Study of Special Algorithms for solving Sturm-Liouville and Schrödinger Equations

Studie van speciale algoritmen voor het oplossen van Sturm-Liouville- en Schrödinger-vergelijkingen

Veerle Ledoux

Promotor: prof. dr. M. Van Daele
Copromotor: prof. dr. G. Vanden Berghe

Proefschrift ingediend tot het behalen van de graad van
Doctor in de Wetenschappen: Informatica

Vakgroep Toegepaste Wiskunde en Informatica
Voorzitter: prof. dr. G. Vanden Berghe
Faculteit Wetenschappen
Academiejaar 2006-2007

UNIVERSITEIT
GENT

Acknowledgements

I would like to thank my promotor Prof. Marnix Van Daele for following my research with interest and for giving me constant support and guidance. I have always valued his ideas, comments and suggestions. I also would like to express my thanks to Prof. Guido Vanden Berghe for his co-promotorship, for his continuous interest and stimulation and for making perfect conditions for my work as a young researcher.

Next, I would like to take this opportunity to thank Prof. Liviu Gr. Ixaru for his many suggestions and positive criticism during his stays in Ghent. I am also grateful to him and to Prof. Margarit Rizea for the cooperation, which contributed a lot to my work.

I thank all my colleagues and ex-colleagues in the department for a friendly atmosphere which makes work pleasant. I also want to express my thanks to the administrative and technical staff which gave me often precious administrative and logistic help.

I am grateful to my parents for their trust and love. Finally, I thank Nele, Bart and Stefan for their support that I could always count on.

This research has been supported by the Fund for Scientific Research - Flanders (Belgium) (F.W.O.-Vlaanderen).

Veerle Ledoux

Contents

Chapter 1

Introduction

1.1 Introductory background

In [92], Murphy describes a *differential equation* as a relation involving one or more derivatives and an unknown function. The problem of solving it is a search for that unknown function. The *solution* of a differential equation is then any relation, free from derivatives, which satisfies the equation identically.

The most general *ordinary differential equation* (frequently called an ODE) is

$$F(x,\ y,\ y',\ y'',\ \ldots,\ y^{(n)}) = 0, \tag{1.1}$$

where x is the independent variable, y is the dependent variable and the notation

$$y', y'',\ \ldots,\ y^{(n)} \tag{1.2}$$

is used to denote the derivatives

$$\frac{dy}{dx},\ \frac{d^2y}{dx^2},\ \ldots,\ \frac{d^ny}{dx^n}. \tag{1.3}$$

The word “ordinary” is used to emphasize that no partial derivatives appear, since there is just one independent variable. If y is a function of more than one independent variable and partial derivatives with respect to those variables are present, the equation is called a *partial differential equation* (PDE).

The *order* of a differential equation is the order n of the highest derivative that appears. Another important concept is that of linearity. An ordinary differential equation is said to be *linear* if it has the form

$$a_0(x)y^{(n)} + a_1(x)y^{(n-1)} + \cdots + a_{n-1}(x)y' + a_n(x)y = Q(x). \tag{1.4}$$

Thus the equation is linear in y and its derivatives.

Differential equations are studied in both pure and applied mathematics. Pure mathematicians study the different types and properties of differential equations, such as whether or not solutions exist, and when they exist, whether they are unique. Applied mathematicians, physicists and engineers are more interested in how to compute solutions to differential equations. However, many of these equations do not have closed form solutions and must be solved using numerical methods.

This dissertation concerns the design of efficient numerical methods for solving numerically one particular class of ordinary differential equations called *Sturm-Liouville equations*. These linear second order differential equations describe a lot of important physical phenomena which exhibit a pronounced oscillatory character; behaviour of pendulum-like systems, vibrations, resonances and wave propagation are all phenomena of this type in classical mechanics, while the same is true for the typical behaviour of quantum particles.

Before considering the Sturm-Liouville equation in more detail, we list some notations which will be used throughout this thesis.

- **Intervals**. The notations $[a, b]$, (a, b), $[a, b)$ $(a, b]$ are used to denote the closed, open and half open intervals. Let $\mathbb{R}$ represent the real line, then

$$
\begin{aligned}
&(i)\ \ (a, b) = \{x \in \mathbb{R} : -\infty \le a < x < b \le +\infty\}\\
&(ii)\ \ [a, b] = \{x \in \mathbb{R} : -\infty < a \le x \le b < +\infty\}\\
&(iii)\ \ [a, b) = \{x \in \mathbb{R} : -\infty < a \le x < b \le +\infty\}\\
&(iv)\ \ (a, b] = \{x \in \mathbb{R} : -\infty \le a < x \le b < +\infty\}.
\end{aligned}
$$

- **Continuity**. A function f is C^0 on an interval if it is continuous there, C^1 if it has a continuous first derivative, C^2 if it has a continuous second derivative and so on. Let I be any interval of $\mathbb{R}$ and let $n \in \{0, 1, 2, \ldots\}$, then

$$C^n(I) = \{f : I \to \mathbb{C} : f^{(r)} \text{ is continuous on } I \text{ for } r = 0, 1, \ldots, n\}.$$

- **Integrability**. A real- or complex-valued function of a real variable is integrable on an interval if the integral of the function over that interval exists and is finite. A real- or complex-valued function of a real variable is square-integrable on an interval if the integral of the square of its absolute value, over that interval, is finite. Let I be any interval of $\mathbb{R}$, then

$$
\begin{aligned}
&(i)\ \ L^1(I) = \{f : I \to \mathbb{C} : \int_I |f(x)| dx < +\infty\}\\
&(ii)\ \ L^2(I) = \{f : I \to \mathbb{C} : \int_I |f(x)|^2 dx < +\infty\}.
\end{aligned}
$$

1.2 The Sturm-Liouville problem

A classical Sturm-Liouville equation, named after Jacques Charles François Sturm (1803-1855) and Joseph Liouville (1809-1882), is a real second-order linear differential equation

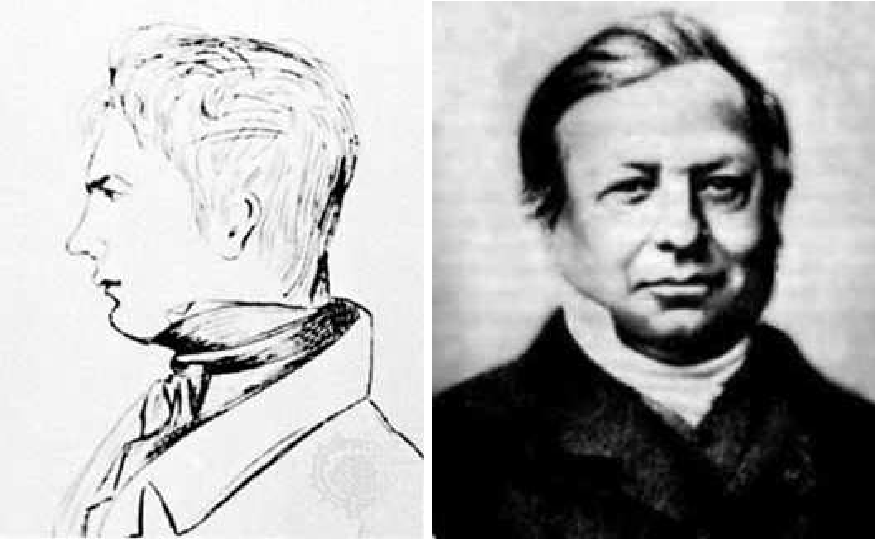


Figure 1.1: The two eponyms of the Sturm-Liouville theory: ***(Left)*** Jacques Charles François Sturm (1803 - 1855), French mathematician, of German extraction. ***(Right)*** Joseph Liouville (1809 - 1882), French mathematician.

of the form

$$-\frac{d}{dx}\left[p(x)\frac{dy(x)}{dx}\right]+q(x)y(x)=Ew(x)y(x), \tag{1.5}$$

where $p(x)$, $q(x)$ and $w(x)$ are given functions and in the simplest of cases are continuous on the finite closed interval $[a,b]$. Often the Sturm-Liouville equation is defined together with boundary conditions, specifying the solution in the endpoints a and b. In the regular Sturm-Liouville theory these boundary conditions have the form

$$a_0y(a)+b_0p(a)y'(a)=0, \quad a_1y(b)+b_1p(b)y'(b)=0 \tag{1.6}$$

where a_0, b_0 are not both zero, nor are a_1, b_1. The value of E is not specified in the equation; finding the values of E for which there exists a nontrivial (nonzero) solution y of (1.5) satisfying the boundary conditions is part of the problem called the *Sturm-Liouville problem*. Such values of E, when they exist, are called the *eigenvalues* of the boundary value problem defined by (1.5) and the prescribed set of boundary conditions. The corresponding solutions $y(x)$ (for such a E) are the *eigenfunctions* of this problem.

Example 1.1 Solve the following equation

$$\frac{d^2y}{dx^2}+Ey=0, \quad 0\le x\le \pi$$

with boundary conditions

$$y(0) = 0, \quad y(\pi) = 0.$$

Here $p(x) = w(x) = 1$, $q(x) = 0$, $a = 0$, $b = \pi$, $a_0 = a_1 = 1$ and $b_0 = b_1 = 0$. The general solution to the differential equation is

$$y(x) = c_1 \cos(\sqrt{E}x) + c_2 \sin(\sqrt{E}x)$$

with $E > 0$. If $E \leq 0$, then the system has only the trivial solution $y = 0$. This is not of interest, since every Sturm-Liouville system has a trivial solution.

The condition $y(0) = 0$ implies that $c_1 = 0$; hence the updated solution becomes

$$y(x) = c_2 \sin(\sqrt{E}x).$$

The second condition $y(\pi) = 0$ implies that either $c_2 = 0$ (which would lead to the trivial solution) or $\sqrt{E}\pi = k\pi$, that is $E = k^2$, $k = 1, 2, 3, \ldots$. The eigenvalues of the system are thus $E_0 = 1$, $E_1 = 4$, $E_2 = 9, \ldots$. The corresponding eigenfunctions are $y_0(x) = \sin(x)$, $y_1(x) = \sin(2x)$, $y_2(x) = \sin(3x), \ldots$, and in general

$$y_k(x) = \sin((k+1)x), \quad k = 0, 1, 2, \ldots,$$

where the arbitrary constants have been set equal to one, since eigenfunctions are unique only upto a multiplicative constant.

1.2.1 Where Sturm-Liouville problems come from

The one-dimensional Sturm-Liouville problem models a large number of important physical processes. The general Sturm-Liouville problem even arose first in the context of the separation of variables method for partial differential equations modelling physical processes in more than one dimension. The separation of variables method was applied on the partial differential equation to obtain a Sturm-Liouville problem for each independent variable.

Example 1.2 We use the separation of variables to show how we can get solutions of the two dimensional Laplace's equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0. \tag{1.7}$$

We look for solutions of the form

$$u(x, y) = X(x)Y(y). \tag{1.8}$$

From Laplace's equation we get

$$Y(y)X''(x) + X(x)Y''(y) = 0. \tag{1.9}$$

Separating variables and assuming $X(x) \neq 0$, $Y(y) \neq 0$ we get

$$\frac{X''(x)}{-X(x)} = \frac{Y''(y)}{Y(y)}. \tag{1.10}$$

Since the left hand side of this equation depends only on x and the right hand side depends only on y we get that

$$\frac{X''(x)}{-X(x)} = \frac{Y''(y)}{Y(y)} = \lambda, \tag{1.11}$$

where λ is a constant. This leads to the Sturm-Liouville differential equations

$$X'' = -\lambda X, \quad Y'' = \lambda Y. \tag{1.12}$$

It follows that if X is a solution of the first differential equation in (1.12) and Y is a solution of the second equation in (1.12) for the same λ value, then

$$u(x,y) = X(x)Y(y) \tag{1.13}$$

is a solution of Laplace's partial differential equation.

Many mechanical systems lead to a general form of a Sturm-Liouville problem. In many applications, the Sturm-Liouville problem describes the oscillation in the physical system. In [105] e.g. it is shown how a Sturm-Liouville problem arises in the context of a vibrating (heavy) string. Also in [118] some examples of physical problems leading to differential equations of Sturm-Liouville type can be found.

1.2.2 The Schrödinger problem

Figure 1.2: The Austrian physicist Erwin Schrödinger (1887 - 1961).

A specific subclass of the Sturm-Liouville equations is formed by the so-called Schrödinger equations [114]. The one-dimensional *Schrödinger equation* is the fundamental equation of quantum mechanics. It arises there in its time-independent form as

$$-\frac{\hbar}{2m}y''(x) + (V(x) - E)y(x) = 0, \tag{1.14}$$

where m is the mass of the particle and $\hbar$ is Planck's constant over 2π. To a quantum physicist or chemist, $V(x)$ is a *potential function* describing a potential field, an eigenvalue E is an *energy level* and its eigenfunction is the corresponding *wave function* of a particle, the two together describing a *bound state*. For a regular Schrödinger problem the boundary conditions take the form

$$a_0 y(a) + b_0 y'(a) = 0, \quad a_1 y(b) + b_1 y'(b) = 0 \tag{1.15}$$

where a_0, b_0 are not both zero, and similarly for a_1, b_1.

In this thesis the Schrödinger equation is considered in natural units (also called *Liouville normal form*):

$$y''(x) = (V(x) - E)y(x). \tag{1.16}$$

1.3 Basic properties of the Sturm-Liouville problem

During time an extensive theory was developed for the regular boundary value problem (1.5)-(1.6), the so-called *Sturm-Liouville theory*. In this section we bring together those facts which seem especially relevant for the subject of this thesis. For a more elaborated study of the Sturm-Liouville theory we can refer to [105, 111, 132].

It may be assumed throughout the following, that $p(x)$ and $w(x)$ are strictly positive on the open interval (a, b) and that $p(x), q(x)$ and $w(x)$ are piecewise continuous on (a, b).

1.3.1 Existence, uniqueness and linearity

From the basic existence and uniqueness theorem for (linear) ordinary differential equations it follows that if $p(x), q(x)$ and $w(x)$ are (piecewise) continuous on an interval, with $p(x)$ strictly positive there, then the Sturm-Liouville equation

$$-\frac{d}{dx}\left[p(x)\frac{dy(x)}{dx}\right] + q(x)y(x) = Ew(x)y(x), \tag{1.17}$$

has a unique solution satisfying any given initial conditions

$$y(c) = \alpha, \quad (py')(c) = \beta \tag{1.18}$$

at a point c of the interval (see [132]).

Proposition 1.1. *Suppose that* (1.17) *is a Sturm-Liouville equation with* $p(x)$, $q(x)$ *and* $w(x)$ *continuous, and* $p(x) > 0$ *for all* $x \in [a, b]$. *Then the set of all functions* $y(x)$ *satisfying* (1.17) *is a vector space of dimension 2. In other words, there exist two linearly independent solutions of* (1.17)*, and any other solution of* (1.17) *is a linear combination of these.*

Proof. The differential equation (1.17) is equivalent to the non-autonomous linear system

$$\begin{cases} y'(x) = \dfrac{1}{p(x)}z(x) \\ z'(x) = [q(x) - Ew(x)]y(x). \end{cases} \tag{1.19}$$

Hence, by the basic existence and uniqueness theorem, there exists a unique solution of (1.17) with initial values $y(a) = 1, p(a)y'(a) = 0$. Similarly, there exists a unique solution of (1.17) with initial values $y(a) = 0, p(a)y'(a) = 1$. Let us denote these solutions by $u(x)$ and $v(x)$. Then $u(x)$ and $v(x)$ are linearly independent (neither function is a constant multiple of the other). Moreover, if $y(x)$ is any solution of (1.17), then

$$y(x) = y(a)u(x) + p(a)y'(a)v(x).$$

To see this, consider the function $\tilde{y}(x) = y(x) - y(a)u(x) - p(a)y'(a)v(x)$. The function $\tilde{y}(x)$ is a solution of (1.17) with initial values $\tilde{y}(a) = y(a) - y(a)u(a) - p(a)y'(a)v(a) = 0$ and $p(a)\tilde{y}'(a) = p(a)[y'(a) - y(a)u'(a) - p(a)y'(a)v'(a)] = 0$. Hence, the uniqueness theorem implies that $\tilde{y}(x) = 0$ for all $x \in [a, b]$. □

One can then say that the Sturm-Liouville equation (1.17) is a *linear* differential equation. That is, if we define the *differential operator*

$$L = \frac{1}{w(x)} \left\{ -\frac{d}{dx} \left(p(x) \frac{d}{dx} \right) + q(x) \right\} \quad \text{on } a < x < b \tag{1.20}$$

then L is a *linear operator*,

$$L(\alpha y + \beta z) = \alpha L(y) + \beta L(z), \tag{1.21}$$

where $L(u)$ is the notation used to denote the function that results from applying L to u. A consequence of the linearity is that the general solution of (1.17) (for any given E) is of the form

$$y(x) = \alpha_1 y_1(x) + \alpha_2 y_2(x), \tag{1.22}$$

where $y_1(x)$ and $y_2(x)$ are any two linearly independent solutions.

