A Collocation/Quadrature-Based Sturm-Liouville Problem Solver

by Raymond A. Adomaitis, Yi-hung Lin

T.R. 99-01

A Collocation/Quadrature-Based Sturm-Liouville Problem Solver

Raymond A. Adomaitis* and Yi-hung Lin
Department of Chemical Engineering and Institute for Systems Research
University of Maryland
College Park, MD 20742

January 4, 1999

Keywords: Sturm-Liouville problems; collocation; quadrature; eigenfunction expansions; computational methods.

Abstract

We present a computational method for solving a class of boundary-value problems in Sturm-Liouville form. The algorithms are based on global polynomial collocation methods and produce discrete representations of the eigenfunctions. Error control is performed by evaluating the eigenvalue problem residuals generated when the eigenfunctions are interpolated to a finer discretization grid; eigenfunctions that produce residuals exceeding an infinity-norm bound are discarded. Because the computational approach involves the generation of quadrature weights and discrete differentiation operations, our computational methods provide a convenient framework for solving boundary-value problems by eigenfunction expansion and other projection methods.

1 Introduction

Eigenfunction expansions are used to solve time-dependent, linear, nonhomogeneous boundary value problems. The method is simple and intuitive to implement, and the eigenfunction and eigenvalues frequently have physically meaningful interpretations. Eigenfunctions also provide a natural basis for trial function expansions when used in conjunction with the Galerkin and other projection methods to solve nonlinear problems. This approach can be further extended to include problems defined in complex, two- and three-dimensional physical domains when the eigenfunctions are computed numerically (see, e.g., [1]). In this paper, we present a set of computational procedures for generating eigenvalues and discrete representations of eigenfunctions, and discuss the relationship between these computational procedures and the implementation of eigenfunction expansion and other projection methods.
The class of second-order, regular, Sturm-Liouville problems to be solved is defined by

$$\frac{1}{x^\alpha v(x)}\frac{d}{dx}\left(x^\alpha p(x)\frac{d\psi}{dx}\right)+q(x)\frac{d\psi}{dx}+g(x)\psi=\lambda\psi \qquad x\in(0,1). \tag{1}$$

The solutions $\psi(x)$ are subject to boundary conditions

$$\begin{aligned} a\frac{d\psi(0)}{dx}+b\psi(0)+a_1\frac{d\psi(1)}{dx}+b_1\psi(1) &= 0,\\ c\frac{d\psi(1)}{dx}+d\psi(1)+c_0\frac{d\psi(0)}{dx}+d_0\psi(0) &= 0. \end{aligned} \tag{2}$$

*Corresponding author, E-mail adomaiti@isr.umd.edu, Ph: 301-405-2969, Fax: (301) 314-9920

Therefore, Dirichlet, Neumann, mixed, periodic, and semiperiodic boundary conditions can be satisfied. The eigenvalue problems are considered regular [2] in the unit interval if

1. $p(x) > 0$, $v(x) > 0$, and $g(x) \leq 0$ for $x \in (0, 1)$;
2. $1/p(x)$, $q(x)$, and $v(x)$ are locally integrable near the endpoints.

We will consider cases where $v(x)$ has one or more finite jump discontinuities.

The format of the Sturm-Liouville problems to be solved (1) was chosen to reflect the form commonly encountered when solving boundary-value problems generated from conservation equations describing the transport of mass, momentum, or energy in engineering modeling problems. In this form, $p(x)$ can represent a spatially-dependent diffusivity, $q(x)$ the contribution of a convective flux, $g(x)$ a spatially nonuniform potential or heat transfer rate, and $v(x)$ a nonuniform capacitive term. The exponent α is related to the problem symmetry: $\alpha = 0$ for slabs, $\alpha = 1$ for cylindrical geometries, and $\alpha = 2$ for spherical geometries. Specific examples will be presented later in the paper.

Considerable research has been devoted to the numerical solution of second-order Sturm-Liouville problems (e.g., [2, 3, 4, 5]). The focus of this paper is development of a simple-to-use, Matlab-based Sturm-Liouville problem solver in the form of computational modules that specifically take into account the interplay between eigenfunction generation and implementation of the eigenfunction expansion methods. Our approach was motivated by the need for computational methods that produce discretized eigenfunctions, adjoint eigenfunctions, eigenvalues, and discretized representations of the weight functions defining (adjoint)eigenfunction orthogonality to simplify implementation of weighted residual methods in engineering system simulation and control applications.

2 Discretized Function Representation

Our numerical approach is based on representing the eigenfunctions in spatially discretized form, using Lagrangian interpolation for computing inter-point eigenfunction values. This approach provides flexibility in representing eigenfunctions that can take polynomial, trigonometric, Bessel, or other special function form. Furthermore, if the M-point discretized set of N eigenfunctions is written as the array $\mathbf{\Psi}^{M\times N}$, the discrete transformation array $\mathbf{\Psi}$ will be composed of columns corresponding to each eigenfunction $\psi_n(x)$ evaluated at the discretization points x_m. This simplifies reconstruction of discretized state variables approximated by eigenfunction expansions, i.e., if the trial function expansion

$$u^N(x,t) = \sum_{n=1}^{N} a_n(t)\psi_n(x)$$

is used to approximate the state variable $u(x,t)$, the discretized equivalent reduces to simple matrix multiplication $\mathbf{u}^{M\times 1} = \mathbf{\Psi}^{M\times N}\mathbf{a}^{N\times 1}$, where $u_m(t) = u(x_m, t)$.

