Solution of Sturm-Liouville problems using modified Neumann schemes

Veerle Ledoux and Marnix Van Daele

In: SIAM Journal on Scientific Computing, 32 (2), pp. 564-584, 2010.

http://dx.doi.org/10.1137/090758398

SOLUTION OF STURM-LIOUVILLE PROBLEMS USING MODIFIED NEUMANN SCHEMES

VEERLE LEDOUX*† AND MARNIX VAN DAELE*

Abstract. The main purpose of this paper is to describe the extension of the successful modified integral series methods for Schrödinger problems to more general Sturm-Liouville eigenvalue problems. We present a robust and reliable modified Neumann method which can handle a wide variety of problems. This modified Neumann method is closely related to the second-order Pruess method, but provides for higher order approximations. We show that the method can be successfully implemented in a competitive automatic general-purpose software package.

Key words. Sturm-Liouville problems, shooting, eigenvalue

AMS subject classifications. 34L16, 65L10, 65L15

1. Introduction. The Sturm-Liouville eigenvalue problem (SLP) considered is to find values for E such that

$$-(p(x)y(x)')' + q(x)y(x) = Ew(x)y(x), \tag{1.1}$$

with p and w strictly positive on (a, b), has a nonzero solution $y(x)$ satisfying boundary conditions of the form

$$a_1y(a) + a_2p(a)y'(a) = 0, \quad b_1y(b) + b_2p(b)y'(b) = 0, \tag{1.2}$$

where a_1, a_2 are not both zero nor are b_1, b_2.
The problem (1.1) is called regular if a and b are finite and $1/p(x)$, $q(x)$ and $w(x)$ are locally integrable near the endpoints. Otherwise, the problem is called singular. Standard mathematical theory assumes that $p(x)$, $q(x)$ and $w(x)$ are at least continuous on (a, b) although a finite number of finite jump discontinuities can be handled by suitably defining an input mesh. For regular problems the eigenvalues can be ordered as an increasing sequence tending to infinity

$$E_0 < E_1 < E_2 < \dots$$

and with this labelling the eigenfunction y_k, corresponding to E_k and unique up to a normalizing constant, has exactly k zeros on the open interval (a, b). More information on the mathematical theory of regular and singular Sturm-Liouville problems can be found in in [25, 29].
Many physical phenomena, both in classical mechanics and in quantum mechanics, are described mathematically by Sturm-Liouville problems. Since most of the problems can not be solved analytically, good numerical approximation methods are essential. Finding the eigenvalues of a Sturm-Liouville problem can however be a computationally challenging task, especially when a large set of eigenvalues is computed, or just when particularly large eigenvalues are sought. The highly oscillatory behavior of the solutions corresponding to high eigenvalues, forces a naive integrator to take increasingly smaller steps. Better results are obtained using methods based on coefficient approximation since, for such methods, the step size is not restricted by the oscillations in the solution. A method that has been successfully used to solve Sturm-Liouville problems is, e.g., the second order coefficient approximation method which was implemented in the SLEDGE package [27, 28]. Pryce called this method the Pruess method in [26, 25]. This Pruess method approximates the coefficient functions $p(x)$, $q(x)$ and $w(x)$ by piecewise-constant approximations, solving the problem analytically on the piecewise-constant intervals. This leads, however, to a method of only second order. To obtain higher order coefficient approximation methods, piecewise polynomial approximations of higher degree should be used. But, for traditional numerical integrators, solving an approximate problem with polynomial coefficient functions of order greater than zero is not

*Vakgroep Toegepaste Wiskunde en Informatica, Ghent University, Krijgslaan 281-S9, B-9000 Gent, Belgium
†VL is a postdoctoral fellow of the Fund of Scientific Research-Flanders (F.W.O.Vlaanderen).

easier than solving the original problem. Thus only piecewise constant polynomials were used in practical software packages as SLEDGE for a long time, because for these eq. (1.1) is easily analytically integrated. More recently some approaches were suggested to construct higher-order methods and thus to realize the approximation of the coefficient functions by higher order polynomials. The so-called piecewise perturbation methods (PPM) use a perturbation technique to construct some correction terms that are added to the known solution of the approximating problem with a piecewise-constant potential. In this way methods up to order 16 were constructed (see [13, 15, 17]) which can efficiently compute the eigenvalues of regular Schrödinger problems

$$y''(x) = (V(x) - E)y(x). \tag{1.3}$$

The PPM were extended to general Sturm-Liouville problems using the Liouville transform (see [14] and [18]) and were implemented in the Fortran package SLCPM12 [14] and the Matlab package MATSLISE [18]. In [5] it was shown that the piecewise perturbation approach may be viewed as the application of a modified Neumann series where in fact each extra PPM correction term is an extra term in the Neumann series. Besides the Neumann series, another integral series has been recognized as being a very effective computational tool for problems with a highly oscillatory solution: the Magnus expansion. Methods based on discrete Magnus expansions are Lie group methods (see [6, 22, 7]), which typically show nice results with regard to accuracy and stability. Also, integrators based on this Magnus expansion can be combined with coefficient approximation and form another extension of the Pruess ideas for high order approximations. Moan [21] was the first to consider a Magnus method in the context of Sturm-Liouville problems in the Schrödinger form (1.3). Later, Degani and Schiff [5] and Iserles [8] showed that, for oscillatory ordinary differential equations, it is better to apply the Magnus series integrator to the so-called modified equation and not to the equation directly. In [19] such a modified Magnus integrator was successfully applied on Schrödinger problems.
Note that in [21, 5, 19], the modified integral (Neumann and Magnus) series methods were only considered for Sturm-Liouville problems in the simpler Schrödinger form. Also the PPM were especially constructed for Schrödinger problems and can only be applied to Sturm-Liouville problems after a Liouville transformation. However the Liouville transformation is rather expensive due to the quadrature which is needed for the conversion between old and new variable. Moreover the transformation can only be realized for sufficiently well-behaved (non-singular) p, q and w functions (see [25]): q must be continuous and p and w should have a continuous second derivative. The first and second order derivative of p and w are needed in the transformation: a user of the SLCPM12 package even has to give their expressions as input. In the present paper, we want to discuss the construction of a modified scheme which can be applied on a Sturm-Liouville problem in its general form. The approach offers a way to approximate the coefficient functions of the SLP by higher order (piecewise) polynomials while still retaining the nice property of the Pruess method that the solution of the approximating problem is integrated explicitly in terms of trigonometric/hyperbolic functions. These new modified Neumann methods can thus be seen as the natural extension of the Pruess method to higher order methods or a generalization of the PPM to a larger class of Sturm-Liouville problems (singular problems, problems with discontinuities,...). All elements will be presented which are needed to allow the automatic solution of a Sturm-Liouville eigenvalue problem.
Our paper is organized as follows. In Section 2 we show how the Neumann expansion can be used in the numerical solution of a Sturm-Liouville problem. Details are given for schemes up to order six. In Section 3, we provide an error analysis of these schemes. We then present in Section 4 a shooting procedure to accurately compute the eigenvalues, including details on error estimation, mesh selection and a Prüfer based zero counting procedure to home in on a particular eigenvalue. We discuss the numerical treatment of (some) singular problems in Section 5. We then implement and apply our numerical methods on some test problems in Section 6.

2. Modified Neumann schemes for the SLP.

2.1. Modified Neumann method.

The SLP can be written in matrix form as

$$\begin{bmatrix} y(x) \\ p(x)y'(x) \end{bmatrix}' = \begin{bmatrix} 0 & 1/p(x) \\ q(x) - Ew(x) & 0 \end{bmatrix} \begin{bmatrix} y(x) \\ p(x)y'(x) \end{bmatrix}. \tag{2.1}$$

This is thus a system of the form

$$\mathbf{y}(x)' = A(x)\mathbf{y}(x), \quad \mathbf{y}(a) = \mathbf{y}_0. \tag{2.2}$$

with

$$\mathbf{y} = \begin{bmatrix} y(x) \\ p(x)y'(x) \end{bmatrix}. \tag{2.3}$$

There is an emerging family of numerical methods based on integral series representation of ODE solutions which can be applied on systems of the form (2.2) (see e.g. [7, 10]). The two most popular integral series are the Neumann series and the Magnus series. When the solution of a linear system $\mathbf{y}' = A(x)\mathbf{y}$ oscillates rapidly, a Neumann (or Magnus) method should however not be applied directly to the problem but modified schemes should be used, as recommended in [5, 8, 9]. We will introduce here such a modified scheme for the SLP.

Suppose that we have already computed $\mathbf{y}_i \approx \mathbf{y}(x_i)$ and that we wish to advance the numerical solution to $x_{i+1} = x_i + h_i$. The first step in the modified scheme is to change the variables locally

$$\mathbf{y}(x) = e^{(x-x_i)\bar{A}}\mathbf{u}(x - x_i), \quad x_i \le x \le x_{i+1} \tag{2.4}$$

where $\bar{A}(E)$ is a constant approximation of the matrix function A

$$\bar{A}(E) = \begin{bmatrix} 0 & \bar{P} \\ \bar{q} - E\bar{w} & 0 \end{bmatrix} \tag{2.5}$$

and $\bar{q}$, $\bar{w}$, $\bar{P}$ are constant approximations of the functions $q(x), w(x), P(x) = 1/p(x)$ over the interval $[x_i, x_{i+1}]$:

$$\bar{f} = \frac{1}{h_i}\int_{x_i}^{x_i+h_i} f(x)dx, \qquad f = q, w, P. \tag{2.6}$$

We treat $\mathbf{u}$ as our new unknown which itself obeys the linear differential equation

$$\mathbf{u}'(\delta) = B(\delta, E)\mathbf{u}(\delta), \quad \delta \in [0, h_i], \quad \mathbf{u}(0) = \mathbf{y}_i \tag{2.7}$$

where

$$B(\delta, E) = e^{-\delta\bar{A}}\left(A(x_i + \delta) - \bar{A}\right)e^{\delta\bar{A}}. \tag{2.8}$$