1.3.2 Reality of eigenvalues and orthogonality of eigenfunctions

The eigenvalue problem (1.17) can be written as

$$L(y) = Ey \tag{1.23}$$

for the linear differential operator L defined above. A fundamental result regarding the linear operator L is Green's identity:

Proposition 1.2. Green's identity

$$\int_a^b \left(L(y_i) y_j^* - y_i L(y_j^*) \right) w \, dx = \Big[p(x) \left[y_i(x) y_j^{*\prime}(x) - y_i'(x) y_j^*(x) \right] \Big]_a^b \tag{1.24}$$

for any well-behaved (real or complex) functions $y_i(x)$ and $y_j(x)$. y_j^* is the complex conjugate of the function y_j.

Proof. The left-hand side of (1.24) can be written as

$$\int_a^b \left[-(p y_i')' y_j^* + q y_i y_j^* \right] dx - \int_a^b \left[-(p y_j^{*\prime})' y_i + q y_i y_j^* \right] dx$$

$$= \int_a^b \left[(py_j^{*\prime})' y_i - (py_i')' y_j^* \right] dx$$

Using partial integration we obtain the required result

$$\Big[py_j^{*\prime} y_i\Big]_a^b - \int_a^b py_j^{*\prime} y_i' dx - \Big[py_i' y_j^*\Big]_a^b + \int_a^b py_i' y_j^{*\prime} dx.$$

□

Let us define the *inner-product* as

$$\langle y_i, y_j \rangle = \int_a^b y_i y_j^* w \, dx. \tag{1.25}$$

Then Green's identity (1.24) can also be written as

$$\langle L(y_i), y_j \rangle - \langle y_i, L(y_j) \rangle = \Big[p(x) \left[y_i(x) y_j^{*\prime}(x) - y_i'(x) y_j^*(x)\right]\Big]_a^b, \tag{1.26}$$

provided p, q and w are real-valued functions (then $L^*(y_j) = L(y_j^*)$).

For a regular Sturm-Liouville problem any two (real) solutions y_i and y_j satisfy the regular boundary conditons (1.6), this means that

$$\begin{cases} a_0 y_i(a) + b_0 p(a) y_i'(a) = 0 \\ a_1 y_i(b) + b_1 p(b) y_i'(b) = 0 \end{cases} \quad \text{and} \quad \begin{cases} a_0 y_j(a) + b_0 p(a) y_j'(a) = 0 \\ a_1 y_j(b) + b_1 p(b) y_j'(b) = 0. \end{cases} \tag{1.27}$$

It follows that

$$\begin{aligned} &\langle L(y_i), y_j \rangle - \langle y_i, L(y_j) \rangle \\ &\qquad = \Big[p(x) \left[y_i(x) {y_j}'(x) - y_i'(x) y_j(x)\right]\Big]_a^b \\ &\qquad = y_i(b) p(b) {y_j}'(b) - p(b) y_i'(b) y_j(b) - y_i(a) p(a) {y_j}'(a) + p(a) y_i'(a) y_j(a) \\ &\qquad = -\frac{a_1}{b_1} y_i(b) y_j(b) + \frac{a_1}{b_1} y_i(b) y_j(b) + \frac{a_0}{b_0} y_i(a) y_j(a) - \frac{a_0}{b_0} y_i(a) y_j(a) \\ &\qquad = 0, \end{aligned} \tag{1.28}$$

when $b_0 \neq 0$ and $b_1 \neq 0$. If $b_0 = 0$ (and/or $b_1 = 0$), then is $y_i(a) = y_j(a) = 0$ (and/or $y_i(b) = y_j(b) = 0$). Thus for a Sturm-Liouville problem with boundary conditions (1.6) and real-valued coefficient functions, the *symmetry* or *formal self-adjointness* property holds, namely that

$$\langle L(y_i), y_j \rangle = \langle y_i, L(y_j) \rangle \tag{1.29}$$

This symmetry relation implies the following simple results:

Proposition 1.3. *With the above assumptions on the coefficient functions and boundary conditions,* *(i) The eigenvalues of a Sturm-Liouville problem are real.*

(ii) The eigenfunctions belonging to distinct eigenvalues are orthogonal (with weighting function $w(x)$) with respect to the inner product (1.25).

Proof. Suppose we have eigenfunctions y_i, y_j satisfying $L(y_i) = E_i y_i$, $L(y_j) = E_j y_j$ where $E_i \neq E_j$. Then

$$E_i \langle y_i, y_j \rangle = \langle E_i y_i, y_j \rangle = \langle L(y_i), y_j \rangle = \langle y_i, L(y_j) \rangle = \langle y_i, E_j y_j \rangle = E_j^* \langle y_i, y_j \rangle .$$

When we take $y_i = y_j$ and $E_i = E_j$ we have $E_i \langle y_i, y_i \rangle = E_i^* \langle y_i, y_i \rangle$ and since y_i is not identically zero we have $\langle y_i, y_i \rangle \neq 0$, so that $E_i = E_i^*$ which proves part (i). Since E_j is real the equation above gives $E_i \langle y_i, y_j \rangle = E_j \langle y_i, y_j \rangle$. This means that $\langle y_i, y_j \rangle = 0$ when $E_i \neq E_j$, proving part (ii). □

For real-valued functions the complex conjugate is a needless complication and the inner-product may be written as

$$\langle y_i, y_j \rangle = \int_a^b y_i(x) y_j(x) w(x)\, dx. \tag{1.30}$$

Notice that the scale or normalization of the eigenfunctions is arbitrary. We can choose it such that the functions are not just orthogonal but *orthonormal*:

$$\langle y_i, y_j \rangle = \int_a^b y_i(x) y_j(x) w(x) dx = \delta_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases} \tag{1.31}$$

with $i, j \in \{0, 1, 2, \dots\}$. It follows that the eigenfunctions of a Sturm-Liouville problem can be used to form an orthonormal set of functions.

The symmetry relation also forms the starting point of the fundamental theorem on regular Sturm-Liouville problems:

Theorem 1.4. *For a regular Sturm-Liouville problem*

(i) The eigenvalues E_k are simple (i.e. there do not exist two linearly independent eigenfunctions with the same eigenvalue).

(ii) The E_k can be ordered as an increasing sequence tending to infinity,

$$E_0 < E_1 < E_2 < \dots \tag{1.32}$$

and with this labelling the eigenfunction $y_k(x)$ corresponding to E_k has exactly k zeros on the open interval (a, b).

(iii) The y_k form a complete orthogonal set of functions over (a, b) with respect to the inner product (1.25). *That is, any reasonable function f can be represented on (a, b) by its Fourier series with respect to the y_k,*

$$f(x) \sim \sum_{k=0}^{\infty} c_k y_k(x) \tag{1.33}$$

where

$$c_k = \langle f, y_k \rangle \,/\, \langle y_k, y_k \rangle \,. \tag{1.34}$$

For a proof of these results see e.g. [30, Chapter 7] or [126]. The integer k in part (ii) is called the *index* of the eigenvalue E_k.

1.3.3 Interlacing

It is a basic feature of the equation (1.17) that, when $p(x)$ and $Q(x) = Ew(x) - q(x)$ are both positive then solutions of the equation are generally of oscillating shape: according to Theorem 1.4 the eigenvalues are indexed by the number of oscillations, i.e. zeros, of their eigenfunctions. A classical result about the relative position of the zeros of different solutions is the *Sturm Comparison theorem*:

Theorem 1.5. Sturm Comparison Theorem *Let $y_i(x)$ be a nontrivial solution on (a, b) of*

$$(p_i(x)y')' + Q_i(x)y = 0$$

and $y_j(x)$ be a nontrivial solution on (a, b) of

$$(p_j(x)y')' + Q_j(x)y = 0$$

where $0 < p_j \leq p_i$ and $Q_j \geq Q_i$ on (a, b). Then (strictly) between any two zeros of y_i lies at least one zero of y_j except when y_j is a constant multiple of y_i. The latter implies $Q_i = Q_j$ and $p_i = p_j$ except possibly in intervals where $Q_i = Q_j = 0$.

For a proof see [30].

The most common application is to a Sturm-Liouville system with different eigenvalues E_k. Then $p_i = p_j$ and $Q_k(x) = E_k w(x) - q(x)$ and the theorem makes a comparison of the different eigenfunctions. We assume that $w(x) > 0$ and $E_j > E_i$. The zeros of the eigenfunction of E_j then lie between the zeros of the eigenfunction of E_i. This property is called *interlacing*. Colloquially we say that the higher eigenfunction is oscillating ‘more rapidly’ than the lower eigenfunction.

Example 1.3 A simple example is shown by the eigenfunctions of

$$-y''(x) = Ey(x), \;\; y(0) = y(\pi) = 0$$

of which the first four are shown in Figure 1.3. The eigenvalues are $E_k = (k+1)^2, k = 0, 1, \ldots$ and the eigenfunctions are $y_k = \sin((k+1)x)$ (see example 1.1). The theorem then tells us that the zeros of $y_j(x)$ lie between the zeros of $y_i(x)$ if $j > i$. If we consider each eigenfunction in turn, then y_0 only has the two zeros at its endpoints, y_1 has one additional zero which must be situated in between the two zeros of y_0. Analogously, y_2 has a zero between each zero of y_1 and y_3 has a zero between each zero of y_2, and so on. The eigenfunctions are thus becoming more oscillatory as the eigenvalue increases.

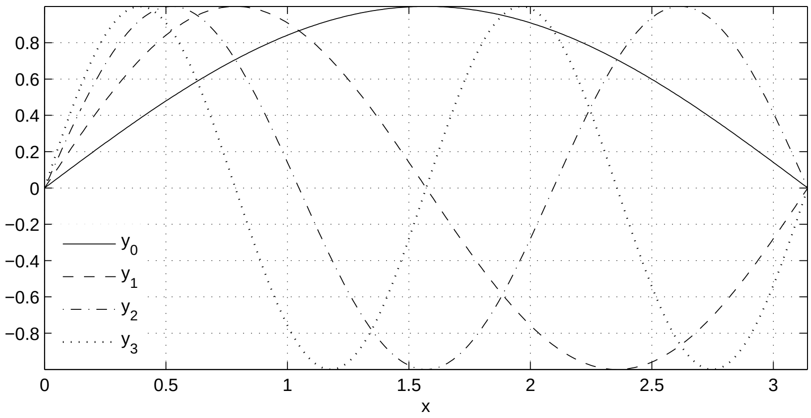


Figure 1.3: Illustration of the interlacing property: the first four eigenfunctions of $-y'' = Ey$.

1.4 Outline

In this first chapter we briefly outlined the subject of interest. We discussed the form and basic properties of the Sturm-Liouville problem and the special case of a Schrödinger problem. In the remaining chapters of the thesis, some specific techniques are considered and implemented for the numerical solution of a Sturm-Liouville or Schrödinger problem. These chapters are organized as follows.

Chapter two discusses into more detail what is meant with the numerical solution of a Sturm-Liouville problem or a Schrödinger problem. Some basic methods and techniques to compute the eigenvalues will be introduced such as the Prüfer transformation, the shooting method and the principle of coefficient approximation.

Chapter three introduces the Constant (reference potential) Perturbation Methods (or CPM in short). These methods were specially devised for the Schrödinger problem by Ixaru and co-workers and are based on a piecewise constant approximation of the potential function. Using a perturbative approach, methods of high order can be constructed. Our contribution exists in the extension of the CPM$\{12,10\}$ algorithm to higher order. This CPM$\{12,10\}$ algorithm was used as the basis for the code SLCPM12 [61] and is of order $\{12, 10\}$ (meaning order 12 at low energies and order 10 at high energies). In the new algorithm the orders $\{14, 12\}$, $\{16, 14\}$ and $\{18, 16\}$ are introduced. Just as the CPM$\{12,10\}$ method, these new higher order algorithms can be used in a shooting procedure to compute the eigenvalues efficiently and accurately. The construction and use of the higher order CPM algorithms was published in [75].

Chapter four examines the Line (reference potential) Perturbation Methods (or LPM in short) and its specific difficulties. For these methods the potential function of the Schrödinger equation is approximated by a piecewise line. We use the approach which was already introduced by Ixaru in [58] to effectively construct a LPM algorithm of order ten. Hereto we compute expressions for the first order and second order corrections. In addition we propose some asymptotic formulae which should be used in order to avoid loss in accuracy due to near-cancellations of like-terms. These results were published in [74].

Chapter five adresses systems of coupled equations. The generalization of the CPM to systems of coupled Schrödinger equations is introduced. It is shown that these generalized CPM preserve the important properties of the one-dimensional CPM discussed in chapter 3. The construction of the generalized CPM was published in [77]. In [78] and [79] we discussed the computation of the eigenvalues of the associated boundary value problem. Using the generalized CPM in a shooting process, the eigenvalues are computed accurately. Atkinson's matrix generalization of the Prüfer transformation is used to improve this shooting procedure.

Chapter six deals with some *singular* Sturm-Liouville problems. A singular problem occurs when at least one of the coefficients p^{-1}, q, w is not integrable up to the endpoint (i.e. is unbounded in a severe way) or if one or both of the endpoints is infinite. These singular problems present particular difficulties both in the determination of well-posed problems and in the numerical calculation of the eigenvalues. We discuss an interval truncation procedure for problems defined on an infinite integration interval. We also consider the important class of radial Schrödinger equations for which an improved truncation algorithm is proposed. Also the algorithm which is applied to deal with the singularity of the radial Schrödinger equations in the origin is explained. This work was published in [73].

Chapter seven discusses the MATSLISE package. We presented an earlier version of MATSLISE in [76]. MATSLISE is a MATLAB package implementing the one-dimensional CPM and LPM algorithms discussed in chapter 3 and 4. Also the truncation algorithms presented in chapter 6 are included. On top of this MATSLISE package a graphical user interface is built, which makes the package more user-friendly and easy to use. We briefly discuss the structure and use of MATSLISE and illustrate this with some examples.

Chapter eight concludes this thesis, summarizes contributions and achievements.

Chapter 2

Numerical solution of the Sturm-Liouville problem

The determination of the eigenvalues of Sturm-Liouville problems is of great interest in mathematics and its applications. However most eigenvalue problems cannot be solved (or are difficult to solve) analytically, and computationally efficient approximation techniques are of great applicability. In this chapter we show that the *numerical* solution of (regular) Sturm-Liouville problems is not trivial. The challenges are to do this cheaply, especially when long runs of higher-order eigenvalues are required.

2.1 Computational methods for the Sturm-Liouville problem

Many numerical methods have been developed for the computation of eigenvalues and eigenfunctions of Sturm-Liouville boundary value problems. Two standard approaches to the numerical approximation of eigenvalues of a boundary value problem can be distinguished: *discretization* and *shooting*. Discretization methods (such as finite differences and finite elements) involve substantial arithmetic and the storage of large matrices. Moreover, the accuracy quickly deteriorates for the higher eigenvalues. Shooting methods require less storage and arithmetic, but usually they do not determine the index of the eigenvalue. For Sturm-Liouville problems, these difficulties are avoided by the Prüfer method, which is a shooting method based on oscillation. This Prüfer-based shooting method has been implemented by Bailey, Gordon and Shampine in the SLEIGN code [21] (and its successor SLEIGN2 [17]) and by Pryce in the NAG library code D02KDF. The Prüfer-based shooting methods have, however, some problems with stiffness when standard initial value solvers are used. This stiffness disappears when the Prüfer transformation is combined with coefficient approximation. The Pruess methods (implemented in the packages SLEDGE [101] and SL02F [87, 88]) combine a piecewise constant midpoint approximation with a Prüfer-based shooting method. Another class of methods using coefficient approximation are the Piecewise Perturbation Methods (PPM). These methods apply a perturbative approach to successively improve the solution of the approximating problem.

