SLEUTH – FORTRAN 77 Library Routine Document

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

1 Purpose

SLEUTH finds a specified eigenvalue of a regular fourth order Sturm-Liouville problem in which the self-adjoint differential equation is cast in a canonical form. The routine uses piecewise constant approximation of the coefficients in the differential equation, coupled with a shooting method based on matrix oscillation theory and Richardson extrapolation to improve the accuracy.

2 Specification

```
 SUBROUTINE SLEUTH(A,B,ELAM,ERR,K,SYM,SL4COF,SL4BCS,TOL,NMESH,
&                 IMATCH,NXTRAP,WORK,IWORK,WORKS,IFAIL)
 INTEGER K,NMESH,IMATCH,NXTRAP,IWORK,IFAIL
 DOUBLE PRECISION A,B,ELAM,ERR,TOL,WORK(0:IWORK,1:6),
&                WORKS(1136)
 LOGICAL SYM
 EXTERNAL SL4COF,SL4BCS
```

3 Description

SLEUTH finds a specified eigenvalue λ_k of an eigenvalue problem consisting of a fourth-order Sturm-Liouville equation

$$(p(x)y'')'' - (s(x)y')' + q(x)y = \lambda w(x)y, \quad x \in [a,b], \tag{1}$$

together with boundary conditions which we shall specify. The functions p and w must be positive throughout the interval $[a,b]$, and all the functions p, s, q and w must be at least piecewise continuous throughout $[a,b]$, although in practice greater smoothness will usually be needed if the routine is to perform well.

To formulate the boundary conditions we introduce vectors u and v defined by

$$u(x) = \left(\; y(x) y'(x) \; \right), \quad v(x) = \left(\; -(p(x)y'')' + s(x)y' p(x)y'' \; \right).$$

The boundary conditions have the form

$$A_1^T u(a) + A_2^T v(a) = 0, \tag{2}$$

$$B_1^T u(b) + B_2^T v(b) = 0, \tag{3}$$

where A_1, A_2, B_1 and B_2 are 2×2 matrices such that $A_1^T A_2$ and $B_1^T B_2$ are symmetric, and such that $A_1 + iA_2$ and $B_1 + iB_2$ are non-singular.

Any self-adjoint fourth order problem with separated boundary conditions can be cast in this form. The standard spectral theory of these problems guarantees the existence of a countably infinite set of real eigenvalues, bounded below by a least eigenvalue λ_0 and increasing to infinity:

$$\lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k \leq \cdots. \tag{4}$$

Each eigenvalue has at most two linearly independent associated eigenfunctions; in other words, its eigenspace has dimension at most two. The dimension of the eigenspace of an eigenvalue is called the multiplicity of that eigenvalue. Each eigenvalue appears in the elencation (4) a number of times equal to its multiplicity. Thus for each k we have $\lambda_k \leq \lambda_{k+1}$ (soft inequality) and $\lambda_k < \lambda_{k+2}$ (strict inequality). For a more detailed discussion of the spectral theory of fourth order problems and of the methods used by this code, see [**?**].

4 References

See the bibliography at the end of this document.

5 Parameters

A – **real** scalar (input).
On entry: A must specify the left-hand endpoint a of the interval on which the problem is posed. Unchanged on exit.

B – **real** scalar (input).
On entry: B must specify the right-hand endpoint b of the interval on which the problem is posed. Unchanged on exit.

ELAM – **real** scalar (input/output).

On entry: ELAM must specify an initial approximation to the eigenvalue λ_k which is sought. This need not be at all accurate since the routine will always find an approximation to λ_k even if this is not the eigenvalue closest to the user's initial approximation. In case of doubt, set ELAM $= 0.D0$.

On exit: ELAM will contain the final approximation to λ_k, unless an error exit occurs.

ERR – **real** scalar (input/output).

On entry: ERR must be set to a **strictly positive** order-of-maginitude estimate of the error in the initial estimate ELAM. Over-estimates are definitely preferable to under-estimates. In case of doubt, set ERR $= 1.D0$.

