SL4EFN – FORTRAN 77 Library Routine Document

NOTE: In the current implementation, all **real** variables are declared to be DOUBLE PRECISION.

1 Purpose

SL4EFN is called after a call to the eigenvalue routine SLEUTH in order to determine an approximation to the corresponding eigenfunction.

2 Specification

```
      SUBROUTINE SL4EFN(X,Y1,Y2,HOT,WORK,IWORK,NMESH,IMATCH,ELAM,
     &                  MULTI,WORKS,YSTOR1,YSTOR2,SL4BCS,IFAIL)
      INTEGER IWORK,NMESH,IMATCH,MULTI,IFAIL
      DOUBLE PRECISION X,Y1(1:4),Y2(1:4),WORK(0:IWORK,1:6),ELAM,
     &                 WORKS(19*NMESH+48),YSTOR1(1:4,0:NMESH),
     &                 YSTOR2(1:4,0:NEMSH)
      LOGICAL HOT
      EXTERNAL SL4BCS
```

\section{ Description \label{sec:Descrip}}

SL4EFN must only be called after a call to SLEUTH to determine the required eigenvalue and other information, which will be passed from SLEUTH to SL4EFN via the appropriate parameters (ELAM, NMESH, IMATCH and WORK). None of these parameters should be changed between the call to SLEUTH and the call to SL4EFN.

SL4EFN computes the eigenfunction and its first three quasi-derivatives (see below) at the user-specified point X and returns these to the user in the array Y1. If the eigenvalue has multiplicity 2, then the values of a second eigenfunction and its quasi-derivatives at X will be placed in Y2.

SL4EFN uses a 'hot start' facility: before the first call the user must set HOT = .FALSE., which forces SL4EFN to set up the eigenfunction values at the mesh nodes, as well as evaluating the eigenfunction at the point specified by the user. After this first call HOT will become .FALSE. so that, on subsequent calls, SL4EFN simply computes the eigenfunction value at the point specified by the user, using the previously determined nodal information to reduce the cost of the computation. Thus, unless the user wishes to evaluate the eigenfunction at the mesh points, SL4EFN must be called once for each point at which the user requires an eigenfunction value.

The user who requires only nodal information about the eigenfunction can obtain this with a single call to SL4EFN: after the first call to SL4EFN with HOT = .TRUE., the required information will be stored in the array YSTOR1 and, if the eigenvalue was of multiplicity 2, in the array YSTOR2. See the descriptions of these variables given below for details.

3 References

See the bibliography at the end of this document.

4 Parameters

X – **real** scalar (input).
On entry: X is the value of x at which the eigenfunction(s) is(are) to be computed.
Unchanged on exit.

Y1 – **real** array of DIMENSION (1:4) (output).
On exit: Y1(1) contains $y(x)$, Y1(2) contains $y'(x)$, Y1(3) contains $-(py'')'(x)+sy'(x)$ and Y1(4) contains $py''(x)$, where y is an eigenfunction associated with the eigenvalue.

Y2 – **real** array of DIMENSION (1:4) (output),
On exit: If the eigenvalue has multiplicity 2 (MULTI = 2 – see below) then Y2(1),..,Y2(4) contain the same quantities as Y1(1),...,Y1(4) for a second eigenfunction linearly independent of the first.

HOT – LOGICAL (input/output).
On entry: HOT should be set to .FALSE. on the first call. After this call, HOT will have the value .TRUE., and this should not be changed until the eigenfunctions associated with a different eigenvalue or different problem are required.
On exit: HOT will always be set to .TRUE..

WORK – **real** array of DIMENSION (0:IWORK,1:6) (workspace).
WORK must be exactly the same as on the preceeding call to SLEUTH which computed the eigenvalue.

IWORK – INTEGER (input).
On entry: IWORK specifies the first dimension of WORK and must be exactly the same as on the preceeding call to SLEUTH.

NMESH – INTEGER (input).
On entry: NMESH must be the same as supplied from the previous call to SLEUTH.

IMATCH – INTEGER (input).
On entry: IMATCH must be the same as supplied from the previous call to SLEUTH.

ELAM –**real** scalar (input).
On entry: ELAM contains the eigenvalue approximation supplied by the previous call to SLEUTH.

MULTI – INTEGER (output).
On exit: MULTI specifies the multiplicity of the eigenvalue under consideration (MULTI = 1 or 2). MULTI must not be changed by the user.

WORKS – **real** array of DIMENSION at least (19*NMESH+48) (workspace).
The contents of WORKS must not be changed by the user between successive calls to SL4EFN with HOT = .TRUE.

YSTOR1 – **real** array of DIMENSION (1:4,0:p) where $p \geq$NMESH.
On exit: After the first call to SL4EFN with HOT = .FALSE., the input values of YSTOR(1,I),..,YSTOR(4,I) specify for each I=0,...,NMESH, the values of, respectively,

$$y(x_I), y'(x_I), -(py'')'(x_I)+sy'(x_I), py''(x_I)$$

for an eigenfunction of the current eigenvalue. Here x_I is the I^{th} meshpoint, whose value may be found in WORK(I,1).

YSTOR2 – **real** array of DIMENSION (1:4,0:p) where $p \geq$NMESH.
On exit: If MULTI=2 then YSTOR2 will be assigned in exactly the same way as YSTOR1 but for a second linearly independent eigenfunction.

SL4BCS – SUBROUTINE (supplied by user)
This is exactly the same subroutine which is supplied to SLEUTH: see SLEUTH documentation for more information.

IFAIL – INTEGER (input/output).