The discretization methods and the different shooting methods will be discussed in more detail in the different sections of this chapter.

2.2 Discretization methods

2.2.1 Simple matrix methods

We consider methods based on finite differences [67]. An equally spaced mesh is used

$$a = x_0 < x_1 < \cdots < x_n = b \tag{2.1}$$

where $x_i = a + ih$ with $h = (b-a)/n$. The finite difference methods typically lead to matrix eigenvalue problems. For instance, the simple centred difference approximation

$$-y_i'' \approx \frac{-y_{i-1} + 2y_i - y_{i+1}}{h^2}, \quad y_i = y(x_i) \tag{2.2}$$

leads to an algebraic matrix eigenvalue problem $\mathbf{A}Y = EY$ where $\mathbf{A}$ is symmetric tridiagonal (see [105]). Another method is derived by applying the Numerov method, leading to a generalized eigenproblem $\mathbf{A}Y = E\mathbf{B}Y$ where $\mathbf{A}, \mathbf{B}$ are tridiagonal matrices. This Numerov method

$$y_{i-1} - 2y_i + y_{i+1} = \frac{h^2}{12}(f_{i-1} + 10f_i + f_{i+1}), \quad f_i = f(x_i, y_i), \tag{2.3}$$

is a well-known method and is used to solve differential equations of the form $y'' = f(x,y)$.

An advantage of the finite difference methods is that they are very simple to set up, especially when we deal with regular problems defined on a finite interval and a uniform mesh. However these simple methods have their limits. They replace an infinite-dimensional problem by a matrix problem of a dimension related to the number of meshpoints n. As a consequence, they can only approximate a certain number of eigenvalues for a given n. Moreover, the quality of the kth eigenvalue deteriorates rapidly as k increases: the error in the kth eigenvalue on a mesh of size h is typically of the form $O(h^p k^q)$ [66].

As an illustration we use the Numerov method to solve a Sturm-Liouville problem in its Liouville normal form

$$-y'' + q(x)y = Ey, \tag{2.4}$$

with boundary conditions $y(a) = y(b) = 0$. Application of Numerov's method leads to the equations

$$y_{i-1} - 2y_i + y_{i+1} = \frac{h^2}{12}\left[(q(x_{i-1}) - E)y_{i-1} + 10(q(x_i) - E)y_i + (q(x_{i+1}) - E)y_{i+1}\right], \tag{2.5}$$

or equivalently

$$\begin{aligned}&\frac{-y_{i-1}+2y_i-y_{i+1}}{h^2}+(q(x_i)-E)y_i=\\&\qquad\frac{1}{12}\left[-(q(x_{i-1})-E)y_{i-1}+2(q(x_i)-E)y_i-(q(x_{i+1})-E)y_{i+1}\right],\end{aligned} \tag{2.6}$$

which with the boundary conditions $y_0 = y_n = 0$ leads to a generalized matrix eigenproblem:

$$\mathbf{A}Y = E\mathbf{B}Y \tag{2.7}$$

where

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} \tag{2.8}$$

and

$$\mathbf{A} = \frac{1}{h^2}\mathbf{M} + \mathbf{B}\mathbf{Q}, \quad \mathbf{B} = \mathbf{I} - \frac{1}{12}\mathbf{M} \tag{2.9}$$

with

$$\mathbf{M} = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & & \ddots & & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} q(x_1) & & & \\ & q(x_2) & & \\ & & \ddots & \\ & & & q(x_{n-1}) \end{pmatrix}. \tag{2.10}$$

The eigenvalues $E_1 < E_2 < E_3 < \ldots$ (in this section we label the eigenvalues from 1 upwards) and eigenfunctions of (2.4) can then be approximated by the eigenvalues and eigenvectors of the generalized matrix eigenvalue problem. It can be shown that the error in the kth eigenvalue is of the form $O(k^6h^4)$ (see [10]). This indicates that the error in the computed eigenvalue approximations increases rapidly with k. Table 2.1 shows the exact eigenvalues E_k and some computed estimates for the problem $-y'' = Ey$, $y(0) = y(\pi) = 0$ and various n.

The reason for the decreasing accuracy is that the finite difference methods are based on approximations of the eigenfunctions by (piecewise) polynomials. These approximations impair with increasing eigenvalue index k, since the eigenfunctions oscillate more rapidly as k increases. Some effort has been done in finding more uniformly valid approximations. For instance, Anderssen, Andrew, de Hoog and Paine constructed a simple correction technique [9, 11, 95, 97]. They showed in [97] that for the case of the simple centred difference discretization, higher eigenvalues can be computed more accurately by adding the errors for the null potential, which are explicitly known, to the corresponding eigenvalues of the discretized problem. The same correction technique was also applied to the Numerov method [12] and finite element methods [13, 89]. Vanden Berghe and De Meyer [121, 125] used another idea to improve the eigenvalue approximations. They

Table 2.1: Eigenvalues obtained with the Numerov method for $-y'' = Ey,\ y(0) = y(\pi) = 0$.

k	E_k	$n = 10$	$n = 20$	$n = 40$
1	1	0.99995926	0.99999746	0.99999984
2	4	3.99736290	3.99983702	3.99998984
3	9	8.96943979	8.99813471	8.99988417
4	16	15.8246732	15.9894516	15.9993481
5	25	24.3170841	24.9594385	24.9975078
6	36	33.9283646	35.8777592	35.9925389

approximate the solution (eigenfunctions) no longer by polynomials only but by a mixed interpolation function containing also trigonometric functions. In [122] a modified Numerov method is discussed which delivers more accurate eigenvalues than the classical one. Also the exponentially-fitted Numerov methods discussed in [124] show a less pronounced increase of the error with the eigenvalue index.

2.2.2 Variational methods

Another class of methods which has been applied to solve the Sturm-Liouville problem is formed by the variational methods [105] such as the finite element methods. These methods are based on variational principles and are not really advantegeous over the finite difference methods. The *Rayleigh-Ritz* method e.g. produces, like the finite difference methods, an approximating matrix eigenproblem and each discretization can only approximate a limited number of eigenvalues. In addition the accuracy of E_k deteriorates with k as fast as with finite differences.

2.3 Shooting methods

2.3.1 Basic idea

The shooting method is a method for solving a boundary value problem by reducing it to the solution of an initial value problem. The differential equation is solved as an initial value problem over the range $[a, b]$ for a succession of trial values of E which are adjusted till the boundary conditions at both ends can be satisfied at once, at which point we have an eigenvalue. The simplest shooting method 'shoots' from one endpoint to the other endpoint, e.g. from a to b. This means that one chooses initial conditions which satisfy the boundary condition (1.6) in a:

$$y(a) = -b_0, \quad p(a)y'(a) = a_0 \tag{2.11}$$

The boundary condition at b determines 'target' values; if the value of y matches the target, we have found an eigenvalue.

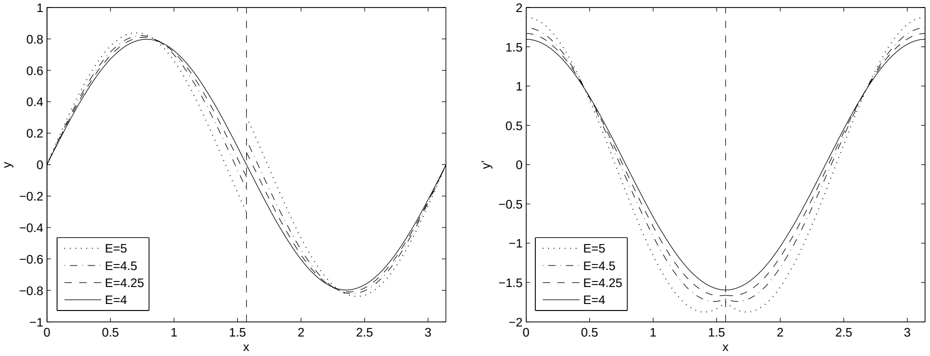


Figure 2.1: The shooting process for the sample problem $-y'' = Ey$ with $y(0) = y(\pi) = 0$. ***(Left)*** The left-hand solution y_L and right-hand solution y_R for different E values. ***(Right)*** The corresponding first derivatives: y'_L and y'_R.

Alternatively, one can shoot from two ends to some interior *matching point* $x_m \in [a, b]$. In this case we define a left-hand solution $y_L(x, E)$ and a right-hand solution $y_R(x, E)$. The left-hand solution is the solution of the initial value problem starting in a with initial conditions

$$y_L(a) = -b_0, \quad p(a)y'_L(a) = a_0, \tag{2.12}$$

while the right-hand solution satisfies the conditions

$$y_R(b) = -b_1, \quad p(b)y'_R(b) = a_1. \tag{2.13}$$

Example 2.1 Figure 2.1 illustrates the shooting process for the problem $-y'' = Ey$ with the boundary conditions $y(0) = y(\pi) = 0$. As seen in example 1.1 this problem has eigenvalues $1, 4, 9, 16, \ldots$. The matching point x_m is chosen in the middle of the integration interval. The left figure shows the left-hand solution $y_L(x, E)$ and right-hand solution $y_R(x, E)$ for different E-values: $E = 5, 4.5, 4.25, 4$. The corresponding first order derivatives are shown on the right figure. The left-hand solution starts in $a = 0$ and is propagated up to the matching point, while the right-hand solution originates in $b = \pi$ and goes down to the matching point. It is clear that only the solutions for $E = 4$ match in the matching point, that is $y_L(x_m, 4) = y_R(x_m, 4)$. However it is possible for the other E-values to rescale e.g. the right-hand solution such that $y_L(x_m, E) = y_R(x_m, E)$. But in this case, the obtained left- and right-hand first order derivatives no longer agree in the matching point. We can conclude that the criterion for E to be an eigenvalue is that the derivatives y' should match, as well as the values y. This criterion is captured in the mismatch function discussed below.

At the matching point we define a *mismatch* function $\phi(E)$. This mismatch function

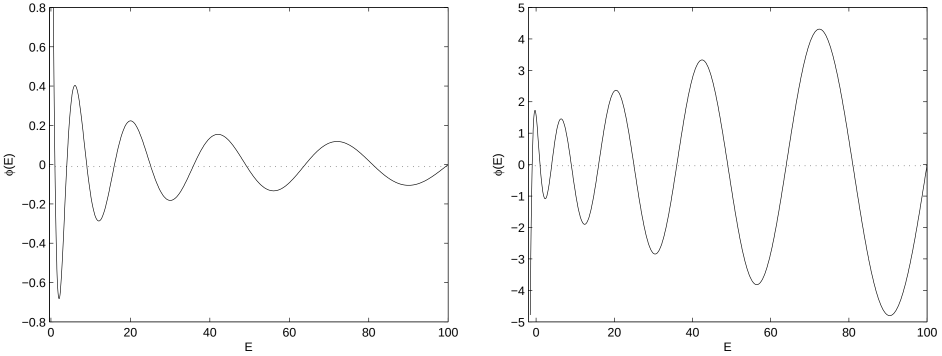


Figure 2.2: Mismatch function for the sample problem $-y'' = Ey$ with boundary conditions ***(Left)*** $y(0) = y(\pi) = 0$ and ***(Right)*** $y(0) + y'(0) = y(\pi) + y'(\pi) = 0$.

can e.g. be written as the determinant

$$\phi(E) = \begin{vmatrix} p(x_m)y_L'(x_m, E) & p(x_m)y_R'(x_m, E) \\ y_L(x_m, E) & y_R(x_m, E) \end{vmatrix}. \tag{2.14}$$

This mismatch function is only zero when E is an eigenvalue. Then y_R can be multiplied by a suitable scalar factor which makes it the continuation of y_L for $x \geq x_m$, producing an eigenfunction. Thus the procedure for finding the numerical value of an eigenvalue, consists in evaluating the mismatch function $\phi(E)$, numerically, and then through a finite series of iterations finding the value of E such that $\phi(E) = 0$ to the required degree of approximation. The usual iterative methods for finding the roots of a function may be employed here to find the zeros of $\phi(E)$.

Example 2.2 Figure 2.2 shows the shape of $\phi(E)$ for the problem $-y'' = Ey$ for the boundary conditions $y(0) = y(\pi) = 0$, which has zeros at $1, 4, 9, 16, \ldots$ (see example 1.1), and for the boundary conditions $y(0) + y'(0) = y(\pi) + y'(\pi) = 0$, which has an additional zero at $E = -1$.

2.3.2 Prüfer-based shooting methods

There are however some difficulties associated with the approach discussed above. The mismatch function $\phi(E)$ is always an oscillating function which makes the rootfinding process more difficult. Moreover, in order to converge on a specific eigenvalue, say the kth, one needs to enhance the algorithm, for instance by counting the zeros of the solution during the integration for each trial E value.

These difficulties can be avoided by using the *Prüfer transformation*. This technique first appeared in a 1923 paper [103] by H. Prüfer. There the change of variables was used to develop oscillation and comparison theorems. The Prüfer transformations reduce a Sturm-Liouville problem to an equivalent, nonlinear boundary value problem of first order. This leads to several useful numerical methods based on some form of the Prüfer transformation. Prüfer based shooting methods can be constructed where the counting of the zeros of $y(x)$ needed to compute the specific eigenvalue with a given index k is built in.

The main idea in the Prüfer method is to introduce polar coordinates (ρ, θ) in the phase plane. For the *simple* Prüfer transformation we take

$$y = \rho \sin\theta, \ \ py' = \rho \cos\theta \tag{2.15}$$

where $\rho = \rho(x; E)$ is called the *amplitude* and $\theta = \theta(x; E)$ is known as the *phase* or *Prüfer angle*. Differentiating (2.15) gives

$$y' = \rho' \sin\theta + \rho\,\theta' \cos\theta, \tag{2.16}$$

and

$$z' = \rho' \cos\theta - \rho\,\theta' \sin\theta, \tag{2.17}$$

where $z = py'$. We can write the Sturm-Liouville differential equation as two first-order equations

$$\begin{cases} y' = \dfrac{1}{p} z, \\ z' = (q - Ew)y. \end{cases} \tag{2.18}$$

and combine (2.16), (2.17) and (2.18) to solve the resulting simultaneous linear equations for ρ', θ'. We find that ρ and θ satisfy the equations

$$\theta' \ = \ \frac{1}{p}\cos^2\theta + (Ew - q)\sin^2\theta, \tag{2.19}$$

$$\frac{\rho'}{\rho} \ = \ \left(\frac{1}{p} - (Ew - q)\right)\sin\theta\cos\theta. \tag{2.20}$$

The Prüfer equations have the property that the eigenproblem is reduced to the solution of the θ equation. Once E and $\theta(x; E)$ are known, ρ can be determined by quadrature

$$\rho(x) = \rho(a) \exp \int_a^x \left[\frac{1}{p(t)} - (Ew(t) - q(t))\right] \sin\theta(t)\cos\theta(t) dt. \tag{2.21}$$

The regular boundary conditions (1.6) at a and b define the conditions

$$\theta(a) = \alpha, \quad \theta(b) = \beta, \tag{2.22}$$

where α and β are values of $\tan^{-1}(-b_0/a_0)$, $\tan^{-1}(-b_1/a_1)$ respectively. As we will see further, the precise value of α and β depends on the eigenvalue searched for (theorem 2.1).

The *scaled* Prüfer transformation is a generalization of the simple Prüfer method and is defined by the equations

$$y = S^{-1/2}\rho \sin\theta, \ \ py' = S^{1/2}\rho\cos\theta, \tag{2.23}$$

where S is a strictly positive scaling function chosen to give good numerical behaviour, and which in general depends upon both x and E. In [105] it it shown that the resulting differential equations for ρ and θ are then of the form

$$\begin{aligned} \theta' &= \frac{S}{p}\cos^2\theta + \frac{(Ew-q)}{S}\sin^2\theta + \frac{S'}{S}\sin\theta\cos\theta, && (2.24) \\ \frac{2\rho'}{\rho} &= \left(\frac{S}{p} - \frac{(Ew-q)}{S}\right)\sin 2\theta - \frac{S'}{S}\cos 2\theta. && (2.25) \end{aligned}$$

The regular boundary conditions (1.6) at a and b define the conditions for θ

$$\theta(a) = \alpha, \;\; \theta(b) = \beta, \tag{2.26}$$

where

$$\tan\alpha = -\frac{S(a)b_0}{a_0}, \;\; \tan\beta = -\frac{S(b)b_1}{a_1}. \tag{2.27}$$

These equations only determine α and β up to a multiple of π. The Prüfer feature is that each (appropriate) choice of this multiple specifies precisely one eigenvalue. The following theorem is proved in [105].