On exit: ERR gives an assessment of the error in the returned value of ELAM, unless an error exit occurs.

K – INTEGER scalar (input)
On entry: K must specify the index $k \geq 0$ of the eigenvalue sought. Unchanged on exit.

SYM – LOGICAL (input)
On entry: SYM should be set to .FALSE. unless the eigenvalue problem is symmetric about the midpoint $(a+b)/2$, in which case SYM may be set to .TRUE. to save computations. Unchanged on exit.

SL4COF – SUBROUTINE, supplied by the user. SL4COF must supply the values of the coefficients in (1). The specification of SL4COF is as follows.

```
SUBROUTINE SL4COF(X,P,S,Q,W)
DOUBLE PRECISION X,P,S,Q,W
```

X – **real** (input).

On entry: X contains the value of a point in $[a, b]$. X must not be changed by SL4COF.

P – **real** scalar (output).

On exit: P must contain the value of p at the point X.

S – **real** scalar (output).

On exit: S must contain the value of s at the point X.

Q – **real** scalar (output).

On exit: Q must contain the value of q at the point X.

W – **real** scalar (output).

On exit: W must contain the value of w at the point X.

SL4BCS – SUBROUTINE, supplied by the user. This subroutine supplies the boundary conditions for the problem as specified by equations (2,3). The specification of SL4BCS is as follows.

```
SUBROUTINE SL4BCS(IEND,ISING,X,U,V,ELAM)
INTEGER IEND
DOUBLE PRECISION X,U(2,2),V(2,2),ELAM
LOGICAL ISING
```

IEND – INTEGER (input).

On entry: IEND specifies the end of the interval $[a, b]$ at which the boundary conditions are to be imposed. If IEND = 0 then boundary conditions are required at the end $x = a$; if IEND = 1 then boundary conditions are required at $x = b$. IEND must not be changed by the routine SL4BCS.

ISING – LOGICAL (output).

On exit: ISING must have the value .FALSE. This parameter is included to allow for a future upgrade of SLEUTH to handle singular problems.

X – **real** scalar (input).

On entry: X specifies the actual endpoint at which the boundary condition is required (i.e. the value A or B). Like ISING, this parameter is included to allow for a future upgrade of SLEUTH to handle singular problems.

U – **real** ARRAY of DIMENSION (2,2) (output).

On exit: U must specify the matrix A_2 if IEND=0 or B_2 if IEND=1.

V – **real** ARRAY of DIMENSION (2,2) (output).

On exit: V must specify the matrix $-A_1$ if IEND=0 or $-B_1$ if IEND=1.

ELAM – **real** scalar (input).

On entry: ELAM specifies the current value of λ, allowing for λ-dependent boundary conditions. The value of ELAM must not be changed by SL4BCS. Note that the standard spectral theory described in Section 3 may not apply if the boundary conditions are λ-dependent.

TOL – **real** scalar (input).

On entry: TOL specifies a target accuracy for the eigenvalue approximation. SLEUTH will attempt to ensure that

$$|ELAM - \lambda_k| \leq TOL. \max(1, |ELAM|).$$

Note that because of the use of Richardson extrapolation, small reductions in TOL may cause large increases in the required computational effort. This is because the amount of computation required is determined largely by the number of extrapolations required, and a small reduction in TOL may necessitate an additional extrapolation to achieve the target accuracy.

NMESH – INTEGER (input/output).

On entry: NMESH specifies the maximum number of mesh-intervals to be used by SLEUTH. Normally, NMESH should have the same value as IWORK (see below). However the user may decide to make NMESH smaller than IWORK to protect against excessive run-times. NMESH should never be set greater than IWORK on entry.

On exit: the actual number of mesh-intervals in the finest mesh used by SLEUTH.

IMATCH – INTEGER (output).

On exit: The index of the mesh-point used as a matching point by the shooting algorithm. Returned for later use by the eigenfunction routine, when the latter is prepared.