The matrix B can be computed explicitly. We first define the functions $\xi(Z)$ and $\eta_0(Z)$ (using the same notation as in the construction of the PPM, see [12, 13]) as

$$\xi(Z) = \begin{cases} \cos(|Z|^{1/2}) & \text{if } Z \le 0\,, \\ \cosh(Z^{1/2}) & \text{if } Z > 0\,, \end{cases} \quad \eta_0(Z) = \begin{cases} \sin(|Z|^{1/2})/|Z|^{1/2} & \text{if } Z < 0\,, \\ 1 & \text{if } Z = 0\,, \\ \sinh(Z^{1/2})/Z^{1/2} & \text{if } Z > 0\,, \end{cases} . \tag{2.9}$$

The entries of B are then given by

$$B_{11}(\delta, E) = -B_{22}(\delta, E) = \Omega(\delta)\delta\eta_0(Z_{2\delta}) \tag{2.10}$$

$$B_{12}(\delta, E) = -\Omega(\delta)\frac{1 - \xi(Z_{2\delta})}{2\bar{\Lambda}(E)} + P(x_i + \delta) - \bar{P} \tag{2.11}$$

$$B_{21}(\delta, E) = \Omega(\delta)\frac{1 - \xi(Z_{2\delta})}{2\bar{P}} + \Lambda(\delta; E) - \bar{\Lambda}(E) \tag{2.12}$$

where

$$\bar{\Lambda}(E) = \bar{q} - E\bar{w}, \quad \Lambda(\delta; E) = q(x_i + \delta) - Ew(x_i + \delta), \tag{2.13}$$

$$\Omega(\delta) = \bar{\Lambda}(E)P(x_i + \delta) - \Lambda(\delta; E)\bar{P} \tag{2.14}$$

and

$$Z_\gamma = Z(\gamma) = \bar{\Lambda}(E)\bar{P}\gamma^2. \tag{2.15}$$

We have thus replaced one linear system by another, i.e (2.2) by (2.7). When the modified equation is solved by an integral series method, the new system (2.7) has one crucial advantage over (2.2): the entries of the matrix B are themselves rapidly oscillating functions (for $Ew(x) - q(x) > 0$). The higher the oscillation in B, the faster the convergence of the integral series method (see [8, 10, 5]). Here we will choose a Neumann series for the integral series method. We can refer to [10] for numerical and theoretical results confirming the success of such a modified Neumann approach for highly oscillatory ODEs.
Over each interval $[x_i, x_{i+1}]$, we apply a Neumann method to the modified equation $\mathbf{u}'(\delta) = B(\delta)\mathbf{u}(\delta), \mathbf{u}(0) = \mathbf{y}_i$. This gives

$$\mathbf{u}_{i+1} = \mathbf{y}_i + \int_0^{h_i} B(x)dx\mathbf{y}_i + \int_0^{h_i}\int_0^{x_1} B(x_1)B(x_2)dx_2dx_1\mathbf{y}_i + \dots \tag{2.16}$$

The solution $\mathbf{y}$ in $x = x_{i+1}$ of the original system is then obtained from $\mathbf{y}(E, x_{i+1}) = e^{h_i\bar{A}}\mathbf{u}(h_i)$, where $e^{h_i\bar{A}}$ is actually the known solution of the system with constant coefficient functions:

$$e^{h_i\bar{A}} = \begin{bmatrix} \xi(Z_{h_i}) & h_i\bar{P}\eta_0(Z_{h_i}) \\ \dfrac{Z_{h_i}\eta_0(Z_{h_i})}{h_i\bar{P}} & \xi(Z_{h_i}) \end{bmatrix}. \tag{2.17}$$

This illustrates that the Neumann scheme forms a natural extension of the Pruess method to higher order methods. When only the first term in the Neumann series (2.16) is retained, one has exactly the second-order Pruess method. In the next section we reconsider this second-order method and construct some higher order methods by including more Neumann terms. In [5] it was shown that for a Schrödinger problem each extra Neumann term is equivalent to a PPM correction term.

2.2. Practical implementation. Practical implementation of the Neumann series method requires the truncation of the integral series and the replacement of multivariate integrals by quadrature. As in [10, 11], a Filon approach can be used to successfully approximate the oscillating integrals in the modified scheme (2.16). Here this means that we have to fit a polynomial to the nonoscillatory part of the integrand. We replace the functions $q, w, P = 1/p$ by interpolating polynomials written as series over shifted Legendre polynomials, as was also done to compute the correction terms in the PPM schemes in [13, 14]. Each coefficient function $f = q, w, P$ is approximated (over each mesh interval $[x_i, x_i + h_i]$) by

$$f(x_i + \delta) \approx \sum_{s=0}^{\nu-1} F_s h_i^s P_s^*(\delta/h_i), \quad \delta \in [0, h_i]. \tag{2.18}$$

The expressions of the first shifted Legendre polynomials $P_s^*(\gamma), \gamma \in [0, 1]$ are as follows

$$P_0^*(\gamma) = 1, \;\; P_1^*(\gamma) = -1 + 2\gamma, \;\; P_2^*(\gamma) = 1 - 6\gamma + 6\gamma^2. \tag{2.19}$$

By the method of least squares the expressions for the coefficients $F_s (F = Q, W, P)$ are obtained:

$$F_s = \frac{(2s+1)}{h_i^{s+1}} \int_0^{h_i} f(x_i + \delta)P_s^*(\delta/h_i)d\delta, \;\; m = 0, 1, 2, \dots. \tag{2.20}$$

It can then be noted that $\bar{q} = Q_0, \bar{P} = P_0, \bar{w} = W_0$. We also define

$$S_0 = 0, \quad S_n = (\bar{q} - E\bar{w})P_n - \bar{P}(Q_n - EW_n), \quad n = 1, \dots, \nu - 1$$

and thus replace $\Omega(\delta)$ by $\sum_{s=0}^{\nu-1} S_s h_i^s P_s^*(\delta/h_i)$.

2.2.1. Method of order 2. For the second order method (or Pruess method) only the first term in the Neumann series (2.16) needs to be retained in the algorithm:

$$\begin{aligned}\mathbf{u}_{i+1} &= \mathbf{y}_i \\ \mathbf{y}_{i+1} &= e^{h_i \bar{A}}\mathbf{u}_{i+1}\end{aligned} \tag{2.21}$$

In this case it is sufficient to have second order piecewise constant approximations in (2.6). As in [28], piecewise constant midpoint approximation can be applied.

2.2.2. Method of order 4. To reach fourth order, the first integral must be included in the algorithm

$$\begin{aligned}\mathbf{u}_{i+1} &= \mathbf{y}_i + \int_0^{h_i} B(x)dx\mathbf{y}_i, \\ \mathbf{y}_{i+1} &= e^{h_i \bar{A}}\mathbf{u}_{i+1}\end{aligned} \tag{2.22}$$

Since $\int_0^{h_i}(\bar{q}-q(x_i+\delta))d\delta = h_i\bar{q} - \int_0^{h_i} q(x_i+\delta)d\delta = h_i\bar{q} - h_i Q_0 = 0$ and analogously $\int_0^{h_i}(\bar{w}-w(x_i+\delta))d\delta = \int_0^{h_i}(\bar{P} - P(x_i+\delta))d\delta = 0$, we can write

$$\int_0^{h_i} B_{12}(\delta, E)d\delta = \frac{1}{2\bar{\Lambda}(E)}\int_0^{h_i}\Omega(\delta)\xi(Z_{2\delta})d\delta \tag{2.23}$$

and

$$\int_0^{h_i} B_{21}(\delta, E)d\delta = -\frac{1}{2\bar{P}}\int_0^{h_i}\Omega(\delta)\xi(Z_{2\delta})d\delta. \tag{2.24}$$

As a result, we only need to compute the two integrals

$$\int_0^{h_i}\Omega(\delta)\delta\eta_0(\delta)d\delta, \quad \int_0^{h_i}\Omega(\delta)\xi(\delta)d\delta. \tag{2.25}$$

For a fourth order method it is sufficient to approximate the functions $q, w, 1/p$ by a first degree polynomial. Solving the resulting integral analytically, we obtain

$$\int_0^{h_i}\Omega(\delta)\delta\eta_0(\delta)d\delta \approx I_1 = \frac{(-2\hat{\eta}_0 + 1 + \hat{\xi})\hat{S}_1}{4Z_{h_i}}, \quad \hat{S}_1 = h_i^3 S_1 \tag{2.26}$$

$$\int_0^{h_i}\Omega(\delta)\xi(\delta)d\delta \approx I_2 = (2\hat{\eta}_0 + \frac{1-\hat{\xi}}{Z_{h_i}})\hat{S}_1/(2h_i) \tag{2.27}$$

with $\hat{\xi} = \xi(Z_{2h_i}) = 2\xi(Z_{h_i})^2 - 1$, $\hat{\eta}_0 = \eta_0(Z_{2h_i}) = \eta_0(Z_{h_i})\xi(Z_{h_i})$ (the same trigonometric or hyperbolic function evaluations as in (2.17) are used). A fourth-order Gauss-Legendre scheme with $\nu = 2$ nodes is sufficient to compute S_1 via formula (2.20).

2.2.3. Method of order 6. To construct a scheme of order 6, we include the double integral:

$$\mathbf{u}_{i+1} = \mathbf{y}_i + \int_0^{h_i} B(x)dx\mathbf{y}_i + \int_0^{h_i}\int_0^{x_1} B(x_1)B(x_2)dx_2dx_1\mathbf{y}_i,$$

and we replace the coefficient functions (piecewisely) by second order polynomials ($\nu = 3$). The resulting analytic expressions for the integrals can then be computed in a symbolic software package. We obtain for the univariate integrals

$$\int_0^{h_i}\Omega(\delta)\delta\eta_0(\delta)d\delta \approx I_1 = \left[\frac{-2\hat{\eta}_0 + 1 + \hat{\xi}}{Z_{h_i}}\hat{S}_1 + \left(\frac{-6\hat{\eta}_0 + \hat{\xi} - 1}{Z_{h_i}} + \frac{3(\hat{\xi}-1)}{Z_{h_i}^2}\right)\hat{S}_2\right]/4 \tag{2.28}$$

$$\int_0^{h_i}\Omega(\delta)\xi(\delta)d\delta \approx I_2 = \left[\left(2\hat{\eta}_0 + \frac{1-\hat{\xi}}{Z_{h_i}}\right)\hat{S}_1 + \left(2\hat{\eta}_0 - \frac{3(1+\hat{\xi}) - 6\hat{\eta}_0}{Z_{h_i}}\right)\hat{S}_2\right]/(2h_i). \tag{2.29}$$

To compute $\hat{S}_1 = h_i^3 S_1$ and $\hat{S}_2 = h_i^4 S_2$, sixth-order Gauss-Legendre with $\nu = 3$ nodes is used.

