY, DGTTRF, DGTTRS, F12FAF, F12FBF,
     +                 F12FCF, F12FDF, F12FEF, MV
*     .. Intrinsic Functions ..
      INTRINSIC        DBLE
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F12FCF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, NEV, NCV
      IF (N.LT.1 .OR. N.GT.MAXN) THEN
         WRITE (NOUT,99999) 'N is out of range: N = ', N
      ELSE IF (NCV.GT.MAXNCV) THEN
         WRITE (NOUT,99999) 'NCV is out of range: NCV = ', NCV
      ELSE
         IFAIL = 0
         CALL F12FAF(N,NEV,NCV,ICOMM,LICOMM,COMM,LCOMM,IFAIL)
*     We are solving a generalized problem
         CALL F12FDF('GENERALIZED',ICOMM,COMM,IFAIL)
*
         H = ONE/DBLE(N+1)
         R1 = (FOUR/SIX)*H
         R2 = (ONE/SIX)*H
         DO 20 J = 1, N
            AD(J) = R1
            ADL(J) = R2
   20    CONTINUE
         CALL DCOPY(N,ADL,1,ADU,1)
         CALL DGTTRF(N,ADL,AD,ADU,ADU2,IPIV,INFO)
*
         IREVCM = 0
         IFAIL = -1
```

```
   40    CONTINUE
         CALL F12FBF(IREVCM,RESID,V,LDV,X,MX,NSHIFT,COMM,ICOMM,IFAIL)
         IF (IREVCM.NE.5) THEN
            IF (IREVCM.EQ.-1 .OR. IREVCM.EQ.1) THEN
*           Perform  X <--- OP*x = inv[M]*A*x.
               CALL AV(N,X,MX)
               CALL DCOPY(N,MX,1,X,1)
               CALL DGTTRS('N',N,1,ADL,AD,ADU,ADU2,IPIV,X,N,INFO)
            ELSE IF (IREVCM.EQ.2) THEN
*           Perform  MX <--- M*x.
               CALL MV(N,X,MX)
            ELSE IF (IREVCM.EQ.4 .AND. IMON.NE.0) THEN
*           Output monitoring information
               CALL F12FEF(NITER,NCONV,D,D(1,2),ICOMM,COMM)
               WRITE (6,99998) NITER, NCONV, DNRM2(NEV,D(1,2),1)
            END IF
            GO TO 40
         END IF
         IF (IFAIL.EQ.0) THEN
*        Post-Process using F12FCF to compute eigenvalues/vectors.
            SIGMA = ZERO
            CALL F12FCF(NCONV,D,V,LDV,SIGMA,RESID,V,LDV,COMM,ICOMM,
     +                  IFAIL)
            WRITE (NOUT,99996) NCONV
            DO 60 J = 1, NCONV
               WRITE (NOUT,99995) J, D(J,1)
   60       CONTINUE
         ELSE
            WRITE (NOUT,99997) IFAIL
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,I5)
99998 FORMAT (1X,'Iteration',1X,I3,', No. converged =',1X,I3,', norm o',
     +       'f estimates =',E16.8)
99997 FORMAT (1X,' NAG Routine F12FBF Returned with IFAIL = ',I6)
99996 FORMAT (1X,/' The ',I4,' generalized Ritz values of largest magn',
     +       'itude are:',/)
99995 FORMAT (1X,I8,5X,F9.1)
      END
*
      SUBROUTINE MV(N,V,W)
*     .. Parameters ..
      DOUBLE PRECISION ONE, FOUR, SIX
      PARAMETER     (ONE=1.0D+0,FOUR=4.0D+0,SIX=6.0D+0)
*     .. Scalar Arguments ..
      INTEGER       N
*     .. Array Arguments ..
      DOUBLE PRECISION V(N), W(N)
*     .. Local Scalars ..
      DOUBLE PRECISION H
      INTEGER       J
*     .. External Subroutines ..
      EXTERNAL      DSCAL
*     .. Intrinsic Functions ..
      INTRINSIC     DBLE
*     .. Executable Statements ..
      H = ONE/(DBLE(N+1)*SIX)
      W(1) = FOUR*V(1) + V(2)
      DO 20 J = 2, N - 1
         W(J) = V(J-1) + FOUR*V(J) + V(J+1)
   20 CONTINUE
      J = N
      W(J) = V(J-1) + FOUR*V(J)
      CALL DSCAL(N,H,W,1)
      RETURN
      END
*
      SUBROUTINE AV(N,V,W)
*     .. Parameters ..
```

```
      DOUBLE PRECISION ONE, TWO
      PARAMETER     (ONE=1.0D+0,TWO=2.0D+0)
*     .. Scalar Arguments ..
      INTEGER       N
*     .. Array Arguments ..
      DOUBLE PRECISION V(N), W(N)
*     .. Local Scalars ..
      DOUBLE PRECISION H
      INTEGER       J
*     .. External Subroutines ..
      EXTERNAL      DSCAL
*     .. Intrinsic Functions ..
      INTRINSIC     DBLE
*     .. Executable Statements ..
      H = ONE/DBLE(N+1)
      W(1) = TWO*V(1) - V(2)
      DO 20 J = 2, N - 1
         W(J) = -V(J-1) + TWO*V(J) - V(J+1)
   20 CONTINUE
      J = N
      W(J) = -V(J-1) + TWO*V(J)
      CALL DSCAL(N,ONE/H,W,1)
      RETURN
      END
```

9.2 Program Data

```
F12FCF Example Program Data
 100 4 10 : Values for N NEV and NCV
```

9.3 Program Results

```
 F12FCF Example Program Results

 The    4 generalized Ritz values of largest magnitude are:

        1      121003.5
        2      121616.6
        3      122057.5
        4      122323.2
```