INSTITUTE FOR ANALYSIS AND SCIENTIFIC COMPUTING

VIENNA UNIVERSITY OF TECHNOLOGY

Numerical Solution of Singular Eigenvalue Problems for ODEs with a Focus on Problems Posed on Semi-Infinite Intervals

R. Hammerling, O. Koch, Ch. Simon, and E. Weinmüller

Summary

This work is concerned with the computation of eigenvalues and eigenfunctions of singular eigenvalue problems (EVPs) arising in ordinary differential equations.
Two different numerical methods to determine values for the eigenparameter such that the boundary value problem has nontrivial solutions are considered.
The first approach incorporates a collocation method. In the course of this work the existing code `bvpsuite` designed for the solution of boundary value problems was extended by a module for the computation of eigenvalues and eigenfunctions.
The second solution approach represents a matrix method.
A code for first order problems is realized in such a way that problems of higher order can also be solved after a transformation to the first order formulation.
Since many eigenvalue problems are of second order, for example Sturm-Liouville problems, we also implemented a code for second order problems and present an empirical error analysis.
For the solution of semi-infinite interval problems a transformation of the independent variable is carried out in such a way that the boundary value problem (BVP) originally posed on a semi-infinite interval is reduced to a singular problem posed on a finite interval. The implementation of this transformation is also incorporated into the `bvpsuite` package.
The time-independent Schrödinger equation serves as an illustrating example.

Contents

Chapter 1

Introduction

The study of eigenvalue problems (EVPs) for singular boundary value problems (BVPs) is a topic of great interest. In many cases eigenvalue problems model important physical processes. For example, the bound state energies of the hydrogen atom can be computed as the eigenvalues of a singular eigenvalue problem. Very often those problems arise due to the use of the method of separation of variables for the solution of classical partial differential equations.
This subclass of boundary value problems also occurs as a model of motion according to Newton's law.
In the course of this work some of these examples are introduced to support the theoretical results.

1.1 Mathematical Notation

By $\mathbb{R}^n$ we denote the space of real vectors of dimension n. $|\cdot|$ is the maximum norm and $||\cdot||_2$ is the Euclidean norm in $\mathbb{R}^n$.
Let $z : (a, b) \to \mathbb{R}^n$, then by $C_n^p(a, b)$ we denote the space of real vector-valued functions which are p times continuously differentiable on (a, b).
For functions $z \in C_n^0[0, 1]$ the maximum norm is denoted by $\|\cdot\|_\infty$.
Thus we define,

$$|z(t)| := \max_{1\le k\le n} |z_k(t)|,\ t \in [0, 1]$$

and

$$\|z\|_\infty := \max_{0\le t\le 1} |z(t)|.$$

For the numerical analysis we define meshes of the form

$$\Delta := (t_0, t_1, \ldots, t_N), \tag{1.1}$$

and

$$h_k := t_k - t_{k-1}, \quad k = 1, \ldots, N, \quad \mathbf{h} := \max_{1\le k\le N} h_k, \quad t_0 = a, t_N = b. \tag{1.2}$$

Vectors $(z_0, \ldots, z_N) \in \mathbb{R}^{(N+1)n}$ corresponding to the grid Δ are called grid vectors. In the following we will restrict ourselves to equidistant meshes,

$$h_k \equiv \mathbf{h}, \quad \mathrm{k} = 1, \ldots, \mathrm{N}. \tag{1.3}$$

In this case we denote the vectors by

$$z_h := (z_0, \ldots, z_N). \tag{1.4}$$

The maximum norm on the space of grid vectors is given by

$$\|z_h\|_\Delta := \max_{0 \le k \le N} |z_k|. \tag{1.5}$$

For a continuous function $z \in C_n[0,1]$, we denote by R_Δ the pointwise projection onto the space of grid vectors,

$$R_\Delta : \begin{cases} C_n[0,1] \to \mathbb{R}^{(N+1)n} \\ z \mapsto (z(t_0), \ldots, z(t_N)). \end{cases} \tag{1.6}$$

For collocation, m points $t_k + h_k\rho_j$, $j = 1, \ldots, m$, are inserted in each subinterval $J_k := [t_k, t_{k+1}]$, where $0 =: \rho_0 < \rho_1 < \rho_2 < \cdots < \rho_m < \rho_{m+1} := 1$.
This yields the grid

$$\Delta^m := \{t_{kj} : \ t_{kj} = t_k + h_k\rho_j, \ k = 0, \ldots, N-1, \ j = 0, \ldots, m+1\}. \tag{1.7}$$

The assumption $\rho_1 > 0$ avoids a special treatment of the left endpoint in case of a singular problem.

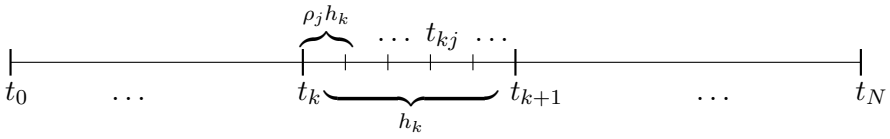

Figure 1.1: The computational grid

1.2 Outline of the Work

In Chapter 1 the mathematical framework that we deal is established.
In Chapter 2 we give the problem formulation of the (singular) eigenvalue problem and introduce the important subclass of Sturm-Liouville problems (SLPs). Furthermore, we consider the time-independent Schrödinger equation and show how singular eigenvalue problems arise by applying the method of separation of variables. Analytical results on the eigenvalues and eigenfunctions for the hydrogen problem are recapitulated.

Then, we present some existing codes for the solution of eigenvalue problems and describe briefly the numerical methods used. In Chapter 3 a transformation of the independent variable for BVPs posed on semi-infinite intervals including the special case of eigenvalue problems is carried out. This is done in such a way that the problem originally posed on a semi-infinite interval is transformed to a problem posed on a finite interval. Subsequently, the collocation code `bvpsuite` can be used.
Motivated by the way parameter dependent problems are treated in [18], a collocation method for the solution of eigenvalue problems for boundary value problems is presented. For regular problems this is described in [1] and for details related to the singular problem, see [2]. The calculation of the eigenvalues and eigenfunctions of the time-independent Schrödinger equation illustrates the technique. Since the exact solution is known, absolute and relative errors for the eigenvalues are presented in Chapter 4. Furthermore, empirical convergence orders are calculated.
In Chapter 5, a second approach for the numerical calculation of eigenfunctions and eigenvalues is presented. This method incorporates a finite difference approximation. By applying Keller's box scheme we replace the original problem by an algebraic eigenvalue problem which can be solved by standard linear algebra packages.
A code for first order problems was implemented in such a way that problems for higher order differential operators can also be solved after transformation to the first order form. Since many eigenvalue problems are of second order, for example Sturm-Liouville problems, we also implemented a code for second order problems and paid special attention to the approximation of the boundary conditions in the singular case. Also for these methods numerical examples are given and empirical convergence orders are presented.
Finally, after analyzing the pros and cons of the two presented numerical methods for the solution of differential eigenvalue problems we conclude that the combination of both techniques results in a very successful and efficient approach. Since for the collocation method a good starting profile is crucial in order to home in on the eigenvalue λ_k for a specified index k, we made use of the matrix method to obtain an accurate starting guess for the eigenvalue as well as an initial profile for the eigenfunction subsequently computed on an adaptive mesh by the collocation method featuring error control. This combined technique proved to be highly successful when applied to three relevant examples from quantum mechanics.

Chapter 2

Eigenvalue Problems for Ordinary Differential Equations

Eigenvalue problems for ordinary differential equations may be stated in various formulations much in the way of 'standard' boundary value problems. In the linear case, for example, an eigenvalue problem for a homogeneous differential operator L is to determine eigenvalues $\lambda \in \mathbb{C}$ for which the BVP,

$$Ly = \lambda y, \tag{2.1}$$

$$B_a y(a) + B_b y(b) = 0, \tag{2.2}$$

has nontrivial solutions, which are called eigenfunctions. The problem is called singular if L denotes a singular differential operator, cf. Chapter 4. One imporant class of eigenvalue problems are Sturm-Liouville problems, where L is of second order and self-adjoint. For those a well-developed theory and various codes for the computation of the numerical solution exist.
However, our aim was not only to focus on SLPs but to consider numerical methods, which are capable of solving a wide range of differential eigenvalue problems including higher order examples. In the course of this report we will also investigate examples of coupled channel Schrödinger equations whose mathematical statement corresponds to an EVP for a system of ODEs. These arise in the context of Hamiltonian systems, cf. [23].

2.1 Sturm-Liouville Problems

Sturm-Liouville problems [27] are linear differential eigenvalue problems of the form

$$-\frac{d}{dt}\left(p(t)\frac{dy(t)}{dt}\right) + q(t)y(t) = \lambda g(t)y(t) \tag{2.3}$$

posed on an interval (a, b) which may be finite or infinite, $-\infty \leq a < b \leq \infty$.
In [27] a Sturm-Liouville problem is called regular if

- a and b are finite,

- p, q and g are defined on $[a, b]$ and continuous except for finitely many jumps,
- p and g are strictly positive almost everywhere and
- regular boundary conditions are posed.

We will later state the assumptions on the problem data necessary to obtain a well-posed problem in the singular case.
A pair consisting of a value of λ for which there is a nontrivial solution $y(t)$ of (2.3) and the solution itself is called *eigensolution*.
Our aim is to calculate such pairs numerically. Depending on the problem, sometimes a large number of such pairs is sought. Consequently, this influences the numerical method that is used to solve the given problem. If only a few eigenvalues and eigenfunctions are sought, a possible way to solve this problem is to recast the Sturm-Liouville problem into the framework of boundary value problems as it is done in Chapter 4.
Typically, Sturm-Liouville problems arise in the context of the separation of variables method for partial differential equations, as it is also the case for the one-particle Schrödinger equation. We then speak of $q(t)$ as a potential function and the eigenvalue is called energy level. The associated eigenfunction is called wave function. The eigenvalue-eigenfunction pair is then referred to as bound state. Many physical phenomena, such as vibration of strings and, as mentioned before, the interaction of atomic particles give rise to Sturm-Liouville problems.
In general, we distinguish between *regular* and *singular* endpoints. We already classified a regular problem. We call a problem singular if one or both of its endpoints are singular. In [27] and related publications the following definition is given:

Definition 1. *The endpoint a of $I = [a, b] \subseteq \mathbb{R}$ is regular if it is finite and*

$$p^{-1}, |q|, g \in L^1[a, \alpha] \quad \text{for some} \quad \alpha \in (a, b).$$

Similarly the endpoint b is regular if it is finite and

$$p^{-1}, |q|, g \in L^1[\beta, b] \quad \text{for some} \quad \beta \in (a, b).$$

An endpoint is called singular if it is not regular. Thus an endpoint is singular if it is infinite, or the endpoint is finite but at least one of p^{-1}, q or g is not integrable in any neighbourhood of the endpoint. Note that the assumptions for p and g to be strictly positive almost everywhere still hold.

It is interesting to note that for $p(t) = 1$ and $q(t) = t^{-\alpha}$ for $\alpha < 1$ in the terminology of Sturm-Liouville problems one still calls 0 a *regular endpoint*.
Additionally, a distinction of singular endpoints into limit-circle (LC) and limit-point (LP) is made:

- Limit-Circle (LC)
  The endpoint a is limit-circle (LC) if a is singular and all solutions of (2.3) are in $L^2((a, \alpha); g)$ for some $\alpha \in I$.

- Limit-Point (LP)
  The endpoint a is limit-point (LP) if a is singular and there is only one nonzero solution in $L^2((a,\alpha);g)$ for some $\alpha \in I$.

A further classification into *oscillatory* and *nonoscillatory* is given by

- Nonoscillatory
  The endpoint a is nonoscillatory (LCNO) if there exists a point $d \in (a,b)$, and a real value $\lambda \in \mathbb{R}$ such that a solution of (2.3) has no zero in (a,d).

- Oscillatory
  The endpoint a is oscillatory (LCO) if for any real $\lambda \in \mathbb{R}$ and any nonzero solution there exists a $d \in (a,b]$ such that this solution has a zero $\xi \in (a,d]$.

Although λ appears in the LP/LC-classification of the endpoint, in fact the classification only depends on the interval (a,b) and the coefficient functions p, q and g.

The Spectrum of a Sturm-Liouville Operator

Regular problems only have a *discrete spectrum*, which consists of all the eigenvalues. However singular problems can also have a *continuous spectrum*. Depending on the problem, different results about the spectrum of the given differential operator hold. For example, under certain assumptions on the coefficient functions and the boundary conditions, the following theorem for regular problems can be shown, cf. [23].

Theorem 2.1. *Consider the Sturm-Liouville problem with real coefficient functions and separated boundary conditions*

$$-(py')'(t) + q(t)y(t) = \lambda g(t)y(t), \quad t \in (0,1], \tag{2.4}$$
$$a_0 y(0) + b_0 (py')(0) = 0, \quad a_1 y(1) + b_1 (py')(1) = 0, \tag{2.5}$$

$a_0^2 + b_0^2 > 0$, $a_1^2 + b_1^2 > 0$. *Assume that* $p, g > 0$ *on* $(0,1]$ *and* $1/p, q$ *and* g *are continuous functions satisfying* $1/p, q, g \in L^1[0,\alpha)$ *for some* $\alpha > 0$. *Then*

- *The eigenvalues* λ_k *have algebraic multiplicity* 1 *and consequently the eigenspace for each eigenvalue is of dimension* 1.

- *The eigenvalues* λ_k *can be ordered as an increasing sequence tending to infinity,*

  $$\lambda_0 < \lambda_1 < \lambda_2 < \dots \tag{2.6}$$

  and the eigenfunction corresponding to λ_k *has exactly* k *zeros on the open interval* $(0,1)$.

The spectrum of a singular problem is a closed infinite subset of the real line. Depending on the type of the endpoints, the spectrum can have different structures. The most common are:

- The eigenvalues are an infinite sequence bounded from below and accumulate at infinity.
- As the first case, but the eigenvalues accumulate at $-\infty$ as well as ∞.
- An infinite sequence of eigenvalues, bounded from below, with one finite accumulation point λ_c such that all $\lambda \geq \lambda_c$ are in the continuous spectrum σ_c.
- A finite, possibly empty, sequence of eigenvalues, which are bounded by λ_c and all $\lambda \geq \lambda_c$ are in σ_c.

The discrete spectrum of the radial equation of the hydrogen problem, which we will consider in more detail in the following chapters, is an infinite set $\lambda_0 < \lambda_1 < \ldots < \lambda_n < \ldots < \lambda_c = 0$, such that $\lim_{n\to\infty} \lambda_n = 0$ and all $\lambda \geq 0$ lie in the continuous spectrum. The fact that the eigenvalues are all simple, makes this singular problem suitable for the solution by the collocation method described in Chapter 4.

2.1.1 Example for a Sturm-Liouville Problem — The Hydrogen Atom

The Schrödinger operator of one particle in d dimensions is given by

$$\hat{H} : \begin{cases} D\left(\hat{H}\right) \subset \mathcal{H} \to \mathcal{H} \\ \hat{H}\psi(x) = \left[-\Delta + V(x)\right]\psi(x), \end{cases} \tag{2.7}$$

with the Hilbert space $\mathcal{H} = L^2\left(\mathbb{R}^d\right)$. Function $V : \mathbb{R}^d \to \mathbb{R}$ denotes the potential acting on the particle. Since in quantum physics the most important cases are $d = 1, 2, 3$ we restrict ourselves to these values of d and in particular for the following calculations we set $d = 3$.
Since we want to determine the discrete spectrum of $\hat{H}$, we further consider the eigenvalue problem

$$\hat{H}\psi(x) = \lambda\psi(x). \tag{2.8}$$

In the context of an electron interacting with a nucleus, an eigenvalue is an energy at which the electron orbits the nucleus stably for an infinite time. If the potential is spherically symmetric, i.e. $V(x) = V(|x|) = V(r)$, then the problem can be transformed into three BVPs for ordinary differential equations. For this purpose the Cartesian coordinates $x = (x_1, x_2, x_3)$ are replaced by spherical coordinates (r, θ, ϕ)

$$\begin{aligned} x_1 &= r\sin(\theta)\cos(\phi), \\ x_2 &= r\sin(\theta)\sin(\phi), \\ x_3 &= r\cos(\theta), \end{aligned}$$

where $r \in [0, \infty)$, $\theta \in [0, \pi]$, $\phi \in (-\pi, \pi]$. Consequently, the operator $\hat{H}$ is transformed to

$$\hat{H} = -\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\right) - \frac{1}{r^2}\frac{1}{\sin(\theta)}\frac{\partial}{\partial\theta}\left(\sin(\theta)\frac{\partial}{\partial\theta}\right) - \frac{1}{r^2}\frac{1}{\sin^2(\theta)}\frac{\partial^2}{\partial\phi^2} + V(r).$$

By separation of variables $\psi(r,\theta,\phi) = R(r)\Theta(\theta)\Phi(\phi)$ we obtain

$$\left(-\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d}{dr}\right) + \frac{l(l+1)}{r^2} + V(r)\right)R(r) = \lambda R(r), \tag{2.9}$$

$$\frac{1}{\sin(\theta)}\left(-\frac{d}{d\theta}\left(\sin(\theta)\frac{d}{d\theta}\right) + \frac{m^2}{\sin(\theta)}\right)\Theta(\theta) = l(l+1)\Theta(\theta), \tag{2.10}$$

$$-\frac{d^2}{d\phi^2}\Phi(\phi) = m^2\Phi(\phi), \tag{2.11}$$

where $l \in \mathbb{N}_0$ and $m \in \mathbb{Z}$.
In the following we will restrict our attention to equation (2.9). Furthermore we carry out a transformation of the dependent variable $R(r) \mapsto u(r) = rR(r)$. Hence, we obtain a Sturm-Liouville problem where it holds that in (2.3) $p(r) = g(r) = 1$ and

$$q(r) = -\frac{l(l+1)}{r^2} + \frac{\gamma}{r}, \tag{2.12}$$

on $[0,\infty)$.

Exact Eigenvalues of the Hydrogen Atom

The hydrogen atom can be modeled by a single electron exposed to the external potential V which is generated by the atom's nucleus. V is called the Coulomb potential. Since the electron is attracted by the nucleus in (2.13) below it holds that $\gamma > 0$. Note that for $\gamma \leq 0$ the discrete spectrum is empty.
Hence, in (2.7) we set the potential $V(r) = \frac{\gamma}{r}$ and therefore the radial equation (2.9) becomes

$$\left(-\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d}{dr}\right) + \frac{l(l+1)}{r^2} - \frac{\gamma}{r}\right)R(r) = \lambda R(r). \tag{2.13}$$

In [31] (2.9) – (2.11) are solved independently and the following theorem is stated.

Theorem 2.2. *The eigenvalues of* (2.7) *are explicitly given by*

$$\lambda_n = -\left(\frac{\gamma}{2n}\right)^2, \quad n \in \mathbb{N}. \tag{2.14}$$

An orthonormal basis for the corresponding eigenspace is given by the n^2 *functions*

$$\psi_{n,l,m}(x) = R_{n,l}(r)Y_l^m(x), \quad |m| \leq l < n,\ r = |x|, \tag{2.15}$$

where

$$R_{n,l}(r) = \sqrt{\frac{\gamma^3(n-1-l)!}{2n^4(n+l)!}}\left(\frac{\gamma r}{n}\right)^l \mathrm{e}^{-\frac{\gamma r}{2}} L_{n+l}^{(2l+1)}\left(\frac{\gamma r}{n}\right), \tag{2.16}$$

$$L_{n+l}^{(2l+1)}\left(\frac{\gamma r}{n}\right) = \sum_{i=0}^{n-l-1}\frac{(-1)^{i+2l+1}}{i!}(n+l)!\binom{n+l}{i+2l+1}\left(\frac{\gamma r}{n}\right)^i. \tag{2.17}$$

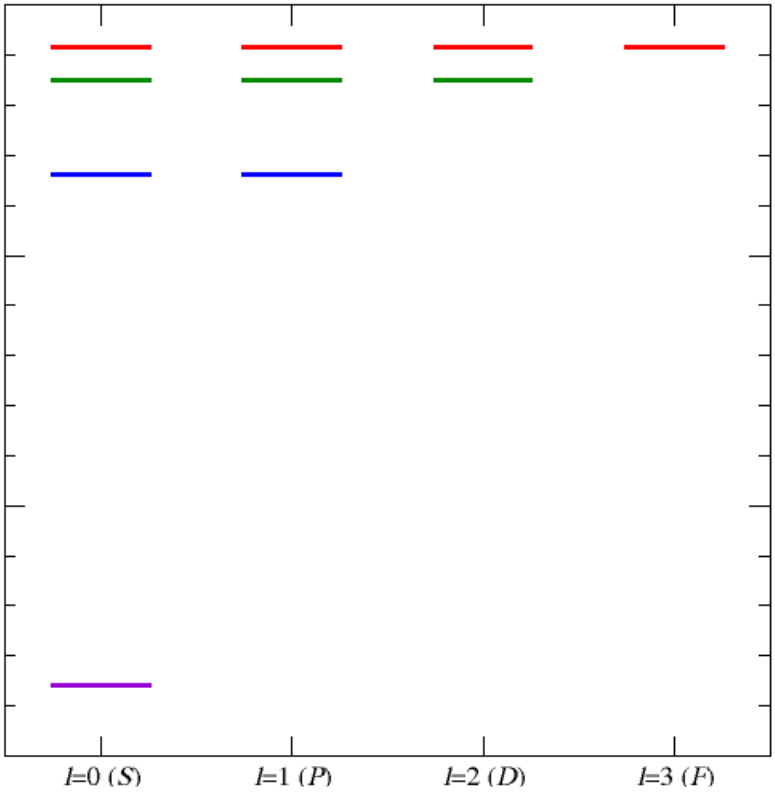

Figure 2.1: Appearance of bound states of the hydrogen atom for various values of the angular momentum quantum number l and the principal quantum number n. $n = 1$ – violet, $n = 2$ – blue, $n = 3$ – green, $n = 4$ – red.

By $L_{n+l}^{(2l+1)}$ *we denote the generalized Laguerre polynomials and* $Y_l^m(x)$ *are the so-called spherical harmonics. In particular, the lowest eigenvalue* $\lambda_1 = -\frac{\gamma^2}{4}$ *is simple and the corresponding eigenfunction* $\psi_{100} = \sqrt{\frac{\gamma^3}{2}} e^{-\frac{\gamma r}{2}}$ *is positive. The radial wave functions are normalized such that*

$$\int_0^\infty r^2 R_{n,l}(r) R_{n,l}(r) dr = 1. \tag{2.18}$$

From the theorem above one clearly sees that the eigenvalues are independent of the choice of l. However, this does not hold for the eigenfunctions.
By using results for self-adjoint operators, cf. [31], one immediately can characterize the spectra of each of the operators given in equations (2.9) – (2.11) for our particular potential. Figure 2.1 shows the dependence of the point spectrum on the angular momentum quantum number l and the principal quantum number n.