On entry: IFAIL must be set to -1, 0 or 1. For users not familiar with this parameter (described in Chapter P01 of the NAG Library documentation) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (see Section 5).

5 Error Indicators and Warnings

Errors detected by the routine:

IFAIL = 1. The value of X is outside the interval [A,B] on which the problem is posed.

IFAIL = 2. During the integration a pair of mtatrices (U,V) has arisen which do not satisfy the conjointness condition. Check the subroutine SL4BCS, making sure that the matrices U and V returned by this routine satisfy

$$U^T V - V^T U = 0.$$

6 Accuracy

Eigenfunction accuracy is difficult to assess because it depends also on the proximity of other eigenvalues. The user is advised to run SLEUTH and SL4EFN again at a tighter tolerance to obtain an assessment of the error.

7 Run times

On the first call with HOT = .FALSE., the run time will be proportional to NMESH; on subsequent calls with HOT = .TRUE. it will be much lower.

8 Keywords

Eigenvalue Problems, Sturm-Liouville Problems, Fourth Order Differential Equations, Matrix Oscillation Theory, Shooting Methods, Coefficient Approximation, Eigenfunctions.

9 Example

We consider the problem on the interval $[0, 100]$ with boundary conditions $y(0) = y'''(0)$, $y'(0) = 0$, $y(100) = y''(100) = 0$, given by the equation

$$y'''' - (2xy')' + x^2 y = \lambda y.$$

The eigenvalues of this problem are the squares of the eigenvalues of the second-order problem on the same interval with boundary conditions $y'(0) = 0 = y(100)$ and equation

$$-y'' + xy = \lambda y.$$

9.1 Example Program Text

```
      PROGRAM MAIN
C     .. Parameters ..
      INTEGER IWORK,IWORKS
      PARAMETER (IWORK=20480,IWORKS=27*IWORK+56)
C     ..
C     .. Local Scalars ..
      DOUBLE PRECISION A,B,ELAM,ERR,TOL,X
      INTEGER I,IFAIL,IMATCH,K,MULTI,NMESH,NXTRAP
      LOGICAL HOT,SYM
C     ..
C     .. Local Arrays ..
      DOUBLE PRECISION WORK(0:IWORK,1:6),WORKS(IWORKS),Y1(1:4),Y2(1:4),
     &                 YSTOR1(1:4,0:IWORK),YSTOR2(1:4,0:IWORK)
C     ..
C     .. External Subroutines ..
      EXTERNAL SL4BCS,SL4COF,SL4EFN,SLEUTH
C     ..
C     .. Intrinsic Functions ..
      INTRINSIC DBLE
C     ..

C Set up endpoints, eigenvalue index, initial eigenvalue approximation
C and tolerance:
      A = 0.D0
      B = 100.D0
      ELAM = 0.D0
      ERR = 1.D0
      K = 1
      TOL = 1.0D-6

      SYM = .FALSE.
      NMESH = IWORK
      IFAIL = 0

C Compute the eigenvalue:

      CALL SLEUTH(A,B,ELAM,ERR,K,SYM,SL4COF,SL4BCS,TOL,NMESH,IMATCH,
     &            NXTRAP,WORK,IWORK,WORKS,IFAIL)

      WRITE (6,FMT=*) 'The approximation to LAMBDA(1) was',ELAM
      WRITE (6,FMT=*) 'and the error estimate was',ERR
      WRITE (6,FMT=*) 'The number of mesh-points was',NMESH

      HOT = .FALSE.
```

```
C Compute the eigenfunction at 1000 equally-spaced points and write out
C to channel 7:
      DO 20 I = 0,999
          X = DBLE(I)/10.D0
          IFAIL = 1
          CALL SL4EFN(X,Y1,Y2,HOT,WORK,IWORK,NMESH,IMATCH,ELAM,MULTI,
     &                WORKS,YSTOR1,YSTOR2,SL4BCS,IFAIL)
          WRITE (7,FMT=*) X,Y1(1)
          IF (IFAIL.NE.0) GO TO 30
   20 CONTINUE

      STOP
      END


      SUBROUTINE SL4COF(X,P,S,Q,W)

C     .. Scalar Arguments ..
      DOUBLE PRECISION P,Q,S,W,X
C     ..

C Coefficients which define the problem:

      P = 1.D0
      S = X + X
      Q = X*X

      W = 1.D0
      RETURN
      END


      SUBROUTINE SL4BCS(IEND,ISING,X,U,V,ELAM)

C     .. Scalar Arguments ..
      DOUBLE PRECISION ELAM,X
      INTEGER IEND
      LOGICAL ISING
C     ..
C     .. Array Arguments ..
      DOUBLE PRECISION U(2,2),V(2,2)
C     ..

C Boundary conditions:

      U(1,1) = 1.D0
      U(2,1) = 0.D0
      U(1,2) = 0.D0
      U(2,2) = 0.D0
      V(1,1) = -1.D0
      V(2,1) = 0.D0
      V(1,2) = 0.D0
```

```
V(2,2) = 1.D0

RETURN
END
```

9.2 Results

```
The approximation to LAMBDA(1) was   10.55078883458672
and the error estimate was  1.7204361741096452E-06
The number of mesh-points was   320
```

References

[1] L. Greenberg and M. Marletta, *Oscillation Theory and Numerical Solution of Fourth Order Sturm-Liouville Problems*, IMA Jour. Num. Anal. 15 (1995), 319-356.

[2] L. Greenberg and M. Marletta, *The code SLEUTH for solving fourth order Sturm-Liouville problems*, ACM Trans. Math. Software 23 (1997), 453-493.