Eigenfunctions computed as the nontrivial solutions to (1) satisfying boundary conditions (2) will be orthogonal with respect to weighted inner product

$$\langle \psi_i(x), \psi_j(x) \rangle_{\mathcal{W}} = \int_0^1 \psi_i(x)\psi_j(x)\,\mathcal{W}(x)x^\alpha\,dx$$

with

$$\mathcal{W}(x) = v(x)\exp\left\{\int_0^x \left[v(x')q(x')/p(x')\right]\,dx'\right\}.$$

Therefore, the projection of the discretized state variable $u(x,t)$ onto the eigenfunctions is computed as the matrix operation

$$a_n = \sum_{m=1}^{M} u_m \Psi_{m,n}\, w_m W_m, \quad n = 1, \ldots, N \qquad \text{or} \qquad \mathbf{a} = \mathbf{Pu}$$

with $P_{i,j} = \Psi_{i,j}\, w_i W_i$. The quadrature weights, computed from the term-by-term product $w_m W_m$, are composed of the discretized weight function W_m defining the eigenfunction orthogonality and the quadrature weights w_m representing the discrete approximation to the differential element $x^\alpha dx$. We do not include the geometry factor x^α in the definition of $\mathcal{W}$ since the former is included in the quadrature weight array $\mathbf{w}$ discussed in the following section.

3 Collocation Points and Quadrature Weights

We have chosen to fix the discretization end point locations as $x_1 = 0$ and $x_M = 1$, and set the interior points as the roots of the $(M-2)$-degree Jacobi polynomial $J^{(1,\alpha+1)}_{(M-2)}(x)$, where α is the geometry factor in (1). The roots are distinct, are all located inside the unit interval [6], and are spaced in such a manner that the discretization point density increases towards each end of the interval. This choice of discretization points forces the residual generated by the projection of an arbitrary function onto the collocation points to have as its primary component the function $J^{(1,\alpha+1)}_{(M-2)}(x)$, giving an excellent approximation to a true projection onto the first $(M-3)$-degree Jacobi polynomials [7]. This choice also guarantees that the quadrature weights $\mathbf{w}$, used to compute

$$\int_0^1 f(x)\, x^\alpha dx = \mathbf{w}^T \mathbf{f}$$

where $\mathbf{f} = [f(x_1), \ldots, f(x_M)]^T$, result in exact integral evaluations when the degree of polynomial $f(x)$ is less than $q = 2M-3$, where M is the total number of collocation points, including the endpoints.

Accurate and efficient algorithms for determining the Jacobi polynomial roots and the associated quadrature weights $\mathbf{w}$ have been developed (e.g., [8, 9]); however, these algorithms can be modified to improve their numerical accuracy and efficiency, and to take advantage of vectorized computational operations [10]. In the first step of our algorithm, points are placed in the unit interval at locations that approximate the spacing of Jacobi polynomial roots of degree significantly higher than M (we use the extrema of a $(3M-1)$-degree Chebyshev polynomial because they can be computed explicitly). Then, $J^{(1,\alpha+1)}_{(M-2)}(x)$ is evaluated at these points using the recursive formula that generates these polynomials; the subintervals of different signs are identified and linear interpolation is used to determine zero-crossing estimates. The polynomial derivatives are computed at these estimated root locations and are used as part of the first step of the Newton-Raphson iterations used to compute the root location vector $\mathbf{x}$.

The Gauss-Lobatto quadrature method generates quadrature weights $\mathbf{w}$ defined by

$$w_i = \frac{C_M K_i}{\left[\dfrac{dP_M(x_i)}{dx}\right]^2}$$

where $P_M(x) = \prod_{j=1}^M (x - x_j)$, $K_i = 1/(\alpha+1)$ for $i = 1$ and $K_i = 1$ for $i \neq 1$. The coefficient C_M is computed indirectly from the known value of the sum of the quadrature weights to find

$$C_M = \frac{1}{(\alpha+1)\sum_{j=1}^M K_j/(dP_M(x_j)/dx)^2}.$$

This avoids computing C_M directly from its definition [8, 11], which can be difficult for large M. Details regarding the quadrature weight formula derivations and numerical methods developed to overcome additional computational limitations can be found in [10]. We have found that accurate computations can be performed using over $M = 500$ discretization points.

4 Discrete Differentiation Operations

Having defined the discretized function representations and derivation of the quadrature arrays, we differentiate the interpolation polynomials explicitly to obtain accurate, discrete, differentiation operations. The eigenfunctions are represented in discretized form in terms of the trial function expansion

$$\psi_n(x) = \sum_{m=1}^{M} \Psi_{m,n} l_m(x)$$

where the $l_m(x)$ are the building block polynomials of the Lagrange interpolation polynomial

$$l_m(x) = \prod_{\substack{j=1 \\ j \neq m}}^{M} \frac{(x - x_j)}{(x_m - x_j)}, \qquad m = 1, \ldots, M. \tag{3}$$

Because the Lagrangian interpolation polynomials $l_i(x)$ are continuous and differentiable, explicit formulas for derivatives of functions represented by these trial function expansions up to order $(M-1)$ can be obtained that are valid at all x_m, $m = 1, \ldots, M$.

In the discrete-ordinate formulation of the Lagrange interpolation function-based collocation method, differentiation is a matrix operation, i.e., $d\boldsymbol{\psi}_n/dx = \mathbf{A}\boldsymbol{\psi}_n$ where the elements of the matrix are defined by $A_{i,j} = dl_j(x_i)/dx$, $i, j = 1, 2, \cdots, M$. The terms in each array are computed by directly differentiating the interpolation equations, and the values of the nodal polynomial derivatives at the M discretization points $\mathbf{x}$ are computed with the recurrence formulas derived by Michelsen and Villadsen [8].

