Variable-step finite difference schemes for the solution of Sturm–Liouville problems

Pierluigi Amodio a,⇑, Giuseppina Settanni b

a Dipartimento di Matematica, Università di Bari, via E. Orabona 4, 70125 Bari, Italy
b Dipartimento di Matematica e Fisica “E. De Giorgi”, Università del Salento, via per Arnesano, 73047 Monteroni di Lecce, Lecce, Italy

Article info
Article history:
Available online 3 July 2014

Keywords:
Second order BVPs
Sturm–Liouville problems
Finite differences
High order
Variable stepsize

Abstract
We discuss the solution of regular and singular Sturm–Liouville problems by means of High Order Finite Difference Schemes. We describe a method to define a discrete problem and its numerical solution by means of linear algebra techniques. Different test problems are considered to emphasize the behavior of a code based on the proposed algorithm. The methods solve any regular or singular Sturm–Liouville problem, providing high accuracy and computational efficiency thanks to the powerful strategy of stepsize variation.

Introduction

Since many applications in quantum physics, quantum chemistry, science and industry are connected to Sturm–Liouville problems (SLPs), their solution has drummed up interest of several researches. Many codes have been developed to solve regular and singular problems. The first Fortran software package SLEIGN, introduced in [14,15,17,22], computes automatically the eigenvalues and the eigenfunctions of some classes of SLPs, using the Prüfer transformation and the oscillatory properties of the eigenfunctions. It requires the linear second order differential equation turned into a nonlinear first-order one of double size. Although the code is reliable, sometimes it is slow and fails on some problems too. For this reason, a new version of this code, named SLEIGN2 [16], has been developed in order to solve any self-adjoint, separated or coupled boundary conditions, and with nearly all the classification of the endpoints of the interval, giving both quantitative and qualitative knowledge of its properties. Based on the approximation of the coefficients introduced by Pruess [25], two other Fortran codes have been also proposed. Both packages produce estimates of the eigenvalues and the eigenfunctions for regular and singular problems classifying the endpoints and the spectrum of the eigenvalues. In particular, the code SLEDGE in [26] fulfils the control of the global error, while SLEIGN2 in [27] uses a Prüfer transformation. Recently, the MATLAB code MATSLISE, see [21], has been implemented for solving regular SLPs, one-dimensional Schrödinger equations and radial Schrödinger equations with a distorted Coulomb potential by using Constant Perturbation Methods [20]. Actually, the first four codes have a larger range of applicability than the last one, although the latter is more efficient in the outlined class.

All these codes use standard ODE techniques to solve the discrete problem associated to SLPs. Across the last years, a class of matrix methods has been also developed, which applies finite differences or finite elements reducing the continuous problem to a matrix eigenvalue problem. In order to improve the numerical estimate of the eigenvalues, asymptotic corrections are also adopted, see [13,24]. Following this approach, in [1–3] a family of boundary value methods (BVMs) [18] has been introduced to approximate simultaneously the eigenvalues and the eigenfunctions of SLPs.

⇑Corresponding author.
E-mail addresses: pierluigi.amodio@uniba.it (P. Amodio), giuseppina.settanni@unisalento.it (G. Settanni).
Based on this idea, in [7,8] we illustrate a different approach able to solve regular and singular SLPs classifying the endpoints of the interval as regular or singular, limit points (LP) or limit circles (LC), oscillatory (LCO) or nonoscillatory (LCNO). The underlying methods approximate the derivatives of SLPs separately, therefore the second order differential equation is not transformed into an equivalent first order system; this means lower computational cost and easier stepsize selection strategy. The mainly used schemes are high order central finite differences, while other additional formulae are considered depending on the Sturm–Liouville classification of the endpoints of the interval. The proposed approach appears efficient and robust to solve with constant stepsize a particular singular Sturm–Liouville problem arising from the numerical computation of the eigenvalues and eigenfunctions of the finite (truncated) Hankel transform (see [9]), important for numerous applications in signal and image processing, and also for the solution of a two-parameter singular Sturm–Liouville problem derived from the numerical simulation of the so called ‘whispering gallery’ modes (WGMs) occurring inside a prolate spheroidal cavity, see [10].

In this paper, stimulated by the previous results, we develop and test a code with variable stepsize and order, where the stepsize variation strategy takes advantage of an equidistribution of the error. Our aim is to point out the better efficiency and the accuracy gained with the variable stepsize. For this reason in Section 2 we introduce the numerical schemes and give some hints on their usage to approximating SLPs. In Section 3 we sketch the code which is tested in the last section for some well known regular and singular problems.