2.2 Standard Codes for the Solution of Eigenvalue Problems in ODEs

SLEIGN2

An advanced solver for Sturm-Liouville eigenvalue problems is the well-known code SLEIGN2. It is inspired by the older code SLEIGN by Bailey, Shampine and Gordon. The code SLEIGN2 (Sturm-Liouville eigenvalue) was published by P.B. Bailey, W.N. Everitt and A. Zettl [6]. It is a code designed for the solution of Sturm-Liouville boundary value problems consisting of one ordinary second order linear differential equation

$$-\frac{d}{dt}\left(p(t)\frac{dy(t)}{dt}\right)+q(t)y(t)=\lambda g(t)y(t) \quad \text{on } (a,b)\subseteq\mathbb{R} \tag{2.19}$$

and associated self-adjoint separated or nonseparated boundary conditions.
The problem can either be singular or regular according to the definitions in Chapter 2. In the singular case the code can also approximate the essential spectrum.
There is one problem type that the code cannot deal with, which is the case where one endpoint is LP and the spectrum is discrete and unbounded in both directions.
The code SLEIGN2 [6] uses the so-called *scaled Prüfer transformation* to transform the dependent variables,

$$y=S^{-1/2}\rho\sin(\theta), \quad py'=S^{1/2}\rho\cos(\theta). \tag{2.20}$$

As a result the linear second order problem is transformed to a nonlinear first order eigenvalue problem believed to be suitable for numerical integration. S is a strictly positive scaling function. The equations for θ and ρ are

$$\theta'=\frac{S}{p}\cos^2(\theta)+\frac{\lambda g-q}{S}\sin^2(\theta)+\frac{S'}{S}\sin(\theta)\cos(\theta), \tag{2.21}$$

$$\frac{2\rho'}{\rho}=\left(\frac{S}{p}-\frac{\lambda g-q}{S}\right)\sin(2\theta)-\frac{S'}{S}\cos(2\theta). \tag{2.22}$$

These equations are partially uncoupled such that we can first deal with equation (2.21). This equation becomes stiff for $\lambda g-q\ll 0$. The numerical integration is realized by a two-sided shooting procedure which is iteratively refined until a matching condition at some suitable interior point is satisfied up to a tolerance. Quite intricate preprocessing is used to prepare for the numerical integration of the initial value problem which is then realized by a 4(5) explicit Runge-Kutta-Fehlberg integrator with local error control called *GERK*. It is well-known that explicit Runge–Kutta methods are not well-suited solvers for singular initial value problems [16] as well as for stiff problems.

SLEDGE

SLEDGE is another code to calculate eigenvalues and eigenfunctions of a Sturm-Liouville problem. It was published by S. Pruess and C. Fulton [26]. The basic idea behind this

approach is to replace the given equation (2.19) and boundary conditions

$$\alpha_1 y(a) + \alpha_2 (py')(a) = \lambda[\alpha_1' y(a) - \alpha_2'(py')(a)], \tag{2.23}$$
$$\beta_1 y(b) + \beta_2 (py')(b) = 0, \tag{2.24}$$

by an approximating problem

$$-(\hat{p}\hat{y}')' + \hat{q}\hat{y} = \hat{\lambda}\hat{g}\hat{y}, \tag{2.25}$$

with the boundary conditions

$$\alpha_1 \hat{y}(a) - \alpha_2 (\hat{p}\hat{y}')(a) = \hat{\lambda}[\alpha_1' \hat{y}(a) - \alpha_2'(\hat{p}\hat{y}')(a)], \tag{2.26}$$
$$\beta_1 \hat{y}(b) + \beta_2 (\hat{p}\hat{y}')(b) = 0. \tag{2.27}$$

Note that the boundary conditions may also depend on λ and $\hat{\lambda}$, respectively, at one side. The functions $\hat{p}$, $\hat{q}$ and $\hat{g}$ are step functions on the mesh $a = t_1 < t_2 < \cdots < t_{N+1} = b$ and the values on each subinterval (t_n, t_{n+1}) are defined by the midpoint value, i.e.

$$\hat{p}(t) = p_n := p\left(\frac{t_n + t_{n+1}}{2}\right), \quad t \in (t_n, t_{n+1}], \tag{2.28}$$

and accordingly for the other two coefficient functions q and g.
Since

$$\hat{p}'(t) = 0 \quad \forall t \in (t_n, t_{n+1}] \tag{2.29}$$

the approximating problem on (t_n, t_{n+1}) reduces to

$$-\hat{y}'' = \omega_n^2 \hat{y}, \tag{2.30}$$

with

$$\omega_n := \sqrt{|\tau_n|}, \tag{2.31}$$
$$\tau_n := \frac{\hat{\lambda} g_n - q_n}{p_n}. \tag{2.32}$$

Because of the simplified coefficient functions we immediately obtain the following exact solution of (2.30)

$$\hat{y}(t) = \hat{y}(t_n)\phi_n'(t - t_n) + \frac{(\hat{p}\hat{y}')(t_n)}{p_n}\phi_n(t - t_n) \tag{2.33}$$

with

$$\phi_n(t) = \begin{cases} \sin(\omega_n t)/\omega_n, & \tau_n > 0, \\ \sinh(\omega_n t)/\omega_n, & \tau_n < 0, \\ t, & \tau_n = 0. \end{cases}$$

By requiring that $\hat{y}$ satisfies the boundary conditions and setting $h_n = t_{n+1} - t_n$, together with (2.33) we obtain the recurrence

$$\hat{y}(t_1) = \alpha_2 - \alpha_2' \hat{\lambda}, \tag{2.34}$$

$$(\hat{p}\hat{y}')(t_1) = \alpha_1 - \alpha_1' \hat{\lambda}, \tag{2.35}$$

$$\hat{y}(t_{n+1}) = \hat{y}(t_n)\phi_n'(h_n) + \frac{(\hat{p}\hat{y}')(t_n)}{p_n}\phi_n(h_n), \tag{2.36}$$

$$(\hat{p}\hat{y}')(t_{n+1}) = -\tau_n p_n \hat{y}(t_n)\phi_n'(h_n) + (\hat{p}\hat{y}')(t_n)\phi_n(h_n). \tag{2.37}$$

In (2.37) the fact that $\phi_n''(t) = -\tau_n \phi_n(t)$ was used.
Roughly speaking, the code proceeds as follows: The values $\hat{y}(t_{n+1})$ for $n = 0, 1, \ldots, N$ are calculated and subsequently a shooting method is applied. The relationship that has to be satisfied such that $\hat{\lambda}$ is an eigenvalue of the approximating problem is

$$\beta_1 \hat{y}(t_{N+1}) + \beta_2 (\hat{p}\hat{y}')(t_{N+1}) = 0. \tag{2.38}$$

Therefore $\hat{\lambda}$ has to be adjusted in order to find a zero of (2.38). This corresponds to a one-sided shooting procedure. Only if eigenfunction approximations are sought, the two-sided shooting algorithm is applied.
In order to separate the desired eigenvalue, say λ_n, from the rest of the point spectrum, an interval $[\xi_k, \eta_k]$ is determined such that this interval only contains λ_n. To find this interval the code exploits the fact that the index of the desired eigenvalue corresponds to the zeros of the eigenfunction. It is possible to count the zeros of the eigenfunction for the current value of λ while shooting from a to b with a trial value for λ_n. This yields appropriate values for ξ_k and η_k. The algorithm described above just gives the basic idea which underlies SLEDGE. Refinements that need to be added such as to avoid numerical instability are not explained here, see [26] for details.

SL02F

SL02F was published by M. Marletta and J. Pryce [24].
This routine approximates the given problem by replacing the coefficient functions with piecewise constant approximations. The aim is to transform the second order problem into a problem that is easy to solve on each subinterval (t_n, t_{n+1}). When the equation is actually solved, the eigenparameter λ is fixed and the problem can be treated as an ordinary differential equation. However, the difference to SLEDGE is that this code transforms $\hat{y}$ and $\hat{p}\hat{y}'$ (defined as in Section 2.2) to *scaled Prüfer's variables* [27]

$$\hat{p}\hat{y}' = S^{\frac{1}{2}} \rho \cos(\theta), \tag{2.39}$$

$$\hat{y} = S^{-\frac{1}{2}} \rho \sin(\theta), \tag{2.40}$$

where the scaling factor S on each subinterval (t_n, t_{n+1}) has the positive constant value S_n and solves the resulting equations

$$\frac{d\theta}{dt} = \frac{S_n}{p_n}\cos^2(\theta) + \frac{\hat{\lambda}g_n - q_n}{S_n}\sin^2(\theta), \tag{2.41}$$

$$\frac{d\log(\rho)}{dt} = \left\{\frac{S_n}{p_n} - \frac{\hat{\lambda}g_n - q_n}{S_n}\right\}\sin(2\theta), \tag{2.42}$$

for $t \in (t_n, t_{n+1})$. The *miss-distance* to determine the two-sided shooting process reads:

$$D(\lambda) = \theta_L(c) - \theta_R(c), \tag{2.43}$$

where $\theta_L(c)$ and $\theta_R(c)$ are the solutions that satisfy the left and right boundary conditions and c is a suitable matching point. For the integration of (2.41), S_n is chosen such that the differential equation becomes easy to solve. Hence, the code treats three different cases, depending on the quantity

$$m = \tau_n h^2, \tag{2.44}$$

where τ_n is defined as in (2.32) and $h = t_{n+1} - t_n$. In SL02F these three cases are

'large positive'	$m > TOL$
'large negative'	$m < -TOL$
'small'	$\lvert m\rvert \leq TOL$

TOL has to be smaller than π^2 and is set to 0.1, cf. [27, pp. 125]. Here TOL does not correspond to the prescribed tolerance of the shooting process. In all cases an explicit solution can be given analytically. Then $D(\lambda)$ has to tend to 0 in order to obtain an approximation for the eigenvalue.
Note that only (2.41) was used to calculate the eigenvalues. For the eigenfunctions, however, also (2.42) has to be integrated. For this purpose the same three cases are used as described above. In this algorithm no numerical integration such as Runge-Kutta is carried out. On each subinterval an ordinary differential equation is solved analytically.

SLEUTH

The code SLEUTH [12] is designed for the solution of fourth-order Sturm–Liouville problems with separated boundary conditions. It implements a shooting procedure to determine a single eigenvalue by replacing the given problem by one with piecewise constant coefficients. The accuracy of the approximation is increased by *Richardson extrapolation.*

MATSLISE

The code MATSLISE [23] incorporates another shooting method for the solution of Sturm-Liouville and Schrödinger equations. As in any other shooting procedure, initial value problems have to be solved repeatedly. The numerical methods which provide the solution approximations of these IVPs are called *Piecewise Perturbation Methods*

(PPMs). Therefore, the coefficient functions of the differential equation are replaced by piecewise polynomials and the resulting equation is referred to as *reference equation*. The method is called *Constant Perturbaton Method* (CPM) if the coefficients are replaced by a piecewise constant function and analogously, it is called *Line Perturbation Method* (LPM) if linear functions are used. Since the PPMs are designed for problems in Sturm-Liouville normal form,

$$\tilde{y}''(t) = (V(t) + \lambda)\tilde{y}(t), \tag{2.45}$$

$$\tilde{y}(a) = \alpha, \quad \tilde{y}'(a) = \beta, \tag{2.46}$$

general Sturm-Liouville problems (2.19) have to be transformed using *Liouville's transformation*, cf. [10]. For these reference equations an exact solution can be found.
In contrast to $SL02F$ and $SLEDGE$, perturbation corrections according to perturbation theory are added to the solution of the reference equation, to increase the accuracy.
Furthermore, a procedure similar to the one used by SLEDGE is carried out to determine the index of the eigenvalue which is calculated.

The reference equation

The notion *reference equation* plays an important role in the context of PPMs. The aim is to replace the coefficient functions in the given differential equations by functions which are represented by polynomials on each subinterval of the underlying mesh. The new equation is called *reference equation*. In case of the CPM the reference equation on the subinterval $[t_{i-1}, t_i]$ has the form

$$\tilde{y}''(\delta) = (\bar{V} - \lambda)\tilde{y}(\delta), \quad \delta \in [0, h_i], \tag{2.47}$$

where $\delta = t - t_{i-1}$, $h_i = t_i - t_{i-1}$ and $\bar{V}$ is a real constant. Hence, the general solution of this reference equation is given by

$$\tilde{y}(\delta) = c_1 \exp((\bar{V} - \lambda)^{1/2}\delta) + c_2 \exp(-(\bar{V} - \lambda)^{1/2}\delta), \tag{2.48}$$

where c_1 and c_2 are free parameters.
In the following we call two linearly independent particular solutions, $\bar{u}$, $\bar{v}$, of this ODE *reference propagators*, if they satisfy the initial conditions $\bar{u}(0) = 1$, $\bar{u}'(0) = 0$ and $\bar{v}(0) = 0$, $\bar{v}'(0) = 1$. They are called propagators because they propagate the solution from t_{i-1} to t_i,

$$\begin{pmatrix} \tilde{y}(t_i) \\ \tilde{y}'(t_i) \end{pmatrix} = \begin{pmatrix} \bar{u}_i(h) & \bar{v}_i(h_i) \\ \bar{u}_i'(h) & \bar{v}_i'(h_i) \end{pmatrix} \begin{pmatrix} \tilde{y}(t_{i-1}) \\ \tilde{y}'(t_{i-1}) \end{pmatrix}, \tag{2.49}$$

where $h_i = t_i - t_{i-1}$ and $\bar{u}_i$ and $\bar{v}_i$ are the propagators of the i-th subinterval $[t_{i-1}, t_i]$.
In MATSLISE both the *Constant Perturbation Method* (CPM) and the *Linear Perturbation Method* (LPM) are implemented.
Coming to the disadvantages of PPMs, we have to mention that the mesh which is used for the numerical calculations is fixed during the whole shooting procedure (at least for

regular problems).
The code also does not offer any possibilities to solve eigenvalue problems for systems of ODEs and also only a limited class of singular problems is covered excluding oscillatory endpoints.

MATSCS

This code was presented in [23] and is a generalization of the code MATSLISE. MATSCS was developed for the solution of coupled channel Schrödinger equations

$$y''(t) = (V(t) - EI)y(t), \quad t \in [a, b], \tag{2.50}$$

where I is the $n \times n$ unity matrix, $V(t)$ is an $n \times n$ symmetric potential matrix which is assumed to be well-behaved in t and y is a set of $nsol \leq n$ column vectors of dimension n. Boundary conditions are typically given as

$$A_0 y(a) + B_0 y'(a) = 0, \tag{2.51}$$
$$A_1 y(b) + B_1 y'(b) = 0, \tag{2.52}$$
$$\tag{2.53}$$

and satisfy the conjointness conditions

$$A_0^T B_0 - B_0^T A_0 = 0, \tag{2.54}$$
$$A_1^T B_1 - B_1^T A_1 = 0, \tag{2.55}$$

and the rank conditions

$$rank(A_0|B_0) = n, \quad rank(A_1|B_1) = n. \tag{2.56}$$

The aim is to solve (2.50) by using a CPM$\{P, N\}$ approach as a propagation method in a shooting procedure. The partition $a = t_0, t_1, t_2, \ldots, t_N = b$ is introduced. A transfer matrix X, which propagates y and y' block wise from one discretization interval to the adjacent, has to be determined. By denoting the current one step interval by $[T, T + h]$, y and y' are propagated by

$$\begin{bmatrix} y(T+h) \\ y'(T+h) \end{bmatrix} = X \begin{bmatrix} y(T) \\ y'(T) \end{bmatrix}. \tag{2.57}$$

The transfer matrix can be written as

$$X = \begin{bmatrix} u(\delta) & v(\delta) \\ u'(\delta) & v'(\delta) \end{bmatrix}, \tag{2.58}$$

where $u(\delta)$ and $v(\delta)$ are two particular solutions of

$$p'' = (V(T + \delta) - EI)p, \quad \delta \in [0, h], \tag{2.59}$$

satisfying the initial conditions $p(0) = 1$, $p'(0) = 0$ and $p(0) = 0$, $p'(0) = 1$, respectively. The matrix $V(T + \delta)$ is an approximate potential matrix

$$V(T + \delta) = \sum_{m=0}^{N} V_m h^m \tilde{P}_m(\delta/h),$$

where $\tilde{P}_m$ denote the shifted Legendre polynomials and the matrix weights V_m result from quadrature. The symmetric matrix V_0 is then diagonalized. It is then denoted as the reference potential V_0^D and the rest of the sum denotes the perturbation.
The actual determination of an eigenvalue of (2.50) is done by a shooting procedure where the miss-distance is given as the determinant

$$\phi(E) = \begin{vmatrix} Y_L & Y_R \\ Y_L' & Y_R' \end{vmatrix}. \tag{2.60}$$

Y_R and Y_L are matrix solutions of (2.50) in a sense that each column of Y_R is a solution and satisfies the right-hand boundary conditions and Y_L is defined correspondingly.
Moreover, a generalization of the Prüfer transformation which is defined for scalar problems makes the whole procedure more robust and allows the code to determine the required eigenvalue by its index.

Chapter 3

Problems Posed on Semi-Infinite Intervals

Boundary value problems in general and eigenvalue problems in particular are very often posed on semi-infinite intervals, see for example the equation for the hydrogen atom in Chapter 2. While the common way of dealing with semi-infinite interval problems is to truncate the interval, see for example [23], another way to solve such problems is to transform them to a finite interval and subsequently apply a suitable numerical method. Here, we present this transformation in a general manner.

3.1 Transformation of the Independent Variable

This chapter deals with problems of the form

$$f(t, z, z', \ldots, z^{(k)}) = 0, \quad a < t < \infty, \tag{3.1}$$

where $a \geq 0$.
By a change of the independent variable, the problem posed on $[a, \infty)$, $a \geq 0$ is transformed to the interval $[0, 1]$. Intervals of the form $[0, \infty)$ are first split into $(0, 1]$ and $[1, \infty)$ and then the latter interval is transformed to $(0, 1]$ via $t \mapsto 1/t$. By applying the transformation $t \mapsto a/t$, intervals of the type $[a, \infty)$ where $a > 0$ can be directly transformed to $(0, 1]$.
To supply the user with the solution on the semi-infinite interval, a graphical back transformation is carried out.
According to [8] and [19], our technique enables the efficient application of collocation methods and classical convergence orders can be observed.
This transformation is not only valuable for eigenvalue problems but also for 'standard' boundary value problems posed on semi-infinite intervals. Therefore we present the technique in a more general setting.
Note that typically this transformation gives rise to essential singularities at 0, see Chapter 4. Since our collocation solver is designed to solve singular problems, the transfor-

mation to a finite interval does not result in any further numerical difficulties. In the following we assume that $z(t)$ is sufficiently smooth.

- **Transformation of a second order equation**

  To illustrate the technique, we consider a second order differential equation posed on $[1, \infty)$. A second order differential equation in its most general form reads

  $$f(t, z_1(t), z_1'(t), z_1''(t), \ldots, z_n(t), z_n'(t), z_n''(t)) = 0, \qquad t \in [1, \infty). \tag{3.2}$$

  Applying the transformation from [9],

  $$z_{i+n}(t) := z_i\left(\frac{1}{t}\right), \tag{3.3}$$

  the problem can be transformed to the finite interval $(0, 1]$. Then we have

  $$z_{i+n}'(t) = -\frac{1}{t^2} z_i'\left(\frac{1}{t}\right), \tag{3.4}$$

  $$\left(z_i'\left(\frac{1}{t}\right)\right)' = -\frac{1}{t^2} z_i''\left(\frac{1}{t}\right). \tag{3.5}$$

  From (3.4) and (3.5) it follows that

  $$\begin{aligned} z_i''\left(\frac{1}{t}\right) &= (-t^2)\left(z_i'\left(\frac{1}{t}\right)\right)' \\ &= (-t^2)(-t^2 z_{i+n}'(t))'. \end{aligned} \tag{3.6}$$

  Inserting (3.6) into the equation

  $$f\left(\frac{1}{t}, z_1\left(\frac{1}{t}\right), z_1'\left(\frac{1}{t}\right), z_1''\left(\frac{1}{t}\right), \ldots, z_n''\left(\frac{1}{t}\right)\right) = 0, \tag{3.7}$$

  yields the new problem formulation on $(0, 1]$,

  $$f\left(\frac{1}{t}, z_{n+1}(t), -t^2 z_{n+1}'(t), (-t^2)(-t^2 z_{n+1}'(t))', \ldots, (-t^2)(-t^2 z_{2n}'(t))'\right) = 0. \tag{3.8}$$

- **Transformation of a third order equation**

  We now transform

  $$f(t, z_1(t), \ldots, z_1^{(o_1)}(t), \ldots, z_n(t), \ldots, z_n^{(o_n)}(t)) = 0, \qquad t \in [1, \infty),\ o_i \leq 3,\ i = 1 \ldots n, \tag{3.9}$$

  to $(0, 1]$. From

  $$\left(z_i''\left(\frac{1}{t}\right)\right)' = -\frac{1}{t^2} z_i^{(3)}\left(\frac{1}{t}\right) \tag{3.10}$$

and the formulas derived for the second order problem, we obtain

$$\begin{aligned} z_i^{(3)}\left(\frac{1}{t}\right) &= -t^2\left(z_i''\left(\frac{1}{t}\right)\right)' \\ &= -t^2(-t^2(-t^2 z_{i+n}'(t))')' \end{aligned} \tag{3.11}$$

and substitute in (3.9) accordingly.
It is important to notice that transformation of boundary conditions other than Dirichlet posed at infinity cannot be carried out in such an automatic manner. In this case further analysis has to be made. Consider for illustration, boundary conditions including the first derivative. According to (3.4) one cannot pose the boundary condition $z_{i+n}'(0) = 0$ without further knowledge of the behaviour of the first derivative of z_{i+n} as it tends to 0.
While for many second order problems typical boundary conditions at infinity are anyway of the Dirichlet type this does not hold in general, if the order of the differential operator is higher than two, cf. [29].
Hence, in the implementation of this automatic transformation problems with such boundary conditions are not accepted to avoid the described difficulties. The user then has to transform the problem manually and supply admissible boundary conditions.

- **Transformation of an arbitrarily high order equation**

  Replacing the highest derivative by

$$(-t^2)\underbrace{[-t^2(-t^2(\ldots(-t^2 z_{i+n}')')')']'}_{\text{(k-1)-products}} = z_i^{(k)} \tag{3.12}$$

  and the lower derivatives accordingly, transforms an equation of arbitrarily high order from $[1, \infty)$ to the finite interval $[0, 1]$.

For illustration we consider the density profile equation which is also treated in [18, p.44].

Example 3.1.

$$z''(t) = -\frac{2}{t}z'(t) + 4(z(t)+1)z(t)(z(t)-0.1), \tag{3.13}$$

$$z'(0) = 0,\ z(\infty) = 0.1. \tag{3.14}$$

Application of the described transformation yields the augmented BVP,

$$z_1''(t) = -\frac{2}{t}z_1'(t) + 4(z_1(t)+1)z_1(t)(z_1(t)-0.1), \tag{3.15}$$

$$z_2''(t) = -\frac{4}{t^4}(z_2(t)+1)z_2(t)(z_2(t)-0.1), \tag{3.16}$$

$$z_1(1) = z_2(1), z_1'(1) = -z_2'(1), z_1(0) = 0, z_2(0) = 0.1. \tag{3.17}$$

3.2 Implementation

The MATLAB Code `bvpsuite`

`bvpsuite` is a code designed for the solution of singular and regular BVPs, cf. [20]. It features a wide range of algorithmic components. The code is capable of computing the solution of implicit nonlinear mixed order systems of ODEs which possibly feature unkown parameters or singularities. For accurate computations a mesh adaptation strategy is realized. Moreover a pathfollowing strategy extends the scope of the code and also index-1 differential algebraic equations can be solved successfully. In the course of this work two new modules were realized in order to augment the existing code. These include

- a driver routine for the interval reduction $[a, \infty)$, $a \geq 0$ to the finite interval $[0, 1]$ and
- a module for the solution of differential eigenvalue problems.

The package `bvpsuite` has been developed in the Matlab versions 7.0-7.1 and requires the Symbolic Math Toolbox. It features a graphical user interface (GUI) which ensures a high level of usability.
The package `bvpsuite` contains the following m-files:

- bvpsuite.m – main routine to start the GUI.
- equations.m – contains the most important parts of the code, e.g. setting up the nonlinear system of equations for the Newton solver.
- solve_nonlinear_sys.m – contains the Newton solver.
- run.m – manages routine calls.
- errorestimate.m – provides error estimates.
- meshadaptation.m – runs the automatic grid control.
- initialmesh.m – provides the initial data for the Newton solver.
- pathfollowing.m – realizes the pathfollowing routine.
- settings.m – opens a window for setting parameters.
- sbvpset.m – sets the options for the Newton solver.
- EVPmodule.m – carries out the reformulation of an EVP to a BVP.
- trafomodule.m – automatically transforms a problem posed on a semi-infinite interval $[a, \infty)$, $a \geq 0$ to a finite domain, $(0, 1/a]$ for $a > 0$, and $[0, 1]$ for $a = 0$.
- backtransf.m – back-transforms the solution to the interval $[a, L] \subset [a, \infty)$, L large.
- plot_results.m – provides graphical solution output.