As an alternative to direct computation of the discretized Laplacian operator $\mathbf{B}$ from the interpolation formulas (using asymptotic values for the operator at the origin for cylindrical and spherical geometries to eliminate the numerical singularity caused by the $1/x^\alpha$ term), the discretized operator can be constructed from matrix operations involving $\mathbf{A}$. In fact, the Laplacian operator in slab geometries is simply the product $\mathbf{B} = \mathbf{A}\mathbf{A}$ to within truncation errors in the computational procedures. This is made possible by the polynomial interpolation: because any $(M-1)$-degree polynomial can be represented exactly in terms of the discretized interpolation functions, its derivatives will exist entirely in the space spanned by the first M Jacobi polynomials, therefore, no accuracy is lost in the discrete-ordinate formulation of the collocation method.

5 Eigenfunction and Eigenvalue Computations

The λ_n and ψ_n are computed as the eigenvalues and eigenvectors of an array defined by the discretized form of (1). In the first step of this procedure, the discretized functions $\mathbf{v}$, $\mathbf{p}$, $\mathbf{q}$, and $\mathbf{g}$, together with the first-order differentiation array $\mathbf{A}$ are used to discretize (1) to obtain

$$\frac{p_m}{v_m}\mathbf{A}_m\mathbf{A} + \left(\frac{\mathbf{A}_m\mathbf{p}}{v_m} + q_m\right)\mathbf{A}_m + \frac{\alpha p_m}{x_m v_m}\mathbf{A}_m + g_m\mathbf{I}_m = \mathbf{L}^{(M-2)\times M} \tag{4}$$

for $m = 2, \ldots, M-1$, where $\mathbf{A}_m$ denotes the mth row of array $\mathbf{A}$, and $\mathbf{I}$ is the identity matrix. We note that the singularity at the origin for $\alpha \neq 0$ does not affect (4).

The boundary conditions then are used to define a linear relationship between the values of $\psi(x)$ at the interval endpoints

$$\begin{aligned} (a\mathbf{A}_1 + b\mathbf{I}_1 + a_1\mathbf{A}_M + b_1\mathbf{I}_M)\boldsymbol{\psi} &= 0, & (5)\\ (c\mathbf{A}_M + d\mathbf{I}_M + c_0\mathbf{A}_1 + d_0\mathbf{I}_1)\boldsymbol{\psi} &= 0. & (6) \end{aligned}$$

These equations are solved for $\psi(x_1)$ and $\psi(x_M)$ in terms of the interior node values and this relationship is used to transform (4) into homogeneous form $\mathbf{J}^{(M-2)\times(M-2)}$. After computing the $M-2$ eigenvalues and eigenvectors of $\mathbf{J}$, the endpoint values of the discretized eigenfunctions that satisfy the boundary conditions are computed using (5,6). This gives the matrix of discretized eigenfunctions $\bar{\boldsymbol{\Psi}}^{M\times(M-2)}$, adjoint eigenfunctions $\bar{\boldsymbol{\Phi}}^{M\times(M-2)}$, and eigenvalues $\boldsymbol{\Lambda}^{(M-2)\times(M-2)}$.

These discretized eigenfunctions will be orthogonal with respect to a properly weighted inner product; to normalized the eigenfunctions, the weight $\mathbf{W}$ is determined by first computing M quadrature weights $\tilde{\mathbf{w}}$ and points $\tilde{\mathbf{x}}$ corresponding to the geometry factor $\alpha = 0$. Then, the weight functions at $x = 0$ are set to $W_1 = v(x_1)$ and $W_{ad_1} = 1/v(x_1)$ and the following, two-step procedure is used to compute the remaining, discretized weight function values. In the first step, the integrand $v(x)q(x)/p(x)$ defined over each subinterval $0 \le x \le x_m$ is interpolated to the points $x_m\tilde{\mathbf{x}}$. The discrete weight functions are computed in the second step of this process as

$$W_m = v_m \exp\left\{\sum_{k=1}^{M} \tilde{w}_k \tilde{v}_k \tilde{q}_k / \tilde{p}_k\right\},$$

and

$$W_{ad_m} = \exp\left\{-\sum_{k=1}^{M} \tilde{w}_k \tilde{v}_k \tilde{q}_k / \tilde{p}_k\right\}/v_m.$$

This two-step procedure is repeated for each point on the original grid x_m. We note that both $W_m = v(x_m)$ and $W_{ad_m} = 1/v(x_m)$, $m = 1, \ldots, M$ if $q(x) = 0$. The discretized weight function $\mathbf{W_{ad}}$ is not computed if $v_m = 0$ for any m.

The normalized eigenfunctions are computed from

$$\mathbf{\Psi}_m = \bar{\mathbf{\Psi}}_m \left[\sum_{k=1}^{M} \bar{\Psi}^2_{k,m}\, w_k W_k\right]^{-1/2}$$

where $\mathbf{\Psi}_m$ is the mth (column) discretized eigenfunction. The discretized adjoint eigenfunctions are computed as the term-by-term product of each discretized eigenfunction $\mathbf{\Psi}_m$ with weight function $\mathbf{W}$, and are normalized using $\mathbf{W_{ad}}$.

5.1 Error control

After the eigenvalues and discretized eigenfunctions are computed, a grid $\hat{\mathbf{x}}$ twice as fine as the original grid $\mathbf{x}$ is computed to estimate the the residuals produced when the eigenfunctions ψ_m are substituted into the eigenvalue problem (1), now discretized on the finer grid (c.f., [12]). The eigenfunctions $\mathbf{\Psi}$ are interpolated to this finer grid, using the Lagrange interpolation functions. Because it is computationally difficult to use the interpolation polynomials directly in cases of high-degree discretizations, we use Neville's algorithm [13] and other computational techniques that avoid direct evaluation of high-degree Jacobi polynomials for interpolation [10].