2. High Order Finite Difference Schemes

In the last few years High Order Finite Difference Schemes have been largely used to solve different problems concerning with differential equations of order greater than 1. We mention Initial Value Problems [4], Boundary Value Problems [5,6,11,12] and Sturm–Liouville problems [7–10]. In all these cases the proposed formulae show good stability properties and the developed codes turn out to be competitive with the existing software, in particular when the problem to be solved is stiff and the solution has a slope different from that of its derivatives.

The reason of the nice behavior depends on the good flexibility of the formulae which are chosen depending on the problem, the additional conditions and even the meshgrid and the point where the problem is discretized.

Suppose that the continuous solution \( y(x) \) of the differential equation is defined on the interval \([a, b]\). Given a generic set of discretization points

\[
X = \{ a = x_0 < x_1 < \cdots < x_n = b \},
\]

we define the following \((s + r)\)-step formula in \( x_i \)

\[
y^{(s)}(x_i) \simeq y^{(s)} = \frac{1}{h_i} \sum_{j=s}^{r} \alpha^{(s,r)}_{i,j} y_{i+j}, \quad h_i = x_i - x_{i-1}
\]

which depends on the integers \( s \) and \( r \), that is, on the stencil \( x_{i-s}, \ldots, x_{i+r} \) associated to (2). The coefficients \( \alpha^{(s,r)}_{i,j} \) are chosen such that the order \( p \) of the formula is maximum, that is \( p = s + r - v + 1 \) for general \( s \) and \( r \), and, due to the symmetry of (2), \( p = s + r - v + 2 \) if \( s = r, v \) is even and the stepsize \( h_i \) is fixed over the stencil.

Starting from second order BVPs, given a general continuous problem

\[
f(x, y, y', y'') = 0,
\]

we define a discrete nonlinear system

\[
f(x_i, y_i, y'_i, y''_i) = 0, \quad \text{for} \quad i = 1, \ldots, n - 1,
\]

that, combined with the boundary conditions, allows to compute a unique solution. In compact form we could consider

\[
Y^{(s)} = A_1 Y, \quad Y^{(s)} = \text{discrete approximation of each derivative, where } Y^{(s)} \text{ and } Y^{(r)} \text{ contain the unknowns } y_i \text{ and the associated derivatives } y^{(r)}_i, \quad \text{for } i = 1, \ldots, n - 1. \quad Y \text{ is the extension of } Y \text{ with the boundary values } y_0 \text{ and } y_n \text{, and } A_1 \text{ is the } (n - 1) \times (n + 1) \text{ matrix built with the coefficients of the approximations. Hence the continuous problem is approximated by its discrete counterpart}
\]

\[
f(X, Y, Y', Y'') = 0.
\]

Matrices \( A_1 \) have essentially a banded structure since the approximation in \( x_i \) of each derivative makes use of values \( y_j \) chosen around \( y_i \). The best choice for \( r \) and \( s \) is \( r = s \) even if in [12], in order to improve stability properties for singular perturbation problems, it is suggested to approximate the first derivative with \( s = r - 2 \) or \( s = r + 2 \) depending on the sign of the term multiplying \( y' \). A different strategy is necessary in the first and last gridpoints since there are not sufficient values to the left (or to the right) of \( x_i \) to obtain a symmetric stencil.

A Sturm–Liouville equation

\[
-(p(x)y')' + q(x)y = \lambda w(x)y
\]

is a second order ODE defined for \( x \in [a, b] \) and positive functions \( p(x) \) and \( w(x) \). The parameter \( \lambda \) is an unknown value (eigenvalue) which must be determined in order nonnull solutions (eigenfunctions) \( y(x) \) of (3) satisfying boundary conditions exist. All the eigenvalues are real. If they are defined in ascending order (\( \lambda_0 \) is the smallest one), then the eigenvector corresponding
to \( \lambda_k \) has \( k \) intersections with the x-axis. Since, for any given eigenvalue, it is possible to compute an infinite number of eigenfunctions, it is required that \( y(x) \) fulfills a normalization condition

\[
\int_a^b |y(x)|^2 \, dx = 1. \tag{4}
\]

We re-write (3) in the standard second order form

\[-p(x)y'' - p'(x)y' + q(x)y = \lambda w(x)y \tag{5}\]

and suppose for a moment that the problem is regular and that its solution is zero in the boundary points, that is

\[y(a) = y(b) = 0. \tag{6}\]

Then, based on the previous formulae, (5) and (4) are approximated by the discrete problem

\[
\begin{aligned}
& \{-PA_2 - P_1A_1 + Q\} Y = \lambda WY, \\
& \tilde{v}^T \tilde{Y}^2 = 1,
\end{aligned} \tag{7}
\]

where \( P, P_1, Q \) and \( W \) are diagonal matrices containing \( p(x_i), p'(x_i), q(x_i) \) and \( w(x_i) \), for \( i = 1, \ldots, n - 1 \), respectively, and \( \tilde{v} \) contains the coefficients of the quadrature formula discretizing (4). We remark that the order of this last formula does not entail a reduction of the order of the overall method since it represents a scaling factor used to fix a particular eigenfunction.