- plotrange.m – defines settings for a solution plot on a subinterval $[a, L] \subset [a, \infty)$, L large.
- err.m – contains error messages.

We refer the reader to the detailed manual of the code cf. [20].

Chapter 4

Collocation Methods for the Computation of Eigenvalues and Eigenfunctions

4.1 Conversion of the Eigenvalue Problem to a 'Standard' BVP

This method is strongly influenced by an approach for the treatment of unknown parameters in the context of BVPs.
To illustrate the approach introduced in [2] we consider a singular first order eigenvalue problem

$$z'(t) - A(t)z(t) = \lambda z(t), \quad t \in (0,1], \tag{4.1}$$
$$B_0 z(0) + B_1 z(1) = 0, \tag{4.2}$$

where $A(t) = \frac{M(t)}{t^\alpha}$, $\alpha \geq 1$ and B_0 and B_1 are constant $n \times n$ matrices. $M(t)$ is required to be sufficiently smooth. Note that $z(t) \in \mathbb{R}^n$.
The problem is said to have a *singularity of the first kind* if $\alpha = 1$ while for $\alpha > 1$ the singularity is called *singularity of the second kind* or *essential* singularity.
Our aim is to find values for λ so that problem (4.1) – (4.2) has nontrivial solutions $z(t)$. Note that, in general, if $z(t)$ is an eigenfunction then so is $\beta z(t)$, $\beta \in \mathbb{R}$, which means that a priori we do not have uniqueness. The approach illustrated below will incorporate uniqueness of the eigenfunctions.
The method presented in [2] can be applied if for each eigenvalue the corresponding eigenspace is of dimension one. If so, the restriction,

$$\int_0^1 ||z(\tau)||_2^2 d\tau = 1, \tag{4.3}$$

ensures the uniqueness of the eigenfunction up to the sign. Here, we denote $||z(t)||_2^2 := \sum_{i=1}^n |z_i(t)|^2$. This requirement is most commonly satisfied in applications, as for example

in regular Sturm-Liouville problems. The main idea of this approach is to reformulate the original EVP as a standard BVP. First we interpret λ as a function of t and add the trivial equation

$$\lambda'(t) = 0. \tag{4.4}$$

Subsequently, we define

$$x(t) := \int_0^t ||z(\tau)||_2^2 d\tau, \tag{4.5}$$

which yields

$$x'(t) = ||z(t)||_2^2, \quad x(0) = 0, \quad x(1) = 1. \tag{4.6}$$

In [2] it was demonstrated that the solution of the augmented problem, (4.1)–(4.2), (4.4) and (4.6), is isolated if and only if the solution of the original eigenvalue problem (4.1)–(4.2) is isolated. This theoretically supports the presented solution approach.
After this reformulation of the problem, any suitable numerical method for singular boundary value problems in ODEs can be used. As mentioned before the method applied here is *polynomial collocation*, cf. [1], which has proven to be a robust technique for solving boundary value problems with singularities, cf. [9] and references therein. See the section about polynomial collocation for more details.
Collocation is also a suitable method for problems with a singularity of the second kind.

4.2 Singular Boundary Value Problems

First order singular boundary value problems in ODEs often take the form

$$t^\alpha z'(t) = f(t, z(t)), \quad 0 < t \leq 1, \tag{4.7}$$

$$b(z(0), z(1)) = 0, \tag{4.8}$$

where $\alpha \geq 1$, $z \in C[0,1] \cap C^1(0,1]$ and f and b are nonlinear mappings on suitable domains. The case $\alpha = 1$, which determines a singularity of the first kind, is often obtained when partial differential equations are reduced to ODEs using their symmetries. A singularity of the second kind, $\alpha > 1$, on the other hand often results from transformation of problems on semi-infinite intervals to a finite interval.
In the second order case a singular problem is given by

$$z''(t) = \frac{1}{t^\alpha} f(t, z'(t)) + \frac{1}{t^{\alpha+1}} g(t, z(t)), \quad 0 < t \leq 1, \tag{4.9}$$

$$b(y(0), y'(0); y(1), y'(1)) = 0. \tag{4.10}$$

As a model example we consider the linear first order system

$$z'(t) = -\frac{M(t)}{t} z(t) + f(t, z(t)), \quad 0 < t \leq 1. \tag{4.11}$$

Since the right-hand side of the system of singular equations is in general neither continuous nor Lipschitz continuous on the closed interval $[0,1]$, the standard theory for 'smooth'

problems to show existence and uniqueness of a solution cannot be adopted. However, one is still interested in computing continuous solutions of (4.11), $z(t) \in C[0,1]$, or even $z(t) \in C^m[0,1]$, $m \geq 1$. We assume that $M(t)$ can be written as $M(t) = M(0) + tC$, $C \in C[0,1]$. Since the direction field is unsmooth the existence of a smooth solution critically depends on the spectrum of $M(0)$. More precisely, in [13] it is shown that the smoothness of $z(t)$ depends on the smoothness of the nonlinear function f and the size of the positive real parts of the eigenvalues of $M(0)$.
For detailed analytical and numerical results on singular boundary value problems for first order systems see [13], [14], and [15]. For second order problems we refer the reader to [32] and [33].

4.3 Polynomial Collocation

As mentioned in Section 4.1 the numerical method to solve the augmented BVP (4.1)–(4.2), (4.4) and (4.6) is collocation. A collocation solution is a piecewise polynomial that satisfies the ODE in predefined collocation points. We present now the method for a first order problem

$$z'(t) = f(t, z(t)), \quad 0 < t \leq 1, \tag{4.12}$$

$$b(z(0), z(1)) = 0. \tag{4.13}$$

cf. [1]. However, extension to higher order problems is straightforward. Here m is the number of points which are placed in each subinterval.
First we consider a polynomial $p_k(t)$ of degree $\leq m$ defined on $[t_k, t_{k+1}]$ by the interpolation conditions,

$$p_k(t_k) = z_k, \tag{4.14}$$

$$p_k'(t_{kj}) = f(t_{kj}, z_{kj}), \quad 1 \leq j \leq m. \tag{4.15}$$

The points t_{kj} are defined as in Chapter 1, cf. Figure 1.1. The aim is to exploit this information in order to obtain an approximation of the solution at t_{k+1}. Since the polynomial can be written in terms of its first derivative,

$$p_k(t) = z_k + \int_{t_k}^{t} p_k'(\xi) d\xi, \tag{4.16}$$

we can further derive

$$\begin{aligned} p_k(t_{kj}) &= z_k + \int_{t_k}^{t_{kj}} \sum_{i=1}^{m} f(t_{ki}, z_{ki}) L_i \left(\frac{\xi - t_k}{h_k} \right) d\xi \\ &= z_k + h_k \sum_{i=1}^{m} \alpha_{ji} f(t_{ki}, z_{ki}) =: z_{kj}, \end{aligned} \tag{4.17}$$

$$p_k(t_{k+1}) = z_k + h_k \sum_{i=1}^{m} \beta_i f(t_{ki}, z_{ki}) =: z_{k+1}, \tag{4.18}$$

and hence obtain a numerical scheme for the approximation $(z_0, \ldots, z_N)$ of the exact solution $z(t)$. The approximations z_{kj} are defined implicitly by a system of (non)linear equations, cf. (4.17). This system of nonlinear equations is typically solved by the Newton method. L_i denotes the i-th Lagrange basis polynomial on the interval $[0, 1]$. The approximating solution is a globally continuous piecewise polynomial function $p(t) := p_k(t)$, $t \in [t_k, t_{k+1}]$, $k = 0, \ldots, N-1$ which satisfies the boundary conditions $b(p(0), p(1)) = 0$. Combining (4.15) and (4.17) one notices that the piecewise polynomial function $p(t)$ satisfies the ODE at the collocation points.
The code `bvpsuite` uses the *Runge-Kutta* basis to represent the polynomials, which offers certain computational advantages (cf. [3]).

4.4 Convergence of Collocation Methods

We state here classical results on the numerical solution obtained by the collocation method presented in [1] and [4] and make use of the notation introduced in Chapter 1. In a numerical algorithm it is of interest to control the *global error* $e_h(t_k)$, that is

$$e_h(t_k) := z(t_k) - z_k, \tag{4.19}$$

where $z(t_k)$ denotes the exact solution evaluated at t_k and z_k is the numerical solution at the k-th mesh point. The notation clearly indicates that the global error depends on the chosen step size. The *order of convergence* characterizes the efficiency for obtaining a desired level of accuracy.

A numerical algorithm is called *convergent* of order p if

$$\max_{0 \le k \le N} |e_h(t_k)| = O(\mathbf{h}^p), \quad \mathbf{h} \to 0. \tag{4.20}$$

In the following the results are presented for linear first order problems:

$$z'(t) = \frac{f(t, z)}{t^\alpha}, \quad t \in (0, 1], \tag{4.21a}$$

$$B_0 z(0) + B_1 z(1) = \beta. \tag{4.21b}$$

For regular problems, where $\alpha = 0$ in (4.21a), with sufficiently smooth data, the following convergence results hold (cf. [1], [4]).

Theorem 4.1. *Let $z(t)$ be an isolated, sufficiently smooth solution of (4.21) and let f be twice continuously differentiable in a neighborhood of z. Then for any collocation scheme there exist constants $r, \boldsymbol{h}_0 > 0$ such that the following statements hold for all meshes Δ with $\boldsymbol{h} \le \boldsymbol{h}_0$:*

- *There exists a unique solution $p(t)$ of the nonlinear system of collocation equations in a tube of radius r around $z(t)$.*

- *This solution can be computed by Newton's method which converges quadratically provided that the initial guess $p^{[0]}(t)$ is sufficiently close to $z(t)$.*
- *The following error estimates hold:*

$$\|R_\Delta(z) - R_\Delta(p)\|_\Delta = O(\boldsymbol{h}^{m+\nu}), \tag{4.22a}$$
$$\|z - p\| = O(\boldsymbol{h}^{m+\mu}), \tag{4.22b}$$
$$\|z^{(l)} - p^{(l)}\| = O(\boldsymbol{h}^{m+1-l}), \quad l = 1, \ldots, m, \tag{4.22c}$$

provided that the m canonical collocation points $\rho_1, \ldots, \rho_m$ satisfy the orthogonality conditions

$$\int_0^1 s^k \prod_{l=1}^{m} (s - \rho_l)\, ds = 0, \quad k = 0, \ldots, \nu - 1, \quad \nu \le m. \tag{4.23}$$

In (4.22b)*, $\mu = 0$ for $\nu = 0$ and $\mu = 1$ otherwise.*

The result in (4.22a) is also called *superconvergence.*
For proofs see for example [1].
For a singularity of the first kind, where $\alpha = 1$ in (4.21), similar existence and uniqueness results can be proven under appropriate assumptions which guarantee the well-posedness of the boundary value problem and the smoothness of its solution. The estimates (4.22a) and (4.22b) have to be replaced by

$$\|R_\Delta(z) - R_\Delta(p)\|_\Delta = O(\mathbf{h}^m |\ln(\mathbf{h})|^{n_0-1}), \tag{4.24a}$$
$$\|z - p\| = O(\mathbf{h}^m |\ln(\mathbf{h})|^{n_0-1}) \tag{4.24b}$$

with $\nu = 0$ and

$$\|R_\Delta(z) - R_\Delta(p)\|_\Delta = O(\mathbf{h}^{m+1} |\ln(\mathbf{h})|^{n_0-1}), \tag{4.25a}$$
$$\|z - p\| = O(\mathbf{h}^{m+1} |\ln(\mathbf{h})|^{n_0-1}) \tag{4.25b}$$

if (4.23) holds with $\nu \ge 1$, see [21], [5] for details. The positive integer n_0 is defined by the structure of the linearization of (4.21), see for example [13]. The perturbation of the convergence order by the logarithmic terms is usually too small to be noticed experimentally.

For problems with an essential singularity, no theoretical results are known for general high-order collocation methods. However, usually the *stage order* $O(\mathbf{h}^m)$ is observed for any choice of symmetric collocation points. The superconvergence orders observed experimentically for $\nu \ge 1$ in (4.23) are

$$\|R_\Delta(z) - R_\Delta(p)\|_\Delta = O(\mathbf{h}^{m+\gamma}), \tag{4.26a}$$
$$\|z - p\| = O(\mathbf{h}^{m+\gamma}), \tag{4.26b}$$

where $0 < \gamma = \gamma(\alpha) < 1$, and γ decreases with increasing α.
For collocation points which are not symmetric, rapid divergence of the numerical solution was observed.

Empirical Convergence Order

In the following consider an equidistant mesh Δ with step size $h_i = \left(\frac{1}{2}\right)^i h_0$.
The vector z_{h_i} is the solution of the numerical method with step size h_i. Accordingly $z_{h_{i+1}}$ denotes the numerical solution with step size $\frac{h_i}{2}$ and $z_{h_{i+2}}$ is the analogue for $\frac{h_i}{4}$. These grid functions constitute a finite-dimensional normed subspace of the solution space of the analytical problem. We now assume that the error is $O(\mathbf{h}^p)$ and additionally

$$z_{h_i} - z(t) \approx C h_i^p, \; h_i \to 0,$$

where z is the exact solution and C is a constant that is independent of t and h. Consequently,

$$z_{h_{i+1}} - z \approx C \left(\frac{h_i}{2}\right)^p,$$
$$z_{h_{i+2}} - z \approx C \left(\frac{h_i}{4}\right)^p,$$

and therefore it holds that

$$z_{h_i} - z_{h_{i+1}} \approx C h_i^p \left(1 - \frac{1}{2^p}\right), \tag{4.27}$$
$$z_{h_{i+1}} - z_{h_{i+2}} \approx C h_i^p \left(1 - \frac{1}{2^p}\right) \frac{1}{2^p}. \tag{4.28}$$

This leads to an empirical estimate for the order of convergence

$$p \approx \frac{\ln\left(\frac{\|z_{h_i} - z_{h_{i+1}}\|_\Delta}{\|z_{h_{i+1}} - z_{h_{i+2}}\|_\Delta}\right)}{\ln(2)}.$$

Another way to estimate the order of convergence is to calculate a so-called *reference solution*, which is a numerical approximation for the exact solution computed with a very small step size h_{ref}. By assuming that the step size h_{ref} is small enough so that

$$\|z_{h_i} - z_{h_{ref}}\|_\Delta \gg \|z_{h_{ref}} - z\|_\Delta \tag{4.29}$$

holds, the estimated order of convergence is

$$p \approx \frac{\ln\left(\frac{\|z_{h_i} - z_{h_{ref}}\|_\Delta}{\|z_{h_{i+1}} - z_{h_{ref}}\|_\Delta}\right)}{\ln(2)}.$$

4.5 Numerical Example

As an illustrating example we consider the radial equation of the hydrogen problem (2.13). For reasons of comparison with other literature, cf. [31], we set $u(r) = rR(r)$.

Example 4.1.

$$-u''(r) + \left(\tfrac{l(l+1)}{r^2} - \tfrac{\gamma}{r}\right) u(r) = \lambda u(r), \quad r \in (0,\infty), \tag{4.30}$$

$$u(0) = 0, \quad u(\infty) = 0. \tag{4.31}$$

Reformulation to a BVP according to the method described in Section 4.1 and a subsequent application of the transformation presented in Chapter 3 yields, $t \in [0,1]$,

$$u_1''(t) = \left(\frac{l(l+1)}{t^2} - \frac{\gamma}{t}\right) u_1(t) - u_2(t)u_1(t), \tag{4.32}$$

$$u_2'(t) = 0, \tag{4.33}$$

$$u_3'(t) = u_1^2(t), \tag{4.34}$$

$$u_4''(t) = -\frac{2}{t}u_4'(t) + \left(\frac{l(l+1)}{t^2} - \frac{\gamma}{t^3}\right) u_4(t) - \frac{1}{t^4}u_5(t)u_4(t), \tag{4.35}$$

$$u_5'(t) = 0, \tag{4.36}$$

$$u_6'(t) = -\frac{1}{t^2}u_4^2(t), \tag{4.37}$$

with the boundary conditions

$$u_1(0) = 0, \quad u_4(0) = 0, \quad u_1(1) = u_4(1), \quad u_1'(1) = -u_4'(1), \quad u_2(1) = u_5(1),$$

$$u_3(1) = u_6(1), \quad u_3(0) = 0, \quad u_6(0) = 1.$$

In the following computations we set $\gamma = 2$ and $l = 0$.
For $l = 0$ this problem is regular if posed on a finite interval and transformed to first order using Euler's transformation. However, the transformation from a semi-infinite interval generically gives rise to singularities. Therefore also this problem becomes singular, cf. (4.32)–(4.37).

We used the MATLAB code `bvpsuite` to calculate the eigenfunctions and eigenvalues. Table 4.1 shows the results received without the mesh adaptation strategy.
Tables 4.2 and 4.3 illustrate the advantages of the mesh adaptation strategy incorporated in `bvpsuite`. For obtaining the same accuracy of the computed eigenvalue far less mesh points are required.

n	$\lambda_n^{(0)}$	λ_n^{exact}	λ_n	abs. err.	rel. err.	Gauss	N
1	-1.0000	-1	-1.00000035e+00	3.5409e−07	3.5409e−07	1	1000
2	-0.2500	$-\frac{1}{4}$	-2.50000198e−01	1.9790e−07	7.9161e−07	1	1000
3	-0.1100	$-\frac{1}{9}$	-1.11110675e−01	4.3595e−07	3.9236e−06	1	1000
4	-0.0625	$-\frac{1}{16}$	-6.24973725e−02	2.6275e−06	4.2040e−05	1	1000
5	-0.0400	$-\frac{1}{25}$	-3.99923145e−02	7.6855e−05	1.9214e−04	1	1000
6	-0.0278	$-\frac{1}{36}$	-2.77605356e−02	1.7242e−05	6.2072e−04	1	1000
7	-0.0204	$-\frac{1}{49}$	-2.03750305e−02	3.31335−05	1.6235e−03	1	1000
8	-0.0156	$-\frac{1}{64}$	-1.55679504e−02	5.7050e−05	3.6512e−03	1	1000
9	-0.0123	$-\frac{1}{81}$	-1.22560174e−02	8.9662e−05	7.2263e−03	1	1000

Table 4.1: Example 4.1: Eigenvalues of the hydrogen atom with $l = 0$.

n	$\lambda_n^{(0)}$	λ_n^{exact}	λ_n	abs. err.	rel. err.	Gauss	N	abs.err (adapt)	rel.err (adapt)
3	-0.11	$-\frac{1}{9}$	-1.11111122e−01	6.1430e−08	5.5287e−07	2	113	2.7256e−09	2.4530e−08
4	-0.0625	$-\frac{1}{16}$	-6.25000293e−02	2.9318e−08	4.6908e−07	2	175	4.2353e−09	6.7765e−08
5	-0.0400	$-\frac{1}{25}$	-3.99998136e−02	1.8634e−07	4.6585e−06	2	219	8.7454e−07	3.4982e−06
6	-0.0278	$-\frac{1}{36}$	-2.77775526e−02	2.2522e−07	8.1078e−06	2	391	2.2522e−07	8.1078e−06

Table 4.2: Example 4.1 with $l = 2$: Eigenvalues of the hydrogen atom for the four lowest values of the principal quantum number n. We applied the mesh adaptation strategy to achieve accurate results and compared those to the errors achieved on equidistant meshes with the same number of points as in the final adapted mesh. The absolute tolerance was set to 10^{-9}.

n	$\lambda_n^{(0)}$	λ_n^{exact}	λ_n	abs. err.	rel. err.	Gauss	N	abs.err (adapt)	rel.err (adapt)	N_{adapt}
3	-0.11	$-\frac{1}{9}$	-1.11111111e−01	1.2040e−10	1.0892e−09	2	535	1.2103e−10	1.0836e−09	240
4	-0.0625	$-\frac{1}{16}$	-6.25000005e−02	4.7800e−10	7.6480e−09	2	488	4.7817e−10	7.6508e−09	397
5	-0.0400	$-\frac{1}{25}$	-3.99999983e−02	1.6866e−09	4.2164e−08	2	750	1.6944e−09	4.2359e−08	469
6	-0.0278	$-\frac{1}{36}$	-2.77777774e−02	5.7082e−10	2.0549e−08	2	1804	5.7082e−10	2.0549e−08	1804

Table 4.3: Example 4.1 with $l = 2$: Eigenvalues and errors computed on equidistant meshes compared to the corresponding results achieved by applying a mesh adaptation strategy. The resulting number of subintervals, N and N_{adapt} respectively, are presented.

Figures 4.1 and 4.2 show the adapted meshes on which the eigenfunctions of Example 4.1 were calculated. Since for higher values of n the eigenfunctions become more oscillatory, the mesh adaptation strategy is forced to insert a large number of mesh points in order to fulfill the prescribed tolerances. This leads to almost equidistant meshes with a large number of mesh points. On the other hand, for lower values of n the automatically selected mesh shows a concentration of points where the function has a steep profile.

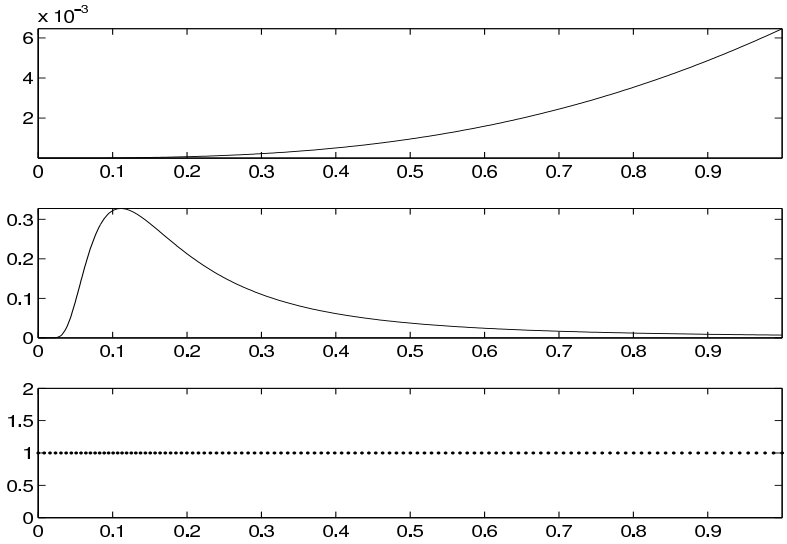

Figure 4.1: Example 4.1 with $l = 2$ and $n = 3$: Approximations of the solution u on $[0, 1]$ (uppermost graph) and u on $[1, \infty)$ transformed to $[0, 1]$ (central graph). The computations were carried out with 2 Gaussian points and $tol_a = 10^{-4}$ and $tol_r = 10^{-2}$. The automatically selected mesh contains 113 points (lower graph).

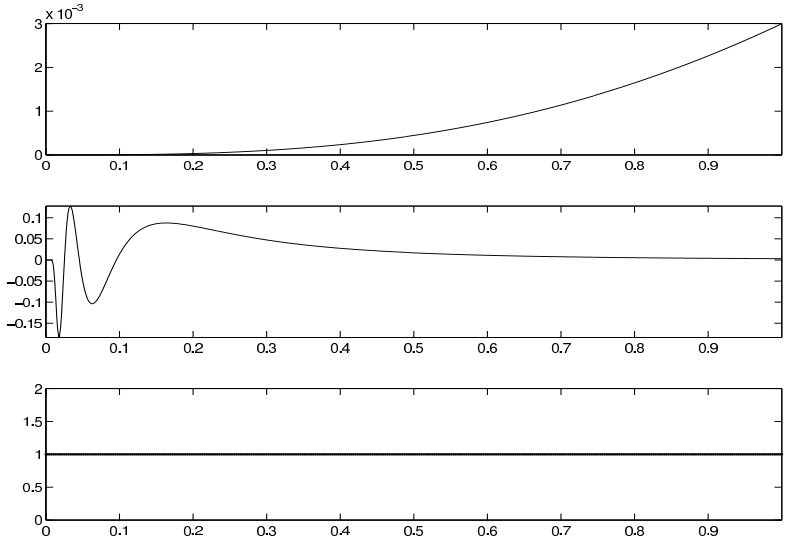

Figure 4.2: Example 4.1 with $l = 2$ and $n = 6$: Approximations of the solution u on $[0, 1]$ (uppermost graph) and u on $[1, \infty)$ transformed to $[0, 1]$ (central graph). The computations were carried out with 2 Gaussian points and $tol_a = 10^{-4}$ and $tol_r = 10^{-2}$. The automatically selected mesh contains 319 points (lower graph).