Theorem 2.1. *Consider the scaled Prüfer equations of a regular Sturm-Liouville problem whose coefficients p, q, w are piecewise continuous with $p > 0$, $q > 0$. Let the boundary values α and β satisfy the following normalization:*

$$\alpha \in [0,\pi), \quad \beta \in (0,\pi]. \tag{2.28}$$

Then the kth eigenvalue is the value of E giving a solution of (2.24) *satisfying*

$$\theta(a,E) = \alpha, \quad \theta(b,E) = \beta + k\pi. \tag{2.29}$$

The main point of this theorem is that the index k of the eigenvalue equals the number of zeros of the associated eigenfunction $y(x)$ on the open interval (a, b). If $\theta(x)$ is a multiple of π at a certain point $x = x_i$, then $\theta'(x_i) = S(x_i)/p(x_i) > 0$ by (2.24). This shows that θ increases through multiples of π as x increases, this means that θ can never be decreasing in a point $x = x_i$ where x_i is a multiple of π. Since $y = 0$ just when θ is a multiple of π, by (2.23), the number of zeros of y on (a, b) is just the number of multiples of π (strictly) between $\theta(a)$ and $\theta(b)$. The normalization (2.28) ensures that for any nonnegative integer k, there are precisely k multiples of π strictly between α and $\beta + k\pi$.

A shooting method can be defined for the θ equation. For simplicity we assume the scaling function has E-independent values in the matching point x_m and in a and b, which makes α and β in (2.28) also conveniently E-independent.

Theorem 2.2. *For any E, let $\theta_L(x;E)$ and $\theta_R(x;E)$ be the solutions of* (2.24) *satisfying*

$$\theta_L(a;E) = \alpha \in [0,\pi), \quad \theta_R(b;E) = \beta \in (0,\pi], \tag{2.30}$$

and define the scaled Prüfer mismatch function by

$$\phi(E) = \theta_L(x_m; E) - \theta_R(x_m; E). \tag{2.31}$$

Then

1. *The eigenvalue E_k is the unique value such that*

   $$\phi(E_k) = k\pi \tag{2.32}$$

   for $k = 0, 1, \ldots$.

2. *The function $\phi(E)$ is stricly increasing and differentiable on $(-\infty, \infty)$.*

For a proof, see [105].

Example 2.3 We look at the form of the Prüfer angle θ for a regular Schrödinger problem $-y'' + V(x)y = Ey$ defined by Paine in [97] with potential function

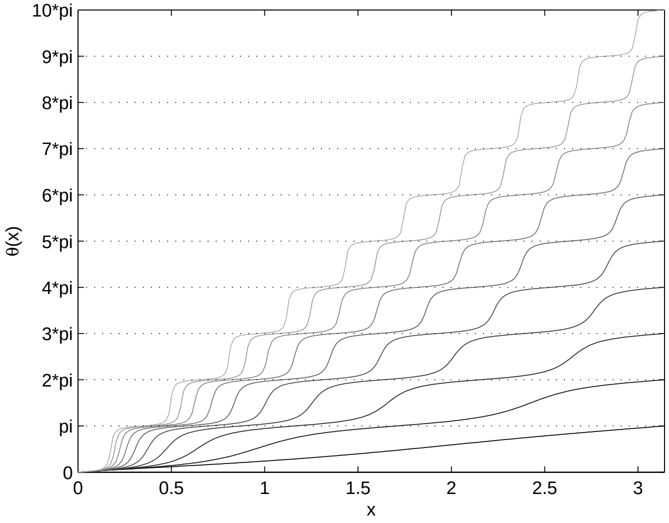
$$V(x) = \frac{1}{(x + 0.1)^2}. \tag{2.33}$$

The problem is defined over the interval $[0, \pi]$ and the boundary conditions are $y(0) = y(\pi) = 0$. The upper figure of Figure 2.3 shows the unscaled Prüfer θ ($S = 1$) for this problem for E running through the first 10 eigenvalues. As predicted by theorem 2.1 the different eigenvalues correspond with different multiples of π in b. Note that as E increases the unscaled Prüfer θ equation has solutions of gradually increasing 'staircase' shape with 'plateaus' at $\theta = k\pi$ and steep slopes around $\theta \approx (k - 1/2)\pi$. The integrator needs to react to the changes in slope of these steps, forcing it to drastically reduce the stepsize. A good choice of the scaling function S can smooth out the oscillatory behaviour, as shown in the lower figure.

Example 2.4 Figure 2.4 shows an example of the left hand (unscaled) Prüfer θ_L and right hand (unscaled) Prüfer θ_R appearing in the shooting method (see theorem 2.2). The upper figure is for the Paine problem (2.33) with matching point $x_m = 1.12$ and eigenvalue E_9. Another example is shown for the Coffey-Evans equation in the lower figure. The Coffey-Evans equation is one of the test problems which frequently appears in the literature (see e.g. [104, 105]). It is a regular Schrödinger equation with

$$V(x) = -2\beta\cos(2x) + \beta^2\sin^2(2x), \tag{2.34}$$

and $y(-\pi/2) = y(\pi/2) = 0$ as boundary conditions. Here the shooting was done for $\beta = 20$, $x_m = 0$ and $E = E_8$. For both problems it is clear that θ_L increases and θ_R decreases through multiples of π. It can also be seen that each pass through a multiple of π corresponds with a zero in the eigenfunction $y(x)$. For the Paine problem shown in the upper figure, $\theta_L(x_m; E_9) = 3.5\pi$ and $\theta_R(x_m; E_9) = -5.5\pi$ in the matching point. Thus the scaled Prüfer mismatch function $\phi(E_9)$ given by (2.31) is equal to 9π, as predicted by (2.32). For the Coffey-Evans problem shown in the lower figure, $\phi(E_8) = 4.5\pi + 3.5\pi = 8\pi$.



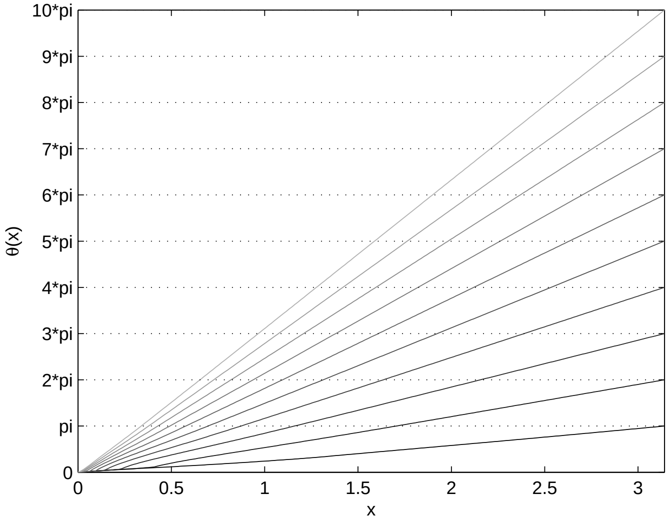
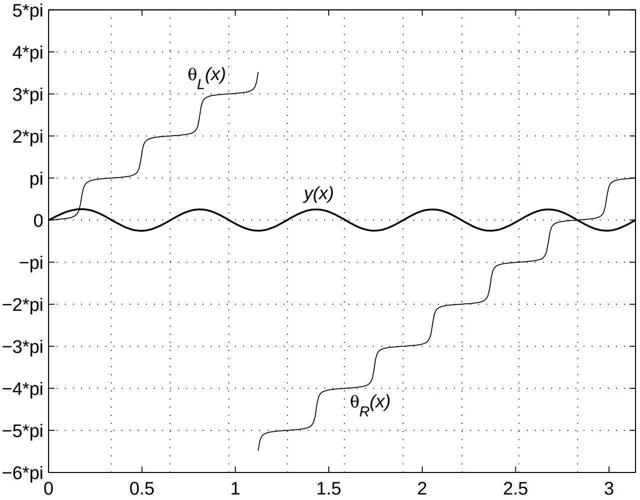


Figure 2.3: ***(Upper)*** The unscaled Prüfer function $\theta(x;E)$ for the Paine problem ($V(x) = 1/(x + 0.1)^2$), with E running through the first 10 eigenvalues. ***(Lower)*** The Prüfer function $\theta(x;E)$ for the same problem as above, using scale factor $S = 1$ where $E - V(x) \leq 1$ and $S = \sqrt{E - V(x)}$ where $E - V(x) > 1$.



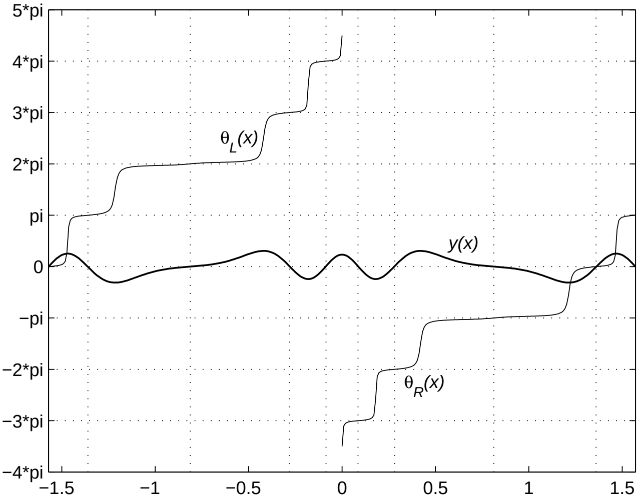


Figure 2.4: ***(Upper)*** The (unscaled) Prüfer $\theta_L(x; E_9)$ and $\theta_R(x; E_9)$ for the Paine problem ($x_m = 1.12$). ***(Lower)*** The (unscaled) Prüfer $\theta_L(x; E_8)$ and $\theta_R(x; E_8)$ for the Coffey-Evans problem ($\beta = 20$, $x_m = 0$).

Each value of the mismatch function $\phi(E)$ involves an integration of (2.24). The aim of the choice of the scaling function $S(x)$ is to reduce the cost of these integrations and allowing the code to take as large steps as possible. Choosing an appropriate scaling function $S(x)$ is however a non-trivial task. We consider first the case $Ew - q \gg 0$, which occurs typically when searching for eigenvalues with large indices. Here the stepsize for a standard library initial value problem solver is limited by the local accuracy requirement defined by some tolerance tol. If p, q and w are constant and $Ew - q > 0$ all numerical problems are eliminated by taking

$$S = \sqrt{(Ew - q)p}. \tag{2.35}$$

This reduces the θ equation simply to the trivial case

$$\theta' = \sqrt{\frac{Ew - q}{p}} = \text{constant}, \tag{2.36}$$

for which there is no stepsize restriction. In general, (2.24) can be written as

$$\begin{aligned} \theta' &= \frac{1}{2}\left[\frac{S}{p} + \frac{(Ew - q)}{S} + \left(\frac{S}{p} - \frac{(Ew - q)}{S}\right)\cos 2\theta + \frac{S'}{S}\sin 2\theta\right] \\ &= A + B\cos 2\theta + C\sin 2\theta, \end{aligned} \tag{2.37}$$

then the scaling function S should be chosen so that B and C are small and $A \approx \sqrt{(Ew - q)/p}$.

At the other extreme, if $Ew - q \ll 0$ the θ equation (2.24) becomes *stiff* and standard library methods need to take small steps to avoid instability (see [72] for more information on stiffness). A scalar differential equation $dy/dx = f(x, y)$ is considered stiff over a range if $\partial f/\partial y$ is large and negative in relation to the length L of the range, i.e. $-L\partial f/\partial y \gg 1$. If we write the right hand side of (2.24) as $F(x, \theta)$ and the right hand side of (2.25) as $G(x, \theta)$, then

$$\frac{\partial F}{\partial \theta} = \left[-\frac{S}{p} + \frac{Ew - q}{S}\right]\sin 2\theta + \frac{S'}{S}\cos 2\theta = -G(x, \theta). \tag{2.38}$$

This implies that it is precisely where ρ increases rapidly that we encounter stiffness in θ. Moreover in this case no choice of $S > 0$ can make $\partial F/\partial\theta$ small. The best one can do is to minimize $| - S/p + (Ew - q)/S|$ which in the constant case means taking

$$S = \sqrt{(q - Ew)p}. \tag{2.39}$$

Then $\partial F/\partial\theta$ is bounded above and below by $\pm 2\sqrt{(q - Ew)/p}$. However, near singular endpoints $(q - Ew)/p$ varies rapidly and even a stiff-ODE solver may be forced to take small stepsizes.

Several variants of the scaled Prüfer transformation have been developed and used in implementations of the shooting method for solving Sturm-Liouville problems (see [82]). Bailey developed a modified Prüfer method [15, 16]

$$y(x) = S^{-1/2}\rho(x)\sin\theta(x) \tag{2.40}$$

$$p(x)y'(x) = S^{1/2}\rho(x)\cos\theta(x). \tag{2.41}$$

where S is a scaling constant which is chosen by the rule $S = k\pi/U$ where k is the eigenvalue index and U is approximately the length of the interval on which $Ew - q$ is positive. This rule was implemented in the SLEIGN code [19, 21] from Sandia Laboratories. Pryce [104] implemented another scaled Prüfer substitution of the form (2.40) where S is a positive piecewise linear function chosen so that both $S/p - |q|/S$ and S'/S are kept small. This method is implemented in the NAG library as D02KDF and D02KEF. More details about e.g. the choice of the scaling function or the matching point used by these methods can also be found in [105].

Both SLEIGN and the NAG codes use an (explicit) Runge-Kutta method to integrate the θ equation. Such shooting methods based on standard initial value libraries often suffer from stepsize restriction when solving for large eigenvalues, or when the potentials are particularly large, and are not suited for computing a large set of eigenvalues. They also have some difficulties caused by stiffness of the θ equation (2.24) in a ‘barrier’ region where $(Ew-q)/p$ is large and negative. Because SLEIGN and D02KEF use Runge-Kutta integrators, stiffness causes very small stepsizes to be taken. In the next sections, we will see the advantages of combining a Prüfer formulation with *coefficient approximation*, in which the coefficient functions are approximated piecewisely by low degree polynomials (constants or lines). Then the integrations may be performed analytically and stiffness is no longer a problem.

2.3.3 Coefficient approximation methods

An important class of methods for the numerical solution of Sturm-Liouville problems is based on coefficient approximation. The basic idea here is to replace the coefficient functions $p(x)$, $q(x)$, $w(x)$ of the Sturm-Liouville equation piecewisely by low degree polynomials so that the resulting equation can be solved analytically.

The idea dates back at least to Gordon [41] and Canosa and De Oliveira [28] and was studied also by Ixaru [55], Paine and de Hoog [96] and Smooke [117]. But the standard reference for convergence in the piecewise polynomial case is due to Pruess [99, 100]. He examined the piecewise constant case and his strategy has been implemented by Pruess and Fulton in the code SLEDGE [101]. There is also another library code by Marletta and Pryce, called SL02F [87, 88]. Both codes use a so-called Pruess method to construct a shooting method which is able to compute a specific eigenvalue. The Pruess method will be discussed briefly in this section. In the next section the piecewise perturbation methods (PPM) will be discussed. These methods use the coefficient approximation in combination with a perturbative procedure which produces correction terms. This perturbative approach makes it possible to define methods of higher order.