After the discrete first-order differentiation array $\hat{\mathbf{A}}$ and quadrature weights $\hat{\mathbf{w}}$ are computed for the finer grid, we discretize the original eigenvalue problem (1) to obtain a discretized operator similar to (4). We note that while $\hat{\mathbf{L}}$ is not defined at the interval end points, it is a function of the eigenfunctions evaluated at the endpoints.

We then compute the normalized infinity norm of the residual function defined by the finely-discretized eigenvalue problem

$$R_m = \frac{\|\hat{\mathbf{L}}\hat{\boldsymbol{\psi}}_m - \lambda_m \hat{\boldsymbol{\psi}}_m\|_\infty}{\|\hat{\mathbf{L}}\hat{\boldsymbol{\psi}}_m - \lambda_m \hat{\boldsymbol{\psi}}_m\|_1}.$$

A search is performed over the indices $1 \ge m \ge M-2$ using interval bisection to determine the eigenfunctions that exceed an error condition defined by $R_m < errtol$; typically, we use $errtol = 0.001$ and find that roughly half of the eigenfunctions fail this test. All of the rejected eigenfunctions correspond to the largest-magnitude eigenvalues.

6 Computational Examples

We have implemented the collocation-based Sturm-Liouville problem solving procedure as part of a Matlab-based set of computational tools for solving boundary-value problems by the method of weighted residuals [10]. The eigenvalue, eigenfunction, and weight arrays are generated by the function

$[\boldsymbol{\lambda}, \boldsymbol{\Psi}, \boldsymbol{\Phi}, \mathbf{W}, \mathbf{W_{ad}}] = \texttt{sl}(\text{'geom'}, \mathbf{A},\mathbf{x}, a,b,c,d,\mathbf{w},\mathbf{v},\mathbf{p},\mathbf{q},\mathbf{g}, errtol,a_1,b_1,c_0,d_0)$.

The output consists of the eigenvalues $\boldsymbol{\lambda}$, discretized, orthonormal eigenfunctions $\boldsymbol{\Psi}$, adjoint eigenfunctions $\boldsymbol{\Phi}$, discretized weight function $\mathbf{W}$ used to define the orthogonality of the eigenfunctions, and discretized weight function $\mathbf{W_{ad}}$ corresponding to the adjoint eigenfunctions. Input includes the problem geometry factor "geom" (geom='slab' corresponds to $\alpha = 0$; geom='disk' to $\alpha = 1$; and geom='sphe' to $\alpha = 2$), and the differentiation, collocation point, and quadrature arrays, $\mathbf{A}$, $\mathbf{x}$, and $\mathbf{w}$, respectively. The remaining parameters correspond directly to the terms in (1) and (2) or should be otherwise self-explanatory.
The differentiation, collocation point, and quadrature arrays are computed with the physical-domain definition function

$[\mathbf{x}, \mathbf{w}, \mathbf{A},\mathbf{B}, \mathbf{Q}] = \texttt{pd}(\text{'geom'},M)$.

This function also computes the discretized set of M Jacobi polynomials $\mathbf{Q}$, a discrete transformation array used for spectral filtering in one of the examples presented in this paper. We chose to split the functions in this manner since the quadrature and differentiation arrays are used in other operations during an eigenfunction expansion solution procedure, and the discretization points are required for plotting results and discretizing the functions v, p, q, and g.
The following examples either compare computed eigenfunctions and eigenvalues to results that can be evaluated exactly, or compare computed results with previously published solutions. In the latter cases, the accuracy of most of our results match previously published results to the reported accuracy.

6.1 The Sinc Functions

We consider the problem of computing nontrivial solutions to

$$\frac{1}{x^\alpha}\frac{d}{dx}\left[x^\alpha \frac{d\psi}{dx}\right] = -\lambda^2\psi$$

subject to $d\psi(0)/dx = 0$ and either $\psi(1) = 0$ or $d\psi(1)/dx = 0$. For the case $\alpha = 2$, the numerically computed eigenfunctions can be compared to the exact eigenfunctions (consisting of the spherical Bessel functions of the first kind of order 0)

$$\psi_m(x) = \frac{\sin \lambda_m x}{x} C_m$$

and eigenvalues $\lambda_m = m\pi$, with $C_m = \sqrt{2}$ for the Dirichlet problem. For the no-flux boundary condition case, the eigenvalues must be computed as solutions to $\lambda \cos\lambda - \sin\lambda = 0$ and the normalization coefficient is

$$C_m = \left[\frac{1}{2} - \frac{\sin 2\lambda_m}{4\lambda_m}\right]^{-1/2}.$$

Numerical computation of the eigenvalues and eigenfunctions begins with calling `pd.m` to define the physical domain discretization points, quadrature weights, and differentiation array. Then, the first call to `sl.m` shown below generates the solutions corresponding to the Dirichlet problem and the second, the Neumann problem:

```
[x,wx,dx] = pd('sphe',40);
[lamD psiD] = sl('sphe',dx,x,1,0,0,1,wx);
```