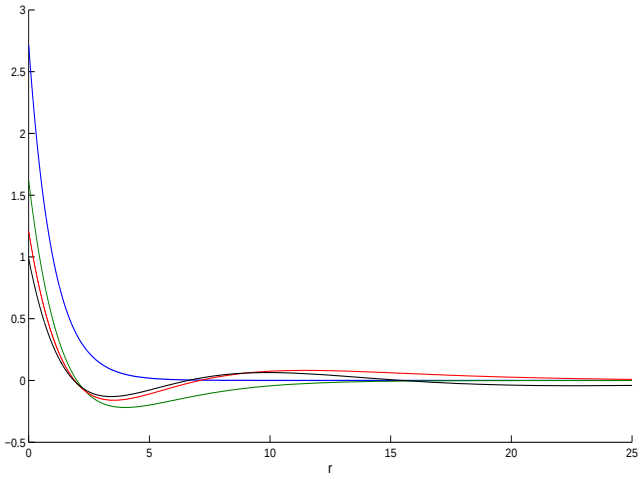

Figure 4.3: Example 4.1: Approximations of $R(r)$ for $\gamma = 2$ and $l = 0$ fixed. $n = 1$ – blue line, $n = 2$ – green line, $n = 3$ – red line, $n = 4$ – black line.

Figure 4.3 and Figure 4.4 show that the wave functions (eigenfunctions) become more oscillatory as n in (2.14) increases. In more detail, the number of zeros of the eigenfunction is given by $n - l - 1$, where n denotes the principal quantum number and l is the

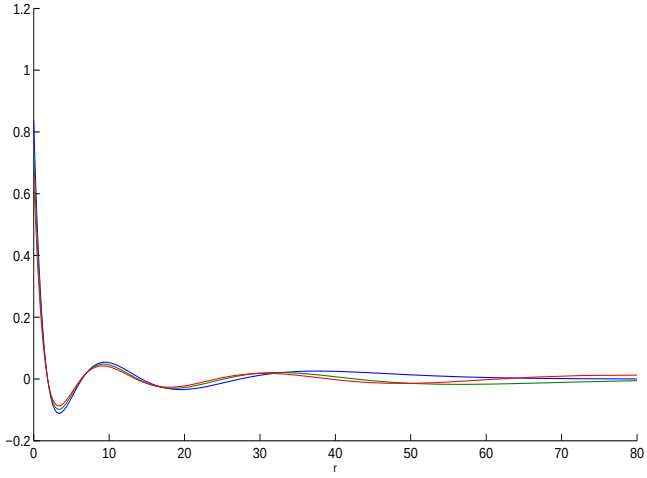

Figure 4.4: Example 4.1: Approximations of $R(r)$ for $\gamma = 2$ and $l = 0$ fixed. $n = 5$ –blue line, $n = 6$ – green line, $n = 7$ –red line.

angular momentum quantum number. As the index increases the eigenvalue tends to zero.

Figure 4.5 illustrates the behaviour of the error as a function of r as $h \to 0$. This experiment shows the expected decrease in the error as h tends to 0.

For the computation of the eigenfunctions no starting profile was used. Therefore a faster convergence of the Newton solver might be achieved if further calculations are done using the given computed solution as starting profile.
As already known from Chapter 2 the eigenvalues accumulate at 0. This makes the solution more sensitive to starting values as n in (2.14) increases, which can be seen in Table 4.1 above. Occasionally the code computes a different solution if the starting value is badly chosen. Furthermore it was observed that the eigenpair computed by our method is not only sensitive with respect to the starting value but also the number of points in the initial grid. Depending on the given problem, it can be rather difficult to choose the right combination of number of intervals and starting values to compute not only one eigenvalue–eigenfunction pair, but rather a sequence of them. For example, this can be observed if the eigenvalues lie close to each other. This problem can be overcome if the starting profiles are provided with reasonable accuracy for instance as the result of the application of the matrix method described in Chapter 5.

Tables 4.4–4.9 below show the estimated order of convergence for the first and fourth solution component of the augmented BVP (4.32)–(4.37) with $l = 0$ and $n = 1$. Tables 4.10–4.15 below show the estimated order of convergence for the first and fourth solution component of the augmented BVP (4.32)–(4.37) with $l = 1$ and $n = 2$. $u_{h_i} \in \mathbb{R}^{(N_i+1)6}$ denotes the numerical approximation. With $u_{h_i}^j$, $j = 1 \dots 6$, we denote the approximations for the solution components respectively.

i	$\|\|u^1_{h_i} - u^1_{h_{i+1}}\|\|$	p_{mesh}	$\|\|u^1_{h_i} - u^1_{h_{i+1}}\|\|_{grid}$	p_{grid}
1	5.085848e−04	1.994396	4.168537e−04	1.993940
2	1.276410e−04	1.998784	1.046521e−04	1.998671
3	3.193717e−05	1.999707	2.618715e−05	1.999679
4	7.985914e−06	1.999927	6.548244e−06	1.999920
5	1.996579e−06	1.999982	1.637151e−06	1.999980
6	4.991510e−07	1.999995	4.092934e−07	1.999995
7	1.247881e−07	1.999999	1.023237e−07	1.999999
8	3.119706e−08	—	2.558095e−08	—

Table 4.4: Example 4.1: Estimated order of convergence for the first solution component with $n = 1$ and $l = 0$ for 1 equidistant point

i	$\|\|u^4_{h_i} - u^4_{h_{i+1}}\|\|$	p_{mesh}	$\|\| u^4_{h_i} - u^4_{h_{i+1}}\|\|_{grid}$	p_{grid}
1	6.968058e−04	2.002786	6.968058e−04	2.002786
2	1.738654e−04	2.005332	1.738654e−04	2.005332
3	4.330600e−05	2.000873	4.330600e−05	2.000873
4	1.081995e−05	2.000137	1.081995e−05	2.000137
5	2.704729e−06	2.000076	2.704729e−06	2.000076
6	6.761465e−07	1.999998	6.761465e−07	1.999998
7	1.690369e−07	2.000005	1.690369e−07	2.000005
8	4.225907e−08	—	4.225907e−08	—

Table 4.5: Example 4.1: Estimated order of convergence for the fourth solution component with $n = 1$ and $l = 0$ for 1 equidistant point

i	$\|\|u^1_{h_i} - u^1_{h_{i+1}}\|\|$	p_{mesh}	$\|\|u^1_{h_i} - u^1_{h_{i+1}}\|\|_{grid}$	p_{grid}
1	1.318707e−06	3.973531	6.673500e−07	3.974119
2	8.394531e−08	3.995758	4.246436e−08	3.995902
3	5.262032e−09	3.998900	2.661572e−09	3.998923
4	3.291279e−10	3.999654	1.664725e−10	3.999557
5	2.057543e−11	3.998747	1.040773e−11	4.002719
6	1.287082e−12	3.985266	6.492584e−13	3.984297
7	8.126833e−14	3.584963	4.102274e−14	4.103166
8	6.772360e−15	—	2.386980e−15	—

Table 4.6: Example 4.1: Estimated order of convergence for the first solution component with $n = 1$ and $l = 0$ for 3 equidistant points

i	$\|\|u^4_{h_i} - u^4_{h_{i+1}}\|\|$	p_{mesh}	$\|\| u^4_{h_i} - u^4_{h_{i+1}}\|\|_{grid}$	p_{grid}
1	1.101407e−05	4.560435	1.101407e−05	4.560435
2	4.667887e−07	4.156966	4.667887e−07	4.156966
3	2.616670e−08	4.023932	2.616670e−08	4.020808
4	1.608513e−09	4.007994	1.612000e−09	4.006897
5	9.997656e−11	4.000844	1.002695e−10	4.001710
6	6.244881e−12	4.000613	6.259424e−12	4.002013
7	3.901393e−13	4.002645	3.906684e−13	4.003573
8	2.433904e−14	—	2.435638e−14	—

Table 4.7: Example 4.1: Estimated order of convergence for the fourth solution component with $n = 1$ and $l = 0$ for 3 equidistant points

i	$\|\|u^1_{h_i} - u^1_{h_{i+1}}\|\|$	p_{mesh}	$\|\| u^1_{h_i} - u^1_{h_{i+1}}\|\|_{grid}$	p_{grid}
1	3.994545e−09	6.303264	3.994545e−09	6.299036
2	5.058198e−11	5.976238	5.073042e−11	5.963423
3	8.034684e−13	5.938533	8.130163e−13	5.931328
4	1.310063e−14	3.182203	1.332268e−14	3.206451
5	1.443290e−15	0.000000	1.443290e−15	-0.106915
6	1.443290e−15	-0.106915	1.554312e−15	-0.099536
7	1.554312e−15	-1.050626	1.665335e−15	-1.000000
8	3.219647e−15	—	3.330669e−15	—

Table 4.8: Example 4.1: Estimated order of convergence for the first solution component with $n = 1$ and $l = 0$ for 3 Gaussian points

i	$\|\|u^4_{h_i} - u^4_{h_{i+1}}\|\|$	p_{mesh}	$\|\| u^4_{h_i} - u^4_{h_{i+1}}\|\|_{grid}$	p_{grid}
1	8.224465e−07	6.711230	9.829328e−07	4.599578
2	7.849211e−09	5.284870	4.054272e−08	4.854216
3	2.013361e−10	5.937768	1.401678e−09	4.979019
4	3.284545e−12	5.949071	4.444408e−11	5.018679
5	5.316506e−14	5.203045	1.371011e−12	5.007493
6	1.443290e−15	-0.884523	4.262216e−14	3.999648
7	2.664535e−15	-0.058894	2.664535e−15	-0.115477
8	2.775558e−15	—	2.886580e−15	—

Table 4.9: Example 4.1: Estimated order of convergence for the fourth solution component with $n = 1$ and $l = 0$ for 3 Gaussian points

i	$\|\|u^1_{h_i} - u^1_{h_{i+1}}\|\|$	p_{mesh}	$\|\|u^1_{h_i} - u^1_{h_{i+1}}\|\|_{grid}$	p_{grid}
1	8.272038e−04	1.729028	8.272038e−04	1.729028
2	2.495302e−04	1.979731	2.495302e−04	1.979731
3	6.326516e−05	1.996560	6.326516e−05	1.996560
4	1.585405e−05	1.999193	1.585405e−05	1.999193
5	3.965728e−06	1.999802	3.965728e−06	1.999802
6	9.915685e−07	1.999951	9.915685e−07	1.999951
7	2.479006e−07	1.999988	2.479006e−07	1.999988
8	6.197568e−08	—	6.197568e−08	—

Table 4.10: Example 4.1: Estimated order of convergence for the first solution component with $n = 2$ and $l = 1$ for 1 equidistant point

i	$\|\|u^4_{h_i} - u^4_{h_{i+1}}\|\|$	p_{mesh}	$\|\|u^4_{h_i} - u^4_{h_{i+1}}\|\|_{grid}$	p_{grid}
1	4.667196e−03	2.127049	4.667196e−03	2.127049
2	1.068441e−03	1.981596	1.068441e−03	1.981596
3	2.705395e−04	2.014268	2.705395e−04	2.014268
4	6.696926e−05	2.003567	6.696926e−05	2.003567
5	1.670097e−05	2.000892	1.670097e−05	2.000892
6	4.172662e−06	2.000223	4.172662e−06	2.000223
7	1.043004e−06	2.000056	1.043004e−06	2.000056
8	2.607410e−07	—	2.607410e−07	—

Table 4.11: Example 4.1: Estimated order of convergence for the fourth solution component with $n = 2$ and $l = 1$ for 1 equidistant point

i	$\|\|u^1_{h_i} - u^1_{h_{i+1}}\|\|$	p_{mesh}	$\|\|u^1_{h_i} - u^1_{h_{i+1}}\|\|_{grid}$	p_{grid}
1	2.116202e−05	3.746490	2.116202e−05	3.746490
2	1.576708e−06	3.869702	1.576708e−06	3.869702
3	1.078587e−07	3.993161	1.078587e−07	3.993161
4	6.773200e−09	3.998092	6.773200e−09	3.998092
5	4.238852e−10	3.999505	4.238852e−10	3.999505
6	2.650193e−11	3.999626	2.650193e−11	3.999626
7	1.656800e−12	3.997983	1.656800e−12	3.997983
8	1.036948e−13	—	1.036948e−13	—

Table 4.12: Example 4.1: Estimated order of convergence for the first solution component with $n = 2$ and $l = 1$ for 3 equidistant points

i	$\|\|u^4_{h_i} - u^4_{h_{i+1}}\|\|$	p_{mesh}	$\|\|u^4_{h_i} - u^4_{h_{i+1}}\|_{grid}$	p_{grid}
1	3.072696e−04	3.357097	3.072696e−04	3.357097
2	2.998698e−05	4.363612	2.998698e−05	4.363612
3	1.456649e−06	4.081981	1.456649e−06	4.081981
4	8.601138e−08	4.005522	8.601138e−08	4.003213
5	5.355175e−09	4.001371	5.363752e−09	4.001545
6	3.343804e−10	4.000304	3.348757e−10	4.001133
7	2.089437e−11	4.000243	2.091330e−11	4.000829
8	1.305678e−12	—	1.306330e−12	—

Table 4.13: Example 4.1: Estimated order of convergence for the fourth solution component with $n = 2$ and $l = 1$ for 3 equidistant points

i	$\|\|u^1_{h_i} - u^1_{h_{i+1}}\|\|$	p_{mesh}	$\|\|u^1_{h_i} - u^1_{h_{i+1}}\|\|_{grid}$	p_{grid}
1	2.203592e−06	9.341077	2.203592e−06	9.341077
2	3.397714e−09	4.979050	3.397714e−09	4.979050
3	1.077317e−10	5.969322	1.077317e−10	5.969322
4	1.719486e−12	6.011198	1.719486e−12	6.011198
5	2.665923e−14	5.024999	2.665923e−14	4.976904
6	8.187895e−16	2.075288	8.465451e−16	2.023847
7	1.942890e−16	-0.440573	2.081668e−16	-0.341037
8	2.636780e−16	—	2.636780e−16	—

Table 4.14: Example 4.1: Estimated order of convergence for the first solution component with $n = 2$ and $l = 1$ for 3 Gaussian points

i	$\|\|u^4_{h_i} - u^4_{h_{i+1}}\|\|$	p_{mesh}	$\|\|u^4_{h_i} - u^4_{h_{i+1}}\|_{grid}$	p_{grid}
1	5.221505e−05	5.280777	5.221505e−05	4.345399
2	1.343146e−06	4.611058	2.568622e−06	4.044302
3	5.496125e−08	6.308793	1.556840e−07	4.930927
4	6.932990e−10	5.928641	5.103721e−09	5.014538
5	1.138209e−11	5.942675	1.578922e−10	5.009676
6	1.850539e−13	5.983217	4.901150e−12	5.009057
7	2.925300e−15	0.434261	1.522025e−13	5.007629
8	2.164935e−15	—	4.731241e−15	—

Table 4.15: Example 4.1: Estimated order of convergence for the fourth solution component with $n = 2$ and $l = 1$ for 3 Gaussian points

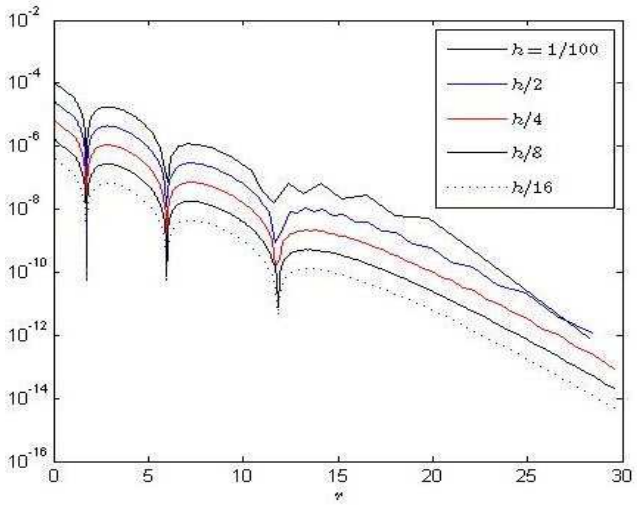

Figure 4.5: Error pattern of the solution $u(r)$ of Example 4.1 for decreasing step sizes.

Figures 4.6 and 4.7 illustrate the results given in the tables. However only the errors at mesh points were used for the illustrations.

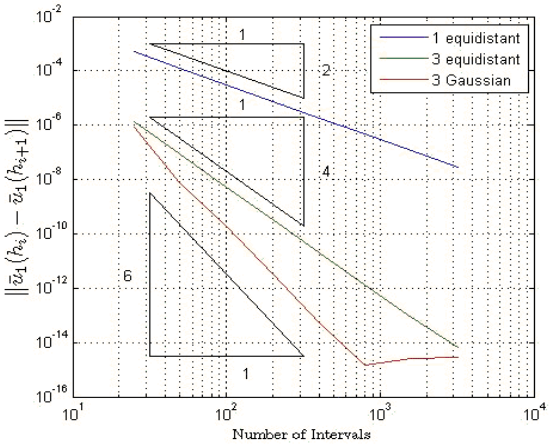

Figure 4.6: Example 4.1: Order of convergence for $u^1_{h_i}$ with $n = 1$ and $l = 0$.

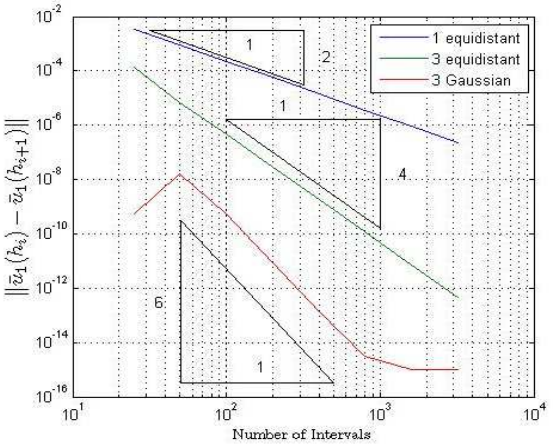

Figure 4.7: Example 4.1: Order of convergence for $u^1_{h_i}$ with $n = 2$ and $l = 0$.

4.6 Discussion

The numerical method described in Chapter 4 requires the solution of a singular BVP for each pair of eigenvalues/eigenfunctions. This is a drawback in cases where several eigenvalues have to be computed.
When solving the one-dimensional, time-independent Schrödinger equation, approximately 10 eigenvalues are sought. In this case it is more efficient to use a so-called *matrix method*, where a larger number of eigenvalues and eigenvectors are obtained simultaneously, see Chapter 5.
The fact that the sequence of eigenvalues lies in $[-1, 0)$ and accumulates at zero makes the computations more delicate.
We encountered the phenomenon that the resulting pair depends critically on the starting value for the nonlinear solver and the initial grid. However if only a few eigenvalues have to be calculated, the technique described above is of advantage because stability and convergence results are already known from the study of polynomial collocation for singular BVPs. As shown in the figures above the empirical convergence results are satisfactory. The order of convergence increases with the number of collocation points. This demonstrates that the given results are reliable.
Another advantage of this technique is that it is generally applicable to a wide range of problems without the need of manual transformation or other preprocessing. The reformulations required for this approach are automatically performed by our code.
Note that by adding the new equations one implicitly requires that the algebraic multiplicity of the eigenvalue is 1. This is indeed a restriction on the problem class which can be treated by this method. According to the presented results this method is highly successful in cases where a particular eigenvalue has to be computed.
Nevertheless, to have a better insight it has to be investigated how the given approach compares to methods which are specifically designed for eigenvalue problems such as in [27], cf. Chapter 2.
An attractive alternative is constituted by *matrix methods*, which turn out to be easily implemented and reliable. They represent a reasonable approach if a sequence of eigenvalues has to be computed or starting profiles for the collocation approach are sought, see Chapter 5 below.

4.7 Implementation of the Eigenvalue Module

Our goal was to augment the existing collocation code `bvpsuite` with a module for solving eigenvalue problems for ODEs. As described above this was done by interpreting the given eigenvalue problem as a related boundary value problem. The module is now able to recast the eigenvalue problem (EVP) according to the method described above automatically. Therefore, the GUI had to be changed, to enable the user to specify that the given problem is an eigenvalue problem. The m-file evpmodule.m of the `bvpsuite` package realizes this reformulation. Since the numerical method which approximates the solution of the BVP is polynomial collocation, in general a set of nonlinear equations

results. These equations are solved by the Newton method. Therefore an initial guess has to be provided. For eigenvalue problems the user can either provide a guess for the eigenvalue only, which corresponds to a constant function of the prescribed value, or in case that an approximiation of the eigenfunction is known, an initial profile can be prescribed.
To illustrate the handling of the GUI, we consider Example 4.1.