We also have to approximate the double integral. The same procedure is followed, i.e. q, w and $1/p$ are replaced by the second degree polynomials and the resulting integrals are computed analytically. For notational reasons it is convenient to rewrite B as

$$B_{11}(\delta, E) = -B_{22}(\delta, E) = \Omega(\delta)\delta\eta_0(Z_{2\delta}) \tag{2.30}$$

$$B_{12}(\delta, E) = \frac{\Omega(\delta)\xi(Z_{2\delta}) + \Sigma(\delta)}{2\bar{\Lambda}(E)} - \bar{P} \tag{2.31}$$

$$B_{21}(\delta, E) = \frac{-\Omega(\delta)\xi(Z_{2\delta}) + \Sigma(\delta)}{2\bar{P}} - \bar{\Lambda}(E) \tag{2.32}$$

where $\Sigma(\delta) = \bar{\Lambda}(E)P(x_i + \delta) + \Lambda(\delta; E)\bar{P}$. The second degree polynomial approximating this Σ function reads

$$\Sigma(\delta) \approx \sum_{s=0}^{\nu-1} T_s h_i^s P_s^*(\delta/h_i) \tag{2.33}$$

where

$$T_s = (\bar{q} - E\bar{w})P_s + \bar{P}(Q_s - EW_s).$$

We show the results (with $\hat{T}_s = h_i^{s+2} T_s$)

$$\begin{aligned}
&\int_0^{h_i}\int_0^{x_1} [B_{11}(x_1)B_{11}(x_2) + B_{12}(x_1)B_{21}(x_2)]dx_2dx_1 = \\
&\Big[\Big(\frac{6\hat{\eta}_0 + 1 - \hat{\xi}}{8Z_{h_i}^2} + \frac{3(1-\hat{\xi})}{8Z_{h_i}^3}\Big)\hat{S}_1 + \Big(\frac{-\hat{\xi} - 1 + 14\hat{\eta}_0}{8Z_{h_i}^2} - \frac{9(1 + \hat{\xi} - 2\hat{\eta}_0)}{4Z_{h_i}^3}\Big)\hat{S}_2\Big]\hat{T}_1 + \\
&\Big[\Big(\frac{10\hat{\eta}_0 - \hat{\xi} - 1}{8Z_{h_i}^2} - \frac{3(1 + \hat{\xi} - 2\hat{\eta})_0}{2Z_{h_i}^3}\Big)\hat{S}_1 + \Big(\frac{18\hat{\eta}_0 - \hat{\xi} + 1}{8Z_{h_i}^2} + \frac{39(1-\hat{\xi}) + 180\hat{\eta}_0}{8Z_{h_i}^3} + \frac{45(1-\hat{\xi})}{4Z_{h_i}^4}\Big)\hat{S}_2\Big]\hat{T}_2 + \\
&\Big(\frac{1 - 4\hat{\eta}_0 + \hat{\xi}}{16Z_{h_i}^2} + \frac{\hat{\xi} - 1}{16Z_{h_i}^3}\Big)\hat{S}_1^2 + \Big(\frac{12\hat{\eta}_0 - \hat{\xi} + 1}{16Z_{h_i}^2} + \frac{36\hat{\eta}_0 - 3 - 15\hat{\xi}}{16Z_{h_i}^3} + \frac{9(1-\hat{\xi})}{16Z_{h_i}^4}\Big)\hat{S}_2^2
\end{aligned} \tag{2.34}$$

$$\begin{aligned}
&\frac{1}{h_i P_0}\int_0^{h_i}\int_0^{x_1} [B_{11}(x_1)B_{12}(x_2) + B_{12}(x_1)B_{22}(x_2)]dx_2dx_1 = \\
&\Big[\Big(\frac{-\hat{\eta}_0}{4Z_{h_i}^2} + \frac{3(1 - 2\hat{\eta}_0 + \hat{\xi})}{8Z_{h_i}^3}\Big)\hat{S}_1 + \Big(\frac{-\hat{\eta}_0}{4Z_{h_i}^2} - \frac{36\hat{\eta}_0 + 7(1-\hat{\xi})}{8Z_{h_i}^3} - \frac{9(1-\hat{\xi})}{4Z_{h_i}^4}\Big)\hat{S}_2\Big]\hat{T}_1 + \\
&\Big[\Big(\frac{-\hat{\eta}_0}{4Z_{h_i}^2} - \frac{5(1-\hat{\xi}) + 24\hat{\eta}_0}{8Z_{h_i}^3} - \frac{3(1-\hat{\xi})}{2Z_{h_i}^4}\Big)\hat{S}_1 + \Big(\frac{-\hat{\eta}_0}{4Z_{h_i}^2} + \frac{9(\hat{\xi} + 1 - 8\hat{\eta}_0)}{8Z_{h_i}^3} + \frac{45(1 + \hat{\xi} - 2\hat{\eta}_0)}{4Z_{h_i}^4}\Big)\hat{S}_2\Big]\hat{T}_2 + \\
&\Big(\frac{-1 - 3\hat{\eta}_0}{24Z_{h_i}^2} + \frac{\hat{\xi} - \hat{\eta}_0}{8Z_{h_i}^3}\Big)\hat{S}_1^2 + \Big(\frac{5\hat{\eta}_0 - 1}{40Z_{h_i}^2} - \frac{3(\hat{\xi} - 5\hat{\eta}_0)}{8Z_{h_i}^3} - \frac{9(\hat{\xi} - \hat{\eta}_0)}{8Z_{h_i}^4}\Big)\hat{S}_2^2
\end{aligned} \tag{2.35}$$

$$
\begin{aligned}
&h_i P_0 \int_0^{h_i}\int_0^{x_1} [B_{21}(x_1)B_{11}(x_2)+B_{22}(x_1)B_{21}(x_2)]dx_2dx_1 = \\
&\Big[\Big(\frac{\hat\eta_0}{4Z_{h_i}}+\frac{3(2\hat\eta_0-\hat\xi-1)}{8Z_{h_i}^2}\Big)\hat S_1+\Big(\frac{\hat\eta_0}{4Z_{h_i}}+\frac{36\hat\eta_0+7(1-\hat\xi)}{8Z_{h_i}^2}+\frac{9(1-\hat\xi)}{4Z_{h_i}^3}\Big)\hat S_2\Big]\hat T_1+\\
&\Big[\Big(\frac{\hat\eta_0}{4Z_{h_i}}+\frac{24\hat\eta_0+5(1-\hat\xi)}{8Z_{h_i}^2}+\frac{3(1-\hat\xi)}{2Z_{h_i}^3}\Big)\hat S_1+\Big(\frac{\hat\eta_0}{4Z_{h_i}}+\frac{9(8\hat\eta_0-\hat\xi-1)}{8Z_{h_i}^2}-\frac{45(1-2\hat\eta_0+\hat\xi)}{4Z_{h_i}^3}\Big)\hat S_2\Big]\hat T_2+\\
&\Big(\frac{-3\hat\eta_0-1}{24Z_{h_i}}+\frac{\hat\xi-\hat\eta_0}{8Z_{h_i}^2}\Big)\hat S_1^2+\Big(\frac{5\hat\eta_0-1}{40Z_{h_i}}-\frac{3(\hat\xi-5\hat\eta_0)}{8Z_{h_i}^2}-\frac{9(\hat\xi-\hat\eta_0)}{8Z_{h_i}^3}\Big)\hat S_2^2
\end{aligned}
\tag{2.36}
$$

$$
\begin{aligned}
&\int_0^{h_i}\int_0^{x_1} [B_{21}(x_1)B_{12}(x_2)+B_{22}(x_1)B_{22}(x_2)]dx_2dx_1 = \\
&\Big[\Big(\frac{-6\hat\eta_0+\hat\xi-1}{8Z_{h_i}^2}-\frac{3(1-\hat\xi)}{8Z_{h_i}^3}\Big)\hat S_1+\Big(\frac{\hat\xi+1-14\hat\eta_0}{8Z_{h_i}^2}+\frac{9(2-\hat\eta_0+\hat\xi)}{2Z_{h_i}^3}\Big)\hat S_2\Big]\hat T_1+\\
&\Big[\Big(\frac{\hat\xi-1-18\hat\eta_0}{8Z_{h_i}^2}-\frac{180\hat\eta_0+39(1-\hat\xi)}{8Z_{h_i}^3}-\frac{45(1-\hat\xi)}{4Z_{h_i}^4}\Big)\hat S_2+\Big(\frac{1+\hat\xi-10\hat\eta_0}{8Z_{h_i}^2}+\frac{3(1+\hat\xi-2\hat\eta_0)}{2Z_{h_i}^3}\Big)\hat S_1\Big]\hat T_2+\\
&\Big(\frac{1-\hat\xi+12\hat\eta_0}{16Z_{h_i}^2}-\frac{3(5\hat\xi+1-12\hat\eta_0)}{16Z_{h_i}^3}+\frac{9(1-\hat\xi)}{16Z_h^4}\Big)\hat S_2^2+\Big(\frac{\hat\xi+1-4\hat\eta_0}{16Z_{h_i}^2}+\frac{\hat\xi-1}{16Z_h^3}\Big)\hat S_1^2
\end{aligned}
\tag{2.37}
$$

In practice, for small Z_{h_i} values one should use alternative formulae in which the $\hat\xi$ and $\hat\eta_0$ functions are replaced by truncated series expansions.

3. Error analysis. The error induced by a modified Neumann method is caused by (i) the approximation of the coefficient functions by a piecewise polynomial (which is in fact applying a Filon-type quadrature method on the integrals in the Neumann series) (ii) taking only a limited number of terms in the Neumann series. In the next two subsections we discuss this error for small stepsize $h>0$ and a fixed value E, and some convergence results when the step size is fixed and E varies.