Matrices \( A_1 \) and \( A_2 \) contain the coefficients of the methods. Due to (6) we delete their first and last columns (and the first and last element of \( \tilde{v} \)) obtaining the square matrices \( A_1 \) and \( A_2 \) (and vector \( v \)). Problem (7) reduces to the algebraic eigenvalue problem

\[
\begin{aligned}
& MY \equiv W^{-1}(-PA_2 - P_1A_1 + Q)Y = \lambda Y, \\
& \tilde{v}^T \tilde{Y}^2 = 1.
\end{aligned} \tag{8}
\]

Matrix \( M \) has a band structure (bandwidth depends on the order of the method) with some additional elements on the first and last rows, because of the initial and final methods (see [8,9] for more details).

In case the problem is regular and separated boundary conditions contain derivatives in \( a \) and \( b \)

\[\alpha_1 y(a) + \alpha_2 (py')(a) = \beta_1 y(b) + \beta_2 (py')(b) = 0, \tag{9}\]

it is convenient, as suggested in [8], to define initial and final methods also involving the values \( y_0 \approx y(a) \) and \( y_n \approx y(b) \). Eq. (5) is approximated in \( x_0, \ldots, x_n \), thus giving a matrix representation of the discrete problem similar to (7), but with \( Y = [y_0, \ldots, y_n]^T \) and \( \tilde{Y} = [y'_0, Y^T, y'_n]^T \). The algebraic eigenvalue problem (8) is now obtained by neglecting the first and the last column from the matrix \( W^{-1}(-PA_2 - P_1A_1 + Q)S^{-1} \), where

\[
S = \begin{pmatrix}
\alpha_2 p(a) & \alpha_1 \\
1 & \\
. & \\
1 & \\
\beta_1 & \beta_2 p(b)
\end{pmatrix}.
\]

In the sequel we will also solve singular problems for which one or both the boundary conditions are useless. Supposing, for example, that \( a \) is a LCNO point or a LP point (no boundary condition needs), we approximate (5) in \( x_0 \) with formula (2) and \( i = 0 \). Clearly, if both the boundary conditions are not given, then it is sufficient to substitute them with the approximation of (5) in \( x_0 \) and \( x_n \). The discrete representation of the problem is again similar to (7), but with \( Y = Y = [y_0, \ldots, y_n]^T, A_1 = A_1 \) and \( A_2 = A_2 \).

3. The HOFiD code for Sturm–Liouville problems

The developed code for solving SLPs uses variable step/order to compute an approximation of a selected eigenvalue and its corresponding eigenfunction. In input, it requires the Sturm–Liouville problem and the index \( k \) of the eigenvalue \( \lambda_k \) to be computed. As optional input, the orders of the schemes and the corresponding exit tolerances to be satisfied, and the length of the initial grid. The default values are 4, 6 and 8 as orders of convergence, and \( 1e - 3, 1e - 6 \) and \( 1e - 8 \), respectively, as output tolerances for the solution. Error is approximated by using two different orders \( p \) and \( p + 2 \). We use relative error for the eigenvalue and absolute error for the eigenvector. The default number of initial points is \( \max(20, 5k) \).

The code, written in the MATLAB language, may be synthesized in the following few lines. It computes a coarse first approximation of the eigenvalue with order 2 formulae and constant stepsize, and then iterates on the input orders. The solution obtained with order \( p \) is used as starting approximation for the order \( p + 2 \). For easy of understanding we have deleted all the controls on the input parameters and the computed solution.
function [lam, y, hh] = solver(problem, k, tol, order, n0)
    [a, b] = problem('interval');
    [lam, y] = init_approx(problem, k, n0);
    hh = (b-a)/n0*ones(n0-1,1);
    for i = 1:length(order)
        [lam, y, hh] = numer_solut(problem, k, order{i}, tol{i}, lam, y, hh);
    end