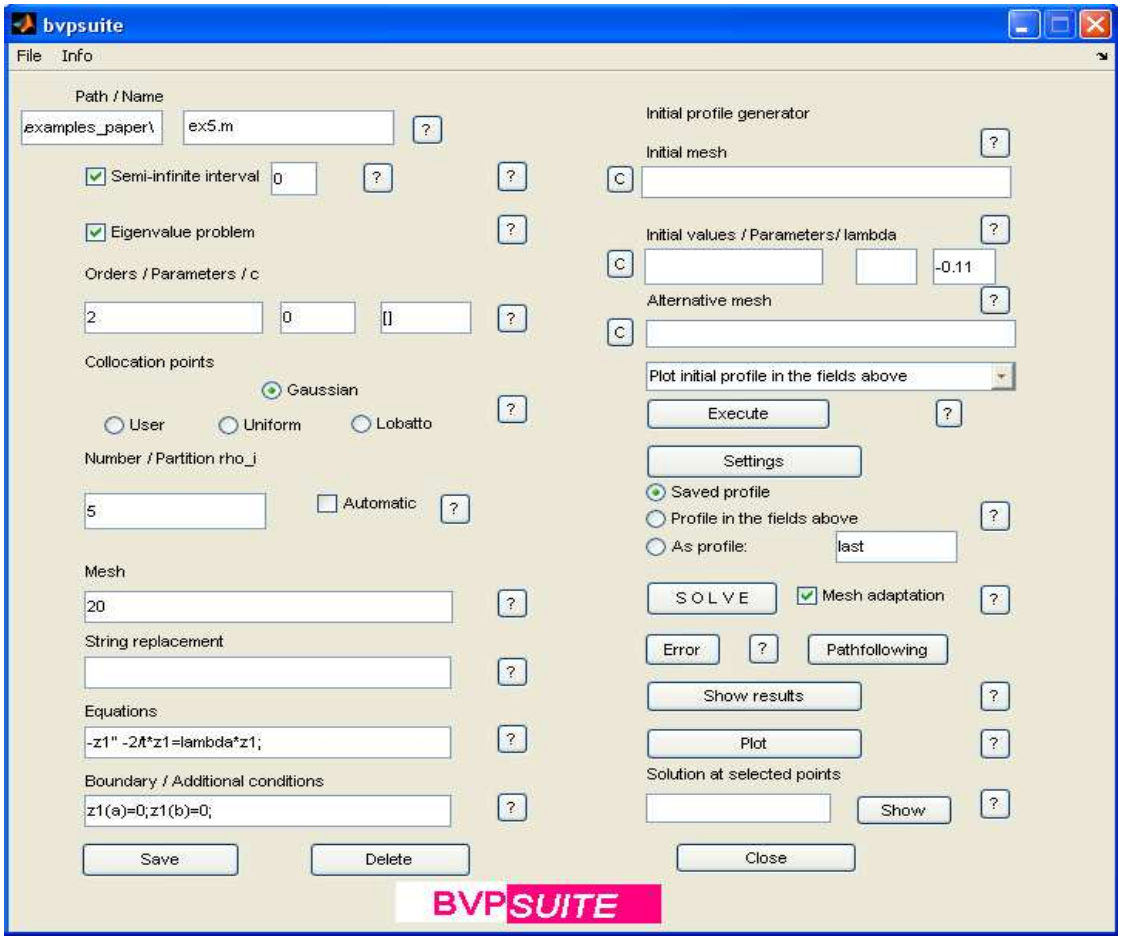

Figure 4.8: GUI input for Example 4.1

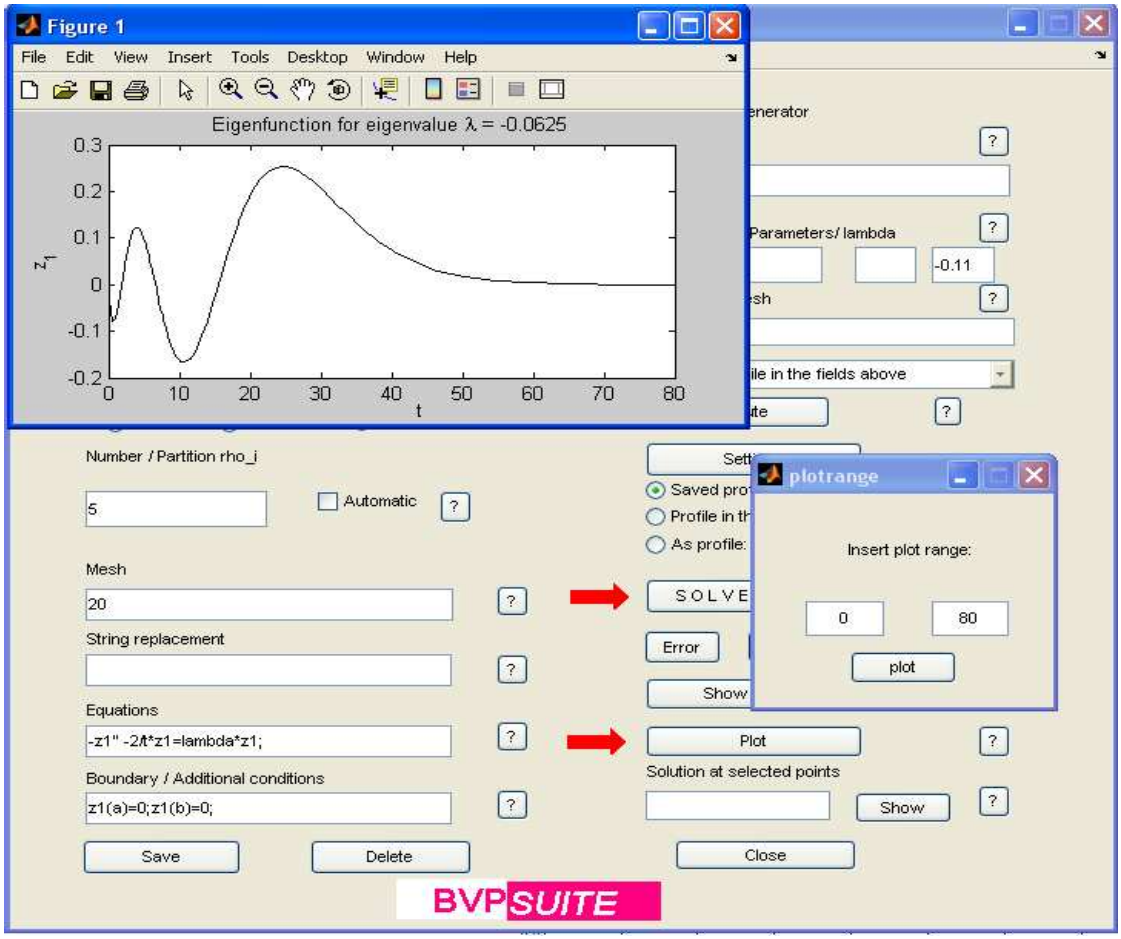

Figure 4.9: Graphical presentation for Example 4.1

Chapter 5

Matrix Methods

A *matrix method* represents an alternative solution method for the calculation of pairs of eigenvalues and eigenfunctions if several of these are sought. The basic idea is to approximate the derivatives with finite differences.
As it is explained below, application of *matrix methods* to eigenvalue problems in ODEs leads either to algebraic eigenvalue problems of the form

$$Ax = \lambda x, \tag{5.1}$$

or to generalized algebraic eigenvalue problems of the form

$$Ax = \lambda Bx, \tag{5.2}$$

where A and B are $(N \cdot n) \times (N \cdot n)$-matrices and x is a vector of length $N \cdot n$. The value of N is determined by the grid Δ and n denotes the dimension of the given system of differential equations. Therefore the eigenvalue problem for differential equations can be considered as the limit $N \to \infty$ of the finite-dimensional problems.

5.1 The Algebraic Eigenvalue Problem

An *eigenvalue* $\lambda \in \mathbb{C}$ of a square matrix $A = (a_{jk}) \in \mathbb{K}^{n \times n}$ ($\mathbb{K} = \mathbb{C}$ or $\mathbb{K} = \mathbb{R}$) is a scalar such that the equation

$$Ax = \lambda x, \tag{5.3}$$

has a nontrivial solution vector $x \in \mathbb{C}$ denoted as the corresponding *eigenvector*.
Given a pair (A, B), where A and B are square $n \times n$ matrices, the generalized (nonsymmetric) eigenvalue problem is to find the eigenvalues λ and corresponding eigenvectors $x \neq 0$ such that

$$Ax = \lambda Bx. \tag{5.4}$$

If the determinant of $A - \lambda B$ is identically zero for all values of λ, the eigenvalue problem is called singular, otherwise we call it regular. If B is singular, then there are p finite

generalized eigenvalues and $n-p$ infinite eigenvalues, where p is the degree of the determinant of $A-\lambda B$. However, if B is nonsingular then the eigenvalues of (5.4) correspond to those of the matrix $B^{-1}A$.
In our implementation, the eigenvalues and eigenfunctions are calculated via MATLAB's built-in function *eig(A,B)*, which solves the generalized eigenvalue problem $Ax = \lambda Bx$ for square matrices A and B. Since in general B is not symmetric, the MATLAB routine *eigs*(A, B, k), which returns the k largest or smallest generalized eigenvalues for symmetric problems, cannot be used. This leads to a high computational effort as the step size h gets smaller because for each h the whole algebraic spectrum $\sigma(A)$ is calculated and the problem size increases as $h \to 0$.

5.2 A Matrix Method for a Singular First Order Differential EVP

In [13] a thorough analysis is given for singular first order EVPs of the form

$$z'(t) - \left(\frac{M}{t} + A_0(t)\right) z(t) = \lambda G(t) z(t), \quad 0 < t \leq 1, \tag{5.5}$$

$$\tilde{B}_0 z(0) + \tilde{B}_1 z(1) = 0, \tag{5.6}$$

where $z(t) \in C[0,1] \cap C^1(0,1]$ is an n-vector, $A_0(t)$, $G(t) \in C[0,1]$ are $n \times n$ matrices, M, $\tilde{B}_0$ and $\tilde{B}_1$ are constant $n \times n$ matrices.
This general formulation also includes the case when eigenvalue problems of higher order equations are transformed to a first order system. For such problems the matrix B in (5.4) is singular. According to the results in [13], B can be any square matrix. However, standard solution routines for the algebraic eigenvalue problem are commonly designed for the case where A is symmetric and B is symmetric and positive definite.
Defining the differential operator L by

$$(Lz)(t) = z'(t) - \left(\frac{M}{t} + A_0(t)\right) z(t), \tag{5.7}$$

we can rewrite (5.5) in the form,

$$(L - \lambda G) z = 0, \quad z \in D(L), \tag{5.8}$$

where $G : C[0,1] \to C[0,1]$ is the bounded linear multiplication operator defined by $(Gz)(t) = G(t)z(t)$.
The numerical scheme proposed in [22] and [13] to approximate (5.5)–(5.6) is the so-called box scheme which is sometimes also denoted as implicit midpoint rule.
By defining $t_{j+1/2} := \frac{t_j + t_{j+1}}{2}$ the approximation scheme for the singular first order problem is given by

$$\begin{aligned} &\frac{z_{j+1} - z_j}{h} - \frac{1}{2}\left(\frac{M}{t_{j+1/2}} + A_0(t_{j+1/2})\right)(z_{j+1} + z_j) = \lambda \frac{1}{2} G(t_{j+1/2})(z_{j+1} + z_j), \\ &j = 0, \ldots, N-1, \\ &\tilde{B}_0 z_0 + \tilde{B}_1 z_N = 0, \quad Q z_0 = 0. \end{aligned} \tag{5.9}$$

Since this numerical scheme uses no evaluation at $t_0 = 0$, it can be applied straight forwardly to singular problems. If the problem has a singularity of the first kind, we assume that the only possible eigenvalues of M lying on the imaginary axis are zero.
The projection matrix Q in (5.9) is defined as

$$Q := I - P, \tag{5.10}$$

where P denotes the projection onto the direct sum of the subspaces X_0 and X_+ of M. Here, X_0 denotes the eigenspace corresponding to the eigenvalue zero and X_+ is the invariant subspace which corresponds to the eigenvalues with positive real parts.
The condition $Qz_0 = 0$ has to be satisfied for a well-posed problem.
In detailed matrix form the box scheme can be written as

$$\begin{pmatrix} D_0 & R_0 & & & & \\ & D_1 & R_1 & & & \\ & & \ddots & \ddots & & \\ & & & & D_{N-1} & R_{N-1} \\ B_0 & & & & & B_1 \end{pmatrix} \begin{pmatrix} z_0 \\ z_1 \\ \vdots \\ z_N \end{pmatrix} =$$

$$\lambda \begin{pmatrix} G_0 & G_0 & & & & \\ & G_1 & G_1 & & & \\ & & \ddots & \ddots & & \\ & & & & G_{N-1} & G_{N-1} \\ 0 & & & & & 0 \end{pmatrix} \begin{pmatrix} z_0 \\ z_1 \\ \vdots \\ z_N \end{pmatrix}, \tag{5.11}$$

where D_j, G_j and R_j are $n \times n$ matrices given by

$$D_j = -h_j^{-1} I - \frac{1}{2}\left(\frac{M}{t_{j+1/2}} + A_0(t_{j+1/2})\right), \tag{5.12}$$

$$R_j = h_j^{-1} I - \frac{1}{2}\left(\frac{M}{t_{j+1/2}} + A_0(t_{j+1/2})\right), \tag{5.13}$$

$$G_j = \frac{1}{2} G(t_{j+1/2}), \quad j = 0, \ldots, N-1. \tag{5.14}$$

Here I denotes the identity matrix of dimension n. B_0 and B_1 correspond to the boundary conditions in (5.9) including condition that are equivalent to the linear independent rows of $Qz_0 = 0$.

Since we approximate the solution (5.5)–(5.6) by solving the numerical scheme (5.9) for a small step size h_i, our aim is to estimate the approximation quality. Therefore, we are interested in how fast the computed value $\lambda_{n,h}$ converges to the exact eigenvalue λ_n^{exact} as h tends to 0. In [13] the following convergence result is given:

Theorem 5.1. *Assume that if M has eigenvalues with positive real part σ_0, then $\sigma_0 > 2$. Then if A_0, $G \in C^2_{n\times n}[0,1]$,*

$$\max_{1\le k\le \tau} |\lambda_n^{exact} - \lambda_{n,h}^k| \le C \cdot (h^2 |\ln(h)|^{d-1})^{\frac{1}{\kappa}},$$

where τ is the dimension of the eigenspace corresponding to λ_n^{exact}, d is the dimension of the largest Jordan box associated with the eigenvalue 0 of M, κ is the smallest integer l for which the nullspaces $(I - \lambda_n^{exact} L^{-1}G)^l$ and $(I - \lambda_n^{exact} L^{-1}G)^{l+1}$ coincide and $\lambda_{n,h}^k$ are the numerical approximations of λ_n^{exact}.

Proof. For the proof see Theorem 5.1 in [13]. □

The assumption on σ_0 is associated with the properties of the fundamental matrix for the solutions of singular boundary value problems, cf. [13]. For $0 < \sigma_0 \le 2$, corresponding results can be stated, where the order of convergence is reduced.

Identification of the Eigenfunctions

One big advantage of matrix methods is that by solving the algebraic generalized eigenvalue problem, approximations for several eigenvalues as well as associated eigenfunctions are obtained simultaneously. Our aim is to reconstruct the approximate eigenfunctions from the eigenvectors of the algebraic eigenvalue problem. We are particularly interested in the realization of this reconstruction for problems that comprise systems of equations and are additionally posed on semi-infinite intervals. In Chapter 3 we described that in our approach for problems that are posed on a semi-infinite interval, the interval is split and the dimension of the system is doubled.
Since the code can also cope with higher order problems, such problems first have to be transformed to a first order system and subsequently the number of equations is doubled by transforming the system to the finite interval $[0,1]$. If the original problem consists of a single equation of order n, then after reducing the order and transforming to the finite interval, the differential eigenvalue problem that is actually solved numerically consists of $2n$ first order equations. Subsequently, these $2n$ equations are approximated by the box scheme which leads to a high dimensional algebraic eigenvalue problem.
For example, for a second order equation the solution components are,

$$z_1(\tau) := z(t), \qquad \text{for} \quad \tau = t \in [0,1], \tag{5.15}$$
$$z_2(\tau) := z'(t), \qquad \text{for} \quad \tau = t \in [0,1], \tag{5.16}$$
$$z_3(\tau) := z(1/t), \qquad \text{for} \quad \frac{1}{\tau} = t \in [1,\infty), \tag{5.17}$$
$$z_4(\tau) := z'(1/t), \qquad \text{for} \quad \frac{1}{\tau} = t \in [1,\infty). \tag{5.18}$$

We point out that there is some freedom of choice in the transformation to the first order. Alternatively it is also possible to transform to first order by using the Euler

transformation $(z_1(t), z_2(t)) = (z(t), tz'(t))$.
Specifically, if

$$z_h = (z_0, z_1, \ldots, z_N) \tag{5.19}$$

denotes the eigenvector corresponding to a specific approximate eigenvalue $\lambda_{n,1/N}$, then

$$z_{1_h} = (z_0, z_4, z_8, \ldots, z_{N-4}), \tag{5.20}$$
$$z_{2_h} = (z_2, z_6, z_{10}, \ldots, z_{N-3}) \tag{5.21}$$

are the two approximating solution components constituting the eigenfunction z_h on $[0, L]$, L large.

Numerical Examples

To illustrate the technique we consider the problem

Example 5.1.

$$z'(t) - \frac{1}{t}\begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix} z(t) = \lambda \begin{pmatrix} 0 & 0 \\ t & 0 \end{pmatrix} z(t),\ t \in (0, 1], \tag{5.22}$$

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} z(0) + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} z(1) = 0. \tag{5.23}$$

Since this problem is singular we have to calculate the eigenvalues of M and determine the projection matrix Q in order to check the well-posedness of the problem. The eigenvalues are $\mu_1 = 0$ and $\mu_2 = -1$. Therefore, $z_2(0) = 0$ has to be satisfied for Example 5.1 to be well-posed. In the tables below we used λ_n^{ref} from [2]. By λ_{n,h_i} we denote the numerical approximation for the eigenvalue λ_n computed with step size h_i.

i	$\lvert\lambda_{1,h_i} - \lambda_1^{ref}\rvert$	$conv_\lambda$
1	5.4863e−04	2.0325
2	1.3410e−04	2.0165
3	3.3144e−05	2.0082
4	8.2392e−06	—

Table 5.1: Example 5.1: Estimated order of convergence for λ_1^{ref}.

Subsequently, we consider the first order formulation of the decoupled one-particle stationary Schrödinger equation transformed to the interval $[0, 1]$, cf. Chapter 4. To obtain a first order formulation we used Euler's transformation.

i	$\lvert\lambda_{2,h_i} - \lambda_2^{ref}\rvert$	$conv_\lambda$
1	1.7120e−02	2.0171
2	4.2298e−03	2.0084
3	1.0513e−03	2.0042
4	2.6206e−04	—

Table 5.2: Example 5.1: Estimated order of convergence for λ_2^{ref}.

i	$\lvert\lambda_{3,h_i} - \lambda_3^{ref}\rvert$	$conv_\lambda$
1	1.0182e−01	2.0153
2	2.5186e−02	2.0074
3	6.2644e−03	2.0036
4	1.5622e−03	—

Table 5.3: Example 5.1: Estimated order of convergence for λ_3^{ref}.

Example 5.2.

$$z'(r) - \frac{1}{r^2}\begin{pmatrix} 0 & 0 & r & 0 \\ 0 & 0 & 0 & r \\ rl(l+1) - r^2\gamma & 0 & r & 0 \\ 0 & rl(l+1) - \gamma & 0 & -r \end{pmatrix} z(r) = \lambda \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -r & 0 & 0 & 0 \\ 0 & -\frac{1}{r^3} & 0 & 0 \end{pmatrix} z(r), \tag{5.24}$$

with the boundary conditions

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} z(0) + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \end{pmatrix} z(1) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{5.25}$$

To analyze the convergence behaviour of the given problem, we calculated the empirical orders of convergence for the first few eigenvalues. The calculations were started with $h_1 = 1/50$. Since $h_1 = 1/50$ only gives the lowest 3 eigenvalues with satisfactory accuracy, the empirical convergence orders for those are presented in Tables 5.5 − 5.7. Furthermore we computed the first 8 eigenvalues for $\gamma = 2$ and $l = 3$ and present absolute and relative errors. By applying the numerical scheme described above and setting $h = 1/400$ we obtain the numerical results given in Table 5.4.

n	λ_n^{exact}	λ_n	abs. error	rel. error
4	$-\frac{1}{16}$	-6.24823563e−02	1.76436805e−05	2.82298887e−04
5	$-\frac{1}{25}$	-3.99078287e−02	9.21712695e−05	2.30428173e−03
6	$-\frac{1}{36}$	-2.75055075e−02	2.72270269e−04	9.80172967e−03
7	$-\frac{1}{49}$	-1.97899027e−02	6.18260564e−04	3.02947676e−02
8	$-\frac{1}{64}$	-1.43720049e−02	1.25299512e−03	8.01916880e−02
9	$-\frac{1}{81}$	-1.00214370e−02	2.32424199e−03	1.88263601e−01
10	$-\frac{1}{100}$	-6.76435637e−03	3.23564363e−03	3.23564363e−00
11	$-\frac{1}{121}$	-2.49806416e−03	5.76639865e−03	6.97734237e−00

Table 5.4: Example 5.2: Eigenvalues of the hydrogen atom with $\gamma = 2$ and $l = 3$.

i	$\lvert\lambda_{4,h_i} - \lambda_{4,h_{i+1}}\rvert$	$conv_\lambda$
1	1.221113e−03	2.383187
2	2.340697e−04	2.121570
3	5.378845e−05	2.021840
4	1.324508e−05	—

Table 5.5: Example 5.2: Estimated order of convergence for $\lambda_4^{exact} = \frac{1}{16}$.

i	$\lvert\lambda_{5,h_i} - \lambda_{5,h_{i+1}}\rvert$	$conv_\lambda$
1	9.395408e−03	2.537248
2	1.618557e−03	2.490702
3	2.879731e−04	2.055563
4	6.927329e−05	—

Table 5.6: Example 5.2: Estimated order of convergence for $\lambda_5^{exact} = \frac{1}{25}$.

i	$\lvert\lambda_{6,h_i} - \lambda_{6,h_{i+1}}\rvert$	$conv_\lambda$
1	6.640375e−03	0.906395
2	3.542750e−03	2.003884
3	8.833061e−04	2.106550
4	2.051051e−04	—

Table 5.7: Example 5.2: Estimated order of convergence for $\lambda_6^{exact} = \frac{1}{36}$.

5.3 A Matrix Method for a Singular Second Order Differential EVP

Since many well-studied and classical eigenvalue problems arise in the context of Sturm-Liouville problems, we are particularly interested in second order problems. We also observed that transformation of a second order problem to first order and subsequent numerical solution leads to high computational effort. This is due to the fact that the resulting algebraic problem which has to be solved becomes very large.
Hence, we also discuss the following problem:

$$-y''(t) + A_1(t)y'(t) + A_0(t)y(t) = \lambda G(t)y(t), \quad 0 < t \leq 1, \tag{5.26}$$
$$B_0(y(0), y'(0))^T + B_1(y(1), y'(1))^T = 0, \tag{5.27}$$

where $y(t) \in C[0,1] \cap C^2(0,1]$ is an n-vector.
We say the problem has a singularity of the first kind if $A_1(t) = \frac{M_1(t)}{t}$ and $A_0(t) = \frac{M_0(t)}{t^2}$ with $M_i \in C^1[0,1]$ and $G \in C[0,1]$. By Taylor expansion,

$$y(t_j \pm h) = y(t_j) \pm hy'(t_j) + \frac{h^2}{2}y''(t_j) \pm \frac{h^3}{6}y^{(3)}(t_j) + O(h^4), \tag{5.28}$$

we can express the first and second derivatives as

$$y'(t_j) = \frac{y(t_{j+1}) - y(t_{j-1})}{2h} + O(h^2) \tag{5.29}$$
$$y''(t_j) = \frac{y(t_{j+1}) - 2y(t_j) + y(t_{j-1})}{h^2} + O(h^3). \tag{5.30}$$

Using (5.29) − (5.30) we can approximate (5.26) in the following way:

$$-\frac{y(t_{i+1}) - 2y(t_i) + y(t_{i-1})}{h^2} + A_1(t_i)\frac{y(t_{i+1}) - y(t_{i-1})}{2h} + A_0(t_i)y(t_i) = \lambda G(t_i), \quad 1 \leq i \leq N-1. \tag{5.31}$$

In the second order case one also has to approximate the first derivative at the endpoints. Hence, we carry out a Taylor expansion about the endpoints t_0 and t_N of the exact solution $y(t)$ which is assumed to exist:

$$y(t_1) = y(t_0) + hy'(t_0) + \tfrac{h^2}{2}y''(t_0) + O(h^3), \tag{5.32}$$
$$y(t_2) = y(t_0) + 2hy'(t_0) + \tfrac{4h^2}{2}y''(t_0) + O(h^3). \tag{5.33}$$

Multiplying (5.32) by 4 and subtracting (5.33) we obtain an approximation for $y'(t_0)$,

$$y'(t_0) = \frac{-3y(t_0) + 4y(t_1) - y(t_2)}{2h} + O(h^2). \tag{5.34}$$

The same argument applies if we expand about t_N and therefore we obtain:

$$y'(t_N) = \frac{y(t_{N-2}) - 4y(t_{N-1}) + 3y(t_N)}{2h} + O(h^2). \tag{5.35}$$

To determine eigenvalues and eigenfunctions we need $N+1$ linear equations for the $N+1$ unknowns. In detailed matrix form the scheme described above reads,

$$\begin{pmatrix} D_1 & E_1 & F_1 & & & D_{N+1} & E_{N+1} & F_{N+1} \\ D_2 & E_2 & F_2 & & & & & \\ & D_3 & E_3 & F_3 & & & & \\ & & & \ddots & \ddots & & & \\ & & & & & D_N & E_N & F_N \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{pmatrix} =$$

$$\lambda \begin{pmatrix} 0 & 0 & 0 & & & 0 & 0 & 0 \\ G_2 & G_2 & & & & & & \\ & G_3 & G_3 & & & & & \\ & & \ddots & \ddots & & & & \\ & & & & & & G_N & G_N \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{pmatrix}, \tag{5.36}$$

where D_j, E_j and F_j, $j = 2, \ldots, N$ are $n \times n$ matrices and $B_0 = (B_{01}|B_{02})$ and $B_N = (B_{N1}|B_{N2})$ such that D_j, E_j and F_j, $j \in \{1, N+1\}$ are $n \times m$ matrices with

$$D_1 = B_{01} - \frac{3}{2h} B_{02}, \tag{5.37}$$

$$E_1 = \frac{2}{h} B_{02}, \tag{5.38}$$

$$F_1 = -\frac{1}{2h} B_{02}, \tag{5.39}$$

$$D_j = -\frac{1}{h^2} I - \frac{1}{2h} A_1(t_j), \tag{5.40}$$

$$E_j = \frac{2}{h^2} I + A_0(t_j), \tag{5.41}$$

$$F_j = -\frac{1}{h^2} I + \frac{1}{2h} A_1(t_j), \tag{5.42}$$

$$G_j = G(t_j), \quad j = 2, \ldots, N, \tag{5.43}$$

$$D_{N+1} = B_{N1} - \frac{2}{h} B_{N2}, \tag{5.44}$$

$$E_{N+1} = \frac{1}{2h} B_{N2}, \tag{5.45}$$

$$F_{N+1} = \frac{3}{2h} B_{N2}. \tag{5.46}$$

$$\tag{5.47}$$

Since for the second order singular eigenvalue problem in its general form no error analysis could be found, we cannot state any results here. However, our examples below also comprise singular problems and show promising values for the estimated errors and convergence orders. However, for the special class of Sturm-Liouville problems, convergence results can be stated. It holds that the approximation quality decreases rapidly as the index of the eigenvalue grows.