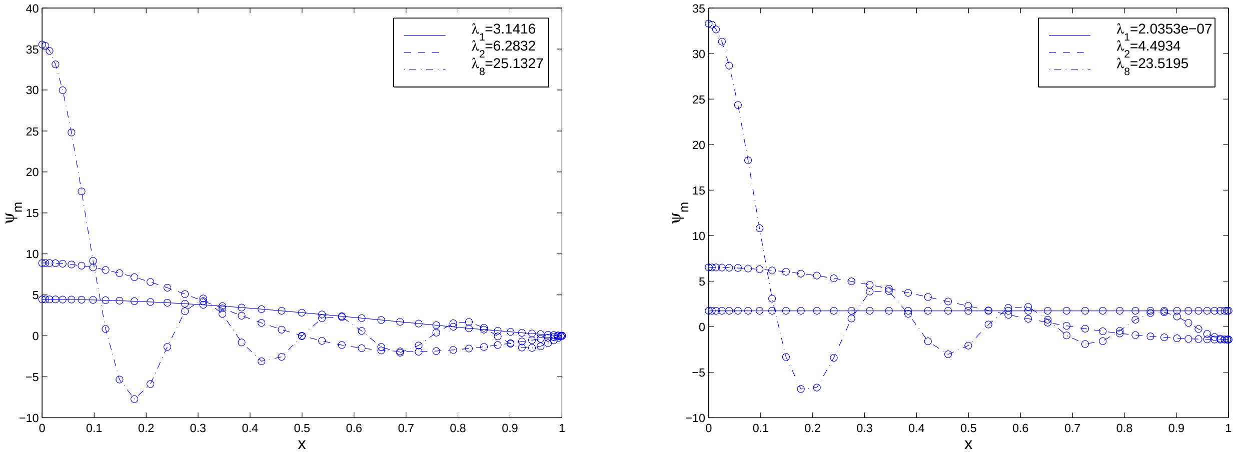

Figure 1: The eigenfunctions for Laplace's equation in a sphere for Dirichlet (left) and Neumann (right) boundary conditions; circles indicate points from the exact solutions. A total of 40 discretization points were used.

```
[lamN psiN] = sl('sphe',dx,x,1,0,1,0,wx);
```

For $M = 40$, the function `sl.m` produced 14 eigenfunctions for the Dirichlet problem with a maximum eigenvalue error magnitude of 1.37×10^{-8}; for the Neumann problem 16 eigenfunctions were produced using the default error tolerance setting. Representative eigenfunctions are plotted in Fig. 1 together with their respective eigenvalues.

6.2 The Graetz-Nusselt Problem

Heat transfer in a Newtonian fluid flowing through a circular tube with a laminar flow profile leads to the eigenvalue problem

$$\frac{1}{x}\frac{d}{dx}\left[x\frac{d\psi}{dx}\right] = -2\lambda(1-x^2)\psi$$

subject to $d\psi(0)/dx = 0$ and $\psi(1) = 0$. The eigenfunctions $\psi(x)$ represent those radial temperature profiles where the radial conductive heat transfer is balanced exactly by streamwise convection [7, 14]. In this problem, $\alpha = 1$ and $v(x) = -2(1-x^2)$ and the eigenfunctions will be orthogonal with respect to inner product weight function $x(1-x^2)$. We note that the singularity at $x = 1$ does not pose a problem because it is located at the boundary. The first three eigenvalues match previously reported results [7]; eigenfunctions and corresponding eigenvalues are presented in Fig. 2. We observe no visible or computational difficulties in the neighborhood of $x = 1$. The results shown in Fig. 2 were computed by

```
[x,wx,dx] = pd('disk',40);
v = -2*(1-x.^2);
[lam psi] = sl('disk',dx,x,1,0,0,1,wx,v);
```

6.3 Non Self-Adjoint Problems

When modeling the temperature profile of a steel sheet-forming process involving multiple-zone heating [15], the sheet temperature profile can be modeled by a boundary-value problem with terms accounting for

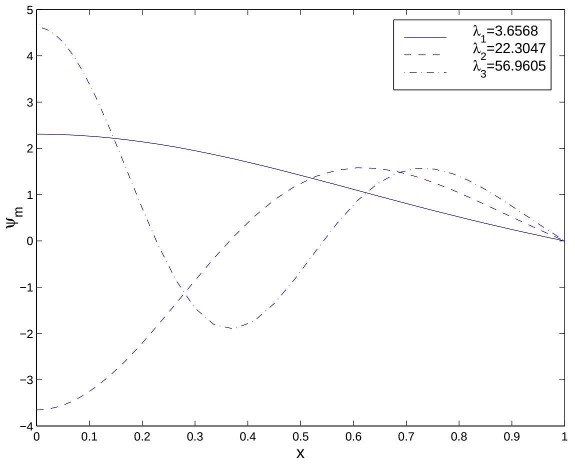

Figure 2: The eigenfunctions and corresponding eigenvalues of the Graetz-Nusselt problem for $M = 40$. Computed results agree with eigenvalues published in Villadsen and Stewart (1967) to the reported accuracy.

thermal conduction, radiant heating, conductive cooling to the surrounding gas phase, and the movement of the sheet relative to the heating lamps and rollers. The linear eigenvalue problem relevant to this simulation and control problem can be written as

$$\frac{d^2\psi}{dx^2} - q\frac{d\psi}{dx} - g\psi = \lambda\psi \qquad x \in (0,1)$$

subject to $d\psi(0)/dx = 0$ and $d\psi(1)/dx = 0$. We consider the case where q and g are constant over $x \in (0,1)$ with values chosen as $q = 10$ and $g = 20$; this creates a challenging computational problem because of the dominance of the convective term ($q\,d\psi/dx$) relative to thermal conduction. The eigenfunctions will be orthogonal with respect to weight $\mathcal{W} = e^{-qx}$; additional checks of the computed solutions are provided by the first eigenfunction $\psi_1 = \sqrt{10/(1-e^{-10})}$ and $\lambda_1 = -g = -20$. Computed results are shown in Fig. 3. For $M = 50$ and the default error tolerance setting, 22 eigenfunctions were produced.
The utility of the discretized weight arrays are demonstrated by evaluating the accuracy of the *array* of weighted inner product computations