3.1. Classical order (fixed E, small h). The following result was already shown by Pruess in [27]: if the coefficient functions are piecewisely approximated by mth degree interpolating polynomials and the set of interpolating points is the set of zeros of the $(m+1)$th degree Legendre polynomial then

$$
|E_k-\hat E_k| = O(h^{2m+2}) \tag{3.1}
$$

where $\hat E_k$ are the eigenvalues of the approximate problem. This means that when the coefficient functions are replaced by a first order piecewise polynomial in the way described in the previous section (eqs. (2.18)-(2.20)) and the resulting approximating problem is solved exactly, the obtained eigenvalues are $O(h^4)$. Similarly for piecewise polynomials of degree 2, an $O(h^6)$ algorithm is obtained when the resulting problem is solved analytically. However it is not really achievable to obtain the analytic solution of a Sturm-Liouville problem with polynomial coefficient functions with degree larger than zero. Therefore this solution is written as a Neumann series in which it is sufficient to retain only a limited number of terms for each order. Terms upto the first integral are sufficient for a scheme of fourth order and the double integral should be added in order to reach a sixth order algorithm. Indeed, by symbolic computation (performing Taylor expansions around $h=0$ in a symbolic software package), one can show that for fixed E and small $h>0$ we have

$$
N_1 \approx \begin{bmatrix} O(h^3) & O(h^4) \\ O(h^4) & O(h^3)\end{bmatrix},\quad N_2 \approx \begin{bmatrix} O(h^6) & O(h^5) \\ O(h^5) & O(h^6)\end{bmatrix},\quad N_3 \approx \begin{bmatrix} O(h^7) & O(h^8) \\ O(h^8) & O(h^7)\end{bmatrix} \tag{3.2}
$$

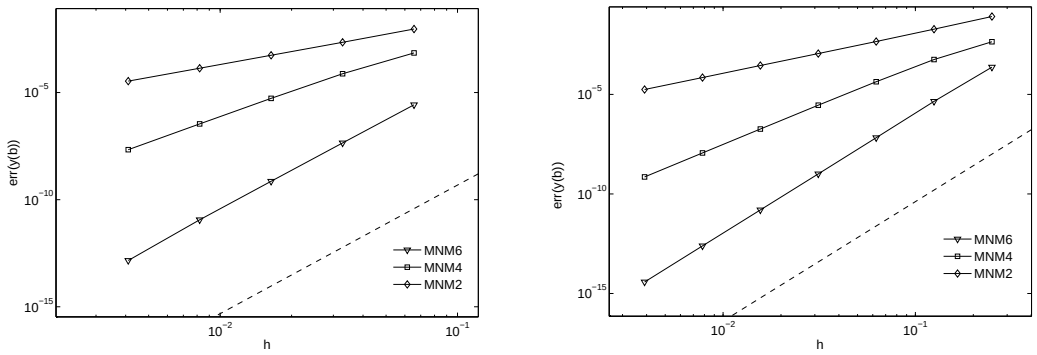

FIG. 3.1. *Illustration of the order of the modified Neumann schemes for two different Sturm-Liouville problems: the Paine problem (left) and the Collatz problem (right). The absolute error in $y(b)$ is shown as a function of h. For the Paine problem we chose $E = E_9 = 102.424988398250$ and for the Collatz problem $E = E_0 = 64\pi^2/9$.*

for our modified Neumann scheme where

$$N_i = \int_0^h \int_0^{x_1} \dots \int_0^{x_{i-1}} B(x_1)B(x_2)\dots B(x_i)dx_i \dots dx_2 dx_1. \tag{3.3}$$

with B as in (2.10). This means that the "classical" order of different truncated Neumann series, as applied to our Sturm-Liouville problem, is as follows:

$$\mathbf{u}_{i+1} = \mathbf{y}_i \quad \text{order 2}, \tag{3.4}$$
$$\mathbf{u}_{i+1} = (I + N_1)\mathbf{y}_i \quad \text{order 4}, \tag{3.5}$$
$$\mathbf{u}_{i+1} = (I + N_1 + N_2)\mathbf{y}_i \quad \text{order 6}. \tag{3.6}$$

We consider two test problems to confirm these results numerically. The first test problem was also used to illustrate the use of SLCPM12 [14]:

$$p(x) = (\gamma + x)^3, \quad q(x) = 4(\gamma + x), \quad w(x) = (\gamma + x)^5, \quad \gamma = \sqrt{0.2} \tag{3.7}$$

over the integration interval $[a, b]$ with $a = 0$ and $b = -\gamma + \sqrt{\gamma^2 + 2\pi}$ and $y(a) = y(b) = 0$. We call this problem the Paine problem, since by the Liouville transformation it can be transformed to a Schrödinger problem introduced by Paine in [23]. The second test problem is the Collatz problem [4]

$$y'' + \frac{3}{4x^2}y = -\frac{1}{x^6}Ey \quad \text{in } [1, 2],\ y(1) = y(2) = 0. \tag{3.8}$$

This problem can be solved in closed form. The solutions are

$$E_k = \frac{64}{9}k^2\pi^2, \quad y_k = \frac{3}{8k\pi}x^{3/2}\sin\frac{4k\pi}{3}\left(1 - \frac{1}{x^2}\right), \quad k = 1, 2, 3, \dots. \tag{3.9}$$

Some numerical results for the two test problems are given in the double logarithmic plot in Figure 3.1. The modified Neumann methods of order two (MNM2), four (MNM4) and six (MNM6) were used to propagate y from a with starting values $y(a) = 0, y'(a) = 1$ to b. The methods were applied with constant step sizes h. The different schemes clearly behave as second, fourth and sixth order. The dashed line is a reference line for a sixth order method.

3.2. Error dependence on E (fixed steps). A basic convergence result by Pruess [27] (see also [25]) states that, provided p, q and w are in $C^{m+1}[a, b]$, using piecewise polynomial interpolants of degree m will give convergence of type

$$|E_k - \hat{E}_k| \le C \max(1, k^2)h^{m+1} \tag{3.10}$$

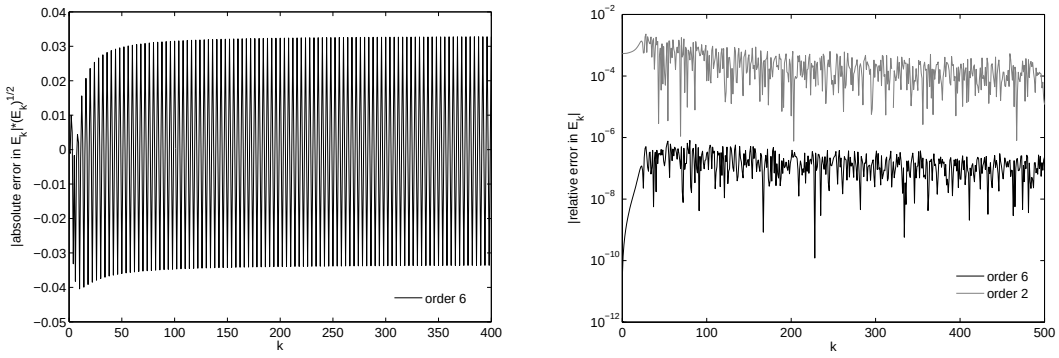

FIG. 3.2. *The error in the eigenvalues computed using the sixth order modified Neumann scheme for the Schrödinger Mathieu problem (left) with $n = 4$ mesh intervals and the Collatz problem (right) with $n = 64$ mesh intervals.*

where h is the maximum mesh size and C is a constant independent of k. For large k this means that $|(E_k - \bar{E}_k)/E_k| = O(h^{m+1})$ and the relative error is thus independent of k (see also [16]). Indeed, by symbolic computation we obtain the following asymptotic expressions (for $Z_h \ll 0$) for the Neumann terms:

$$N_1 \approx \begin{bmatrix} O(1) & O(1) \\ O(E) & O(1) \end{bmatrix}, \quad N_2 \approx \begin{bmatrix} O(1) & O(1) \\ O(E) & O(1) \end{bmatrix}, \quad N_3 \approx \begin{bmatrix} O(E) & O(1) \\ O(E) & O(E) \end{bmatrix}. \tag{3.11}$$

For problems in the Schrödinger form ($p = w = 1$), one can use larger h as k increases for a given relative error. As shown in [13, 5, 19], the error behaves like $O(E^{-1/2})$ for these Schrödinger problems. In figure 3.2, the E dependence of the error is illustrated for both a Schrödinger problem and a Sturm-Liouville problem. The eigenvalues were computed using the shooting method discussed in the next section. For the Schrödinger Mathieu problem

$$y'' = (2\cos(2x) - E)y, \quad y(0) = y(\pi) = 0 \tag{3.12}$$

we show the absolute error in the eigenvalue approximation for E_k multiplied by $E_k^{1/2}$ for different k values. For the Collatz problem we show the relative error in the eigenvalue estimates.

4. Eigenvalue computation. The modified Neumann schemes can be used to locate the eigenvalues by shooting. As for the Pruess method [28], some special care is needed in the shooting algorithm to avoid difficulties when the solutions show rapid exponential growth. Also we discuss a procedure to count the oscillations of the solution, as to home in on a particular eigenvalue.