The input parameter problem is a function which contains the data of the problem (5)–(9). A prototype of this function is the following.

function [a, b, c, d] = test_problem(flag, x, param)
    % param contains possible parameters inside the problem
    switch flag
        case 'interval'
            % define a and b as the extremes of (1)
        case 'bound_cond'
            % define a = alpha_1, b = alpha_2, c = beta_1, d = beta_2
        case 'functions'
            % define a = p(x), b = p'(x), c = q(x), d = w(x)
    end

The main program is based on the following modules:

- [lam, y] = init_approx(problem, k, n) computes an initial approximation for the kth eigenvalue and eigenfunction by using n constant steps of the order 2 method. Since this matrix is not symmetric, the subroutine computes all the eigenvalues and then selects the kth checking if the eigenvector matches the number of zeros.
- [lam, y, hh] = numer_solut(problem, k, ord, tol, lam, y, hh) computes an approximate solution with a prescribed tolerance starting from a suitable meshgrid previously obtained together with the approximation of the eigenvalue and eigenvector. Order of the used method is fixed inside this subroutine.

This second subroutine is summarized as follows:

function [lam, y, hh] = numer_solut(problem, k, order, tol, lam, y, hh)
    M = matrix(problem, order, hh);
    [lam0, y] = eig_compute(M-lam*speye(n), y);
    lam = lam + lam0;
    e = err(problem, order, lam, y, hh);
    while norm(e) > tol
        hh = equidistribute(hh, e);
        M = matrix(problem, order, hh);
        [lam0, u] = eig_compute(M-lam*speye(n), y);
        lam = lam + lam0;
        e = err(problem, order, lam, y, hh);
    end

It takes advantages of the following subroutines:

- M = matrix(problem, order, hh) computes the sparse matrix M of the problem discretized with fixed order and variable stepsize.
- [lam0, y] = eig_compute(M-lam*speye(n), y) computes a new approximation of the solution as the smallest eigenvalue of the matrix M – J by means of the inverse power method.
- e = err(problem, order, lam, y, hh) computes a discrete error function on the considered grid by using the subsequent even order to that used in eig_compute.
- hh = equidistribute(hh, e) computes a new grid by means of an equidistribution of the error function. In order to simplify the definition of the methods, only one stepsize variation is allowed in each stencil. This means that a variation in the stepsize is allowed at most every r + 5 steps.
4. Numerical experiments

This section is devoted to illustrate the behavior of the code on some regular and singular Sturm–Liouville problems, defined on bounded or unbounded intervals, see [19,21,28]. All these tests allows us to try out the code on all the different numerical strategies proposed in Section 2, and they confirm as both stepsize and order variation work very well. We also provide the first nine digits of any computed eigenvalue. These results are in perfect agreement with those produced by the MATSLISE package, see [21].

When a problem is defined on an unbounded interval, then we perform a change of variable in order to obtain a new problem on a bounded one. Moreover, as suggested in [17,23,26], if the coefficients of the Eq. (5) are not defined in the endpoints of the interval [a, b], we consider the truncated interval [x, β] with a < α < β < b.

In the following, the default parameters introduced in the previous section can be also modified accordingly to the features and difficulties of the problem. In particular we fix the order of the methods to compare the behavior of the code on singular problems. In fact, it is known (see, for example, [2]) that finite differences suffer of an order reduction on such equations.

For the first cluster of 6 eigenvalues we use orders 6, 8 and 10 to reach the exit tolerance 10⁻⁸. For the estimate of successive clusters we adopt the same order and stepsize variation strategy, but a different combination of orders and exit tolerances. Since clusters become wider, the problem is easier to solve and we start with order 4 (and then use orders 6 and 8) to reach the final exit tolerance 10⁻⁶, obtaining a good compromise between accuracy, number of mesh points and computational cost. As in general, the code reaches a better accuracy faster in computing the eigenvalues rather than the eigenfunctions.

In Fig. 1 we depict the eigenfunctions associated to the first and the last eigenvalue belonging to the first cluster, their absolute error and the stepsize variation. We observe how in this case it is quite important to use small exit tolerances since the numerical solution is often quite near the x-axis.

### Problem 1. The Mathieu equation

\[-y''(x) + \cos(x)y(x) = \lambda y(x), \quad x \in [0, 40]\]

has regular boundary conditions \(y(0) = y(40) = 0\) and oscillatory coefficients. It is known that lower eigenvalues are grouped in clusters of 6, and more and tighter clusters appear as \(c\) increases.