The Pruess method

We review first briefly some of Pruess’s convergence results for coefficient approximation methods (see also [70]). Consider the approximation of a regular Sturm-Liouville problem

$$-(p(x)y'(x))' + q(x)y(x) = Ew(x)y(x), \quad x \in (a,b) \tag{2.42}$$

with

$$a_0 y(a) + b_0 p(a)y'(a) = 0, \quad a_1 y(b) + b_1 p(b)y'(b) = 0 \tag{2.43}$$

by another regular problem

$$-(\tilde{p}(x)\tilde{y}'(x))' + \tilde{q}(x)\tilde{y}(x) = E\tilde{w}(x)\tilde{y}(x), \quad x \in (a,b) \tag{2.44}$$

$$a_0 \tilde{y}(a) + b_0 \tilde{p}(a)\tilde{y}'(a) = 0, \quad a_1 \tilde{y}(b) + b_1 \tilde{p}(b)\tilde{y}'(b) = 0. \tag{2.45}$$

Here $\tilde{p}$, $\tilde{q}$ and $\tilde{w}$ are approximations to p, q and w, generally taken to be piecewise polynomials over a mesh $a = x_0 < x_1 < \cdots < x_n = b$. Both problems, being regular, have an infinite sequence of eigenvalues (E_k and $\tilde{E}_k$ respectively, $k = 0, 1, \ldots$) and associated eigenfunctions (y_k and $\tilde{y}_k$). The basic convergence result given by Pruess in [99] states that if p, q and w are in $C^{m+1}[a,b]$, then using piecewise polynomial interpolants of degree m will give convergence of the type

$$|E_k - \tilde{E}_k| \leq Ch^{m+1} \max(1, k^2), \tag{2.46}$$

where h is the maximum stepsize in the mesh and C is a constant independent of k. An enhanced convergence result (also given in [99]), states that if $\tilde{p}$, $\tilde{q}$ and $\tilde{w}$ interpolate to p, q and w at the Gauss points (see e.g. [109]) of each subinterval $[x_{i-1}, x_i]$ then (2.46) may be replaced by

$$|E_k - \tilde{E}_k| \leq C_k h^{2m+2}. \tag{2.47}$$

As pointed out in [96], the analysis of Pruess may be followed in detail to show that in (2.47) the constant C_k will grow with k:

$$C_k \leq Ck^{\max(3, 2m+2)}. \tag{2.48}$$

For piecewise constant approximations at the mesh centres (midpoints) $(x_{i-1} + x_i)/2$ (that is Gaussian interpolation for $m = 0$) this means

$$|E_k - \tilde{E}_k| \leq Ch^2 k^3. \tag{2.49}$$

Knowing the asymptotic behaviour of the eigenvalues $E_k \sim O(k^2)$ (see [131]), we obtain for large k (see [88, 105])

$$\frac{|E_k - \tilde{E}_k|}{\max(1, |E_k|)} \leq C(h\sqrt{k})^2. \tag{2.50}$$

Thus one would expect E_{100} to need ten times as many meshpoints to compute to a given relative tolerance than E_1. However, as mentioned in [88, 105], there are two reasons why this is not seen in practice. Firstly, (2.46) gives

$$|E_k - \tilde{E}_k| \leq Ch \max(1, k^2), \tag{2.51}$$

so the bound given by (2.50) cannot be tight for large k. Secondly, many problems occur in Liouville normal form (Schrödinger form) where $p = w = 1$ and for these there is an improved error bound (shown in [55, 96])

$$|E_k - \tilde{E}_k| \leq Ch^2 \sqrt{\max(1, E_k)}, \tag{2.52}$$

which implies for large k

$$\frac{|E_k - \tilde{E}_k|}{\max(1, |E_k|)} \leq Ch^2 k^{-1}. \tag{2.53}$$

Thus we can actually use larger h for large k for a given relative error. For this reason, Paine suggested that all problems be transformed to Liouville normal form before the Pruess approximation is applied. An idea which was however not incorporated in library software packages as SLEDGE or SL02F.

Accurate eigenvalue approximations are obtained by dividing each mesh interval into a number of equal parts and using Richardson extrapolation. Pruess [100] shows that the eigenvalue error is expandable in even powers of h when the mesh is uniform, and his analysis extends easily to nonuniform meshes.

Numerical solution of the approximating problem

As said before, the aim of using coefficient approximation is to obtain an approximating problem which can be integrated exactly. For the piecewise constant approximation in particular, the $\tilde{y}(x)$ of the approximating problem (2.44) can be integrated explicitly in terms of trigonometric and hyperbolic functions.

Let $\tilde{p}$, $\tilde{q}$, $\tilde{w}$ have the constant values $\tilde{p}_i$, $\tilde{q}_i$, $\tilde{w}_i$ in the ith interval (x_{i-1}, x_i), $i = 1, \dots, n$. In $[x_{i-1}, x_i]$ a solution of (2.44) has the form

$$\tilde{y}(x) = c_i F_i(x) + d_i G_i(x) \tag{2.54}$$

where F_i, G_i are fundamental solutions of $-\tilde{y}'' = \Lambda_i \tilde{y}$ and Λ_i is a constant

$$\Lambda_i = \frac{E\tilde{w}_i - \tilde{q}_i}{\tilde{p}_i}. \tag{2.55}$$

Take

$$\omega_i = \sqrt{|\Lambda_i|}. \tag{2.56}$$

Convenient definitions are then

$$F_i(x) = \begin{cases} \cos(\omega_i(x - x_{i-1})) & \Lambda_i \geq 0, \\ \cosh(\omega_i(x - x_{i-1})) & \Lambda_i < 0, \end{cases} \tag{2.57}$$

and

$$G_i(x) = \begin{cases} \dfrac{\sin(\omega_i(x - x_{i-1}))}{\tilde{p}_i \omega_i} & \Lambda_i > 0, \\ x - x_{i-1} & \Lambda_i = 0, \\ \dfrac{\sinh(\omega_i(x - x_{i-1}))}{\tilde{p}_i \omega_i} & \Lambda_i < 0. \end{cases} \tag{2.58}$$

Note that these functions also depend on E.

At meshpoints we have the *matching conditions* that $\tilde{y}$ and $\tilde{p}\tilde{y}'$ are continuous. The solution over $[x_{i-1}, x_i]$ is then advanced by the relation

$$\begin{bmatrix} \tilde{p}(x_i)\tilde{y}'(x_i) \\ \tilde{y}(x_i) \end{bmatrix} = T_i \begin{bmatrix} \tilde{p}(x_{i-1})\tilde{y}'(x_{i-1}) \\ \tilde{y}(x_{i-1}) \end{bmatrix} \tag{2.59}$$

where the propagation matrix (also called transfer matrix) T_i is given by

$$T_i = \begin{bmatrix} \tilde{p}_i G_i'(x_i) & \tilde{p}_i F_i'(x_i) \\ G_i(x_i) & F_i(x_i) \end{bmatrix} \tag{2.60}$$

When boundary conditions (2.45), equations (2.54) and their derivatives, and the matching conditions at the meshpoints x_i are used, we obtain a system of $4n+2$ linear equations between the $2n$ unknowns c_i, d_i and the $2n+2$ unknowns $\tilde{y}(x_i)$, $\tilde{p}(x_i)\tilde{y}'(x_i)$. The coefficients depend on E and eigenvalues are just the values of E for which this system is singular.

In [28] the $\tilde{y}(x_i)$, $\tilde{p}(x_i)\tilde{y}'(x_i)$ are eliminated to obtain an order $2n$ system for the c_i, d_i and zeros of the determinant are looked for. In [96] the c_i, d_i are eliminated producing an order $2n+2$ system consisting of equations (2.59) and the boundary conditions. Then the $\tilde{p}(x_i)\tilde{y}'(x_i)$ are eliminated between adjacent equations (2.59) to yield a tridiagonal system for $\tilde{y}(x_0), \ldots, \tilde{y}(x_n)$. These methods have the disadvantage that it is difficult to home in on the eigenvalue E_k for a specified index k. It is then more convenient to treat the equations as a method for explicitly integrating $(\tilde{y}, \tilde{p}\tilde{y}')$ over the x range, and to use a shooting method. This is done in both SLEDGE and SL02F and combined with the ideas based on the Prüfer substitution to be able to home in on a particular eigenvalue.

The SLEDGE algorithm uses the transfer matrix T_i to propagate the solution $(\tilde{p}\tilde{y}', \tilde{y})$. The zero count is kept during propagation by noting that if $\Lambda_i \leq 0$, $\tilde{y}$ has a (single) zero in (x_{i-1}, x_i) if and only if $\tilde{y}_{i-1}\tilde{y}_i < 0$, while if $\Lambda_i > 0$ the number of zeros equals the number of integers in the interval $(\theta/\pi, (\theta + \omega_i h_i)/\pi)$ where

$$\theta = \arctan \frac{\omega_i \tilde{y}_{i-1}}{\tilde{p}\tilde{y}'_{i-1}}, \tag{2.61}$$

is an implicit conversion to the Prüfer variable.

SL02F uses a bit different approach and applies an explicit scaled Prüfer transformation (2.23) of the form

$$\tilde{p}\tilde{y} = S^{1/2}\rho\cos\theta, \quad \tilde{y} = S^{-1/2}\rho\sin\theta, \tag{2.62}$$

where over each mesh interval (x_{i-1}, x_i) the scale factor S has the (positive) constant value S_i. Equations (2.24) and (2.25) then take the form

$$\frac{d\theta}{dx} = \frac{S}{\tilde{p}}\cos^2\theta + \frac{(E\tilde{w} - \tilde{q})}{S}\sin^2\theta, \tag{2.63}$$

$$\frac{d}{dx}\log\rho = \left(\frac{S}{\tilde{p}} - \frac{(E\tilde{w} - \tilde{q})}{S}\right)\sin 2\theta, \tag{2.64}$$

within each mesh interval.

The mismatch is then as in (2.31) given by

$$\phi(E) = \theta_L(x_m) - \theta_R(x_m) \tag{2.65}$$

where θ_L and θ_R are the solutions satisfying the left and right boundary conditions. The method used in SL02F to integrate (2.63) relies on judiciously choosing S on each subinterval, to make the change in θ easy to compute and is discussed into more detail in [105] and [87, 88].

At the meshpoints a rescaling formula is needed to compute the jumps in θ and ρ caused by the jumps in S. We will discuss the construction of these rescaling formulae hereafter.

Rescaling at jumps in S

At a meshpoint where S changes, θ and ρ need to be adjusted. Let S, θ, ρ be the old values and $\hat{S}, \hat{\theta}, \hat{\rho}$ the new ones. From (2.62) we see that

$$(\hat{\rho}\cos\hat{\theta}, \hat{\rho}\sin\hat{\theta}) = (\sigma^{-1/2}\rho\cos\theta, \sigma^{1/2}\rho\sin\theta), \tag{2.66}$$

where $\sigma = \hat{S}/S > 0$. In this form, given θ this defines $\hat{\theta}$ only up to a multiple of 2π. But, $(\cos\hat{\theta}, \sin\hat{\theta})$ and $(\cos\theta, \sin\theta)$ lie strictly in the same quadrant, so we get an unambiguous definition of $\hat{\theta}$ by imposing the condition $|\hat{\theta} - \theta| < \pi/2$. From (2.66), the following formula can then be derived (see [105]):

$$\hat{\theta} = \theta + \arg(1 + (\sigma - 1)\sin^2\theta, (\sigma - 1)\sin\theta\cos\theta). \tag{2.67}$$

Here $\arg(x, y)$ denotes the polar angle of the point (x, y), i.e. the argument of $x+iy$ in the range $-\pi < \arg \leq \pi$. The $\arg$ function in (2.67) can be coded in Fortran or MATLAB as

$$\mathrm{ATAN2}((\sigma - 1)\sin\theta\cos\theta, 1 + (\sigma - 1)\sin^2\theta). \tag{2.68}$$

The adjustment formula for ρ follows immediately from (2.66): the square of the formulae (2.66) gives

$$\left(\frac{\hat{\rho}}{\rho}\right)^2 = \frac{\cos^2\theta}{\sigma\cos^2\hat{\theta}}, \tag{2.69}$$

and

$$\left(\frac{\hat{\rho}}{\rho}\right)^2 = \frac{\sigma\sin^2\theta}{1 - \cos^2\hat{\theta}}. \tag{2.70}$$

Elimination of the $\cos^2\hat{\theta}$ term gives us then

$$\frac{\hat{\rho}}{\rho} = \left(\frac{\cos^2\theta}{\sigma} + \sigma\sin^2\theta\right)^{1/2}. \tag{2.71}$$

Advantages of the Pruess method

The convergence results already show that the Pruess methods have significant advantages over finite difference and variational methods. Another obvious benefit of Pruess methods (and coefficient approximation methods in general) is that they produce an approximating problem with, like the original, a (potentially) infinite spectrum, unlike matrix methods. Moreover the accuracy is maintained or even improves as k increases. In addition, for finite differences a uniform mesh is used, and this is rarely a good idea e.g. when dealing with (truncated) singular problems.

In comparison with shooting methods based on a standard initial-value solver, the following holds.

1. Pruess methods are of low order: $O(h^2)$ convergence for fixed k as $h \to 0$ if the constant midpoint approximation is used. But repeated h^2 extrapolation is valid for p, q and w sufficiently smooth, to give $O(h^4)$, $O(h^6), \ldots$ accuracy. Repeated extrapolation is the basic method of SLEDGE, whereas SL02F uses only one extrapolation for h^4 extrapolation (for reasons to do with the interface to the rest of the package).
2. The overall shooting process consists of a number of integrations with different values of E. Unlike a method based on a standard initial-value solver, it is practical with this method to fix the mesh and evaluate the coefficient midpoint values once for all before the start of the shooting process. This can give a big speed advantage.
3. Pruess methods are relatively unaffected by the stiffness/instability which can force a very small stepsize on an initial-value solver in regions where $q - Ew \gg 0$.

A drawback of the Pruess methods is the difficulty in obtaining higher order methods. It is usual to implement them using Richardson extrapolation. For the piecewise perturbation methods and integral series methods (both related to the Pruess methods) higher order methods can be constructed directly. Both classes of methods will be discussed next.

2.3.4 Piecewise perturbation methods

Linear second-order differential equations describe a lot of important physical phenomena and it is therefore not surprising that physicists contributed with their own special numerical techniques. The contribution of physicists was to make use of some ideas originating in mathematical physics. Such an idea is the *perturbation approximation*. Using this perturbative approach, methods of higher efficiency can be constructed, called the Piecewise (reference potential) Perturbation Methods (PPM). Some PPM are discussed in [58] for the general case of linear second-order differential equations.

As for the Pruess method, the original differential equation is replaced (piecewisely) by another differential equation (called the *reference equation* by Ixaru in [58]), which can be solved exactly. But now the perturbation theory is used, to estimate the deviation between the solution of the reference equation and the solution of the original equation.

Some perturbation corrections can then be added to the solution of the reference equation to obtain a more accurate approximation to the solution of the original equation.

The motivation of the perturbative approach lies in the fact that by taking the approximate solution as e.g. a piecewise zeroth order solution of the approximating problem plus a number of corrections to this zeroth order solution, we will obtain a more accurate approximation to y than if we had not included any perturbation corrections. This will manifest itself in a smaller number of mesh intervals needed to solve the problem numerically. Hence, providing the complexity of evaluating the perturbation corrections is not too prohibitive, the overall cost of the calculation will be reduced.

The PPM are identified by the type of piecewise approximation. For instance if the coefficients are approximated by piecewise constants the method is referred to as a *constant* perturbation method (CPM) while if piecewise lines are used the method is called a *line* perturbation method (LPM). CPM as well as LPM have been constructed for the Liouville normal (Schrödinger) form. We will discuss these methods in detail in the next chapters and give here the main ideas of the general piecewise perturbation approach.

The reference equation

We focus on the initial value problem for the one-dimensional regular Schrödinger equation,

$$y'' = (V(x) - E)y\,, \quad x \in [a,b]\,, \tag{2.72}$$

with given initial conditions in one of the endpoints, e.g.

$$y(a) = y_0, \quad y'(a) = y'_0\ . \tag{2.73}$$

The potential function $V(x)$ is supposed to be a well behaved (i.e. real, bounded and continuous) function and E, the energy, is a constant.

A partition of the integration interval $[a,b]$ is introduced

$$a = x_0 < x_1 < x_2 < \cdots < x_n = b. \tag{2.74}$$

This partition is in general non-equidistant. Let us focus on the current interval $I_k = [x_{i-1}, x_i]$ of steplength h_i. Our aim is to construct a piecewise perturbation algorithm which propagates the solution from one endpoint of this interval x_{i-1} to the other endpoint x_i. We introduce the variable $\delta = x - x_{i-1}$, $\delta \in [0, h_i]$ and denote generically $X = x_{i-1}$ and $h = h_i$. The local one-step problem is then

$$y''(X+\delta) = (V(X+\delta) - E)y(X+\delta)\,, \quad \delta \in [0,h] \tag{2.75}$$

with some known initial conditions $y(X) = \alpha$, $y'(X) = \beta$.