4.1. The shooting algorithm. As described before, the following recursive scheme is used to propagate the solution over some mesh $a = x_0 < x_1 < \cdots < x_n = b$

$$\mathbf{y}_{i+1} = e^{h_i \bar{A}_i}(I + N_1 + N_2)\mathbf{y}_i, \quad i = 0, \ldots, n-1. \tag{4.1}$$

The spectral radius of the matrix $e^{h_i \bar{A}_i}$ is $\rho_i = 1$ when $(\bar{q}_i - E\bar{w}_i)\bar{P}_i \le 0$ and $\rho_i = e^{\sqrt{(\bar{q}_i - E\bar{w}_i)\bar{P}_i}h_i}$ when $(\bar{q}_i - E\bar{w}_i)\bar{P}_i > 0$. The recursion (4.1) can be unstable when many $(\bar{q}_i - E\bar{w}_i)\bar{P}_i > 0$ because of the exponential growth. To avoid problems, we stabilize our shooting algorithm in the same way as for the second order method in [28]. We divide $e^{h_i \bar{A}_i}$ by its spectral radius at each step. We set then

$$z_i^L = \gamma y_i / [\rho_0 \rho_1 \ldots \rho_{i-1}] \tag{4.2}$$

and

$$z_i^{L'} = \gamma p y_i' / [\rho_0 \rho_1 \ldots \rho_{i-1}] \tag{4.3}$$

with γ a factor which will be used to normalize the eigenfunction. In terms of these new variables, we have the recursion

$$\mathbf{z}_{i+1}^L = (e^{h_i \bar{A}_i}(I + N_1 + N_2)/\rho_i)\mathbf{z}_i^L, \quad i = 0, \ldots, n-1.$$

To start the recursion we take

$$z_0^L = -a_2/\max(|a_1|, |a_2|), \quad z_0^{L'} = a_1/\max(|a_1|, |a_2|),$$

which satisfies the boundary condition (1.2) in a. At b a so-called miss-distance (or mismatch) function $\phi(E)$ can be defined which describes the extent to which the boundary condition (1.2) fails to be satisfied there:

$$\phi(E) = b_1 z_n^L + b_2 z_n^{L'}. \tag{4.4}$$

The eigenvalues are the zeros of ϕ which may be found by a standard rootfinder.
Analogs of all of the above forward equations can be defined for the backwards recurrence. We define

$$z_i^R = \kappa y_i/[\rho_i \ldots \rho_{n-1}], \quad z_i^{R'} = \kappa p y'_i/[\rho_i \ldots \rho_{n-1}] \tag{4.5}$$

to have the stable recursion (for $i = n-1, \ldots, 1$)

$$\mathbf{z}_i^R = (e^{h_i \bar{A}_i}(I + N_1 + N_2)/\rho_i)^{-1}\mathbf{z}_{i+1}^R. \tag{4.6}$$

Note that the inverse of the transfer matrix $T = (e^{h_i \bar{A}_i}(I + N_1 + N_2)/\rho_i)$ can be obtained as (see [13])

$$T^{-1} = \begin{pmatrix} T_{2,2} & -T_{1,2} \\ -T_{2,1} & T_{1,1} \end{pmatrix}$$

since $\det(T) = 1$. The recursion is started with

$$w_n = -b_2/\max(|b_1|, |b_2|), \quad w'_n = b_1/\max(|b_1|, |b_2|).$$

If only an eigenvalue estimate is desired, then using only the forward recursions is sufficient and no storage of the scaling parameters ρ is necessary. For the computation of an eigenfunction, we combine above scaled recurrences and the ρ scale factor at each recursion step needs then to be stored. Shooting from the two ends towards the middle, yields an output satisfying the boundary conditions exactly, and lessens the possibility of underflows and overflows in the recursions. Let m be the index of an interior mesh point, called the matching point. We calculate z_i^L for $i = 0, \ldots, m$ and z_i^R for $i = n, n-1, \ldots, m$. In order to have smooth solutions in the matching point, the following must be satisfied

$$\rho_0 \rho_1 \ldots \rho_{m-1} z_m^L/\gamma = \rho_m \rho_{m+1} \ldots \rho_{n-1} z_m^R/\kappa \tag{4.7}$$

and

$$\rho_0 \rho_1 \ldots \rho_{m-1} z_m^{L'}/\gamma = \rho_m \rho_{m+1} \ldots \rho_{n-1} z_m^{R'}/\kappa. \tag{4.8}$$

Combining (4.2),(4.5) and (4.7) we have

$$y_i = \begin{cases} \rho_0 \rho_1 \ldots \rho_{i-1} z_i^L/\gamma, & i \le m \\ (\rho_0 \rho_1 \ldots \rho_{m-1})(z_m^L/z_m^R) z_i^R/(\gamma \rho_m \ldots \rho_{i-1}), & m < i \end{cases}$$

with an analogous formula for $(py')_i$. Since we expect z_m^L to be O(1) because of the induced stability of the recursions, a good choice of γ to minimize problems of scale is

$$\gamma = \rho_0 \ldots \rho_{m-1}$$

so that $y_m = O(1)$ as well. This corresponds to normalize the approximate eigenfunction so that

$$y_i = \begin{cases} z_i^L/(\rho_i\rho_{i+1}\dots\rho_{m-1}), & i < m \\ z_m^L, & i = m \\ (z_m^L/z_m^R)z_i^R/(\rho_m\dots\rho_{i-1}), & m < i \end{cases}$$

and similarly for the approximate derivative.
It is preferable to choose the index m where matching occurs to be in a stable region (where $(E\bar{w}-\bar{q})\bar{P} > 0$). Therefore we choose m as the index of the interval for which $(E\bar{w}-\bar{q})\bar{P}$ is maximized. For problems in Liouville normal (i.e. Schrödinger) form this corresponds to matching near the minimum of the potential.

4.2. Zero counting procedure. The search for the eigenvalues uses a standard zero finder applied to a mismatch function $\phi(E)$. First, however, the desired eigenvalue, say E_k must be isolated from the remaining ones. We need a way to find a good initial guess for the eigenvalue which can then be passed to the zero finder procedure.
Counting the number of zeros of the solution during the integration, allows us to converge on a specific eigenvalue E_k. We generalize here the procedure from [14] for Schrödinger problems to Sturm-Liouville problems. For more details on the Prüfer theory [24] used, see [25].
We write the solution and its derivative in the scaled Prüfer form in the following way:

$$y(x) = S^{-1/2}\varrho\sin\theta, \quad y'(x) = S^{1/2}\varrho\cos\theta$$

see [25]. Both ϱ and θ depend on x and E. We choose as a global scaling function S

$$S = \begin{cases} 1, & \text{if } (E\bar{w}_t - \bar{q}_t)/\bar{P}_t < 1, \\ \sqrt{(E\bar{w}_t - \bar{q}_t)/\bar{P}_t} & \text{if } (E\bar{w}_t - \bar{q}_t)/\bar{P}_t \geq 1 \end{cases}$$

where t is the index of the step where $(E\bar{w}_t-\bar{q}_t)/\bar{P}_t$ is maximal. The choice of this scaling function is based on the observations discussed in [25]. The knowledge of θ can help us for our purpose since the number of zeros over the current step $[x_i, x_i + h_i]$ equals the number of integers in the interval $[\theta(x_i)/\pi, \theta(x_{i+1})/\pi]$. Suppose now θ is known in the endpoint x_i and we want to obtain $\theta(x_{i+1})$. We distinguish two cases: (i) the "well" case $(E\bar{w}_i - \bar{q}_i)\bar{P}_i > 0$ and (ii) the "barrier" case $(E\bar{w}_i - \bar{q}_i)\bar{P}_i \leq 0$.

(i) We take as local scale factor $S_i = \omega_i = \sqrt{(E\bar{w}_i - \bar{q}_i)\bar{P}_i}$. The Prüfer phase θ_i over the interval $[x_i, x_i + h_i]$ is of the form

$$\theta_i(x) = \omega_i(x - x_i) + \varphi(x), \tag{4.9}$$

where $\varphi(x)$ is close to the constant value $\varphi(x_i) = \arctan(\omega_i y(x_i)/y'(x_i))$. If it is assumed that $\varphi(x)$ remains unchanged over $[x_i, x_{i+1}]$, then the number of zeros of y in $[x_i, x_{i+1}]$ is the number of integers in the interval $(\varphi(x_i)/\pi, (\omega h_i + \varphi(x_i))/\pi)$ (this is the procedure used in SLEDGE). Ixaru suggested in [14] to add a correction in the phase. This means, that we write θ_i as

$$\theta_i(x) = \omega_i(x - x_i) + \varphi_i(x_i) + \Delta\varphi(x),$$

with $\Delta\varphi(x_i) = 0$. The value of $\Delta\varphi(x_{i+1})$ is calculated using the available data $y(x_i)$, $y'(x_i)$, $y(x_{i+1})$ and $y'(x_{i+1})$ (actually we use the approximations given by the modified Neumann method). Specifically, we compute

$$\varphi_i(x_{i+1}) = \arctan(\omega_i y(x_{i+1})/y'(x_{i+1})).$$

If n_φ is the integer part of $(\omega_i h_i + \varphi_i(x_i))/\pi$, then $\bar{\varphi} = \omega_i h_i + \varphi_i(x_i) - n_\varphi\pi$ lies between 0 and π. $\Delta\varphi(x_{i+1})$ is then given by

$$\Delta\varphi(x_{i+1}) = \begin{cases} \varphi_i(x_{i+1}) - \bar{\varphi} + \pi, & \text{if } \varphi_i(x_{i+1}) - \bar{\varphi} < -\pi/2, \\ \varphi_i(x_{i+1}) - \bar{\varphi} - \pi, & \text{if } \varphi_i(x_{i+1}) - \bar{\varphi} > \pi/2, \\ \varphi_i(x_{i+1}) - \bar{\varphi}, & \text{otherwise,} \end{cases}$$

Once the values of θ_i at x_i and x_{i+1} are known, the values of θ corresponding to the original global S are obtained by the rescaling procedure described in Section 5.2.4 of [25].

(*ii*) In this case the values of θ corresponding to the global S are obtained directly. We have

$$\theta(x_i) = \arctan(Sy(x_i)/y'(x_i))$$

If $y(x_i)y(x_{i+1}) < 0$ then y has a (single) zero in the interval (x_i, x_{i+1}) and thus we take

$$\theta(x_{i+1}) = \begin{cases} \theta^1 + \pi, & \text{if } \theta(x_i)\theta^1 > 0, \\ \theta^1 & \text{otherwise} \end{cases}$$

with $\theta^1 = \arctan(Sy(x_{i+1})/y'(x_{i+1}))$. If $y(x_i)y(x_{i+1}) \geq 0$ then there are no zeros of y in (x_i, x_{i+1}) and therefore we take

$$\theta(x_{i+1}) = \begin{cases} \theta^1 + \pi, & \text{if } \theta(x_i) > 0 \text{ and } \theta^1 < 0, \\ \theta^1 - \pi, & \text{if } \theta(x_i) < 0 \text{ and } \theta^1 > 0, \\ \theta^1 & \text{otherwise} \end{cases}$$

Using the one step increment of $\theta, \Delta_i = \theta(x_{i+1}) - \theta(x_i)$, the global Prüfer phase can be constructed in a simple way. The values of $\theta(x_m)$ obtained from the forward (from a to x_m) and backward directions (from b down to x_m) are given by

$$\theta_L(x_m) = \theta_a + \sum_{i=0}^{m-1} \theta_i, \quad \theta_R(x_m) = \theta_b - \sum_{i=m}^{n-1} \Delta_i$$

where θ_a and θ_b correspond to the values of θ_L and θ_R in a and b such that $\theta_a \in [0, \pi), \theta_b \in (0, \pi]$:

$$\theta_a = \begin{cases} \theta(a), & \text{if } \theta(a) \geq 0, \\ \theta(a) + \pi, & \text{if } \theta(a) < 0, \end{cases}$$

and

$$\theta_b = \begin{cases} \theta(b), & \text{if } \theta(b) > 0, \\ \theta(b) + \pi, & \text{if } \theta(b) \leq 0. \end{cases}$$

The number of zeros in the solution is then given by $n_z = (\theta_L(x_m) - \theta_R(x_m))/\pi$. The eigenvalue E_k is thus identified as that E-value for which $\Delta\theta(E) = \theta_L(x_m) - \theta_R(x_m) = k\pi$.