This problem is a numerical challenge for codes which have to compute one particular eigenvalue, and the information given by the eigenfunction is necessary to check if the obtained eigenvalue is the right one. Therefore, their estimate reveals the reliability and the efficiency of the code described in Section 3. Even if the problem is regular, we need a good starting accuracy to compute the first eigenvalue. Therefore, as shown in Table 1 for \(c = 5\), we apply the order and stepsize variation strategy, starting with 251 equidistant mesh points.

For the first cluster of 6 eigenvalues we use orders 6, 8 and 10 to reach the exit tolerance 10⁻⁸. For the estimate of successive clusters we adopt the same order and stepsize variation strategy, but a different combination of orders and exit tolerances. Since clusters become wider, the problem is easier to solve and we start with order 4 (and then use orders 6 and 8) to reach the final exit tolerance 10⁻⁶, obtaining a good compromise between accuracy, number of mesh points and computational cost. As in general, the code reaches a better accuracy faster in computing the eigenvalues rather than the eigenfunctions.

In Fig. 1 we depict the eigenfunctions associated to the first and the last eigenvalue belonging to the first cluster, their absolute error and the stepsize variation. We observe how in this case it is quite important to use small exit tolerances since the numerical solution is often quite near the x-axis.

### Problem 2. The Truncated Gelfand–Levitan equation

\[-y''(x) + \frac{2(T(x) \sin(2x) + \cos^4(x))}{T''(x)} y(x) = \lambda y(x), \quad x \in [0, 100],\]

\[T(x) = 1 + \frac{x}{2} + \frac{\sin(2x)}{4}, \quad x \in [0, 100],\]

The Mathieu equation with \(c = 5\). Exit tolerances are 10⁻³, 10⁻⁶ and 10⁻⁸ for the orders 6, 8 and 10, respectively; 10⁻³, 10⁻⁴ and 10⁻⁶ for the orders 4, 6 and 8, respectively. \(n_0 = 251\).

| \(k\) | \(p\) | \(\lambda_k\) | \(n\) | \(E_s(\lambda_k)\) | \(E_s(y_{\lambda_k})\) |
|---|---|---|---|---|---|
| 0 | 6-8-10 | -3.48423886 | 981 | 1.08e−14 | 3.06e−09 |
| 1 | 6-8-10 | -3.48422191 | 1642 | 3.29e−14 | 7.46e−09 |
| 2 | 6-8-10 | -3.48419799 | 986 | 1.15e−14 | 4.12e−09 |
| 3 | 6-8-10 | -3.48417260 | 1315 | 1.33e−14 | 6.25e−09 |
| 4 | 6-8-10 | -3.48415155 | 1416 | 1.22e−13 | 9.90e−09 |
| 5 | 6-8-10 | -3.48413967 | 1158 | 9.55e−14 | 9.08e−09 |
| 6 | 6-8-10 | -3.50954355 | 695 | 9.94e−10 | 1.05e−07 |
| 12 | 4-6-8 | 1.93291488 | 501 | 1.97e−09 | 1.67e−09 |
has regular boundary conditions \( y(0) - y'(0) = 0 \) and \( y(100) = 0 \). For the more general boundary condition on the left endpoint we use the numerical approach discussed in Section 2. We start with 50 equidistant mesh points and use orders 6, 8 and 10 to reach the exit tolerance \( 10^{-8} \). As shown in Table 2, the adopted strategy allows us to reach a good accuracy in both the eigenvalues and eigenfunctions approximations.

In Fig. 2 we depict the eigenfunctions associated to the eigenvalues computed in Table 2. It is clear that the eigenfunctions have nonuniform oscillations of decreasing size which depends on \( q(x) = 2(T(x) \sin(2x) + \cos^4(x))/T^2(x) \), and more oscillations with tiny width appear for the first eigenvalues.

Problem 3. The Pruess equation

\[
-y''(x) + \ln(x)y(x) = \lambda y(x), \quad x \in [0, 4]
\]

has regular boundary conditions \( y(0) = y(4) = 0 \). It is defined as a regular problem that looks singular in [28] since for any high order finite difference scheme the order of convergence reduces to 2 as when these formulae are applied to singular problems. Consequently, it is simple to infer that there is no advantage in their use.