We consider two particular solutions of (2.75) $u(\delta)$ and $v(\delta)$ which satisfy the initial conditions

$$u(0) = 1,\ u'(0) = 0, \tag{2.76}$$

and

$$v(0) = 0,\ v'(0) = 1. \tag{2.77}$$

The functions u and v are linear independent and their wronskian

$$W(u,v) = uv' - u'v \tag{2.78}$$

is equal to 1. It follows that a solution of (2.75) has the form

$$y(X+\delta) = c_1 u(\delta) + c_2 v(\delta), \tag{2.79}$$

where c_1 and c_2 are two constants. From (2.76) and (2.77) we know that $c_1 = y(X)$ and $c_2 = y'(X)$. The solution of Eq. (2.75) can thus be written in matrix form as follows

$$\begin{bmatrix} y(X+\delta) \\ y'(X+\delta) \end{bmatrix} = \begin{bmatrix} u(\delta) & v(\delta) \\ u'(\delta) & v'(\delta) \end{bmatrix} \begin{bmatrix} y(X) \\ y'(X) \end{bmatrix}. \tag{2.80}$$

Taking the inverse of this formula, we obtain

$$\begin{bmatrix} y(X) \\ y'(X) \end{bmatrix} = \begin{bmatrix} v'(\delta) & -v(\delta) \\ -u'(\delta) & u(\delta) \end{bmatrix} \begin{bmatrix} y(X+\delta) \\ y'(X+\delta) \end{bmatrix}. \tag{2.81}$$

The role of the functions u and v is thus to propagate the (exact) solution from X to $X+\delta$ and vice versa. Therefore u and v are called exact *propagators*.

It is clear that the knowledge of the propagators u, v and their first derivatives u', v' is sufficient to advance the solutions in both directions. However, analytic forms of these u and v are known only for a restricted number of expressions for the function $V(x)$, let such functions be denoted by $\bar{V}(x)$. The idea behind the perturbation approach is to replace $V(x)$ piecewisely by a $\bar{V}(x)$. To further improve the accuracy, the corrections derived from the perturbation $\Delta V = V(x) - \bar{V}(x)$ are added (also piecewisely).

More concrete, we associate to Eq. (2.75) an equation of the same form

$$y''(X+\delta) = \left[\bar{V}(X+\delta) - E\right] y(X+\delta)\,, \quad \delta \in [0,h], \tag{2.82}$$

which is called the *reference equation*. The function $\bar{V}(x)$ (the so-called *reference potential*) is chosen is such a way that this equation has known analytic solutions. In particular we are interested in the two solutions $\bar{u}(\delta)$ and $\bar{v}(\delta)$ which are the propagators of Eq. (2.82). Our purpose is thus to construct the unknown propagators u and v of the original equation (2.75) in terms of the known reference propagators $\bar{u}$ and $\bar{v}$. Actually, the reference propagators form the zeroth order approximations of u and v and some perturbation corrections derived from the *perturbation*

$$\Delta V(\delta) = V(X+\delta) - \bar{V}(X+\delta) \tag{2.83}$$

will successively improve this approximation.

The perturbation corrections

As explained in [58] by Ixaru, the parameter dependent function $F(\delta;\gamma)$, $\gamma \in [0,1]$ can be introduced as

$$F(\delta;\gamma) = \bar{V}(\delta) + \gamma \Delta V(\delta). \tag{2.84}$$

This function reproduces the given potential function $V(\delta)$ and the reference potential $\bar{V}(\delta)$ when γ takes its extreme values, i.e. $F(\delta;1) = V(\delta)$ and $F(\delta;0) = \bar{V}(\delta)$. The propagators u, v and $\bar{u}$, $\bar{v}$ are the particular cases $\gamma = 1$ and $\gamma = 0$ of the propagators $u(\delta,\gamma)$ and $v(\delta,\gamma)$ of the differential equation

$$y''(X+\delta;\gamma) = [F(\delta;\gamma) - E]\, y(X+\delta;\gamma). \tag{2.85}$$

The propagators $u(\delta;\gamma)$ and $v(\delta;\gamma)$ are written as power series in the parameter γ:

$$u(\delta;\gamma) = \sum_{q=0}^{\infty} u_q(\delta)\gamma^q, \quad v(\delta;\gamma) = \sum_{q=0}^{\infty} v_q(\delta)\gamma^q. \tag{2.86}$$

We will denote the propagators generically by p, thus Eqs. (2.86) can be written as

$$p(\delta;\gamma) = \sum_{q=0}^{\infty} p_q(\delta)\gamma^q, \tag{2.87}$$

where $p = u$ if $p(0;\gamma) = 1$, $p'(0;\gamma) = 0$ and $p = v$ if $p(0;\gamma) = 0$, $p'(0;\gamma) = 1$.

To calculate p_q we introduce $p(\delta;\gamma)$ into Eq. (2.85),

$$p''(\delta;\gamma) = \big[(\bar{V}(\delta) - E) + \gamma\Delta V(\delta)\big]p(\delta;\gamma). \tag{2.88}$$

and organize the terms in powers of γ:

$$[p_0'' - (\bar{V}(\delta) - E)p_0] + \sum_{q=1}^{\infty} \gamma^q [p_q'' - (\bar{V}(\delta) - E)p_q - \Delta V(\delta)p_{q-1}] = 0. \tag{2.89}$$

Since this has to be satisfied for every $\gamma \in [0,1]$, the δ-dependent weights of γ^q must vanish for any $q = 0, 1, 2, \ldots$, i.e.

$$\begin{aligned} p_0'' &= (\bar{V}(\delta) - E)p_0, & (2.90)\\ p_q'' &= (\bar{V}(\delta) - E)p_q + \Delta V(\delta)p_{q-1}, \quad q = 1, 2, 3, \ldots. & (2.91) \end{aligned}$$

From Eq. (2.87) we can derive that $p(\delta;0) = p_0(\delta)$. On the other hand, we know that $p(\delta;0) = \bar{p}(\delta)$, so we get $p_0(\delta) = \bar{p}(\delta)$. And since the initial values for $p(\delta;\gamma)$ are the same as for $\bar{p}(\delta)$, the differences $p(0;\gamma) - p_0(0)$ and $p'(0;\gamma) - p_0'(0)$ must vanish:

$$\sum_{q=1}^{\infty} p_q(0)\gamma^q = 0, \quad \sum_{q=1}^{\infty} p_q'(0)\gamma^q = 0, \quad \gamma \in [0,1]. \tag{2.92}$$

This means that

$$p_q(0) = p_q'(0) = 0, \tag{2.93}$$

for any $q = 1, 2, 3, \ldots$.

In short, we obtained the following results:

Theorem 2.3. *The solution of Eq.* (2.75) *with the initial conditions* $y(X) = \alpha$ *and* $y'(X) = \beta$ *can be written as Eq.* (2.80) *where the propagators* u *and* v *are written as perturbation series*

$$p(\delta) = p_0(\delta) + p_1(\delta) + p_2(\delta) + p_3(\delta) + \dots \tag{2.94}$$

where p *stands for* u *or* v*. The zeroth order propagator* $p_0(\delta)$ *is exactly the reference propagator* $\bar{p}(\delta)$ *and the* qth *correction* $p_q(\delta)$, $q = 1, 2, 3, \dots$ *is the solution of the problem*

$$p_q'' = (\bar{V}(\delta) - E)p_q + \Delta V(\delta)p_{q-1}, \;\; p_q(0) = p_q'(0) = 0. \tag{2.95}$$

If the potential function is approximated by piecewise constants, the method is referred to as a *Constant* (reference potential) Perturbation Method (CPM in short) while if piecewise straight lines are used the method is called a *Line* Perturbation Method (or LPM). The CPM will be considered in detail in chapter 3. The LPM and its specific difficulties will be discussed in chapter 4. The use of polynomials of a degree higher than one is problematic in so much that the accurate computation of the two linear independent solutions u and v is difficult.

In the chapters 3 and 4, it will become clear that the advantages of the Pruess methods remain valid for the discussed PPM versions, that is (i) the integrations are performed analytically, so stiffness is not a problem, (ii) the mesh has to be computed only once and can be fixed before the start of the shooting process, moreover many information related to this mesh can be computed before the shooting and stored. (iii) an approximation to the kth eigenvalue can be found for a user-specified k rather than finding a range of eigenvalues (as a finite difference method might do).

Software packages

Also for the PPM a completely automatic software code can be constructed: the user has to specify only the information which defines the problem, plus a tolerance tol. The user does not have to set up a mesh or deal with other algorithmic inconveniences. In [61] the Fortran package SLCPM12 (available under the identifier ADJV_v1_0 in the CPC library [1]) was presented. This package uses the power of a high order CPM to solve regular Schrödinger and Sturm-Liouville problems. Later we implemented some higher order CPM versions in a MATLAB package, called MATSLISE [76].

Table 2.2 shows a comparison between different (Fortran) software packages applied on the regular Schrödinger problem with $V(x) = 1/(x + 0.1)^2$ on $[0, \pi]$ (the Paine problem of example 2.3). In [60] also some comparisons were done for other regular problems, but they lead to the same conclusion, namely that the CPM software has the power to outperform the other well-known Sturm-Liouville solvers when it comes to regular problems.

In the experiment shown in Table 2.2, the CPM code SLCPM12 is compared with the SLEDGE package [101] and the SLEIGN package [21] as they appear in the SLDRIVER code of Pryce [108]. All codes were run on a 2.4GHz PC. We asked to compute the first 21 eigenvalues at a user input tolerance 10^{-8}. Table 2.2 shows the exact eigenvalues E_k

Table 2.2: Comparison of different Sturm-Liouville solvers for the Paine problem ($tol = 10^{-8}$). $nfev$ is the number of function evaluations and T the CPU time in seconds.

		$\lvert\Delta E_k\rvert$		
k	E_k	SLCPM12	SLEDGE	SLEIGN
0	1.5198658210993471	$5.8(-14)$	$8.2(-9)$	$5.3(-9)$
4	26.7828631583287419	$2.4(-13)$	$1.4(-8)$	$9.2(-9)$
8	83.3389623741632420	$3.3(-13)$	$1.2(-7)$	$2.1(-8)$
12	171.6126448515666790	$3.1(-13)$	$2.4(-7)$	$1.8(-8)$
16	291.7629324611350560	$5.1(-13)$	$4.1(-7)$	$4.3(-8)$
20	443.8529598351504081	$1.1(-12)$	$8.7(-7)$	$9.9(-8)$
$nfev$		1080	34718	86089
T		0.03	0.07	0.5

and the (absolute value of the) errors ΔE_k in the obtained eigenvalue approximations. For shortness, the table contains only details for some selected eigenvalues and the notation $z(-q)$ is used for $z10^{-q}$. We also give the total number of function evaluations ($nfev$) of the potential function $V(x)$ required by each program to compute the whole set of eigenvalues and the associated CPU time T (in seconds).

The SLCPM12 code clearly needs much less function evaluations than the other codes. The main reason is the E-independent partition (mesh): the partition is only constructed once and then used in all eigenvalue computations. During the shooting process no extra function evaluations have to be performed. This is very different to the SLEIGN code where the computation of each eigenvalue is treated as a separate problem. Moreover the number of function evaluations increases dramatically with the eigenvalue. When asking the SLEIGN code to compute the first eigenvalue E_0, 2657 function evaluations are performed, while for the computation of the eigenvalue E_5 already 18414 function evaluations are needed. Another reason is the small number of meshpoints needed by the CPM code: only 15 steps were needed in the partition. The lower order method SLEDGE needs a lot more.

We can conclude that the SLCPM12 code is more efficient than the other codes, however we must add that SLEDGE and SLEIGN cover a wider range of problems than SLCPM12 (that is SLEDGE and SLEIGN are able to handle also some non-regular problems). The higher order CPM implemented in the MATLAB package MATSLISE are even more efficient (less meshpoints, less function evaluations) and MATSLISE also covers more problems than SLCPM12. MATSLISE comes with a Graphical User Interface (GUI) which makes the package more user-friendly than the Fortran packages, for which the access via a driver routine can be a non-trivial task. MATSLISE can be accessed and modified comparatively easily, but the language being an interpreter language also has inherent speed disadvantages when compared to the other (compiled) packages. The MATSLISE package, its structure and use will be discussed into more detail in chapter 7.

2.3.5 Magnus and Neumann series methods

A relatively new approach in the numerical solution of Sturm-Liouville eigenvalue problems is based on Magnus or Neumann expansions. These algorithms are related to the Pruess ideas, but provide high order approximations. We first introduce the Magnus and Neumann expansion and consider afterwards some applications to the Sturm-Liouville equation.

Neumann and Magnus expansions

There is an emerging family of numerical methods based on integral series representation of ODE solutions. We consider the linear differential equation

$$y' = A(t)y, \quad y(0) = y_0. \tag{2.96}$$

The simplest integral series is obtained by applying Picard iteration [44] to obtain the fundamental solution of the matrix linear ODE

$$\begin{aligned} y(t) = & y_0 \Big[1 + \int_0^t A(\tau)d\tau + \int_0^t A(\tau) \int_0^\tau A(\tau_1)d\tau_1 d\tau \\ & + \int_0^t A(\tau) \int_0^\tau A(\tau_1) \int_0^{\tau_1} A(\tau_2)d\tau_2 d\tau_1 d\tau + \ldots \Big] \end{aligned} \tag{2.97}$$

This series is known as the Feynman-Dyson path ordered exponential in quantum mechanics, in mathematics it is known as the *Neumann* series or Peano series.

The Magnus and Cayley expansions are two other examples. They are obtained by transforming Eq. (2.96) to the suitable Lie algebra and applying the Picard iteration to the transformed ODE. Details on both approaches can be found in [51]. The Cayley expansion is based on the Cayley transform while the Magnus expansion is based on the exponential map. The approach of Magnus [83] aims at writing the solution of Eq. (2.96) as

$$y(t) = \exp(\Omega(t))y_0 \tag{2.98}$$

where $\Omega(t)$ is a suitable matrix. The *Magnus expansion* says that

$$\begin{aligned} \Omega(t) = & \int_0^t A(\tau)d\tau - \frac{1}{2}\int_0^t \left[\int_0^\tau A(\tau_1)d\tau_1, A(\tau) \right] d\tau \\ & + \frac{1}{4}\int_0^t \left[\int_0^\tau \left[\int_0^{\tau_1} A(\tau_2)d\tau_2, A(\tau_1) \right] d\tau_1, A(\tau) \right] d\tau \\ & + \frac{1}{12}\int_0^t \left[\int_0^\tau A(\tau_1)d\tau_1, \left[\int_0^\tau A(\tau_2)d\tau_2, A(\tau) \right] \right] d\tau + \ldots \end{aligned} \tag{2.99}$$

where $[\cdot,\cdot]$ denotes the matrix commutator defined by $[X,Y] = XY - YX$.

Numerical methods based on this expansion are reviewed by Iserles et al. [51]. They are of the form

$$y_{n+1} = \exp(\Omega_n)y_n \tag{2.100}$$

to give an approximation to $y(t_{n+1})$ at $t_{n+1} = t_n + h$. Here Ω_n is a suitable approximation of $\Omega(h)$ given by (2.99), with $A(t_n + \tau)$ instead of $A(\tau)$. This approximation involves first truncating the expansion, and second approximating the integrals.

The Magnus expansion converges in the Euclidean 2-norm provided (see [51])

$$\int_0^t \|A(\tau)\|\, d\tau < \frac{r_0}{\nu} \tag{2.101}$$

where

$$r_0 = \int_0^{2\pi} \left(2 + \frac{1}{2}\tau(1 - \cot(\frac{1}{2}\tau))\right)^{-1} d\tau = 2.173737\ldots \tag{2.102}$$

and $\nu \leq 2$ is the smallest constant such that

$$\|[A_1, A_2]\| \leq \nu \|A_1\| \, \|A_2\|, \tag{2.103}$$

for any two elements A_1 and A_2 in the underlying Lie algebra. Taking the crudest case $\nu = 2$ we get

$$\int_0^t \|A(\tau)\|\, d\tau < 1.08686\ldots \tag{2.104}$$

Hence, a Magnus-based integrator appears to have an inherent time-step restriction. Especially when $A(t)$ is non-oscillatory with large norm a small step size is forced. For differential equations with a highly oscillatory matrix $A(t)$ however, Magnus series numerical methods were shown to be very suitable. In such cases it is speculated that the convergence interval of the Magnus series will be much larger than predicted by (2.104), since integration of the norm of the matrix ignores the favourable effects of high oscillation in its entries. Moreover, Hochbruck and Lubich [46] showed that Magnus integrators perform well in situations where the stiffness of the system originates from the time-independent part of the coefficient matrix. Further, by factoring out the flow of the time-independent part of the coefficient matrix, Iserles [49] and Degani and Schiff [32] introduced a right correction Magnus series (RCMS) which has a uniform radius of convergence and uniformly bounded global errors as stiffness is increased (see further).