If we denote the lowest eigenvalue by λ_1 and so forth, according to [17] the following result holds,

Theorem 5.2. *For the Sturm-Liouville problem*

$$-(py')' + qy = \lambda y, \tag{5.48}$$
$$y(0) = y(\pi) = 0, \tag{5.49}$$

there exists a constant C such that the kth eigenvalue λ_k and the approximating eigenvalue λ_{k,h_i} satisfy

$$|\lambda_k^{exact} - \lambda_{k,h_i}| \leq Ch^2k^4, \tag{5.50}$$

provided that $p^{(3)}$ and q'' are continuous.

To understand the statement above, we need the Lemma below. In the following we denote the eigenfunction corresponding to the k-th eigenvalue by $y^{[k]}(t)$.

Lemma 1. *The k-th eigenfunction of the Sturm-Liouville problem*

$$-(py')' + qy = \lambda y, \tag{5.51}$$
$$y(0) = y(\pi) = 0, \tag{5.52}$$

can be asymptotically represented by

$$y^{[k]}(t) = c_k \frac{\sin\left(k\frac{\pi}{l}\int_0^t \sqrt{\frac{1}{p}}d\tau\right)}{\sqrt[4]{p}} + O\left(\frac{1}{k}\right), \tag{5.53}$$

where

$$l = \int_0^\pi \sqrt{\frac{1}{p}}dt \tag{5.54}$$

and c_k is a normalizing factor in the L^2-norm.

Proof. For the proof see [10, pp. 336-339]. □

For the proof of Theorem 5.2 we need the notion of *local truncation error*,

$$\tau_h[y^{[k]}] = (\tau_h[y^{[k]}(t_1)], \ldots, \tau_h[y^{[k]}(t_N)]), \tag{5.55}$$

which is defined as the residual when the exact solution $y^{[k]}(t)$ is substituted into the numerical scheme.

Proof. The numerical scheme

$$-\left(p_{i+1/2}(y_{i+1} - y_i)/h - p_{i-1/2}(y_i - y_{i-1})/h\right)/h + q_i y_i = \lambda y_i, \tag{5.56}$$

yields an algebraic EVP with eigenvalues $\lambda_{k,h}$ satisfying, according to Keller's convergence theorem (cf. [17, Theorem 1]),

$$|\lambda_k^{exact} - \lambda_{k,h}| \leq \frac{||\tau_h[y^{[k]}]||_2}{||y^{[k]}||_2}, \tag{5.57}$$

where $\tau_h[y^{[k]}]$ denotes the local truncation error.
Hence, we have to find a bound for $\tau_h[y_k(t_i)]$.
To estimate $\tau_h[y^{[k]}(t_i)]$, Taylor's theorem is used.

$$y^{[k]}(t_i \pm h) = y^{[k]}(t_i) \pm h y^{[k]\prime}(t_i) + \frac{h^2}{2} y^{[k]\prime\prime}(t_i) \pm \frac{h^3}{6} y^{[k]\,(3)}(t_i) + \frac{h^4}{24} y^{[k]\,(4)}(t_i) + O(h^5), \tag{5.58}$$

$$p(t_i \pm \frac{h}{2}) = p(t_i) \pm \frac{h}{2} p'(t_i) + \frac{h^2}{8} p''(t_i) \pm \frac{h^3}{48} p^{(3)}(t_i) + O(h^4). \tag{5.59}$$

By inserting (5.58)–(5.59) into (5.56) and collecting like powers of h, the leading term of the local truncation error has the form

$$\tau_h[y^{[k]}(t_i)] = -\left(2p(t_i) y^{[k]\,(4)}(t_i) + 4p'(t_i) y^{[k]\,(3)}(t_i) + 3p''(t_i) y^{[k]\prime\prime}(t_i) + p^{(3)}(t_i) y^{[k]}(t_i)\right) \frac{h^2}{24}. \tag{5.60}$$

According to (5.53), the fourth derivative of $y^{[k]}$ in the error expansion gives rise to the factor k^4. Therefore it follows that

$$|\lambda_k^{exact} - \lambda_{k,h}| = O(h^2 k^4). \tag{5.61}$$

□

For a detailed proof cf. [17], [25] and [27].
For problems of the more general form $-y'' + qy = \lambda g y$ the same estimate holds with the additional factor $||B^{-1}||$, with B from (5.2), on the right-hand side of (5.50), provided that also g'' is continuous. Hence, we can conclude that B must be invertible and therefore this result is not applicable in the more general setting. Due to (5.61), the error grows rapidly with increasing k. These observations are also confirmed in our setting.

Numerical Examples

Calculations with the matrix method module implemented in MATLAB backed up the results stated above. See, for example, Table 5.8, which shows the dependence of the error on the eigenvalue index for the following regular example,

Example 5.3.

$$-y''(t) + e^t y(t) = \lambda y(t), \quad t \in [0, \pi], \tag{5.62}$$

$$y(0) = y(\pi) = 0, \tag{5.63}$$

for a fixed $h = 1/80$. The reference solution λ_k^{ref} was calculated using $h = 1/1500$.
This problem often occurs in the literature for comparisons of different numerical methods for the solution of Sturm-Liouville problems. To determine p in $O(k^p h^2)$ empirically we made the ansatz

$$|\lambda_k^{exact} - \lambda_{k,h_i}| \approx Ch^2 k^p, \tag{5.64}$$
$$|\lambda_{k+j}^{exact} - \lambda_{k+j,h_i}| \approx Ch^2 (k+j)^p. \tag{5.65}$$
$$\tag{5.66}$$

Thus p can be approximated by

$$p \approx \frac{\ln \frac{|\lambda_k^{exact} - \lambda_{k,h}|}{|\lambda_{k+j}^{exact} - \lambda_{k+j,h}|}}{\ln \frac{k}{k+j}}. \tag{5.67}$$

For problems with a singular matrix B, as for the hydrogen atom, this behavior could

Index	$\lvert\lambda_{k,h} - \lambda_k^{ref}\rvert$	p
1	0.0007	2.5825
2	0.0043	–
5	0.0827	3.8980
6	0.1683	–
10	1.2773	3.9824
11	1.8670	–
15	6.4152	3.9739
16	8.2907	–

Table 5.8: Example 5.3: Empirical value of p with a fixed step size $h = 1/80$.

not be observed.
Although we are not aware of any theoretical background for such problems, based on numerical experiments we conjecture that also in this case the error grows with the index of the eigenvalue.
To illustrate this phenomenon we consider the equation for the hydrogen atom, cf. Example 5.4 in Chapter 4.

Example 5.4.

$$-u''(r) + \left(\frac{l(l+1)}{r^2} - \frac{\gamma}{r}\right) u(r) = \lambda u(r), \quad r \in (0, \infty), \tag{5.68}$$
$$u(0) = 0, \quad u(\infty) = 0. \tag{5.69}$$

In the following we set $\gamma = 2$ and $l = 3$. With a step size of $h = 1/1000$ we obtained the results in Table 5.9, where n denotes the principal quantum number as introduced in Chapter 2. For the angular momentum quantum number $l = 2$ the lowest eigenvalue exists for $n = 4$.

n	λ_n^{exact}	λ_n	abs. error	rel. error
4	$-\frac{1}{16}$	-6.25018751e−02	1.8751021e8−06	3.00016348e−05
5	$-\frac{1}{25}$	-4.00086284e−02	8.62841398e−06	2.15710350e−04
6	$-\frac{1}{36}$	-2.78015623e−02	2.37845506e−05	8.56243823e−04
7	$-\frac{1}{49}$	-2.04597484e−02	5.15851809e−05	2.52767386e−03
8	$-\frac{1}{64}$	-1.57223228e−02	9.73227865e−05	6.22865834e−03
9	$-\frac{1}{81}$	-1.25135678e−02	1.67888738e−04	1.35989878e−02
10	$-\frac{1}{100}$	-1.02736283e−03	2.73628306e−04	2.73628306e−02
11	$-\frac{1}{121}$	-8.69513057e−03	4.30667756e−04	5.21107984e−02

Table 5.9: Example 5.4: Eigenvalues of the hydrogen atom with $\gamma = 2$ and $l = 3$.

n	$\lvert\lambda_{k,h} - \lambda_k^{exact}\rvert$	p
4	1.87510218e−06	2.2021
5	8.62841398e−06	2.5008
6	2.37845506e−05	2.6912
7	5.15851809e−05	2.8448
8	9.73227865e−05	2.9907
9	1.67888738e−04	3.1688
10	2.73628306e−04	3.3967
11	4.30667756e−04	–

Table 5.10: Example 5.4: Empirical value of p with a fixed step size $h = 1/1000$ of the hydrogen equation.

Table 5.10 gives empirical values for p similar to those in Table 5.8 which do not give a clear picture on the actual dependence.

Tables 5.11 – 5.13 show the estimated order of convergence for the first 3 eigenvalues of the hydrogen equation with $l = 2$.

i	$\lvert\lambda_{4,h_i} - \lambda_4^{exact}\rvert$	$conv_\lambda$
1	7.5708e−04	2.0180
2	1.8692e−04	1.9935
3	4.6940e−05	2.0015
4	1.1723e−05	2.0004
5	2.9299e−06	—

Table 5.11: Example 5.4: Estimated order of convergence for λ_4^{exact}. The computations were started with $h_1 = 1/50$.

Since we are not aware of theoretical results on the quality of the approximation scheme for singular second order problems, we want to point out in Table 5.14 that

i	$\lvert\lambda_{5,h_i} - \lambda_5^{exact}\rvert$	$conv_\lambda$
1	2.1916e−03	1.2730
2	9.0685e−04	2.0577
3	2.1782e−04	2.0110
4	5.4041e−05	2.0027
5	1.3485e−05	—

Table 5.12: Example 5.4: Estimated order of convergence for λ_5^{exact}. The computations were started with $h_1 = 1/50$.

i	$\lvert\lambda_{6,h_i} - \lambda_6^{exact}\rvert$	$conv_\lambda$
1	6.2692e−03	0.8224
2	3.5453e−03	2.5795
3	5.9311e−04	1.9850
4	1.4983e−04	2.0102
5	3.7194e−05	—

Table 5.13: Example 5.4: Estimated order of convergence for λ_6^{exact}. The computations were started with $h_1 = 1/50$.

accurate results for important examples as the hydrogen equation can be obtained under significantly less computational cost than for the first order formulation.

l	n	abs. err	rel. err	dim (1st order)	dim (2nd order)
1	4	$3.3 \cdot 10^{-5}$	$5.0 \cdot 10^{-4}$	2804	1062
2	4	$3.0 \cdot 10^{-5}$	$4.9 \cdot 10^{-4}$	2204	862
3	4	$3.1 \cdot 10^{-5}$	$5.0 \cdot 10^{-4}$	1204	492

Table 5.14: Computational cost comparison between Example 5.4 and 5.2: We applied the matrix method to the first and second order formulations of the hydrogen equation for various values of the quantum numbers l. In the last two columns the dimension of the square matrix B of the resulting algebraic eigenvalue problem is given.

Typically, physicists are particularly interested in the first few eigenvalues, which correspond to the lowest bound energies. Table 5.10 shows that for these the accuracy deteriorates with order > 2 as the index of the eigenvalue increases.

To deal with this undesired behavior, there exist correction techniques for Sturm-Liouville problems in normal form with the result that the eigenvalue approximation deteriorates only linearly with k, cf. [27]. Application of this correction technique could also be advantageous for problems with a singular matrix B.

In the following we want to point out and demonstrate that in contrast to other standard approaches for the solution of eigenvalue problems in ODEs, the matrix method immediately incorporates the numerical treatment of vector eigenvalue problems. The problem formulations for the first order problem (5.5)–(5.6) as well as for the second order problem (5.26)–(5.27) in Chapter 5 already include the n-dimensional case.
We also want to mention that in principle we could as well apply the collocation method described in Chapter 4. However, this requires that all eigenvalues have multiplicity 1.

To demonstrate the success of the approximation by finite differences, we consider three test examples of higher dimension also presented in [23]. We solved Example 5.5, which is posed as a regular problem on $[0.1, 1]$ in [23], also on the domain $[0, 1]$ so that a singularity arises at the left endpoint.

Example 5.5. *We consider the example*

$$y''(t) = \begin{pmatrix} \cos(t) + \frac{1}{t} - \lambda & \frac{\cos(t)}{2} & \frac{\cos(t)}{3} & \frac{\cos(t)}{4} \\ \frac{\cos(t)}{2} & \frac{\cos(t)}{2} + \frac{1}{t^2} - \lambda & \frac{\cos(t)}{3} & \frac{\cos(t)}{4} \\ \frac{\cos(t)}{3} & \frac{\cos(t)}{3} & \frac{\cos(t)}{3} + \frac{1}{t^3} - \lambda & \frac{\cos(t)}{4} \\ \frac{\cos(t)}{4} & \frac{\cos(t)}{4} & \frac{\cos(t)}{4} & \frac{\cos(t)}{4} + \frac{1}{t^4} - \lambda \end{pmatrix} y(t), \tag{5.70}$$

with the boundary conditions

$$y(0.1) = y(1) = 0 \quad \textit{and alternatively} \quad y(0) = y(1) = 0. \tag{5.71}$$

In the computations we set $h = 1/200$. The numerical results are given in Tables 5.15 and 5.16. Since we are not aware if this problem was originally posed as either regular or singular, we just want to point out that the eigenvalues change significantly when the problem is solved on $[0, 1]$.
Note the huge difference in the $5th$ eigenvalue.

Index	$\lambda_{k,h}$
1	14.9415
2	17.0432
3	21.3799
4	26.9199
5	51.8217
6	55.7994

Table 5.15: Example 5.5: First 6 eigenvalues solved on $[0.1, 1]$.

As a second test example we consider

Example 5.6.

$$y''(t) = \begin{pmatrix} \frac{3t}{2} - \lambda & -\frac{t}{2} \\ -\frac{t}{2} & \frac{3t}{2} - \lambda \end{pmatrix} y(t), \quad t \in [0, 1], \tag{5.72}$$

Index	$\lambda_{k,h}$
1	13.0509
2	16.3685
3	21.3537
4	26.9181
5	43.3868
6	52.0515

Table 5.16: Example 5.5: First 6 eigenvalues solved on $[0, 1]$.

with the boundary conditions

$$y(a) = y(b) = 0. \tag{5.73}$$

This example is of particular interest because the eigenvalues appear pairwise. Hence, we choose a very small step size, $h = 1/1600$, in order to obtain accurate results.The numerical results are given in Table 5.17.

Example 5.7 is particularly suitable to be solved by a matrix method approach, because

Index	$\lambda_{k,h}$
1	10.3812
2	10.8782
3	40.0284
4	40.5297
5	89.4378
6	89.9387

Table 5.17: Example 5.6: First 6 eigenvalues.

the eigenvalues have multiplicity 2. This property places both examples outside the scope of the collocation method considered in Chapter 4. The discretization step size is set to $h = 1/400$.

Example 5.7.

$$y''(t) = \begin{pmatrix} t^2 - \lambda & 0 \\ 0 & t^2 - \lambda \end{pmatrix} y(t), \quad t \in [0, 10], \tag{5.74}$$

with the separated boundary conditions

$$y(a) = 0 \quad y(b) = 0. \tag{5.75}$$

The analytical eigenvalues of Example 5.7 are $3, 7, 11, \ldots$.

Index	$\lambda_{k,h}$
1	2.9998
2	2.9998
3	6.9998
4	6.9998
5	10.9998
6	10.9998

Table 5.18: First 6 eigenvalues of Example 5.7.

5.4 Discussion

The matrix method yields an appropriate solution method if several pairs of eigenvalues and eigenfunctions are sought.
If the collocation method from Chapter 4 is used to obtain a new eigenvalue-eigenfunction pair, a new starting value and starting profile have to be chosen and a new calculation has to be performed. As mentioned there the selection of the starting value is even sensitive to changes in the number of mesh points.
An advantage of this matrix method is that it yields several approximate eigenvalues in one sweep. Moreover, no guess for the starting value has to be provided. Therefore hardly any information about the eigenvalues or eigenfunctions of the given analytical problem is required.
However, there are also some drawbacks of this technique. The given infinite-dimensional problem is replaced by a matrix problem of dimension roughly the number of mesh points N times the dimension of the system of ODEs. Solving this algebraic eigenvalue problem demands an enormous computational effort as N increases.
Another drawback is that the box scheme is only of second order. This means that accurate approximations can only be computed on relatively dense meshes. Acceleration techniques such as extrapolation could be applied to increase the order of convergence, cf. [22].
However the biggest disadvantage of these finite differences techniques is that the error of the eigenvalue approximation increases rapidly with the index of the eigenvalue as already mentioned in Section 5.3. These observations are our motivation to propose the scheme described below, which exploits the advantages of both schemes described so far.

5.5 Matrix Method – a Procedure for Obtaining Accurate Starting Profiles

We propose now a hybrid method based on both introduced solution approaches for the solution of singular eigenvalue problems in ordinary differential equations. Since the nonlinear collocation equations are solved by the Newton method, a starting profile is required in order to resolve the eigenpair of interest by the iterative solver used in the

computation. To obtain a good starting profile we considered the matrix method which was described in previous sections of this chapter.
We consider three important applications which are of great interest in the field of quantum mechanics.
All of these examples are posed on a semi-infinite interval. While physicists commonly truncate and adapt the right endpoint of the interval depending on the requested eigenvalue, we apply here the transformation to a finite interval as recapitulated in Chapter 3.
To obtain accurate results we first solve the eigenvalue problems with the matrix method and subsequently used the approximate eigenvalue and eigenfunction as initial guess for the collocation method implemented in `bvpsuite`.
As a first example we consider the Schrödinger equation for one particle with the radially symmetric Yukawa potential with parameter $\alpha > 0$. In the following we multiply the Schrödinger operator as formulated in Chapter 2 by $\frac{1}{2}$ which results in the so-called Hartree formulation so that we can compare our results to those reported in [11]. Furthermore, in the following it holds that n denotes the principal quantum with $n \in \mathbb{N}$.

Example 5.8.

$$-\frac{1}{2}u''(r) + \left(\frac{l(l+1)}{2r^2} - \frac{e^{-\alpha r}}{r}\right) u(r) = \lambda u(r), \quad 0 \leq r < \infty, \tag{5.76a}$$

$$u(0) = 0,\ u(\infty) = 0, \tag{5.76b}$$

where $l \in \mathbb{N}_0$, the parameter α is called screening parameter and as usual λ denotes the eigenvalue. Bound states exist only for certain values of α below a threshold α_c. The number of elements in the point spectrum varies with α. Note that for $\alpha = 0$, Example 5.8 corresponds to the hydrogen problem which has infinitely many eigenvalues and was already discussed in more detail in Chapter 2 and Chapter 4. We are interested in

- the values of α for which the number of eigenvalues changes and
- the value of α_c so that for $\alpha > \alpha_c$ no bound states exist. In [28] this value is given as $1.19061227 \pm 0.00000004$.

In the tables below n denotes the principle quantum number, *no.coll* is the number of collocation points used, *no.mesh* is the number of mesh points in the final mesh, $\lambda^{(0)}$ is the starting value for the eigenparameter obtained by the matrix method, $errest_\lambda$, $errest_\lambda^{rel}$ are the error estimates for the eigenvalue and $errest_u$ and $errest_u^{rel}$ are the error estimates for the eigenfunction all obtained by the package `bvpsuite`.
Table 5.19 shows the comparison of eigenvalues for the Yukawa potential with values cited in [28]. Table 5.20 compares our results with those in [11].

As a second example we consider again the one particle Hamiltonian but this time containing a Hulthén potential.

Example 5.9.

$$-\frac{1}{2}u''(r) + \left(\frac{l(l+1)}{2r^2} - \frac{\alpha e^{-\alpha r}}{1-e^{-\alpha r}}\right) u(r) = \lambda u(r), \quad 0 \le r < \infty, \tag{5.77a}$$

$$u(0) = 0,\ u(\infty) = 0. \tag{5.77b}$$

According to [7] for $l = 0$ analytical results for the eigenvalues are given by

$$\lambda_n^{exact} = -\frac{1}{2}\left(\frac{1}{n} - \frac{n\alpha}{2}\right)^2 \tag{5.78}$$

This relation shows how the eigenvalues compare to those of the Coulomb potential. Table 5.21 compares the computed eigenvalues for the Hulthén potential for several values of α with those presented in [28]. Tables 5.19 – 5.21 show that we are capable of producing accurate results for Examples 5.8 and 5.9. The numerical results for the eigenvalues correspond with those reported in [28] and [11] up to the 13th (or even 15th) decimal place. In contrast to the results reported in [28] and [11] we also present error estimates which make our results trustworthy.

l	α	n	N (MM)	$\lambda^{(0)}$ (MM)	no. coll.	no. mesh	`bvpsuite`	$errest_\lambda$	$errest_\lambda^{rel}$	$errest_u$	$errest_u^{rel}$	[28]
0	0.1	1	50	-0.4073	8	70	-0.40705803061326	$2.2 \cdot 10^{-19}$	$5.3 \cdot 10^{-19}$	$2.1 \cdot 10^{-17}$	$2.9 \cdot 10^{-17}$	-0.40705803061340
1	0.01	2	50	-0.1154	8	167	-0.11524522409056	$1.1 \cdot 10^{-19}$	$9.4 \cdot 10^{-19}$	$1.5 \cdot 10^{-18}$	$3.5 \cdot 10^{-18}$	-0.11524522409056
1	0.01	3	50	-0.0468	8	188	-0.04615310482916	$8.2 \cdot 10^{-20}$	$1.8 \cdot 10^{-18}$	$1.7 \cdot 10^{-18}$	$5.5 \cdot 10^{-18}$	-0.04615310482916
2	0.01	3	50	-0.0463	8	176	-0.04606145416066	$5.4 \cdot 10^{-20}$	$1.2 \cdot 10^{-18}$	$1.5 \cdot 10^{-18}$	$4.7 \cdot 10^{-18}$	-0.04606145416065
0	0.01	9	400	-0.00111	8	350	-0.00058524761250	$7.2 \cdot 10^{-21}$	$1.2 \cdot 10^{-17}$	$4.9 \cdot 10^{-18}$	$4.4 \cdot 10^{-17}$	-0.00058524761250
1	0.01	9	400	-0.00106	8	1034	-0.0005665076261	$2.5 \cdot 10^{-21}$	$4.5 \cdot 10^{-18}$	$2.4 \cdot 10^{-18}$	$2.2 \cdot 10^{-17}$	-0.00056650762617

Table 5.19: Example 5.8: Comparison of the eigenvalues of the Yukawa potential for several values of α reported in [28]. For the computations we prescribed $tol_a = 10^{-15}$ and $tol_r = 10^{-10}$.

l	α	n	N (MM)	$\lambda^{(0)}$ (MM)	no. coll.	no. mesh	`bvpsuite`	$errest_\lambda$	$errest_\lambda^{rel}$	$errest_u$	$errest_u^{rel}$	[11]
0	0.5	1	25	-0.1489	8	85	-0.14811702188982	$9.8 \cdot 10^{-19}$	$6.6 \cdot 10^{-18}$	$1.8 \cdot 10^{-17}$	$2.7 \cdot 10^{-17}$	-0.14811702
0	0.8	1	25	-0.0452	8	158	-0.04470430449730	$2.7 \cdot 10^{-19}$	$6.1 \cdot 10^{-18}$	$2.6 \cdot 10^{-18}$	$4.6 \cdot 10^{-18}$	-0.447043
0	1	1	25	-0.0106	8	265	-0.01028578998998	$5.3 \cdot 10^{-19}$	$5.2 \cdot 10^{-17}$	$9.6 \cdot 10^{-18}$	$2.2 \cdot 10^{-17}$	-0.0102858
0	1.1	1	25	-0.00264	8	257	-0.00228724423404	$2.1 \cdot 10^{-19}$	$9.1 \cdot 10^{-17}$	$5.9 \cdot 10^{-18}$	$1.8 \cdot 10^{-17}$	-0.00228724

Table 5.20: Example 5.8: Comparison of the eigenvalues of the Yukawa potential for several values of α reported in [11]. For the computations we prescribed $tol_a = 10^{-15}$ and $tol_r = 10^{-10}$.

l	α	n	N (MM)	$\lambda^{(0)}$ (MM)	no. coll.	no. mesh	`bvpsuite`	$errest_\lambda$	$errest_\lambda^{rel}$	$errest_u$	$errest_u^{rel}$	[11]
0	0.002	1	50	-0.4993	8	67	-0.49900049999985	$8.7 \cdot 10^{-18}$	$1.7 \cdot 10^{-17}$	$1.8 \cdot 10^{-17}$	$2.4 \cdot 10^{-17}$	-0.4900050000000
2	0.15	3	50	-0.00153	8	235	-0.00139659246573	$1.9 \cdot 10^{-20}$	$1.3 \cdot 10^{-17}$	$1.5 \cdot 10^{-18}$	$5.7 \cdot 10^{-18}$	-0.00139659246573
1	0.02	8	500	-0.0012	6	1441	-0.0009868327076	$2.6 \cdot 10^{-21}$	$2.6 \cdot 10^{-18}$	$1.5 \cdot 10^{-18}$	$1.2 \cdot 10^{-17}$	-0.0009868327076

Table 5.21: Example 5.9: Comparison of the eigenvalues of the Hulthén potential for several values of α reported in [11]. For the computations we prescribed $tol_a = 10^{-15}$ and $tol_r = 10^{-10}$.