### Table 2

Truncated Gelfand–Levitan equation. Default exit tolerances for variable orders 6, 8 and 10, \( n_0 = 51 \).

| \( k \) | \( p \) | \( \lambda_k \) | \( n \) | \( E_x(\lambda_k) \) | \( E_y(y_k) \) |
|---|---|---|---|---|---|
| 0 | 6-8-10 | 9.58281381e−4 | 1015 | 3.32e−10 | 2.83e−11 |
| 4 | 6-8-10 | 2.39854803e−2 | 1235 | 1.17e−10 | 1.12e−10 |
| 9 | 6-8-10 | 9.5849295e−2 | 1019 | 3.92e−11 | 1.28e−10 |
| 14 | 6-8-10 | 2.15695603e−1 | 1253 | 1.25e−12 | 9.11e−11 |
In Table 3 we just analyze the convergence behavior for fixed order. We depict the required number of steps for the first, the fifth and the tenth eigenvalue and eigenfunction using orders 4, 6 and 8 and stepsize variation with the exit tolerance $10^{-8}$. As the results highlight, the convergence is not saved from the order reduction. Actually, order 4 is sufficient to guarantee a good accuracy with a small number of mesh points for the first eigenvalue, but when the index of the eigenvalue increases, then order 8 becomes competitive. Obviously, for simple problems the use of one order is enough to gain the accuracy of the lowest eigenvalues, while an order variation is recommended for the larger ones.

Problem 4. The Airy equation

$$-y''(x) + xy(x) = \lambda y(x), \quad x \in [0, +\infty]$$

is a singular problem with left boundary condition $y(0) = 0$ while $b = +\infty$ is LP. The eigenvalues are the zeros of the Airy function.

In Table 3 we just analyze the convergence behavior for fixed order. We depict the required number of steps for the first, the fifth and the tenth eigenvalue and eigenfunction using orders 4, 6 and 8 and stepsize variation with the exit tolerance $10^{-8}$. As the results highlight, the convergence is not saved from the order reduction. Actually, order 4 is sufficient to guarantee a good accuracy with a small number of mesh points for the first eigenvalue, but when the index of the eigenvalue increases, then order 8 becomes competitive. Obviously, for simple problems the use of one order is enough to gain the accuracy of the lowest eigenvalues, while an order variation is recommended for the larger ones.

**Problem 4.** The Airy equation

$$-y''(x) + xy(x) = \lambda y(x), \quad x \in [0, +\infty]$$

is a singular problem with left boundary condition $y(0) = 0$ while $b = +\infty$ is LP. The eigenvalues are the zeros of the Airy function.

| $k$ | $p$ | $\lambda_k$ | $n$ | $E_r(\lambda_k)$ | $E_d(\lambda_k)$ |
|-----|-----|-------------|-----|----------------|---------------|
| 1   | 4   | 1.12481680  | 290 | $3.6e-09$     | $2.6e-09$     |
|     | 6   | 1.12481680  | 435 | $1.4e-09$     | $1.7e-09$     |
|     | 8   | 1.12481678  | 324 | $4.1e-09$     | $5.1e-09$     |
| 4   | 4   | 15.8644571  | 1201| $1.1e-09$     | $1.3e-09$     |
|     | 6   | 15.8644571  | 363 | $1.4e-09$     | $7.6e-09$     |
|     | 8   | 15.8644571  | 395 | $9.4e-10$     | $5.3e-09$     |
| 9   | 4   | 62.0987975  | 1930| $3.2e-10$     | $6.7e-09$     |
|     | 6   | 62.0987973  | 503 | $8.9e-10$     | $3.9e-09$     |
|     | 8   | 62.0987972  | 429 | $7.6e-10$     | $8.8e-09$     |

**Fig. 2.** Truncated Gelfand-Levitan equation.
\[ A(\lambda) = (J_{1/3} + J_{-1/3}) \left( \frac{2}{3} x^{1/3} \right), \]

where \( J_{\alpha} \) is the Bessel function. Since the problem is defined on a semi-infinite interval, we perform the following transformation

\[ \tau = 1 - \frac{1}{\sqrt{1 + x}} \in [0, 1]. \] (16)

Consequently, with the change of variable \( u(\tau) = y(\tau(x)) \), problem (15) is rewritten as

\[ -\frac{(1 - \tau)^6}{4} u''(\tau) + \frac{3}{4} (1 - \tau)^5 u'(\tau) + \frac{2\tau - \tau^2}{1 - \tau^2} u(\tau) = \lambda u(\tau), \quad \tau \in [0, 1]. \] (17)

It is regular in \( a = 0, \) while \( b = 1 \) is LP. Moreover, since the problem is singular in 1, the interval is truncated to \( \beta = 1 - \delta \) with \( \delta = 10^{-4}. \)

As illustrated in Table 4, also this example shows a reduction of the convergence order. Effectively order 4 is the best choice both for accuracy and minimum computational cost. This behavior justifies the failure of the order variation strategy, and underlines as it makes sense to apply only a variable stepsize strategy with the smaller order.