The numerical schemes based on the Magnus expansion received a lot of attention due to their preservation of Lie group symmetries (see [51, 52] and references therein). The Feynmann (Neumann) series does not respect Lie group structure but avoids the use of the matrix exponential. The use of Neumann series integrators has been proved successfull for certain large, highly oscillatory systems in [50].

Applying Magnus or Neumann Expansions to eigenvalue problems

Moan discusses in [90] an approach to the numerical solution of Sturm-Liouville eigenvalue problems based on Magnus expansions. A scheme is constructed in [90] for the Schrödinger equation

$$-y'' + V(x) = Ey, \tag{2.105}$$

which can be adapted to the general Sturm-Liouville problem. The Schrödinger problem is written as

$$\begin{aligned} Y'(x) &= \begin{pmatrix} 0 & 1 \\ V(x)-E & 0 \end{pmatrix} Y(x) = A(x,E)Y(x), \\ B_aY(a) &+ B_bY(b) = 0, \end{aligned} \tag{2.106}$$

where $B_a, B_b \in \mathbb{R}^{2\times 2}$ and $Y(x)^T = [y(x), y'(x)]$. The fundamental matrices Φ_a and Φ_b of the equation $Y'(x) = A(x,E)Y(x)$ are defined as the matrices satisfying the linear differential equations

$$\Phi_a'(x) = A(x,E)\Phi_a(x), \ \ \Phi_a(a) = I, \tag{2.107}$$

and

$$\Phi_b'(x) = A(x,E)\Phi_b(x), \ \ \Phi_b(b) = I. \tag{2.108}$$

The eigenvalues are determined using some iterative technique as the solution of

$$\phi(E) = \det[B_a\Phi_a(x_m) + B_b\Phi_b(x_m)] \tag{2.109}$$

with the matching point $a < x_m < b$. Approximations to Φ_a and Φ_b are obtained using the Magnus expansion (over each meshinterval). In fact the Magnus expansion is truncated after replacing A by a interpolating approximation $\tilde{A}$ (over each meshinterval) and evaluating the integrals. More details can be found in [90].

As mentioned in [90], poor approximations can be expected for large eigenvalues. The reason is the finite radius of convergence which implies a relation between the maximum allowable stepsize and the magnitude of the required eigenvalues. The deterioration in accuracy is improved by adding correction functions to the discrete Magnus expansions (see [90]). These asymptotic corrections are however rather complex and alternative approaches may be preferred. It is therefore suggested to use a different numerical scheme especially in the large E regime as the modified Magnus method [49] or a right correction Magnus series integrator (RCMS) [32]. Both methods use the same basic approach, namely application of a Magnus series integrator to the right correction equation. These RCMS form a subclass of the more general class of right correction integral series (RCIS) integrators.

The coefficient matrix $A(x,E)$ is decomposed into its natural constant and varying parts

$$A(x,E) = A_0(E) + A_1(x). \tag{2.110}$$

The part of the coefficient matrix responsible for the frequency oscillations, namely A_0 is thus isolated. The eigenvalues of A_0 are zero or purely imaginary scaling linearly with E. The RCIS integrators from Degani et al. [32] transform the original equation

$$y' = [A_0(E) + A_1(x)]y \tag{2.111}$$

to the right correction equation and approximate its solution by an integral series (e.g. Magnus for the RCMS). With the constant approximation $\bar{A}_1 = \int_{x_n}^{x_{n+1}} A_1(x)dx$ of A_1

on $[x_n, x_{n+1}]$, Eq. (2.111) can be written as

$$y' = [(A_0(E) + \bar{A}_1) + (A_1(x) - \bar{A}_1)]y, \quad x \in [x_n, x_{n+1}]. \tag{2.112}$$

The *right correction* u is defined by $y = zu$ where z is the fundamental solution of $z' = (A_0(E) + \bar{A}_1)z$. u is a solution of the *right correction equation* $u' = [z^{-1}(A_1(x) - \bar{A}_1)z]u$. The function $u(x)$, $x \in [x_n, x_{n+1}]$ is thus defined by the equation $y(x) = \exp(x[A_0(E) + \bar{A}_1])u(x)$ and so

$$u' = \left[\exp\left(-x[A_0(E) + \bar{A}_1]\right)\left(A_1(x) - \bar{A}_1\right)\exp\left(x[A_0(E) + \bar{A}_1]\right)\right] u. \tag{2.113}$$

The new coefficient matrix $[\exp(-x[A_0(E) + \bar{A}_1])(A_1(x) - \bar{A}_1)\exp(x[A_0(E) + \bar{A}_1])]$ is uniformly bounded in E, as are the rescaled solution u and correspondingly the radius of convergence of its Magnus series. The same conclusion can be drawn from the observation that $z(x) = \exp(x[A_0(E) + \bar{A}_1])$ is highly oscillatory as a function of x (since the eigenvalues of A_0 are pure imaginary eigenvalues which grow in absolute value as $E \to \infty$). Thus the matrix in (2.113) has entries which are highly oscillatory. Moreover the difference $(A_1(x) - \bar{A}_1)$ makes the norm of the matrix small. Therefore, integral series representations are ideal for the solution of (2.113). The RCMS methods apply an integrator based on the Magnus series to the right correction equation, whereas Moan applied a Magnus series integrator directly to (2.106) with piecewise polynomial $V(x)$. As said before, the latter is not recommended because of the large norm of the matrix in (2.106) when E is large.

PPM as right correction Neumann series

In [32] it is shown that the RCMS integrators, the modified Magnus method from [49], the piecewise perturbation methods (PPM) and the integrators for near adiabatic propagation in quantum dynamics discussed in [64] are examples of RCIS schemes. The first two use the Magnus series and the others use the Neumann series to integrate the right correction equation. In all such RCIS integrators, $V(x)$ is replaced by polynomial approximations and the resulting series terms are evaluated analytically. Taking a large number of terms very high order integrators are obtained.

To understand how a PPM can be seen as a right correction Neumann series, we consider the case of a constant reference potential $\bar{V}$ (CPM) for the Schrödinger problem which can then be written in the form

$$Y'(x) = \left[\begin{pmatrix} 0 & 1 \\ \bar{V} - E & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \Delta V(x) & 0 \end{pmatrix}\right] Y(x). \tag{2.114}$$

The fundamental solution Y on the ith interval is constructed as the limit of the series

$$Y(E, x, x_{i-1}) = P_0(x) + P_1(x) + P_2(x) + \dots, \quad x \in [x_{i-1}, x_i], \tag{2.115}$$

where

$$P_0 = \exp\left[(x - x_{i-1})\begin{pmatrix} 0 & 1 \\ \bar{V} - E & 0 \end{pmatrix}\right] \tag{2.116}$$

and for $q = 1, 2 \dots,$ (see (2.95))

$$P'_q = \begin{pmatrix} 0 & 1 \\ \bar{V} - E & 0 \end{pmatrix} P_q + \begin{pmatrix} 0 & 0 \\ \Delta V(x) & 0 \end{pmatrix} P_{q-1}, \quad P_q(x_{i-1}) = 0. \tag{2.117}$$

With

$$R(x) = -(x - x_{i-1}) \begin{pmatrix} 0 & 1 \\ \bar{V} - E & 0 \end{pmatrix}, \tag{2.118}$$

equation (2.117) can be written as

$$\frac{d}{dx} [\exp(R(x)) P_q(x)] = \exp(R(x)) \begin{pmatrix} 0 & 0 \\ \Delta V(x) & 0 \end{pmatrix} P_{q-1} \tag{2.119}$$

or equivalently

$$P_q(x) = \exp(-R(x)) \int_{x_{i-1}}^{x} \exp(R(s)) \begin{pmatrix} 0 & 0 \\ \Delta V(s) & 0 \end{pmatrix} P_{q-1}(s) ds \tag{2.120}$$

or

$$P_q(x) = P_0(x) \int_{x_{i-1}}^{x} P_0^{-1}(s) \begin{pmatrix} 0 & 0 \\ \Delta V(s) & 0 \end{pmatrix} P_{q-1}(s) ds. \tag{2.121}$$

This means that the ‘correction matrices’ P_q can be written as

$$\begin{aligned} P_1(x) &= P_0(x) \int_{x_{i-1}}^{x} B_i(s_1) ds_1 \\ P_q(x) &= P_0(x) \int_{x_{i-1}}^{x} B_i(s_1) \int_{x_{i-1}}^{s_1} B_i(s_2) \dots \int_{x_{i-1}}^{s_{q-1}} B_i(s_q)\, ds_q \dots ds_2\, ds_1, \end{aligned} \tag{2.122}$$

where B_i is precisely the right correction equation matrix of coefficients for the right correction defined as

$$y(E, x, x_{i-1}) = \exp\left[(x - x_{i-1}) \begin{pmatrix} 0 & 1 \\ \bar{V} - E & 0 \end{pmatrix} \right] u_i(x), \quad x \in [x_{i-1}, x_i]. \tag{2.123}$$

The right correction equation is then

$$u'_i = B_i(x) u_i, \tag{2.124}$$

where

$$B_i(x) = \\ \exp\left[-(x - x_{i-1}) \begin{pmatrix} 0 & 1 \\ \bar{V} - E & 0 \end{pmatrix} \right] \begin{pmatrix} 0 & 0 \\ \Delta V(x) & 0 \end{pmatrix} \exp\left[(x - x_{i-1}) \begin{pmatrix} 0 & 1 \\ \bar{V} - E & 0 \end{pmatrix} \right].$$

Thus the CPM approach may be viewed as a Neumann series applied to the right correction equation (2.124).

2.4 Conclusion

The discretization methods (both variational and finite difference methods) reduce the Sturm-Liouville problem to a matrix eigenproblem. There have been many advances in such methods, especially the correction formulae of Paine, de Hoog, Andrew and Anderssen. However, they inherently produce an approximating problem with finite spectrum, and the accuracy falls off with the increasing index of the eigenvalue. Another drawback of many of these methods is that one cannot proceed directly to the computation of a particular eigenvalue, the prior computation of all the preceding eigenvalues is necessarily required.

The Prüfer methods are more complicated but give good accuracy, and can calculate any specified eigenvalue without consideration of other eigenvalues. When the Prüfer approach is combined with coefficient approximation larger stepsizes can be taken and the mesh has to be computed only once and is fixed before the shooting process. In addition these methods allow a variable mesh which makes them more suited for general-purpose software with an automatic mesh-selection and error control.

The next chapters of this thesis discuss the Piecewise Perturbation Methods (PPM), which is a class of methods specially devised for the Sturm-Liouville problem in Schrödinger form. These PPM are methods based on coefficient approximation which are applied in a shooting procedure to obtain accurate eigenvalue estimates. The coefficient approximation is improved by a perturbative approach which produces corrections to be added to the solution of the low order approximating problem.

The PPM can be placed in the general framework of right correction integral series integrators, covering a whole class of powerfull methods.

Chapter 3

Constant Perturbation Methods

As mentioned in the previous chapter, the Piecewise Perturbation Methods (PPM) are a class of methods specially devised for the numerical solution of the (regular) Schrödinger equation. In this chapter we will consider the PPM approach in more detail and construct the algorithm for the simplest case where the potential is approximated by a piecewise constant. This algorithm is then used as propagation method in a shooting procedure to compute eigenvalues of the boundary value problem. In addition it is shown how the algorithm devised for the Schrödinger equation can be extended to numerically solve regular Sturm-Liouville problems .

3.1 A Constant Perturbation Method for the Schrödinger equation

Let the potential function $V(x)$ be approximated by $\bar{V}$ which is a constant in each subinterval $[x_{i-1}, x_i]$ of the mesh $\pi : a = x_0 < x_1 < \cdots < x_n = b$. We then say that $\bar{V}$ is a piecewise constant approximation over the mesh π. The PPM obtained on this basis are referred to as forming the CPM (Constant Perturbation Method) family. Some early work in this direction was already described by Ixaru in [53, 54]. The CPM algorithm in the form we will discuss here, was already discussed in [58] in the context of more general linear second-order differential equations and has later been applied on Schrödinger and Sturm-Liouville problems in [60] and [61].

Example 3.1 Let us illustrate the piecewise constant approximation technique with an example. The Coffey-Evans equation is a regular Schrödinger equation $-y'' + V(x)y = Ey$ with

$$V(x) = -2\beta \cos(2x) + \beta^2 \sin^2 2x, \tag{3.1}$$

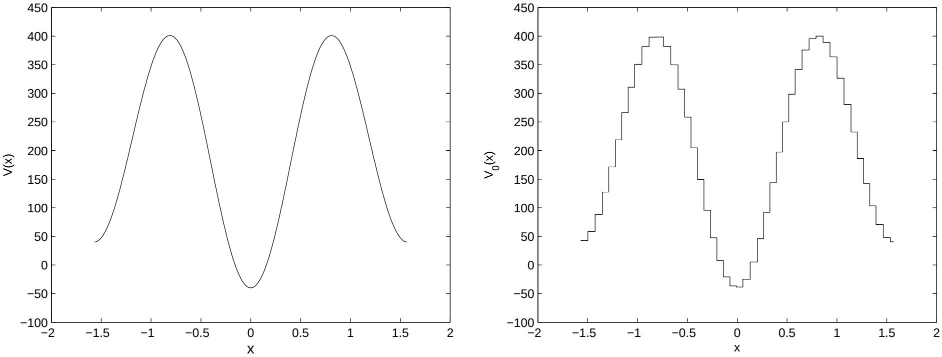


Figure 3.1: The potential $V(x)$ and a piecewise constant approximation $V_0(x)$ for the Coffey-Evans equation with $\beta = 20$.

and $y(-\pi/2) = y(\pi/2) = 0$ as boundary conditions. For $\beta = 20$ the potential $V(x)$ looks like Figure 3.1. This figure also shows a piecewise constant approximation of this potential function. This approximation is constructed by the CPM{12,10} method — which will be discussed later— at a user tolerance $tol = 10^{-12}$.

3.1.1 The reference equation

For a constant perturbation method, we use the reference equation

$$y''(\delta) = (\bar{V} - E)y(\delta), \quad \delta \in [0, h], \tag{3.2}$$

where $\bar{V}$ is a real constant. The general solution of the reference equation is

$$y(\delta) = c_1 \exp((\bar{V} - E)^{1/2}\delta) + c_2 \exp(-(\bar{V} - E)^{1/2}\delta), \tag{3.3}$$

with c_1 and c_2 arbitrary constants. The reference propagators $\bar{u}(\delta)$ and $\bar{v}(\delta)$ are two particular solutions which satisfy the intial conditions $\bar{u}(0) = 1, \bar{u}'(0) = 0$ and $\bar{v}(0) = 0, \bar{v}'(0) = 1$. For $\bar{u}$ this means that $c_1 = c_2 = 1/2$, while for $\bar{v}$ one obtains $c_1 = -c_2 = 1/2(\bar{V} - E)^{1/2}$. Thus, as Ixaru did in [58], we define the functions (note the correspondence with the functions F_i and G_i in section 2.3.3.)

$$\xi(Z) = \begin{cases} \cos(|Z|^{1/2}) & \text{if } Z \leq 0, \\ \cosh(Z^{1/2}) & \text{if } Z > 0, \end{cases} \tag{3.4}$$

$$\eta_0(Z) = \begin{cases} \sin(|Z|^{1/2})/|Z|^{1/2} & \text{if } Z < 0, \\ 1 & \text{if } Z = 0, \\ \sinh(Z^{1/2})/Z^{1/2} & \text{if } Z > 0. \end{cases} \tag{3.5}$$

Figure 3.2 shows the functions $\xi(Z)$ and $\eta_0(Z)$. For negative Z values the functions are oscillating, while for positive Z the functions increase exponentially.