α	n	N (MM)	$\lambda^{(0)}$ (MM)	no. coll.	no. mesh	`bvpsuite`	abs. err	rel. err
0.002	1	50	-0.4993	8	62	$-4.9900 \cdot 10^{-01}$	$1.5 \cdot 10^{-13}$	$2.9 \cdot 10^{-13}$
0.05	1	50	-0.4876	8	67	$-4.8758 \cdot 10^{-01}$	$1.4 \cdot 10^{-13}$	$3.0 \cdot 10^{-13}$
0.05	2	50	-0.1132	8	152	$-1.1281 \cdot 10^{-01}$	$2.8 \cdot 10^{-14}$	$2.5 \cdot 10^{-13}$
1.97	1	400	-0.00011293	8	371	$-1.1250 \cdot 10^{-04}$	$3.1 \cdot 10^{-15}$	$2.7 \cdot 10^{-11}$

Table 5.22: Example 5.9 with $l = 0$: Comparison of the computed eigenvalues for several values of α with the corresponding analytical results given in (5.78).

The third example models the so-called Stark effect, cf. [30]. This effect occurs when a hydrogen atom is placed in a static electric field. In our case the electric field is along the z direction. This effect can be modeled by two coupled radial Schrödinger equations resulting from transforming a PDE to a ODE system and its truncation.

Example 5.10.

$$\left[-\frac{1}{2}\frac{d^2}{dr^2} + \begin{pmatrix} -\frac{1}{r} & -\frac{\varepsilon}{\sqrt{3}} r f_b(r) \\ \frac{\varepsilon}{\sqrt{3}} r f_b(r) & -\frac{1}{r} + \frac{1}{r^2} \end{pmatrix}\right] u(r) = \lambda u(r),\ 0 \le r < \infty, \tag{5.79a}$$

$$u(0) = 0,\ u(\infty) = 0, \tag{5.79b}$$

where $f_b(r) = e^{-br^2}$.
For the calculations we set $\varepsilon = 10^{-4}, 10^{-5}$ and $b = 0, 10^{-10}$. We used the matrix method with $N = 200$ to obtain accurate starting profiles. To our knowledge this problem is usually not solved as a matrix valued eigenvalue problem. This distinguishes our approach from others. Furthermore we are capable of solving Example 5.10 with b set to 0. We compared our numerical results with those obtained from perturbation theory as can be seen in Tables 5.25–5.28. For bigger values of ε perturbation theory is not valid. Figures 5.5–5.7 show the approximations for the eigenfunctions of Example 5.10. Table 5.29 shows the dependence of the eigenvalues on the parameter b. The errors between the results for $b = 0$ and the corresponding values for $b > 0$ are presented. As b gets smaller the eigenvalues converge to those for $b = 0$. The starting values for the collocation method were obtained by the matrix method with $N = 200$ subintervals on $[0, 1]$.

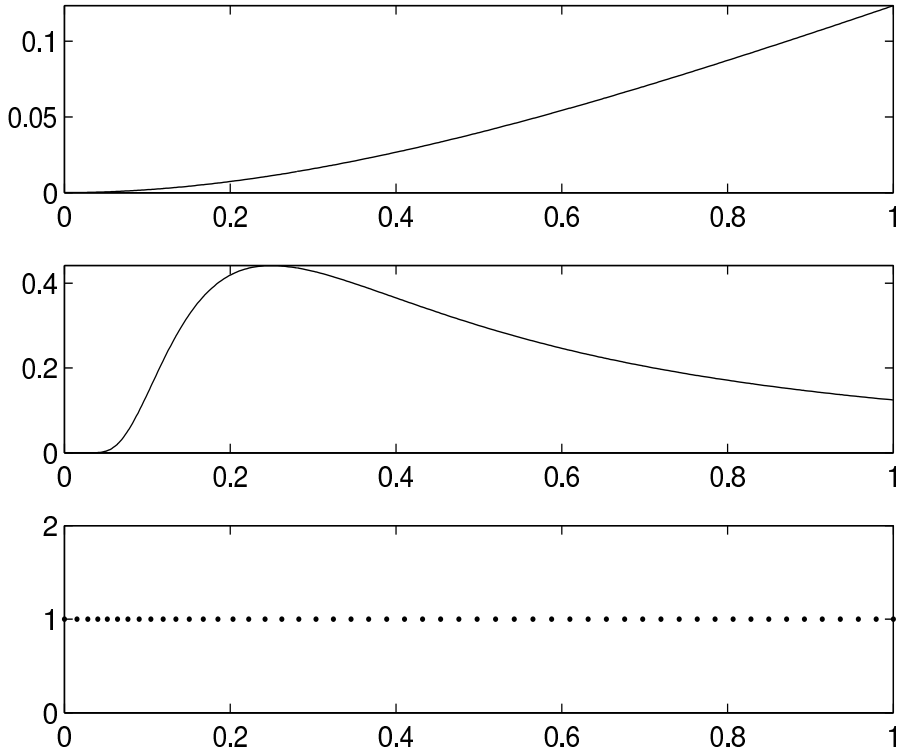

Figure 5.1: Example 5.8 with $\alpha = 0.01$, $l = 2$ and $n = 3$: Approximations of the solution u on $[0, 1]$ (uppermost graph) and u on $[1, \infty)$ transformed to [0, 1] (central graph). The computations were carried out with 8 Gaussian points and $tol_a = 10^{-10}$ and $tol_r = 10^{-6}$. The automatically selected mesh contains 52 points (lower graph).

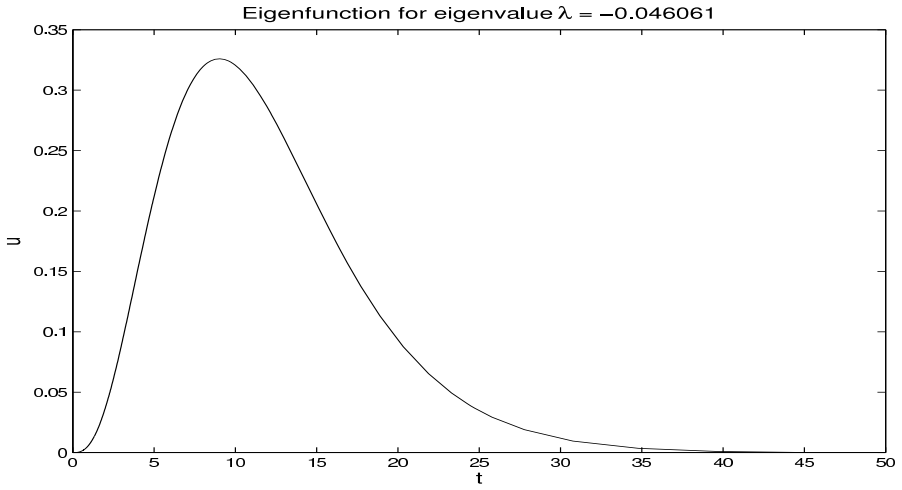

Figure 5.2: Example 5.8 with $\alpha = 0.01$, $l = 2$ and $n = 3$: Approximation of the solution u on $[0, 50]$.

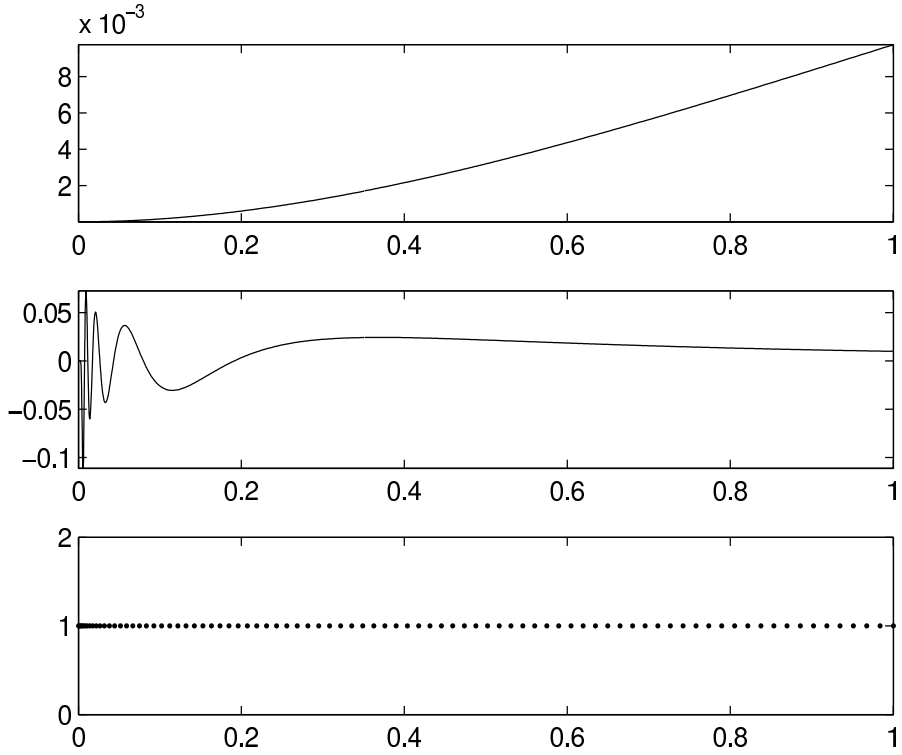

Figure 5.3: Example 5.8 with $\alpha = 0.001$, $l = 1$ and $n = 9$: Approximations of the solution u on $[0, 1]$ (uppermost graph) and u on $[1, \infty)$ transformed to [0, 1] (central graph). The computations were carried out with 6 Gaussian points and $tol_a = 10^{-6}$ and $tol_r = 10^{-2}$. The automatically selected mesh contains 85 points (lower graph).

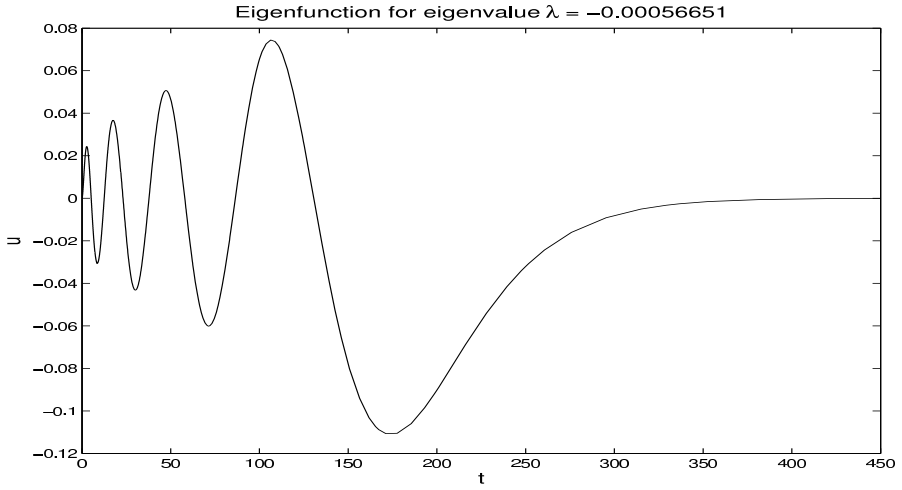

Figure 5.4: Example 5.8 with $\alpha = 0.001$, $l = 1$ and $n = 9$: Approximation of the solution u on $[0, 450]$.

n	$\lambda^{(0)}$ (MM)	no. coll.	no.mesh	`bvpsuite`	$errest_{\lambda}$	$errest_{\lambda}^{rel}$	λ_{per}
1	-0.5001	8	100	-0.500000022499859	$4.4 \cdot 10^{-19}$	$8.7 \cdot 10^{-19}$	-0.500000022500000
2	-0.1253	8	163	-0.125300320116262	$2.2 \cdot 10^{-19}$	$1.7 \cdot 10^{-18}$	-0.125300840000000
3	-0.1247	8	163	-0.124700319903047	$1.1 \cdot 10^{-19}$	$8.7 \cdot 10^{-19}$	-0.124700840000000

Table 5.23: Example 5.10: The computed eigenvalues λ for $\varepsilon = 10^{-4}$ and $b = 0$. The tolerances were set to $tol_a = 10^{-15}$ and $tol_r = 10^{-10}$.

n	no. coll.	no.mesh	$errest_{u_1}$	$errest_{u_1}^{rel}$	$errest_{u_2}$	$errest_{u_2}^{rel}$
1	8	100	$4.4 \cdot 10^{-18}$	$5.9 \cdot 10^{-18}$	$4.7 \cdot 10^{-21}$	$3.5 \cdot 10^{-17}$
2	8	163	$2.2 \cdot 10^{-17}$	$7.0 \cdot 10^{-17}$	$2.3 \cdot 10^{-17}$	$7.3 \cdot 10^{-17}$
3	8	163	$6.9 \cdot 10^{-17}$	$2.2 \cdot 10^{-16}$	$6.8 \cdot 10^{-17}$	$2.2 \cdot 10^{-16}$

Table 5.24: Example 5.10: The computed error estimates for u_1 and u_2 corresponding to the eigenvalues presented in Table 5.27. The tolerances were set to $tol_a = 10^{-15}$ and $tol_r = 10^{-10}$.

n	$\lambda^{(0)}$ (MM)	no. coll.	no.mesh	`bvpsuite`	$errest_\lambda$	$errest_\lambda^{rel}$	λ_{per}
1	-0.5000	8	99	-0.500000000224856	0	0	-0.500000000225000
2	-0.1251	8	166	-0.125030003200092	$1.1 \cdot 10^{-19}$	$8.7 \cdot 10^{-18}$	-0.125030008400000
3	-0.1250	8	217	-0.124970003199879	$2.2 \cdot 10^{-19}$	$1.7 \cdot 10^{-18}$	-0.124970008400000

Table 5.25: Example 5.10: The computed eigenvalues λ for $\varepsilon = 10^{-5}$ and $b = 0$. The tolerances were set to $tol_a = 10^{-15}$ and $tol_r = 10^{-10}$.

n	no. coll.	no.mesh	$errest_{u_1}$	$errest_{u_1}^{rel}$	$errest_{u_2}$	$errest_{u_2}^{rel}$
1	8	99	$3.1 \cdot 10^{-18}$	$4.1 \cdot 10^{-18}$	$8.0 \cdot 10^{-22}$	$6.0 \cdot 10^{-17}$
2	8	166	$5.7 \cdot 10^{-16}$	$1.8 \cdot 10^{-15}$	$5.8 \cdot 10^{-16}$	$1.8 \cdot 10^{-15}$
3	8	217	$1.1 \cdot 10^{-15}$	$3.7 \cdot 10^{-15}$	$1.2 \cdot 10^{-15}$	$3.7 \cdot 10^{-15}$

Table 5.26: Example 5.10: The computed error estimates for u_1 and u_2 corresponding to the eigenvalues presented in Table 5.25. The tolerances were set to $tol_a = 10^{-15}$ and $tol_r = 10^{-10}$.

n	$\lambda^{(0)}$ (MM)	no. coll.	no.mesh	`bvpsuite`	$errest_\lambda$	$errest_\lambda^{rel}$	λ_{per}
1	-0.5001	8	100	-0.500000022499858	$4.4 \cdot 10^{-19}$	$8.7 \cdot 10^{-19}$	-0.500000022500000
2	-0.1253	8	163	-0.125300320114753	$2.2 \cdot 10^{-19}$	$1.7 \cdot 10^{-18}$	-0.125300840000000
3	-0.1247	8	164	-0.124700319904538	$1.6 \cdot 10^{-19}$	$1.3 \cdot 10^{-18}$	-0.124700840000000

Table 5.27: Example 5.10: The computed eigenvalues λ for $\varepsilon = 10^{-4}$ and $b = 10^{-10}$. The tolerances were set to $tol_a = 10^{-15}$ and $tol_r = 10^{-10}$.

n	no. coll.	no.mesh	$errest_{u_1}$	$errest_{u_1}^{rel}$	$errest_{u_2}$	$errest_{u_2}^{rel}$
1	8	100	$3.5 \cdot 10^{-18}$	$4.7 \cdot 10^{-18}$	$4.7 \cdot 10^{-21}$	$3.5 \cdot 10^{-17}$
2	8	163	$3.3 \cdot 10^{-18}$	$1.1 \cdot 10^{-17}$	$2.8 \cdot 10^{-18}$	$9.1 \cdot 10^{-18}$
3	8	164	$2.2 \cdot 10^{-16}$	$3.1 \cdot 10^{-16}$	$2.2 \cdot 10^{-16}$	$3.1 \cdot 10^{-16}$

Table 5.28: Example 5.10: The computed error estimates for u_1 and u_2 corresponding to the eigenvalues presented in Table 5.27. The tolerances were set to $tol_a = 10^{-15}$ and $tol_r = 10^{-10}$.

b	λ_1	abs. err	λ_2	abs. err	λ_3	abs. err
10^{-10}	-0.500000022499858	$1.0 \cdot 10^{-17}$	-0.125300320114753	$1.5 \cdot 10^{-12}$	-0.124700319904538	$1.5 \cdot 10^{-12}$
10^{-12}	-0.500000022499858	$1.0 \cdot 10^{-17}$	-0.125300320116246	$1.5 \cdot 10^{-14}$	-0.124700319903062	$1.5 \cdot 10^{-14}$
10^{-14}	-0.500000022499858	$1.0 \cdot 10^{-17}$	-0.125300320116261	$1.0 \cdot 10^{-17}$	-0.124700319903047	$1.0 \cdot 10^{-17}$

Table 5.29: Example 5.10: The computed eigenvalues λ for $\varepsilon = 10^{-4}$ and various values of b. The tolerances were set to $tol_a = 10^{-15}$ and $tol_r = 10^{-10}$.

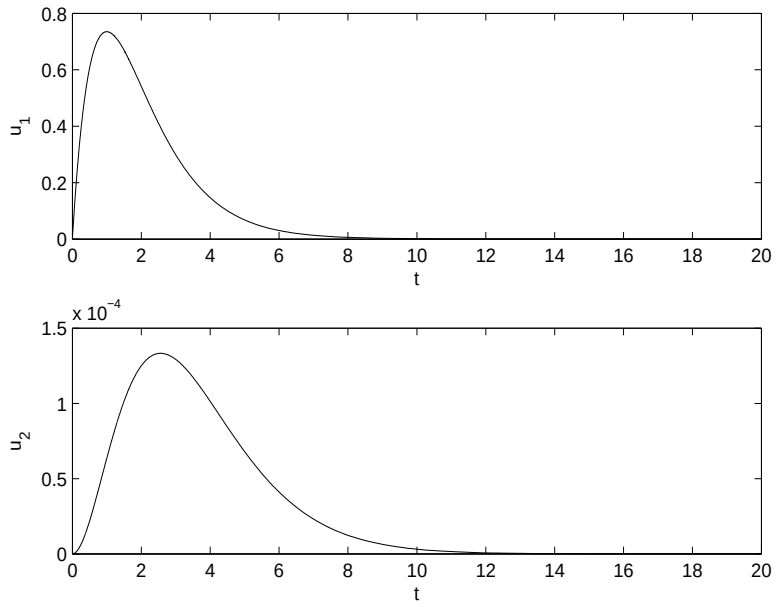

Figure 5.5: Example 5.10: Eigenfunction for $\lambda_1 = -0.500000022499859$ with $\varepsilon = 10^{-4}$ and $b = 0$.

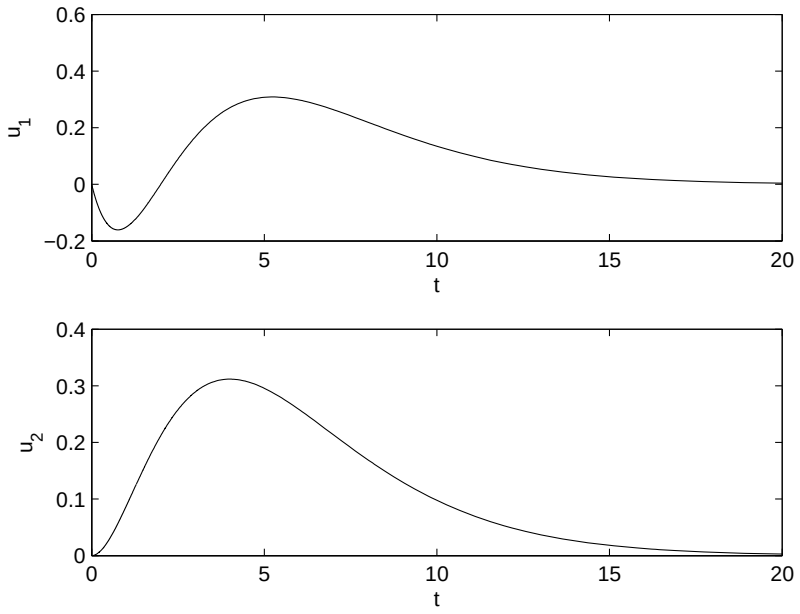

Figure 5.6: Example 5.10: Eigenfunction for $\lambda_2 = -0.125300320116262$ with $\varepsilon = 10^{-4}$ and $b = 0$.

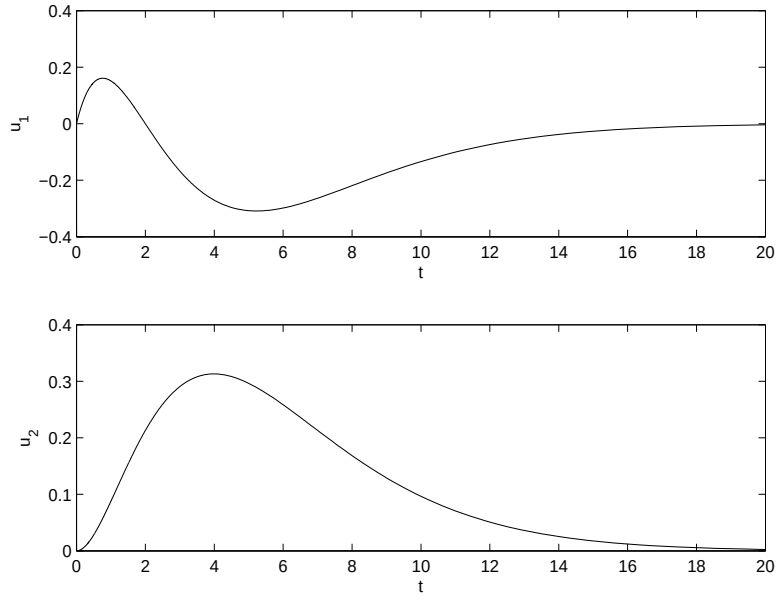

Figure 5.7: Example 5.10: Eigenfunction for $\lambda_3 = -0.124700319903124$ with $\varepsilon = 10^{-4}$ and $b = 0$.