$$I_{p1} = \langle \boldsymbol{\psi}, \boldsymbol{\psi} \rangle_{\mathcal{W}} \qquad I_{p2} = \langle \boldsymbol{\psi}, \boldsymbol{\phi} \rangle \qquad I_{p3} = \langle \boldsymbol{\phi}, \boldsymbol{\phi} \rangle_{\mathcal{W}_{ad}}$$

which would all give the identity matrix if performed exactly. The function `wip.m` is a weighted inner product subprogram described in [10].

```
[x,wx,dx] = pd(’slab’,50);
[lam psi phi W W_ad] = sl(’slab’,dx,x,1,0,1,0,wx,1,1,-10,-20);
Ip1 = wip(psi,psi,W.*wx);
Ip2 = wip(psi,phi,wx);
Ip3 = wip(phi,phi,W_ad.*wx);
```

In all of the cases above, we found the maximum magnitude off-diagonal element of the inner production computations was $I_{p_{i,j}} = 2.76 \times 10^{-5}$ which is sufficient for a typical process control simulation application. Further reductions in these errors can be produced by increasing M.

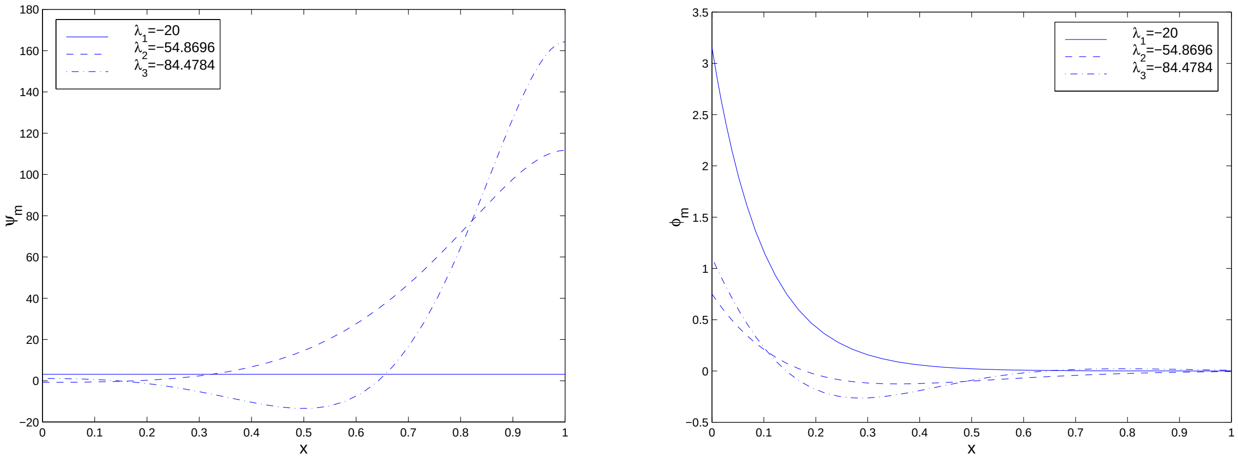

Figure 3: The eigenfunctions (left), adjoint eigenfunctions (right) and corresponding eigenvalues of the metal-slab heating problem for $M = 50$.

6.4 Periodic Boundary Conditions

In Vanden Berge and co-workers [5], modified finite difference methods were developed as an accurate solver of periodic and semi-periodic Sturm-Liouville problems. One test case considered in the cited work consisted of

$$\frac{1}{\pi^2}\frac{d^2\psi}{dx^2} - \pi^3 x^2(1-x)\psi = -\lambda\psi$$

(scaled to the unit interval) subject to the periodic boundary conditions $\psi(0) = \psi(1)$ and $d\psi(0)/dx = d\psi(1)/dx$. For the case of $M = 50$, we present our results in Fig. 4. These results are computed by the function calls

```
[x,wx,dx] = pd(’slab’,80);
v = -pi^2;  g = pi^3*x.^2.*(1-x);
[lam psi] = sl(’slab’,dx,x,0,1,1,0,wx,v,1,0,g,[],0,-1,-1,0);
```

The eigenvalues computed with our collocation algorithm match the nineteen eigenvalues listed in [5] exactly to the reported accuracy; additionally, we have found that relatively accurate eigenvalues can be computed with significantly fewer discretization points, e.g., $M = 40$.

6.5 Coffey-Evans Equation

As a more severe test of the computational techniques, we consider the Coffey-Evans equation used as a test case by Pruess and Fulton [2]

$$\frac{1}{\pi^2}\frac{d^2\psi}{dx^2} - (2\beta\cos 2\pi x + \beta^2\sin^2 2\pi x)\psi = -\lambda\psi$$

with $\psi(0) = \psi(1) = 0$ after shifting and scaling to the unit interval. Representative results are shown in Fig. 5 for $M = 160$. We note that this fine discretization level was required to resolve the eigenfunction details; converged eigenvalues matching the results published in [2] could be achieved with significantly fewer points.