4.3. Mesh selection and error estimation. Of course using a uniform mesh is rarely a good idea, e.g., when dealing with (truncated) singular problems. For automatic software a step size selection algorithm should be used, which places a higher concentration of mesh points in regions with a large variation in the coefficient functions and selects larger steps elsewhere. Here we present a way to construct a good initial mesh which is bisected for a-posteriori error control.
The error in the eigenvalue can typically be expressed in terms of the corresponding eigenfunction. When in the mesh selection this error is equidistributed, some kind of iteration is needed with approximate eigenvalues and eigenfunctions. For reasons of efficiency, we try to avoid this redistribution and to use a different heuristic. The initial mesh is chosen to equidistribute

$$\epsilon_i = h_i^2 \max\Big(|1/\hat{p} - 1/p|, |\hat{q} - q|, |\hat{w} - w|\Big)_i$$

where $1/\hat{p}, \hat{q}$ and $\hat{w}$ are the second degree polynomials which approximate $1/p, q$ and w in the ith interval. This error in approximating the coefficient functions appears as a major factor in both eigenvalue and eigenfunction errors. Note that this error is independent of E and $y(x)$ and can thus be computed a priori. The algorithm uses the standard idea of equidistribution: to try to choose the mesh $x_i, i = 0, \ldots, n$ so that for all i, $\epsilon_i \approx tol$ with tol a user input tolerance. Meshing is done once and for all for a sequence of integrations for the shooting method. Also data associated to the mesh, as $\bar{P}, \bar{q}, \bar{w}, P_1, Q_1, \ldots, W_2$, can

be computed before the actual shooting, which means that no further function evaluations are needed during shooting. Once the mesh has been generated, this mesh is bisected (i.e. the midpoint of each subinterval is inserted as mesh point) to generate a second (reference) mesh. When an eigenvalue approximation $E_k(h)$ is computed, a second approximation $E_k(h/2)$ is obtained over the reference mesh. The shooting algorithm for $E_k(h/2)$ is started using $E_k(h)$ as initial guess. The difference between the two approximations $E_k(h)$ and $E_k(h/2)$ is used for error control. If this difference exceeds a *tol*-related value then the mesh is refined, i.e. bisected.

5. Singular problems. Both problems defined on an infinite integration interval and problems with singular endpoints require a special numerical treatment. In these cases an interval truncation procedure must be adopted. Different algorithms are implemented in the available SLP library codes to determine a truncated endpoint and appropriate boundary conditions to give a prescribed accuracy (see [1, 2, 3, 20, 25, 28]). The SLEDGE package even has algorithms for automatically classifying the nature of the problem, regular or singular, limit-circle singularity or limit-point singularity and so on. This classification information is important to determine whether or not there is a continuous spectrum, when there are eigenvalues and how many, and what boundary condition should be imposed at a singular endpoint. The SLEDGE classification algorithm and other techniques to handle singular endpoints automatically are essentially independent of the shooting method and can equally well be combined with higher order modified Neumann methods as with the second order method.
The higher order modified Neumann methods also preserve the interesting advantange of the Pruess method that they allow a very simple interval truncation algorithm for singular problems. Evaluating the coefficients only at the Legendre nodes effectively regularizes the problem and can be regarded as truncating the integration interval at the first and last Legendre node of the initial and final intervals respectively. Every time a mesh interval is bisected, these implicit truncation points move closer to the singular endpoint. The boundary conditions however are always applied at the original endpoints.
We handled the boundary conditions for a singular endpoint a as follows (see [25]). Suppose $\bar{q}_1$, $\bar{w}_1$ and $\bar{P}_1$ are the constant approximations of the $q, w, P = 1/p$ functions in the first mesh interval. If $E\bar{w}_1 - \bar{q}_1 < 0$ then there are two solutions, one growing exponentially, and one decaying. The boundary condition $y(a) = 0$ is then taken in an attempt to capture the subdominant behaviour of the eigenfunction near $x = a$. If $E\bar{w}_1 - \bar{q}_1 \geq 0$, we look which of $\bar{P}_1$ and $E\bar{w}_1 - \bar{q}_1$ is bigger. If $\bar{P}_1 > E\bar{w}_1 - \bar{q}_1$, we assume that $1/p$ (and not $|Ew - q|$) fails to be integrable at a and we take $p(a)y'(a) = 0$ as boundary condition. When $\bar{P}_1 < E\bar{w}_1 - \bar{q}_1$ we take $y(a) = 0$. Similarly for a singular endpoint b, we take $y(b) = 0$ when $E\bar{w}_{n-1} - \bar{q}_{n-1} < 0$ or $\bar{P}_{n-1} < E\bar{w}_{n-1} - \bar{q}_{n-1}$ and $p(b)y'(b) = 0$ in the other case.
An infinite endpoint is transformed to zero by the local change of variable $t = -1/x$. This change of variable is used on the first (when $a = -\infty$) or last (when $b = \infty$) initial mesh interval.

6. Results.

6.1. Matlab code. The sixth order modified Neumann scheme and the shooting method discussed in Section 4 are implemented in a Matlab package. This package can be downloaded from
`http://www.nummath.ugent.be/SLsoftware`
and allows the automatic solution (i.e. computation of eigenvalues and eigenfunctions) of Sturm-Liouville problems. Singular problems are handled as discussed in the previous section. Some test problems are predefined, illustrating the usage of the different routines. The package was used to obtain the results presented in this section.

6.2. Experiments.

6.2.1. Eigenvalue computations. We consider again the two test problems from Section 3: the Collatz problem (3.8) and the Paine problem (3.7). Table 6.1 illustrates how much the sixth order modified Neumann method is an improvement over the second order method: the Pruess method. Both methods were applied on an equidistant mesh. The Pruess method needs only one function evaluation for each coefficient function per mesh interval, where the sixth order method needs three. However the higher order method needs much less mesh intervals to attain (approximately) the same accuracy and as a result the total number of function evaluations is substantially smaller. All function evaluations are

TABLE 6.1

The relative errors in some eigenvalue computations for the Collatz Problem and the Paine problem, computed by the second order Pruess method and the sixth order Modified Neumann Method (MNM6). nsteps is the number of mesh intervals in the equidistant mesh, nfevs the number of function evaluations of the coefficient functions needed to compute the eigenvalues over this equidistant mesh and T(s) denotes the CPU time needed in seconds (Matlab implementation). The notation $a(-b)$ stands for $a\,10^{-b}$.

	Collatz				Paine		
k	E_k	Pruess	MNM6	k	E_k	Pruess	MNM6
0	70.1836836876	2.1(-6)	1.9(-9)	0	1.5198658211	3.4(-6)	1.2(-9)
25	47444.18579306	2.1(-6)	5.0(-6)	5	37.9644258619	5.6(-6)	6.0(-8)
50	182548.2030025	2.1(-6)	3.3(-6)	10	123.4977068009	6.0(-6)	2.1(-7)
75	405381.9379249	2.1(-6)	3.1(-6)	20	443.8529598352	6.2(-6)	7.1(-7)
100	715945.489746	2.2(-6)	4.1(-6)	30	963.9644462621	6.2(-6)	1.7(-6)
125	1114238.858465	2.2(-6)	3.0(-6)	40	1684.0120143379	6.2(-6)	2.4(-6)
150	1600262.044083	2.3(-6)	7.5(-7)	50	2604.0363320246	6.2(-6)	5.2(-6)
nsteps		1024	32			1024	48
nfevs		3072	288			3072	432
T(s)		2.37	1.33			0.89	0.58

TABLE 6.2

The relative error in some eigenvalue computations for the Collatz problem. Three different software codes were used: the Fortran packages SLEDGE and SLCPM12 and the Matlab implementation of the sixth order modified Neumann method. For SLCPM12 and MATSLEMN the number of steps (nsteps) is shown for the mesh on which the first, resp. 151th eigenvalue is computed, as well as the number of function evaluations (nfevs) of the coefficient functions needed. For SLEDGE the number of steps (ns_0) in the 'level 0 mesh' and the number of extrapolation levels ([nl]) needed to compute the first, resp. 151th eigenvalue is displayed.

k	E_k	SLEDGE	SLCPM12	MATSLEMN
0	70.1836836876	1.0(-7)	2.3(-10)	1.4(-10)
25	47444.18579306	9.7(-8)	1.5(-9)	1.2(-8)
50	182548.2030025	1.2(-7)	2.9(-9)	1.6(-7)
75	405381.9379249	7.1(-8)	3.4(-9)	2.1(-7)
100	715945.489746	9.3(-8)	2.3(-8)	3.4(-7)
125	1114238.858465	2.1(-7)	4.5(-9)	5.4(-7)
150	1600262.044083	8.8(-8)	2.3(-8)	3.7(-7)
ns_0[nl] / nsteps (0)		4[4]	1	26
ns_0[nl] / nsteps (150)		4[9]	1	52
nfevs(0)			3780	991
nfevs(150)			3780	1927

performed before the actual shooting. Due to the smaller number of mesh intervals the shooting process itself is also faster for the sixth order method.