5.6 Implementation

5.6.1 Files for this Module

The matrix method module is part of the package `bvpsuite` and contains the following files:

- boxscheme.m – set up and solution of the algebraic eigenvalue problem which approximates a differential eigenvalue problem of first order.
- approx2ndorder.m – set up and solution of the algebraic eigenvalue problem which approximates a differential eigenvalue problem of second order.
- evpsolve.m – main routine to start the GUI.
- evpsolve.fig – the corresponding GUI file.

5.6.2 Getting started

To run `bvpsuite`, start MATLAB, change to the folder where `bvpsuite` is installed, and type `bvpsuite` in the command line. The GUI window opens. To run `evpsolve` enter a file name in the edit field 'Name' of the `bvpsuite` GUI and tick the box 'Eigenvalue problem - matrix method'. Consequently a new GUI opens. Henceforth, the `bvpsuite` GUI controls are obsolete, since all inputs related to the solution of an EVP by the matrix method are defined in the `evpsolve` GUI. The GUI windows are shown in Figure 5.8.

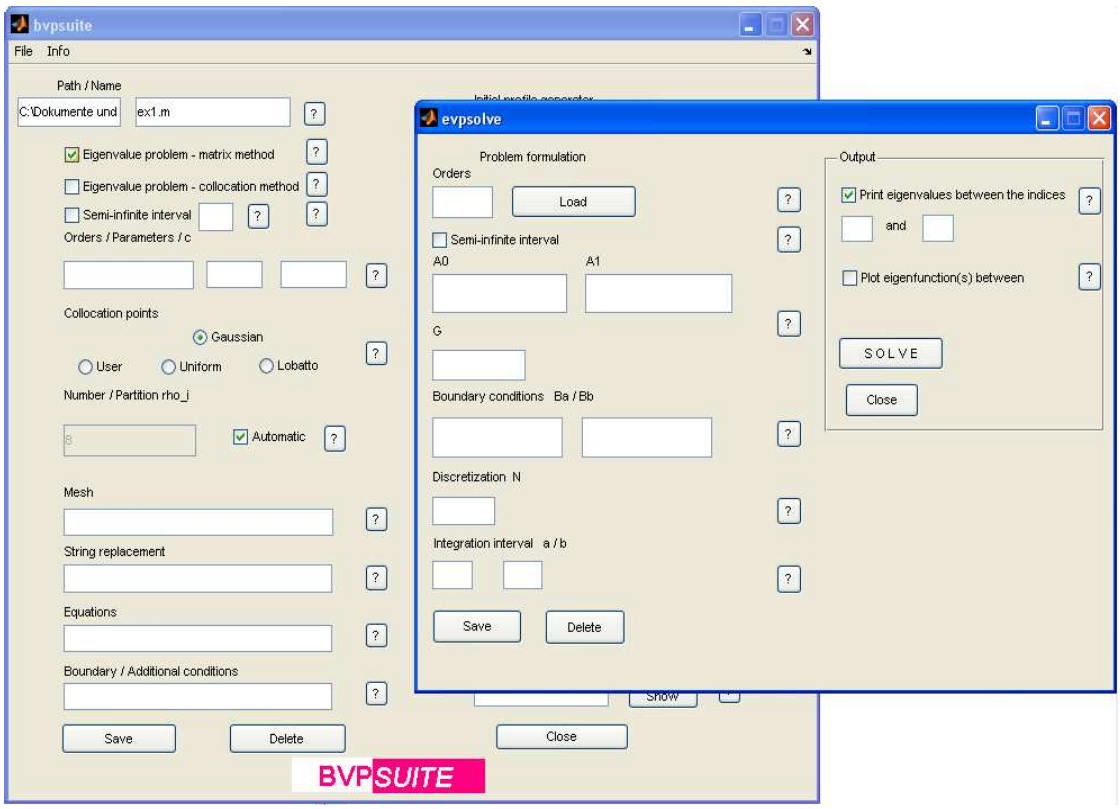

Figure 5.8: Start `evpsolve`

5.6.3 Graphical User Interface Tutorial

The following example demonstrates how the GUI can be used to solve an EVP of second order posed on $[0, \infty)$. For problems posed on semi-infinite intervals the code performs an automatic transformation to a finite interval. Consequently, the code solves the problem on the finite domain first and provides the user with a back-transformation of the numerical solution to a truncated interval.

Example 5.11.

$$-y''(t) + 2\left(\frac{l(l+1)}{2t^2} - \frac{e^{-\alpha t}}{t}\right) y(t) = 2\lambda y(t), \quad 0 \le t < \infty, \tag{5.80a}$$

$$y(0) = 0, \; y(\infty) = 0, \tag{5.80b}$$

and set $\alpha = 0.01$ an $l = 2$. The inputs are shown in Figure 5.9.

- Tick the check box 'Eigenvalue problem - matrix method' in the `bvpsuite` GUI. Subsequently the `evpsolve` GUI pops up. Note that if you open an already existing file you are asked to press the 'Load' button in order to load the data saved in the evpfile into the GUI.

- Since the given problem consists of one equation of order 2, insert the row vector `[2]` in the field 'Orders'.
- Tick the check box 'Semi-infinite interval'.
- Insert the matrices `A0=[+2/t^2-2*exp(-0.01*t)/t]`, `A1=[0]` and `G=[2]` to set up the eigenvalue problem, cf. (5.26)–(5.27).
- Fill in the constant matrices `Ba=[1,0;0,0]` and `Bb=[0,0;1,0]` to specify the linear boundary conditions $B_a y(a) + B_b y(b) = 0$.
- In the field 'Discretization N' insert `200` for the number of equidistantly spaced mesh points in the interval $(0, 1]$.
- Insert the left endpoint of the integration interval: `a=0`. Since the right endpoint is ∞, type the string `infinity` in the corresponding edit field.
- Press 'Save' to store the input in the evpfile.
- The right-hand column of fields in the GUI handles the output. Check the box for printing eigenvalues and insert `1` in the left edit field and `3` in the right one. Consequently, the code generates a table whose entries are the eigenvalues with indices between 1 and 3 and prints it in the MATLAB workspace.
- Tick 'Plot eigenfunction(s)' for a plot of the eigenfunctions corresponding to the requested eigenvalues. The plot range is determined by the discretization parameter N because the computed solution on $[0, 1]$ is back transformed to $[1, N]$.
- Finally, press 'SOLVE'. See Figure 5.10 for the graphical output.

5.6.4 Graphical User Interface Controls

In this section all options for the GUI fields corresponding to the matrix methods are described in detail.

Field	Description
Orders	Specify the orders of the differential equations. The code can only deal with systems of order one or order two, e.g. a system of 2 equations of order 2 should be denoted by the row vector [2,2].
Load	If your file has already been saved once, then you are asked to press the 'Load' button in order to load the problem specifications, which were saved in the evpfile, into the GUI.

A0/A1/G	Your system has either to be of second order, $-y'' + A1y' + A0y = \lambda Gy$, or of first order, $-y' + A0y = \lambda Gy$. Insert the matrices $A1$, $A0$ and G in the corresponding fields. If $A1$ or $A0$ is empty, insert a zero matrix of the corresponding size.
Boundary Conditions Ba/Bb	The boundary conditions have to be linear and homogeneous, i.e. $B_a y(a) + B_b y(b) = 0$. Insert the constant matrices B_a and B_b. If the differential operator is of order 2, then B_a and B_b are of dimension $(2n) \times (2n)$, where n is the number of equations. In this case the first n columns of B_a correspond to $y_1(a), \ldots, y_n(a)$ and the last n rows correspond to $y_1'(a), \ldots, y_n'(a)$. The same holds for B_b analogously.
Discretization N	Insert the number of subintervals used for the discretization. In case of a semi-infinite interval N defines the number of subintervals on $(0, 1]$. Therefore the solution on the truncated interval is provided at $2N - 1$ points.
Integration interval a/b	Insert the left endpoint `a` and the right endpoint `b` of the interval on which the problem is posed. If your problem is posed on a semi-infinite interval, write `infinity` instead. The code is designed to deal with problems posed on intervals $[a, \infty)$, $a \geq 0$.
Print eigenvalues between the indices	The eigenvalues with indices between the two inserted integers are printed in the MATLAB command window. They are sorted in ascending order. If the spectrum contains complex eigenvalues, whose imaginary part is not equal to 0, they are listed at the end. Complex numbers are sorted by their absolute values and matches are further sorted by their angles.
Plot eigenfunctions	Tick this checkbox if you want to have a plot of the corresponding eigenfunctions.
SOLVE	Starts the computations. Depending on the orders of the differential equations either the function boxscheme.m, for first order problems, and approx2ndorder.m for second order problems are called.

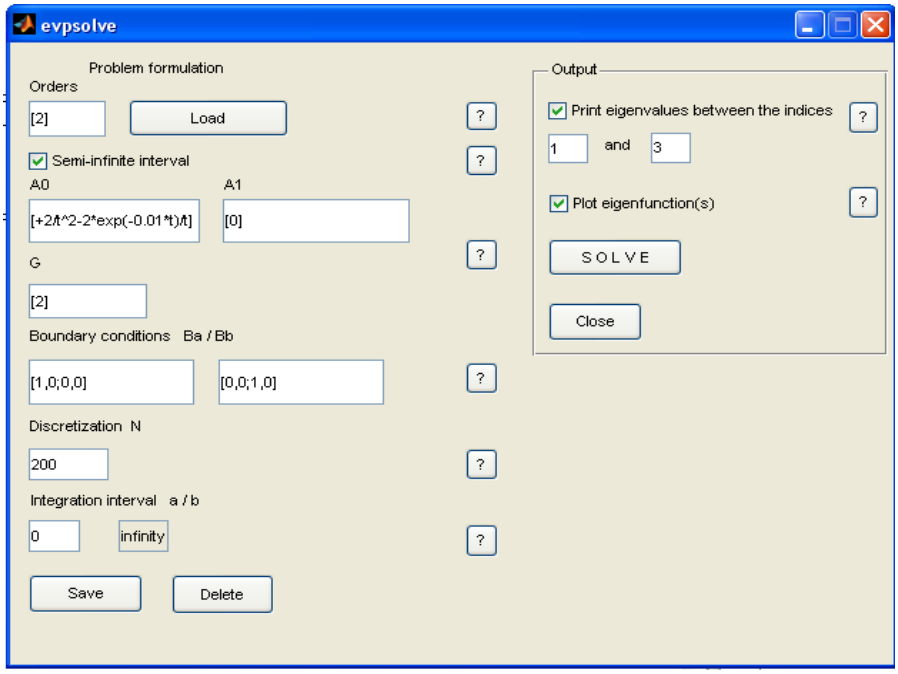

Figure 5.9: Example 5.11: GUI input data

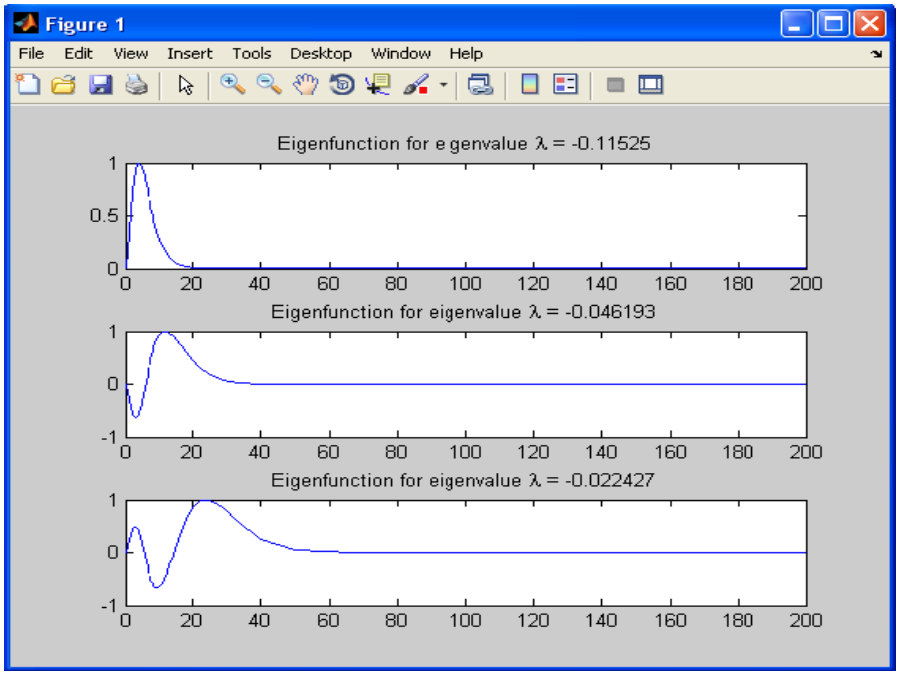

Figure 5.10: Example 5.11: Results

5.6.5 The evpfile

This section describes the general form of the evpfiles. Those files are generated automatically by the GUI. In general, the inputs correspond to the inputs of the GUI.

```
function [ret] = evp_ex1(nr,t)
switch nr
% Define whether the problem is an EVP, which is solved by the matrix method.
      case 'EVP_mm'
            ret=true;

% Your problem is of the type -y'' + A1y' + A0y = λGy (or -y' + A_0 y = λGy).
% Define the matrix G.
      case 'G'
            ret=[2,0;0,(1/t^4)*(2);];

% Define the matrix A0.
      case 'A0'
            ret=[+2/t^2-2*exp(-0.01*t)/t,0;
                  0,(1/t^4)*(+2/(1/t)^2-2*exp(-0.01*(1/t))/(1/t));];

% Define the matrix A1.
      case 'A1'
            ret=[0,0;0,(-1/t^2)*(0)-2/t;];

% The linear boundary conditions – the matrix Ba corresponds to the boundary conditions at the left endpoint.
      case 'Ba'
            ret=[1+0*(-3/2),0,+0*(2),0,+0*(-1/2),0;
                  0+0*(-3/2),1,+0*(2),0,+0*(-1/2),0;
                  0,0,0,0,0,0;0,0,0,0,0,0;];

% The matrix Bb corresponds to the boundary conditions at the right endpoint
      case 'Bb'
            ret=[0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,1,-1;
                  +1/2,+1/2,-2,-2,+3/2,+3/2;];

% a is the left endpoint of the integration domain. If your problem is posed on a
semi-infinite interval the code automatically transforms the problem to (0, 1].
      case 'a'
            ret=0;

% The right endpoint of the integration domain.
      case 'b'
```

```
            ret=1;
```

% The number of subintervals in $[a, b]$ which implicitly determine the dimension of the discretized problem. If your problem is posed on a semi-infinite interval, N is the number of subintervals in $(0, 1]$

```
      case 'N'
            ret=400;
```

% The left endpoint of the semi-infinite interval $[a, \infty)$, $a \geq 0$.

```
      case 'antwort_endpoint'
            ret=0;
```

% boolean; determines if your problem is posed on a semi-infinite interval

```
    case 'infinite'
            ret=true;
```

% number of equations

```
    case 'n'
            ret=2;
```

% orders of differential equations to be solved;

```
    case 'orders'
            ret=2;
end
%Values read by bvpsuite GUI:
%EVP_mm
%true%#EVP_mm
%Values read by evpsolve GUI:
%infinite
%true%#infinite
%orders
%[2]%#orders
%a
%0%#a
%b
%1%#b
%N
%400%#N
%A0
%[+2/t^2-2*exp(-0.01*t)/t]%#A0
%A1
%[0]%#A1
%G
%[2]%#G
```

```
%Ba
%[1,0;0,0]%#Ba
%Bb
%[0,0;1,0]%#Bb
%up_ind
%8%#up_ind
%low_ind
%1%#low_ind
```

Chapter 6

Conclusions

In this work numerical methods for the solution of linear eigenvalue problems for systems of ordinary differential equations are considered.
Two different approaches were implemented for this relevant and nontrivial problem.
We studied a collocation method in order to take advantage of the already existing code `bvpsuite` which features an automatic mesh selection and an asymptotically correct error estimate. The reformulation of the eigenvalue problem as a related boundary value problem makes it accessible to the routines of the code.
Additionally, we considered the matrix method. One reason why we followed this approach is that it incorporates the possibility to solve not only Sturm-Liouville type but also more general problems. For this purpose, in Chapter 5 coupled vector eigenvalue problems were considered and accurate solutions were computed. This distinguishes matrix methods from most of the available standard codes described in Chapter 2. Moreover, the matrix method serves very well as a procedure for finding good starting profiles for the solution of the nonlinear collocation equations.
Unfortunately, the dimension of the resulting algebraic problem increases rapidly as the discretization parameter tends to zero. This leads to a high computational effort, because even for these small step sizes the current MATLAB implementation only allows to compute the whole discrete spectrum of the discretized problem. Since we are only interested in a small number of eigenvalues, performance improvements could be achieved by developing faster solver routines for the solution of the algebraic eigenvalue problem, providing only the relevant fraction of the eigenvalues, and suitable for a high-performance (parallel) implementation.
According to results stated in Chapter 5 the accuracy of the eigenvalue approximation decreases with the index of the eigenvalue. This means that for eigenvalues with a higher index small step sizes have to be taken in order to achieve satisfactory accuracy. Whereas the numerical solution of one-dimensional Sturm-Liouville problems has been extensively studied in the literature, methods for coupled high dimensional eigenvalue problems are not so well-known. By a combination of the two presented numerical methods we were able to solve such high dimensional EVPs, which frequently arise in quantum mechanics, to an accuracy comparable with recent results in the literature, cf. Section 5 in Chapter

5. These problems were in addition singular and posed on a semi-infinite interval. Since `bvpsuite` offers an asymptotically correct error estimate, we are able to assess the accuracy of our results.
From the experiences we made in the course of this work we want to point out that the applicability of a certain method to solve eigenvalue problems arising in ordinary differential equations depends significantly on the problem type. However, matrix methods show advantageous properties that seem relevant in applications for Schrödinger eigenvalue problems.

Bibliography

[1] U. Ascher, R. Mattheij, and R. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[2] W. Auzinger, E. Karner, O. Koch, and E. Weinmüller, *Collocation methods for the solution of eigenvalue problems for singular ordinary differential equations*, Opuscula Math., 26 (2006), pp. 229–241.

[3] W. Auzinger, G. Kneisl, O. Koch, and E. Weinmüller, *A collocation code for boundary value problems in ordinary differential equations*, Numer. Algorithms, 33 (2003), pp. 27–39.

[4] W. Auzinger, O. Koch, D. Praetorius, and E. Weinmüller, *New a posteriori error estimates for singular boundary value problems*, Numer. Algorithms, 40 (2005), pp. 79–100.

[5] W. Auzinger, O. Koch, and E. Weinmüller, *Analysis of a new error estimate for collocation methods applied to singular boundary value problems*, SIAM J. Numer. Anal., 42 (2005), pp. 2366–2386.

[6] P. Bailey, W. Everitt, and A. Zettl, *Algorithm 810: The SLEIGN2 Sturm-Liouville code*, ACM Trans. Math. Software, 27 (2001), pp. 143–192.

[7] O. Bayrak and I. Boztosun, *Analytical Solutions of the Hulthén and the Morse Potentials by using the Asymptotic Iteration Method*, Journal of Molecular Structure, THEOCHEM, 802 (2007), pp. 17–21.

[8] C. Budd, O. Koch, and E. Weinmüller, *From nonlinear PDEs to singular ODEs*, Appl. Numer. Math., 56 (2006), pp. 413–422.

[9] J. Cash, G. Kitzhofer, O. Koch, G. Moore, and E. Weinmüller, *Numerical solution of singular two-point BVPs*, JNAIAM J. Numer. Anal. Indust. Appl. Math., 4 (2009), pp. 129–149.

[10] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Interscience Publishers INC., New York, 1953.

[11] O. Gomes, H. Chacham, and J. Mohallem, *Variational calculations for the bound-unbound transition of the Yukawa potential*, Phys. Rev. A, 50 (1994), pp. 228–231.

[12] L. Greenberg and M. Marletta, *Algorithm 775: The code SLEUTH for solving fourth-order Sturm-Liouville problems*, ACM Trans. Math. Software, 23 (1997), pp. 453–493.

[13] F. d. Hoog and R. Weiss, *Difference methods for boundary value problems with a singularity of the first kind*, SIAM J. Numer. Anal., 13 (1976), pp. 775–813.

[14] ——, *The numerical solution of boundary value problems with an essential singularity*, SIAM J. Numer. Anal., 16 (1979), pp. 637–669.

[15] ——, *On the boundary value problem for systems of ordinary differential equations with a singularity of the second kind*, SIAM J. Math. Anal., 11 (1980), pp. 41–60.

[16] ——, *The application of Runge-Kutta schemes to singular initial value problems*, Math. Comp., 44 (1985), pp. 93–103.

[17] H. Keller, *On the accuracy of finite difference approximations to the eigenvalues of differential and integral operators*, Numer. Math., 7 (1965), pp. 412–419.

[18] G. Kitzhofer, *Numerical Treatment of Implicit Singular BVPs*, Ph.D. Thesis, Inst. for Anal. and Sci. Comput., Vienna Univ. of Technology, Austria, 2009. In preparation.

[19] G. Kitzhofer, O. Koch, P. Lima, and E. Weinmüller, *Efficient numerical solution of the density profile equation in hydrodynamics*, J. Sci. Comput., 32 (2007), pp. 411–424.

[20] G. Kitzhofer, O. Koch, G. Pulverer, C. Simon, and E. Weinmüller, *Treatment of Singular BVPs: The New Matlab Code bvpsuite*, 2009. Available at `http://www.math.tuwien.ac.at/~ewa/`.

[21] O. Koch, *Asymptotically correct error estimation for collocation methods applied to singular boundary value problems*, Numer. Math., 101 (2005), pp. 143–164.

[22] H.-O. Kreiss, *Difference approximations for boundary and eigenvalue problems for ordinary differential equations*, Math. Comp., 26 (1972), pp. 605–624.

[23] V. Ledoux, *Study of Special Algorithms for Solving Sturm-Liouville and Schrödinger Equations*, Ph. D. Thesis, Department of Applied Mathematics and Computer Science, Ghent University, 2007.

[24] M. Marletta and J. Pryce, *A new multipurpose software package for Schrödinger and Sturm-Liouville computations*, Comput. Phys. Commun., 62 (1991), pp. 42–52.

[25] J. Paine, F. de Hoog, and R. Anderssen, *On the correction of finite difference eigenvalue approximations for Sturm-Liouville problems*, Computing, 26 (1981), pp. 123–139.

[26] S. Pruess and C. Fulton, *Mathematical software for Sturm–Liouville problems*, ACM Trans. Math. Software, 19 (1900), pp. 360–376.

[27] J. Pryce, *Numerical Solution of Sturm-Liouville Problems*, Oxford University Press, New York, 1993.

[28] A. Roy, *The generalized pseudospectral approach to the bound states of the Hulthén and the Yukawa potentials*, PRAMANA – J. Phys., 65 (2005), pp. 1–15.

[29] H. Schlichting, *Boundary Layer Theory*, McGraw-Hill, New York, 1968.

[30] F. Schwabl, *Quantenmechanik*, Springer Verlag, Berlin; Heidelberg, 1993.

[31] G. Teschl, *Mathematical Methods in Quantum Physics*, Graduate Studies in Mathematics, Amer. Math. Soc., Providence, 2009.

[32] E. Weinmüller, *A difference method for a singular boundary value problem of second order*, Math. Comp., 42 (1984), pp. 441–464.

[33] ——, *On the boundary value problems for systems of ordinary second order differential equations with a singularity of the first kind*, SIAM J. Math. Anal., 15 (1984), pp. 287–307.