**Table 4**
Airy equation. Exit tolerance \( 10^{-8} \) for fixed orders, default exit tolerances for variable orders. \( n_0 = 21. \)

| \( k \) | \( p \) | \( \lambda_k \) | \( n \) | \( E_l(\lambda_k) \) | \( E_l(y_k) \) |
|---|---|---|---|---|---|
| 0 | 4 | 2.33810740 | 872 | 1.00e-09 | 4.98e-09 |
| 6 | 2.33810741 | 739 | 1.36e-09 | 8.51e-09 |
| 8 | 2.33810741 | 912 | 1.60e-09 | 1.00e-08 |
| 4-6-8 | 2.33810741 | 938 | 1.19e-09 | 7.44e-09 |
| 4 | 7.94413358 | 3648 | 2.00e-11 | 4.54e-09 |
| 6 | 7.94413358 | 3657 | 5.28e-11 | 1.79e-09 |
| 8 | 7.94413358 | 5169 | 4.33e-11 | 1.47e-09 |
| 4-6-8 | 7.94413358 | 5337 | 7.94e-11 | 5.99e-11 |

**Table 5**
Laguerre’s equation. Exit tolerance \( 10^{-8} \) for fixed orders, default exit tolerances for variable orders. \( n_0 = 21. \)

| \( k \) | \( p \) | \( \lambda_k \) | \( n \) | \( E_l(\lambda_k) \) | \( E_l(y_k) \) |
|---|---|---|---|---|---|
| 0 | 4 | 3.99999999 | 1018 | 3.65e-10 | 8.71e-09 |
| 6 | 3.99999999 | 969 | 1.19e-11 | 8.63e-09 |
| 8 | 3.99999999 | 748 | 3.03e-11 | 9.04e-09 |
| 4-6-8 | 3.99999999 | 569 | 2.22e-11 | 6.77e-09 |
| 4 | 20.00000000 | 597 | 1.33e-11 | 9.04e-09 |
| 9 | 39.99999999 | 753 | 2.95e-11 | 9.37e-09 |
| 24 | 99.99999999 | 2067 | 8.52e-12 | 1.09e-09 |

**Fig. 3.** Laguerre’s equation. Numerical approximation of the tenth eigenfunction.
Problem 5. The Laguerre’s equation

\[-y''(x) + \left(x^2 + \frac{3}{4x^2}\right)y(x) = \lambda y(x), \quad x \in [0, +\infty]\]  \hspace{1cm} \text{(18)}

is singular and both the endpoints \(a = 0\) and \(b = +\infty\) are LP. The eigenvalues satisfy the relation \(\lambda_k = 4(k + 1)\), for \(k \geq 0\). As in the previous problem, the equation is defined on a semi-infinite interval, therefore a change of variable is needed to reformulate (18) in the finite interval \([0, 1]\). Moreover, due to the singularities in 0 and 1, the new problem is solved in the truncated interval \([\delta, 1 - \delta]\), with \(\delta = 10^{-4}\).

As shown in Table 5, the results for this problem are completely different from the previous ones since now order of accuracy is preserved and order (and stepsize) variation allow to greatly reduce the required number of points. In Fig. 3 we depict the tenth eigenfunction, both for the transformed problem and the original one truncated in the interval \([0, 20]\), where it is clear that stepsize variation is necessary to gain high accuracy with a limited number of steps.

5. Conclusions

In this paper we propose a code for solving regular and singular Sturm–Liouville problems which takes advantage of stepsize and order variation strategies. The code works well for any considered problem even if the use of high orders is sometimes useless because of a known deficiency of finite differences applied to singular equations. The successive step of the research will be the application of this code to some challenging problems for which the use of a robust stepsize variation is necessary.