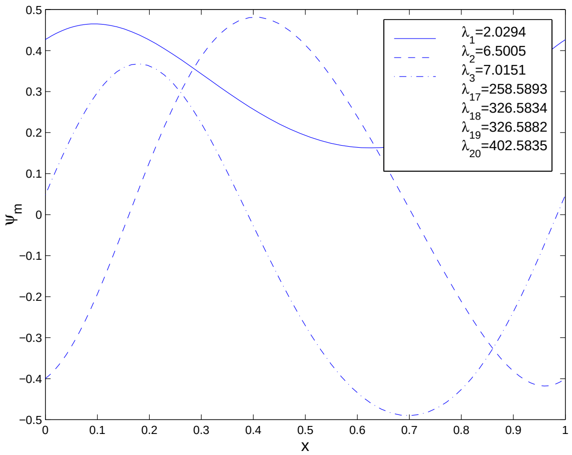

Figure 4: The eigenfunctions and corresponding eigenvalues of the periodic Sturm-Liouville problem test case for $M = 50$.

6.6 Square-Well Potential

As the final example, we consider the radial Schrödinger equation with a square-well potential function $V(x)$ in the unit interval

$$\frac{1}{4}\frac{d^2\psi}{dx^2} - V(x)\psi = -\lambda^2\psi$$

with $V(x) = 1000$ for $|x| > 1/4$ and $V(x) = 0$ otherwise. The boundary conditions are $\psi(0) = \psi(1) = 0$. This problem was devised to test the stiff boundary-value problem solver created by Lee and Greengard [16]. The discontinuous $V(x)$ presents the most severe computational challenge to our discretization method. For example, using $M = 60$, we see significant differences between the eigenvalues computed by our techniques (Fig. 6, left) compared to results published in [16], where $\lambda_{1,2,3} = 2.95446, 5.90736, 8.85702$ in the cited reference.

The oscillatory behavior of the eigenvalue convergence can be traced directly to the collocation approximation of the discontinuous potential function. The collocation-point spacing results in twice the number of collocation points in the region where $V(x) \neq 0$ relative to the central region where $V(x) = 0$. As collocation points are added, two will fall into the $V \neq 0$ region resulting in eigenvalues larger in magnitude than the converged values, followed by a point falling into the central region, resulting in computed eigenvalues smaller in magnitude than the converged values.

The convergence behavior can be improved using a spectral filtering technique [17]. To implement this method, we first project the discontinuous potential function onto the Jacobi polynomials orthogonal to the polynomial defining the roots $\hat{\mathbf{x}}$ (used in the eigenfunction error control step); a second-order Fourier-space filter is then used to smooth the interpolated potential function, and this smoothed function is then interpolated to the original discretization grid $\mathbf{x}$. As seen by the solid curves of Fig. 6 (right) the oscillations in the eigenvalue plots are diminished considerably, and computational experiments show that the eigenvalues converge to the values reported in [16].

7 Conclusions

One of the factors that motivated developing the numerical techniques discussed in this paper was to have available a set of computational modules that simplify the implementation of eigenfunction expansion and other projection methods applied to a range of engineering simulation problems. This approach consists of using the subprograms described in this paper as a simple and flexible method for generating

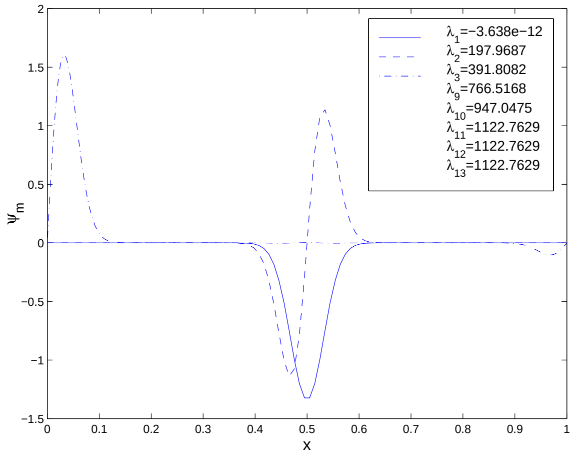

Figure 5: The eigenfunctions and corresponding eigenvalues of the Coffey-Evans problem for $M = 160$. Additional eigenvalues are shown, including one of the triplets characteristic of this problem.

spatially discretized representations of eigenfunctions to be used as the globally defined basis functions representing solutions to boundary-value problems. The subsequent implementation of eigenfunction expansion, Galerkin, collocation, or other projection methods and the analysis of the computed solutions is simplified by the differentiation arrays, quadrature weights, and discretized weight functions produced by the Matlab-based `pd.m` and `sl.m` subprograms. These and related functions can be obtained from `http://www.ench.umd.edu/software/MWRtools`.
These methods primarily have been used for simulation of chemical vapor deposition reactors and other unit operations in semiconductor device fabrication [18] and model reduction studies [19, 20]. In the latter, the eigenfunction expansion methods are instrumental for identifying optimized trial functions for reduced-basis discretizations as well as in the implementation of nonlinear Galerkin methods [21] for the reducing the dynamic degrees of freedom in discretized boundary-value problems. We have also found educational applications for our numerical methods [22], where the utility stems from the reduction of details of numerically implementing the methods of weighted residuals.

Acknowledgments

This work was supported by the National Science Foundation through grant NSF EEC 95-27 576 and the Petroleum Research Fund of the American Chemical Society through grant 31391-G9.

References

[1] Bangia, A. K., P.F. Batcho, I.G. Kevrekidis, G.Em. Karniadakis, Unsteady 2-D flows in complex geometries: comparative bifurcation studies with global eigenfunction expansions, *SIAM J. Sci. Comput.* **18** (3), 775-805, 1997.

[2] Pruess, S. and C. T. Fulton, Mathematical software for Sturm-Liouville problems. *ACM Trans. Math. Software* **19** (3) 360-376, 1993.

[3] Hinton, D. and P. W. Schaefer, Eds. *Spectral Theory and Computational Methods of Sturm-Liouville Problems.* Lecture Notes in Pure and Applied Mathematics **191** Marcel Dekker, New York, 1997.