In Tables 6.2 and 6.3 results are given which were obtained with three different software packages, all three in some way implementing a modified Neumann method or a related method: the Fortran packages SLEDGE and SLCPM12 and the matlab package implementing the sixth order method discussed in this paper (called MATSLEMN in short here). The input tolerances for the different packages were set in a way to obtain approximately the same accuracy in the results. It is clear that all packages have the power to compute eigenvalues upto high accuracy, also those with a high eigenvalue index. Due to the use of different programming languages, it is difficult to compare timings. Moreover the authors of SLEDGE chose not to store function evaluations [1], which complicates a fair comparison between the packages even more. However from table 6.1 we already know that the higher order methods need a substantially smaller number of mesh intervals, and as a consequence a smaller number of function evaluations to reach the same accuracy. To give some indication, we show for the SLCPM12 and MATSLEMN package the number of steps in the mesh on which the lowest and largest eigenvalue was computed. The SLEDGE algorithm gets its accuracy from repeated mesh bisection and Richardson extrapolation: solutions are computed for a sequence of meshes (called levels), until sufficient accuracy is attained. Each level's mesh consists of the points from the previous level's mesh united with its midpoints. For SLEDGE we display

[1] due to the absence of dynamical array allocation in Fortran 77

TABLE 6.3

The relative error in some eigenvalue computations for the Paine problem in Sturm-Liouville form. Three different software codes were used: the Fortran packages SLEDGE and SLCPM12 and the Matlab implementation of the sixth order modified Neumann method. For SLCPM12 and MATSLEMN the number of function evaluations (nfevs) and the number of steps (nsteps) in the mesh is shown for the first, resp. 51th eigenvalue. For SLEDGE the number of steps (ns_0) in the 'level 0 mesh' is displayed as well as the number of levels ([nl]) needed.

k	E_k	SLEDGE	SLCPM12	MATSLEMN
0	1.5198658211	3.6(-9)	2.9(-9)	7.6(-11)
5	37.9644258619	4.2(-9)	1.3(-9)	6.6(-9)
10	123.4977068009	4.6(-9)	5.0(-10)	8.4(-9)
20	443.8529598352	4.7(-9)	1.1(-9)	9.7(-10)
30	963.9644462621	5.0(-9)	1.0(-9)	1.9(-9)
40	1684.0120143379	2.9(-9)	1.1(-10)	3.1(-9)
50	2604.0363320246	5.0(-9)	7.3(-11)	5.0(-9)
ns_0[nl] / nsteps (0)		4[5]	6	80
ns_0[nl] / nsteps (50)		4[9]	6	160
nfevs(0)			39458	2665
nfevs(50)			39458	5545

TABLE 6.4

The absolute error in some eigenvalue computations for the Schrödinger Mathieu problem. Three different software codes were used: the Fortran packages SLEDGE and SLCPM12 and the Matlab implementation of the sixth order modified Neumann method. For SLCPM12 and MATSLEMN the number of function evaluations (nfevs) and the number of steps (nsteps) in the mesh is shown for the first, resp. 51th eigenvalue. For SLEDGE the number of steps (ns_0) in the 'level 0 mesh' is displayed as well as the number of levels ([nl]) needed.

k	E_k	SLEDGE	SLCPM12	MATSLEMN
0	-0.11024881665	2.5(-8)	4.7(-7)	2.0(-8)
5	36.01428991063	4.2(-7)	1.9(-8)	2.6(-7)
10	121.0041667613	6.7(-6)	5.5(-8)	5.6(-7)
20	441.0011363655	2.5(-5)	6.8(-9)	2.3(-7)
30	961.0005208335	5.4(-5)	6.3(-9)	2.7(-8)
40	1681.0002976191	9.2(-5)	1.9(-9)	2.5(-7)
50	2601.0001923077	1.4(-4)	3.0(-10)	5.1(-7)
ns_0[nl] / nsteps (0)		4[5]	4	33
ns_0[nl] / nsteps (50)		4[8]	4	33
nfevs(0)			264	1435
nfevs(50)			264	1435

in the tables the number of steps in the 'level 0 mesh' and the number of extrapolation levels needed.

To compare our current scheme with a PPM, we use the SLCPM12 package and not the matlab software package MATSLISE. The reason why we didn't include this package in our comparison, is that the Liouville transformation is performed by symbolic computation in MATSLISE using the matlab symbolic toolbox, which makes it difficult to measure its cost. The SLCPM12 code applies the Liouville transformation to convert the Sturm-Liouville problem in a Schrödinger problem, and it is then this Schrödinger problem which is solved by a PPM of order twelve. The SLCPM12 method needs less mesh intervals than the sixth order method in MATSLEMN to reach a similar accuracy. This is a consequence of the higher order of the method and the fact that a Schrödinger problem with a smooth potential is solved for these problems: e.g for the Collatz problem a Schrödinger problem with zero potential is obtained after Liouville's transformation. However the Liouville transformation is expensive and requires extra function evaluations and computational time, as can be seen from the number of function evaluations of the coefficient functions shown in Tables 6.2 and 6.3. Note that in contrast to the experiment in Table 6.1, the coefficient functions are now also evaluated during mesh selection and error control.

In Table 6.4 the experiment was repeated for a Sturm-Liouville problem in the Schrödinger form: the Mathieu problem (3.12). The PPM (used in SLCPM12) were especially constructed for regular Schrödinger problems and form clearly the best choice to solve these problems. The higher order method in SLCPM12 needs less mesh intervals than the sixth order method in MATSLEMN and only evaluates the potential function. MATSLEMN and SLEDGE are both aimed for a more general class of problems and

also evaluate the p and w function. Where a PPM is the method to be prefered for a regular Schrödinger problem, our modified Neumann methods is expected to perform better on Sturm-Liouville problems where the Liouville's transformation is expensive or impossible to apply, e.g. problems with discontinuities, with a large variation in the coefficient functions, singular problems,.... As mentioned before, the higher order modified Neumann methods share the interesting property with the Pruess method that singular problems can be easily dealt with. Table 6.5 shows results for some typical but very different singular problems. The set of chosen problems is:

1. The hydrogen atom equation
$$y'' = (-1/x + 2/x^2 - E)y, \quad x \in [0, +\infty[. \tag{6.1}$$
with eigenvalues $E_k = -1/(2k+4)^2,\ k = 0, 1, \ldots$. The main numerical difficulty for an automatic code is that a suitable value for the right-hand truncation point depends strongly on k.
2. The harmonic oscillator
$$y'' = (x^2 - E)y, \quad x \in [-\infty, \infty] \tag{6.2}$$
with eigenvalues $E_k = 2k + 1, k = 0, 1, \ldots$.
3. The Morse oscillator
$$y'' = \left(\frac{2}{x^2} - 2000(2e^{-1.7(x-1.3)} - e^{-3.4(x-1.3)}) - E\right) y, \quad x \in [0, +\infty] \tag{6.3}$$
with precisely 26 eigenvalues, all negative. The right-hand truncation point does not depend very much on the eigenvalue index, in contrast to the hydrogen problem.
4. The Legendre equation
$$-((1 - x^2)y')' = Ey \quad x \in [-1, 1] \tag{6.4}$$
with eigenvalues $E_k = k(k + 1), k = 0, 1 \ldots$. Both ends are limit-circle.
5. The Bessel equation in its standard (non-Schrödinger) form
$$-(xy')' + \frac{1/4}{x} y = Exy, \quad x \in [0, 1] \tag{6.5}$$
with eigenvalues $E_k = ((k + 1)\pi)^2,\ k = 0, 1, \ldots$.

The same test problems were used in the numerical experiments in chapter 8 of [25]. The table includes results produced by SLEDGE and MATSLEMN. These singular problems cannot be solved by SLCPM12. In the table 'error' is $|E_{\text{exact}} - E_{\text{approx}}|/\max(1, |E_{\text{exact}}|)$. * indicates an $IFLAG = -1$ exit of SLEDGE, meaning that too many levels were needed for the eigenvalue calculation. The default maximum number of levels, which is 10 levels, was not changed. Good accuracy is obtained for both low and higher eigenvalues. Since a higher eigenvalue needs a larger truncated integration interval, it can be expected that the number of steps in the mesh increases with k. However in most cases the higher order modified Neumann method needs a substantially smaller number of steps than the SLEDGE algorithm (especially for the higher eigenvalues), which often needs a high number of extrapolation levels on these problems to reach a certain accuracy.

6.2.2. Eigenfunction computations. An eigenfunction y_k is represented directly by the solution of the differential equation for the final converged E_k value. Figure 6.1 shows approximations for a selection of eigenfunctions of the Collatz problem, obtained by applying the sixth order modified Neumann method over a mesh with only 31 mesh points. For this Collatz problem the explicit expressions of the eigenfunctions are known, which allows us to study the error in our approximations. Even for the higher eigenfunctions, e.g. y_{500}, good approximations are obtained even though the mesh step size is larger than the solution wave length. To give an idea about the exact error in the eigenfunctions shown, Table 6.6 contains the maximum of the absolute errors in the mesh points.

Figure 6.2 shows some eigenfunctions of the hydrogen atom equation. Note the great variation in horizontal scale between the different eigenfunctions of this problem.

TABLE 6.5

The numerical solution of some singular problems in SLEDGE and MATSLEMN (matlab implementation of sixth order modified Neumann method).

k	E_k	SLEDGE		MATSLEMN	
		error	ns_0[nl]	error	nsteps
		hydrogen problem			
		tol=10^{-9}		tol=10^{-6}	
0	-0.062500000000	1.4(-10)	20[4]	1.2(-11)	208
10	-0.001736111111	2.2(-10)	20[6]	1.0(-13)	312
100	-0.000024029220	4.7(-11)	20[8]	2.1(-17)	564
1000	-0.000000249003	7.5(-11)	20[6]	2.6(-15)	758
		Bessel problem			
		tol=10^{-8}		tol=10^{-8}	
0	9.8696044011	4.3(-9)	8[4]	1.2(-10)	148
10	1194.22213253	4.7(-10)	8[6]	1.5(-10)	258
100	100679.8344955	5.3(-10)	8[9]	4.0(-10)	264
		Harmonic oscillator			
		tol=10^{-8}		tol=10^{-8}	
0	1	4.5(-12)	4[6]	1.6(-9)	52
10	21	2.6(-10)	4[7]	6.5(-11)	102
100	201	5.5(-8)*	5[10]	6.8(-9)	102
1000	2001	4.7(-4)*	5[10]	5.3(-9)	402
		Morse oscillator			
		tol=10^{-10}		tol=10^{-6}	
0	-1923.529655114	1.4(-9)	9[8]	3.3(-11)	223
10	-721.2590105685	3.7(-8)	9[9]	1.0(-9)	304
20	-97.04816409520	2.0(-7)	9[10]	9.8(-10)	354
		Legendre equation			
		tol=10^{-8}		tol=10^{-8}	
0	0	1.2(-10)	12[2]	1.0(-17)	211
10	110	1.2(-10)	12[7]	2.6(-11)	217
100	10100	4.7(-10)	12[9]	5.2(-10)	432

TABLE 6.6

The error in the eigenfunctions of the Collatz problem shown in Figure 6.1. For each eigenfunction the maximum (absolute) error over the different meshpoints is shown.

k	E_k	$\max(\lvert y-\bar{y}\rvert)$
0	70.1836836876	1.7(-13)
10	8492.246275782	2.7(-11)
50	182548.20300255	2.7(-8)
100	715945.48974587	3.7(-8)
250	4421652.955542	6.3(-8)
500	17616217.41709	2.8(-8)

7. Conclusion. In this paper we presented a class of coefficient approximation methods which extend the ideas of the Pruess method to higher order methods. Where many algorithms are defined for problems in Schrödinger form and can only be applied on Sturm-Liouville problems after a Liouville transformation, the presented method is applied directly on the Sturm-Liouville problem. This makes it especially interesting for those problems where a Liouville transformation is problematic or expensive. Where the second-order Pruess method needs many extrapolation levels to reach a certain accuracy, our higher order methods seem to be more efficient. We showed that the presented method is well suited to be applied in a automatic code to solve a large class of Sturm-Liouville eigenvalue problems.