References

[1] Aceto L, Ghelardoni G, Gheri G. An algebraic procedure for the spectral corrections using the miss-distance functions in regular and singular Sturm–Liouville problems. SIAM J Numer Anal 2006;44:2277–43.
[2] Aceto L, Ghelardoni P, Magherini C. BVMs for Sturm–Liouville eigenvalue estimates with general boundary conditions. J Numer Anal Ind Appl Math 2009;4:113–27.
[3] Aceto L, Ghelardoni P, Magherini C. Boundary value methods as an extension of Numerov’s method for Sturm–Liouville eigenvalue estimates. Appl Numer Math 2009;59:1644–56.
[4] Amodio P, Budd C, Koch O, Sattan G, Weinmüller EB. Asymptotical computations for a model of flow in saturated porous media. Appl Math Comput 2014;237:155–67.
[5] Amodio P, Sattan G. Variable step/order generalized upwind methods for the numerical solution of second order singular perturbation problems. J Numer Anal Ind Appl Math 2009:4:65–76.
[6] Amodio P, Sattan G. A deferred correction approach to the solution of singularly perturbed BVPs by high order upwind methods: implementation details. In: Simos TE, Psihos G, Tsitouras Ch, editors. Numerical analysis and applied mathematics – ICNAAM 2009, AIP Conf. Proc. 1168, issue 1; 2009. p. 711–714.
[7] Amodio P, Sattan G. High order finite difference schemes for the numerical solution of eigenvalue problems for IVPs in ODEs. In: Simos TE, Psihos G, Tsitouras Ch, editors. Numerical analysis and applied mathematics – ICNAAM 2010, AIP Conf. Proc. 1281, issue 1; 2010. p. 202–204.
[8] Amodio P, Sattan G. A matrix method for the solution of Sturm–Liouville problems. JNAIAM J Numer Anal Ind Appl Math 2011:6:1–13.
[9] Amodio P, Levitina T, Sattan G, Weinmüller EB. On the calculation of the finite Hankel transform eigenfunctions. J Appl Math Comput 2013:43:151–73.
[10] Amodio P, Levitina T, Sattan G, Weinmüller EB. Numerical simulation of the whispering gallery modes in prolate spheroids. Comput Phys Commun 2014:185:1200–6.
[11] Amodio P, Sgura I. High-order finite difference schemes for the solution of second-order BVPs. J Comput Appl Math 2005;176:59–76.
[12] Amodio P, Sgura I. High order generalized upwind schemes and numerical solution of singular perturbation problems. BIT 2007;47:241–57.
[13] Andrew AL, Paine JW. Correction of Numerov’s eigenvalue estimates. Numer Math 1985;47:289–300.
[14] Bailey PB. SLEIGN: an eigenfunction–eigenvalue code for Sturm–Liouville problems. SAND77-2044, Sandia Laboratories, Albuquerque; 1978.
[15] Bailey PB, Everitt WN, Zettl A. Algorithm 810: the SLEIGN2 Sturm–Liouville code. ACM Trans Math Software 2001;27:143–92.
[16] Bailey PB, Gordon MK, Shampine LF. Automatic solution of the Sturm–Liouville problem. ACM Trans Math Software 1978;4:113–27.
[17] Bailey PB, Gordon MK, Shampine LF. Automatic solution of the Sturm–Liouville problem. SIAM J Numer Anal 2006;44:2277–43.
[18] Brugnano L, Trigiante D. Solving differential problems by multistep initial and boundary value methods. Amsterdam: Gordon and Breach Science Publishers; 1998.
[19] Everitt WN. A catalogue of Sturm–Liouville differential equations. In: Amrein WO, Hinz AM, Pearson DP, editors. Sturm–Liouville theory. Past and present. Birkhäuser; 2005.
[20] Ixaru LGr, De Meyer H, Vanden Berghe G. SLCPM12 – a program for solving regular Sturm–Liouville problems. Comput Phys Commun 1999;118:259–77.
[21] Ledoux V, Van Daele M, Vanden Berghe G. MATSLISE: a MATLAB package for the numerical solution of Sturm–Liouville and Schrödinger equations. ACM Trans Math Software 2005;31:532–54.
[22] Marletta M. Certification of algorithm 700: numerical tests of the SLEIGN software for Sturm–Liouville problems. ACM Trans Math Software 1991:17(4):481–90.
[23] Marletta M, Pryce JD. LCNO Sturm–Liouville problems: computational difficulties and examples. Numer Math 1995;69(3):303–20.
[24] Paine JW, De Hoog FR, Anderssen RS. On the correction of finite difference eigenvalue approximations for Sturm–Liouville problems. Computing 1981;30:123–39.
[25] Pruess S. Estimating the eigenvalues of Sturm–Liouville problems by approximating the differential equation. SIAM J Numer Anal 1973:10:55–68.
[26] Pruess S, Fulton CT. Mathematical software for Sturm–Liouville problems. ACM Trans Math Software 1993;19:360–76.
[27] Pruess S, Marletta M. Automatic solution of Sturm–Liouville problems using the Pruess method. J Comput Appl Math 1992;39:57–78.
[28] Pryce JD. A test package for Sturm–Liouville solvers. ACM Trans Math Software 1999;25(1):21–57.