NXTRAP – INTEGER (output).

On exit: The number of Richardson extrapolations carried out by SLEUTH (always at least 3).

WORK – **real** array of DIMENSION (0:IWORK,1:6) (workspace).

IWORK – INTEGER (input).

On entry: IWORK must specify the first dimension of WORK as declared in the calling (sub)program; IWORK $\geq$ 20480 in the current implementation.

WORKS – **real** array of DIMENSION (1136) (workspace).

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 6).

6 Error Indicators and Warnings

Errors detected by the routine:

IFAIL = 1. A parameter error on input. This error may be trigerred by all of the following: TOL less than machine precision to the power 0.6, A greater than or equal to B, IFAIL greater than 1 in absolute value, ERR equal to zero in absolute value, K less than zero, IWORK less than 40. If IFAIL was -1 or 0 on entry then an error message will give more details.

IFAIL = 2. An error has occurred while attempting to compute the 'correction parameter' σ at the matchpoint. Check the boundary conditions, although this is more likely to indicate a serious internal error in SLEUTH.

IFAIL = 3. The matrices U and V (see [?]), at the matchpoint, have determinants which indicate either that a serious overflow has occurred or else that the boundary conditions were not set up correctly.

IFAIL = 4. An error has occurred while finding eigenvalues of a Θ-matrix (see [?]) at the matchpoint. The same remarks apply as for IFAIL = 2.

IFAIL = 5. Not currently used.

IFAIL = 6. Indicates the same problem as for IFAIL = 2, but at a general mesh point. This is most likely to be caused by incorrect boundary conditions.

IFAIL = 7. Indicates the same problem as for IFAIL = 3, but at a general mesh point. This is most likely to be caused by incorrect boundary conditions.

IFAIL = 8. The eigenvalue approximations are not converging as the mesh is refined. This may be an indication that the coefficients in the differential equation have been coded incorrectly.

IFAIL = 9. No interval $[\lambda_{min}, \lambda_{max}]$ can be found which contains the eigenvalue with the required index. This usually means that ERR has been set too small on entry.

IFAIL = 10. An eigenvalue has been bracketed but the rootfinding process cannot locate it with sufficient accuracy. This may be an indication that the eigenvalue is one of a close cluster and that C05AZF is rootfinding, not very effectively, on a function which is almost discontinuous.

IFAIL = 11. To achieve the target accuracy, NMESH needs to be at least twice as large as the input value.

7 Accuracy

The error assessment ERR is generally a very pessimistic indicator of the size of the error in the final eigenvalue approximation.

8 Run times

Run times depend mainly on the number of extrapolations NXTRAP required to achieve the target accuracy; they may therefore increase suddenly with a small reduction in TOL, if the reduction necessitates an additional extrapolation to achieve the required accuracy.

9 Keywords

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

10 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

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

10.1 Example Program Text

```
      PROGRAM main
      INTEGER IWORK
      PARAMETER (IWORK=20480)
      INTEGER K,NMESH,IMATCH,NXTRAP,IFAIL
      DOUBLE PRECISION A,B,ELAM,ERR,TOL,WORK(0:IWORK,1:6),
     & WORKS(1136)
      LOGICAL SYM
      EXTERNAL SL4COF,SL4BCS

      A = 0.D0
      B = 100.D0
      ELAM = 0.D0
      ERR = 1.D0
      K = 1
      SYM = .FALSE.
      TOL = 1.0D-6
      NMESH = IWORK
      IFAIL = 0

      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

      STOP
      END


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

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

      W = 1.D0
      RETURN
      END


      SUBROUTINE SL4BCS(IEND,ISING,X,U,V,ELAM)
      INTEGER IEND
      DOUBLE PRECISION X,U(2,2),V(2,2),ELAM
      LOGICAL ISING
```

```
      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
```

10.2 Results

```
 The approximation to LAMBDA(1) was   10.55078883458672
 and the error estimate was  1.7204361741096452E-06
```

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.