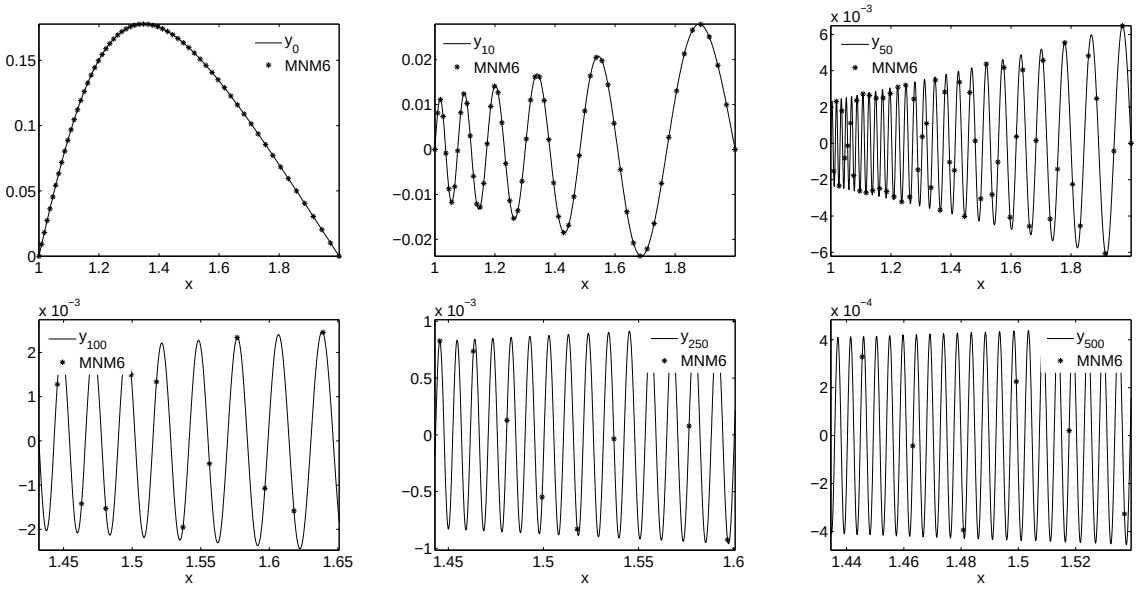

FIG. 6.1. *Sample eigenfunction computations of Collatz problem for* $k = 0, 10, 50, 100, 250, 500$. *For* $k = 100, 250, 500$ *only part of the eigenfunction is shown. The mesh contains only 31 mesh points.*

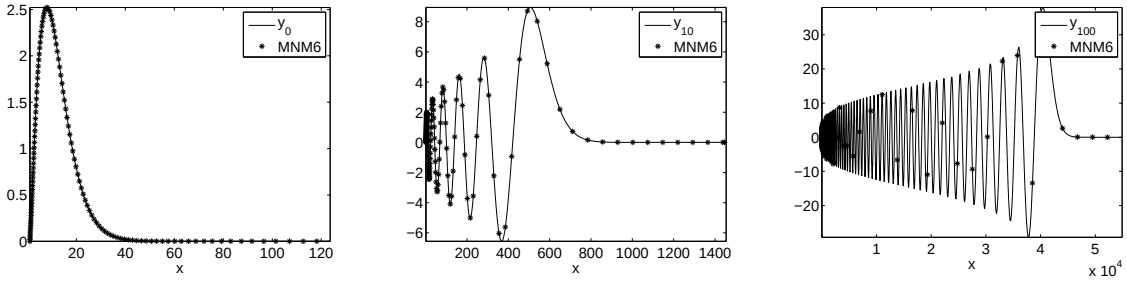

FIG. 6.2. *Sample eigenfunction computations of hydrogen atom equation, for* $k = 0, 10, 100$.

REFERENCES

[1] P. B. BAILEY, M. K. GORDON AND L. F. SHAMPINE, *Solving Sturm-Liouville eigenproblems*, Technical Report SAND76-0560, Sandia National Laboratories, Albuquerque, (1976).

[2] P. B. BAILEY, M. K. GORDON AND L. F. SHAMPINE, *Automatic solution of the Sturm-Liouville problem*, ACM Trans. Math. Software, 4 (1978), pp. 193–207.

[3] P. B. BAILEY, P. B. EVERITT AND A. ZETTL, *The SLEIGN2 Sturm-Liouville code*, ACM Trans. Math. Software, 21 (2001), pp. 143–192.

[4] L. COLLATZ, *Eigenwertprobleme und ihre numerische Behandlung*, Chelsea, New York, 1948.

[5] I. DEGANI AND J. SCHIFF, *RCMS: right correction Magnus series approach for oscillatory ODEs*, J. Comput. Appl. Math., 193 (2006), pp. 413–436.

[6] A. ISERLES AND S. P. NØRSETT, *On the solution of linear differential equations in Lie groups*, Phil. Trans. R. Soc. Lond. A, 357 (1999), pp. 983–1019.

[7] A. ISERLES, H.Z. MUNTHE-KAAS, S.P. NØRSETT AND A. ZANNA, *Lie-group methods*, Acta Numerica, 9 (2000), pp. 215–365.

[8] A. ISERLES, *On the global error of discretization methods for highly-oscillatory ordinary differential equations*, BIT, 42 (2002), pp. 561–599.

[9] A. ISERLES, *Think globally, act locally: Solving highly-oscillatory ordinary differential equations*, Appld Num. Anal., 43 (2002), pp. 145–160.

[10] A. ISERLES, *On the method of Neumann series for highly oscillatory equations*, BIT, 44 (2004), pp. 473–488.

[11] A. ISERLES, *On the numerical quadrature of highly oscillatory integrals I: Fourier transforms*, IMA J. Numer. Anal., 24 (2004), pp. 365–391.

[12] L. GR. IXARU, *Numerical Methods for Differential Equations and Applications*, Reidel, 1984.

[13] L. Gr. Ixaru, H. De Meyer and G. Vanden Berghe, *CP methods for the Schrödinger equation, revisited*, J. Comput. Appl. Math., 88 (1997), pp. 289–314.
[14] L. Gr. Ixaru, H. De Meyer and G. Vanden Berghe, *SLCPM12 - A program for solving regular Sturm-Liouville problems*, Comput. Phys. Comm., 118 (1999), pp. 259–277.
[15] L. Gr. Ixaru, *CP methods for the Schrödinger equation*, J. Comput. Appl. Math., 125 (2000), pp. 347–357.
[16] P. Kosowski, *The relative error in the Pruess method for Sturm-Liouville problems*, Linear Alg. Appl., 309 (2000), pp. 325–337.
[17] V. Ledoux, M. Van Daele and G. Vanden Berghe, *CP methods of higher order for Sturm-Liouville and Schrödinger equations*, Comput. Phys. Commun., 162 (2004), pp. 151–165.
[18] V. Ledoux, M. Van Daele and G. Vanden Berghe, *MATSLISE: a MATLAB package for the numerical solution of Sturm-Liouville and Schrödinger equations*, ACM Trans. Math. Software, 31 (2005), pp. 532–554.
[19] V. Ledoux, M. Van Daele and G. Vanden Berghe, *Efficient numerical solution of the one-dimensional Schrödinger eigenvalue problem using Magnus integrators*, IMA J. Numer. Anal., to appear.
[20] M. Marletta and J. D. Pryce, *A new multipurpose software package for Schrödinger and Sturm-Liouville computations*, Comput. Phys. Commun., 62 (1991), pp. 42–52.
[21] P.C. Moan, *Efficient approximation of Sturm-Liouville problems using Lie group methods*, Technical report, Cambridge: Department of Applied Mathematics and Theoretical Physics, University of Cambridge (1998).
[22] H. Munthe-Kaas and B. Owren, *Computations in a free Lie algebra*, Phil. Trans. R. Soc. Lond. A, 357 (1999), pp. 957–981.
[23] J. W. Paine, F. R. de Hoog and R. S. Anderssen, *On the correction of finite difference eigenvalue approximations for Sturm-Liouville problems*, Computing, 26 (1981), pp. 123–139.
[24] H. Prüfer, *Neue herleitung der Sturm-Liouvilleshen reihenentwicklung stetiger funktionen*, Math. Ann., 95 (1923), pp. 499–518.
[25] J. D. Pryce, *Numerical Solution of Sturm-Liouville Problems*, Oxford University Press, 1993.
[26] J. D. Pryce and M. Marletta, *Automatic solution of Sturm-Liouville problems using the Pruess method*, J. Comput. Appl. Math., 39 (1992), pp. 57–78.
[27] S. Pruess, *Estimating the Eigenvalues of Sturm-Liouville Problems by Approximating the Differential Equation*, SIAM J. Numer. Anal., 10 (1973), pp. 55–68.
[28] S. Pruess and C. T. Fulton, *Mathematical software for Sturm-Liouville problems*, ACM Trans. Math. Software, 19 (1993), pp. 360–376.
[29] A. Zettl, *Sturm-Liouville Theory*, American Mathematical Society, 2005.