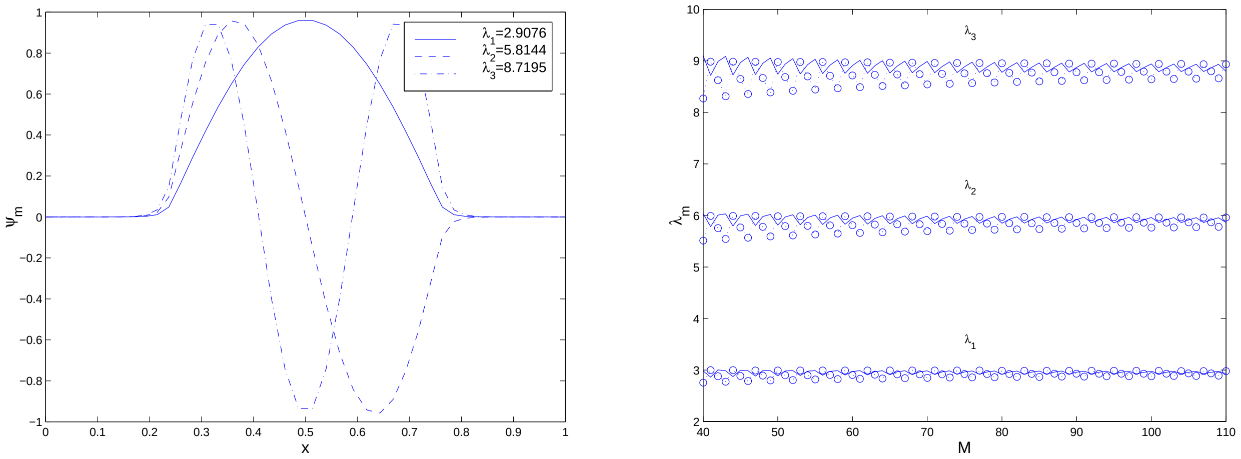

Figure 6: The eigenfunctions and corresponding eigenvalues of the square-well potential problem for $M = 60$ (left). A plot demonstrating eigenvalue convergence as a function M (right) for unfiltered (circles and dashed curves) and filtered (solid curves) computations.

[4] Pryce, J. D., *Numerical Solution of Sturm-Liouville Problems.* Oxford University Press, Oxford, UK, 1994.

[5] Vanden Berghe, G., M. Van Daele, and H. De Meyer, A modified difference scheme for periodic and semiperiodic Sturm-Liouville problems. *Appl. Num. Math.* **18**, 69-78, 1995.

[6] Stroud, A. H., *Numerical Quadrature and the Solution of Ordinary Differential Equations.* Springer-Verlag Applied Mathematical Science Series, Vol. 10, New York, 1974.

[7] Villadsen, J. and W. E. Stewart, Solution of boundary-value problems by orthogonal collocation. *Chem. Engng Sci.* **22**, 1483-1501, 1967.

[8] Michelsen, M. L. and J. Villadsen, A convenient computational procedure for collocation constants. *Chem. Eng. J.* **4**, 1972.

[9] Villadsen, J. and M. L. Michelsen, *Solution of Differential Equation Models by Polynomial Approximation* Prentice-Hall, Int Series in Phys. and Chem. Engng Sci. Englewood Cliffs, NJ, 1978.

[10] Lin, Y. -h., Chang, H. -Y., and R. A. Adomaitis, MWRtools: A library for weighted residual methods, *Comp. & Chem Engng*, submitted; also ISR TR 98-24, 1998.

[11] Rice, R. G. and D. D. Do,*Applied Mathematics and Modeling for Chemical Engineers*, John Wiley & Sons, New York, 1995.

[12] Adomaitis, R. A. and Y. -h. Lin, A technique for accurate collocation residual calculations, *Chem. Engng J.*, accepted for publication; also ISR TR 98-6, 1998.

[13] Joyce, D. C., Survey of extrapolation processes in numerical analysis. *SIAM Rev.* **13**, 435-490 (1971).

[14] Jakob, M., *Heat Transfer* J. Wiley & Sons, New York, 1949.

[15] Ray, W. H., *Advanced Process Control* McGraw-Hill Chemical Engineering Series, New York, 1982.

[16] Lee, J. -Y. and L. Greengard, A fast adaptive numerical method for stiff two-point boundary value problems, *SIAM J. Sci. Comput.* **18** (2) 403-429, 1997.

[17] Gottlieb, D. and C. -W. Shu, On the Gibbs phenomenon and its resolution. *SIAM Rev.* **39** (4) 644-668, 1997.

[18] Chang, H.-Y., R. A. Adomaitis, Analysis of heat transfer in a chemical vapor deposition reactor: an eigenfunction expansion solution approach. *Int. J. Heat and Fluid Flow*, in press; also ISR TR 97-84, 1997.

[19] Chang, H. Y. and R. A. Adomaitis, Model reduction for a tungsten chemical vapor deposition system, *Proc. DYCOPS-5* Corfu, Greece, 35-40; also ISR TR 98-28, 1998.

[20] Chang, H.-Y., Y.-h. Lin, and R. A., Adomaitis, Statistically-determined trial functions for RTCVD system model reduction and simulation, AIChE Annual Meeting Los Angeles, 1997.

[21] Shvartsman S. Y. and I. G. Kevrekidis, Nonlinear model reduction for feedback control of distributed parameter systems; A computer-assisted Study *AIChE J.*, **44** 1579, 1998.

[22] Adomaitis, R. A., Software and other teaching tools applied to modeling and analysis of distributed parameter systems. Proc. ASEE Conf., Oct. 1997.