We can then write the reference propagators as

$$\bar{u}(\delta) = \xi(Z(\delta)), \;\; \bar{v}(\delta) = \delta\eta_0(Z(\delta)), \tag{3.6}$$

where $Z(\delta) = (\bar{V} - E)\delta^2$. It is easy to show that the corresponding derivatives are

$$\bar{u}'(\delta) = Z(\delta)\eta_0(Z(\delta))/\delta, \;\; \bar{v}'(\delta) = \xi(Z(\delta)). \tag{3.7}$$

These reference propagators $\bar{u}, \bar{v}, \bar{u}'$ and $\bar{v}'$ form the zeroth order propagators in our perturbation method. As described in section 2.3.4, corrections of different order can be added in order to approximate the unknown propagators u and v more accurately. These corrections p_q, $q = 1, 2, \ldots$ ($p = u, v$) obey the equation

$$p_q'' = (\bar{V} - E)p_q + \Delta V(\delta)p_{q-1}, \;\; p_q(0) = p_q'(0) = 0. \tag{3.8}$$

3.1.2 The construction of the perturbation corrections

First we define some additional functions derived via the following recurrence relations:

$$\eta_1(Z) = [\xi(Z) - \eta_0(Z)]/Z, \tag{3.9}$$

$$\eta_m(Z) = [\eta_{m-2}(Z) - (2m-1)\eta_{m-1}(Z)]/Z, \;\; m = 2, 3, \ldots. \tag{3.10}$$

Note that each η_m function is a linear combination of the reference propagators $\bar{u} = \xi$ and $\bar{v} = \delta\eta_0$. The functions $\xi(Z), \eta_0(Z), \eta_1(Z), \ldots,$ were already described in [58] (and denoted there as $\bar{\xi}, \bar{\eta}_0, \bar{\eta}_1, \ldots$). They satisfy some basic properties which we will summarize here briefly.

1. **Series expansion**

$$\eta_m(Z) = 2^m \sum_{q=0}^{\infty} \frac{g_{mq} Z^q}{(2q + 2m + 1)!}, \tag{3.11}$$

with

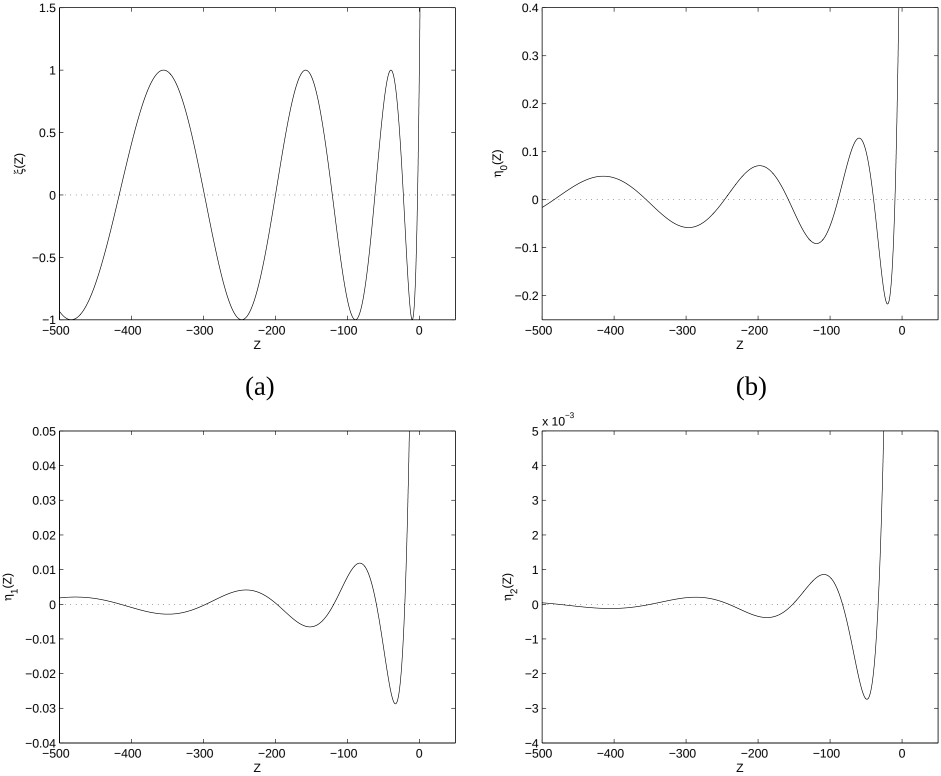
$$g_{mq} = \begin{cases} 1 & \text{if } m = 0, \\ (q+1)(q+2)\ldots(q+m) & \text{if } m > 0. \end{cases} \tag{3.12}$$

In particular

$$\eta_m(0) = \frac{1}{(2m+1)!!} = \frac{1}{(2m+1)(2m-1)(2m-3)\ldots 1}. \tag{3.13}$$

2. **Differentiation with respect to Z**

$$\xi'(Z) = \frac{1}{2}\eta_0(Z), \quad \eta_m'(Z) = \frac{1}{2}\eta_{m+1}(Z), \;\; m = 0, 1, 2, \ldots. \tag{3.14}$$



(a) (b)

(c) (d)

Figure 3.2: The basic functions for the constant perturbation method: $\xi(Z)$, $\eta_0(Z)$, $\eta_1(Z)$ and $\eta_2(Z)$.

3. **Differentiation with respect to δ**

$$\frac{\partial \xi(Z(\delta))}{\partial \delta} = Z(\delta)\eta_0(Z(\delta))/\delta, \quad \frac{\partial \delta \eta_0(Z(\delta))}{\partial \delta} = \xi(Z(\delta)), \tag{3.15}$$

$$\frac{\partial \delta^{2m+1}\eta_m(Z(\delta))}{\partial \delta} = \delta^{2m}\eta_{m-1}(Z(\delta)), \quad m = 0, 1, 2, \ldots. \tag{3.16}$$

The set of functions $\xi(Z), \eta_0(Z), \eta_1(Z), \ldots$ exhibits a certain hierarchy with respect to the numerical importance. As illustrated in Figure 3.2, $\xi(Z)$ is the largest member in the set, followed by $\eta_0(Z), \eta_1(Z), \eta_2(Z), \ldots$. For negative Z (i.e. $E > \bar{V}$) the function $\eta_m(Z)$ is an oscillating function whose amplitude damps out when $Z \to -\infty$. For positive Z however, all these functions increase exponentially with Z.

The functions ξ and η_m form a set of basic functions for the CPM and are used in the construction of the perturbation corrections. That is, to construct the qth correction p_q, we assume that the inhomogeneous term in (3.8) is a linear combination of the functions ξ and $\delta\eta_0, \delta^3\eta_1, ..., \delta^{2M+1}\eta_M$, i.e.,

$$\Delta V(\delta)p_{q-1}(\delta) = Q(\delta)\xi(Z(\delta)) + R_0(\delta)\delta\eta_0(Z(\delta)) + ... + R_M(\delta)\delta^{2M+1}\eta_M(Z(\delta)) \tag{3.17}$$

where $Q, R_0, R_1, ...$ are polynomials in δ. Now we search for p_q of the form

$$p_q(\delta) = \sum_{m=0}^{+\infty} C_m(\delta)\delta^{2m+1}\eta_m(Z(\delta)) \ , \tag{3.18}$$

and show that this sum has a finite number of terms and that the coefficients $C_m(\delta), m \geq 0$ are polynomials in δ. In fact, we differentiate (3.18) with respect to δ and use (3.15)-(3.16) to obtain:

$$\begin{aligned} p_q'(\delta) = C_0(\delta)\xi(Z(\delta)) + [C_0'(\delta) + \delta C_1(\delta)]\delta\eta_0(Z(\delta)) + \dots \\ +[C_m'(\delta) + \delta C_{m+1}(\delta)]\delta^{2m+1}\eta_m(Z(\delta)) + \dots \end{aligned} \tag{3.19}$$

Differentiating this again with respect to δ and using (3.15)-(3.16), one can construct an expression for $p_q''(\delta) - (\bar{V} - E)p_q(\delta)$:

$$\begin{aligned} p_q'' - (\bar{V} - E)p_q = 2C_0'\xi(Z(\delta)) + [C_0'' + 2\delta C_1' + 2C_1]\delta\eta_0(Z(\delta)) + \dots \\ +[C_m'' + 2\delta C_{m+1}' + 2(m+1)C_{m+1}]\delta^{2m+1}\eta_m(Z(\delta)) + \dots \end{aligned} \tag{3.20}$$

From (3.8) we know that this should be equal to $\Delta V(\delta)p_{q-1}$. Then, upon identifying the coefficients of $\xi, \eta_0, \eta_1, ...$ of the expressions (3.17) and (3.20), one gets

$$2C_0'(\delta) = Q(\delta) \tag{3.21}$$

$$C_m''(\delta) + 2[\delta C_{m+1}'(\delta) + (m+1)C_{m+1}(\delta)] = R_m(\delta), \quad m = 1, 2, \dots, M. \tag{3.22}$$

$$C_m''(\delta) + 2[\delta C_{m+1}'(\delta) + (m+1)C_{m+1}(\delta)] = 0, \quad m = M+1, \dots . \tag{3.23}$$

These equations can be solved iteratively; for C_0 the following formula is obtained

$$C_0(\delta) = \frac{1}{2}\int_0^{\delta} Q(\delta_1)d\delta_1, \tag{3.24}$$

while for $C_1(\delta), C_2(\delta), ...$ we get

$$C_m(\delta) = \frac{1}{2}\delta^{-m}\int_0^{\delta} \delta_1^{m-1}P_{m-1}(\delta_1)d\delta_1, \tag{3.25}$$

where

$$P_m(\delta) = \begin{cases} R_m(\delta) - C_m''(\delta) & \text{if } m = 0, 1, 2, ..., M \\ -C_m''(\delta) & \text{if } m = M+1, M+2, ... \end{cases} \tag{3.26}$$

Let us denote the degree of a polynomial P by $d(P)$. Eqs. (3.24) and (3.26) imply that $P_0(\delta)$ is a polynomial with maximal degree $d(P_0) = \max(d(R_0), d(C_0''))= \max(d(R_0), d(Q)-1)$, so that $C_1(\delta)$ wich results from (3.25) is a polynomial of the same degree as P_0. Also for higher m we can say that the degree of $C_m(\delta)$ is equal to the degree of $P_{m-1}(\delta)$: $d(P_{m-1}) = \max(d(R_{m-1}), d(C''_{m-1})) = \max(d(R_{m-1}), d(C_{m-1}) - 2)$ for $m \leq M+1$, but $d(P_{m-1}) = d(C''_{m-1}) = d(C_{m-1})-2$ for $m > M+1$. It follows that $d(C_{M+2}) = d(C_{M+1}) - 2, d(C_{M+3}) = d(C_{M+1}) - 4$ and so on. Thus, upon denoting the integer part of $\frac{1}{2}d(C_{M+1})$ by $\bar{M}$, it results that $C_m(\delta) = 0$ for any $m > M + \bar{M} + 1$, i.e. the last term in the sum (3.18) is $C_{M+\bar{M}+1}(\delta)\eta_{M+\bar{M}+1}(Z(\delta))$.

Upon this point we have shown that if $\Delta V(\delta)p_{q-1}(\delta)$ can be written as (3.17) then $p_q(\delta)$ results in the form (3.18) with a finite number of terms, and also that the coefficients are polynomials in δ which can be calculated by Eqs. (3.24)-(3.26). The only remaining question is whether the assumed form for (3.17) is valid. The answer is positive provided $V(\delta)$ is a polynomial in δ. In fact, for $q = 1$, the expression in (3.17) consists of a single term. This is the first term, with $Q(\delta) = \Delta V(\delta)$ (which is a polynomial as V is) for $p = u$, and the second term, with $R_0(\delta) = \Delta V(\delta)$ for $p = v$. This guarantees that $p_1(\delta)$ will be of the form (3.18). In turn, $\Delta V(\delta)p_1(\delta)$ will also be of the form (3.17), and so on.

Finally, we can summarize the previous in the following theorem.

Theorem 3.1. *If the potential function $V(\delta)$ is a polynomial in δ, then the qth correction p_q for the propagator p ($p = u, v$) is of the form*

$$p_q(\delta) = \sum_{m=0} C_m(\delta)\delta^{2m+1}\eta_m(Z(\delta)) \ , \tag{3.27}$$

$$p'_q(\delta) = C_0(\delta)\xi(Z(\delta)) + \sum_{m=0} [C'_m(\delta) + \delta C_{m+1}(\delta)]\delta^{2m+1}\eta_m(Z(\delta)) \tag{3.28}$$

with a finite number of terms. This means that the product $\Delta V(\delta)p_{q-1}(\delta)$ is of the form

$$\Delta V(\delta)p_{q-1}(\delta) = Q(\delta)\xi(Z(\delta)) + \sum_{m=0} R_m(\delta)\delta^{2m+1}\eta_m(Z(\delta)), \tag{3.29}$$

and the coefficients $C_0(\delta), C_1(\delta)$ are then polynomials in δ which are given by quadrature

$$C_0(\delta) = \frac{1}{2}\int_0^\delta Q(\delta_1)d\delta_1, \tag{3.30}$$

$$C_m(\delta) = \frac{1}{2}\delta^{-m}\int_0^\delta \delta_1^{m-1}[R_{m-1}(\delta_1) - C''_{m-1}(\delta_1)]d\delta_1, \ \ m = 1, 2, \ldots . \tag{3.31}$$

The starting functions in $\Delta V(\delta)u_0(\delta)$ are $Q(\delta) = \Delta V(\delta), R_0(\delta) = R_1(\delta) = \cdots = 0$, while for v_0 they are $Q(\delta) = 0, R_0(\delta) = \Delta V(\delta), R_1(\delta) = R_2(\delta) = \cdots = 0$.

It is interesting to know that each new correction p_q starts with a term of higher order than the previous one. In [58] the following result was obtained:

Proposition 3.2. *The first nonvanishing term in the series* (3.27) *is the term corresponding to η_{q-1} in u_q and to η_q in v_q. In series* (3.28) *it corresponds to ξ in u_1' to η_{q-2} in u_q' for $q \geq 2$, and to η_{q-1} in v_q'.*

Also important to remark is that the magnitude of the corrections typically decreases in magnitude with q.

3.1.3 A pilot reference equation

Theorem 3.1 assumes that $V(\delta)$ is a polynomial in δ. Also we can remark that the successive quadratures (3.30)–(3.31) are difficult to deal with when $\Delta V(\delta)$ starting from is not of the polynomial form. However, we want a procedure which is suitable for any well-behaved $V(X+\delta)$. This suggests to add an extra stage, in which $V(X+\delta)$ is approximated by $V^*(X+\delta)$, a polynomial in δ. As in [58], $V^*(X+\delta)$ is called the *pilot* reference function. The pilot reference potential $V^*(X+\delta)$ can be expressed as an expansion over a set of orthogonal polynomials. More exactly, we assume that $V(X+\delta)$ can be written as a series over shifted Legendre polynomials $P_n^*(\delta/h)$ in the following way:

$$V(X+\delta) = \sum_{n=0}^{+\infty} V_n h^n P_n^*(\delta/h)\ . \tag{3.32}$$

The shifted Legendre polynomials $P_n^*(z)$ are a set of functions defined on the interval $[0,1]$ (see [3]). They obey the orthogonality relationship

$$\int_0^1 P_i^*(z) P_j^*(z) dz = \frac{1}{2i+1}\delta_{ij}, \tag{3.33}$$

where δ_{ij} denotes the Kronecker delta. The first few are

$$\begin{aligned} P_0^*(z) &= 1,\\ P_1^*(z) &= 2z-1,\\ P_2^*(z) &= 6z^2-6z+1,\\ P_3^*(z) &= 20z^3-30z^2+12z-1. \end{aligned} \tag{3.34}$$

The original $V(X+\delta)$ is then approximated by the truncated series

$$V^*(X+\delta) = V^{(N)}(X+\delta) = \sum_{n=0}^{N} V_n h^n P_n^*(\delta/h)\ . \tag{3.35}$$

The option for shifted Legendre polynomials was proposed in [60] and is based on the fact that $V^{(N)}(X+\delta)$ represents the best approximation to V in $L^2(X, X+h)$ by a polynomial of degree $\leq N$. As will be shown further, the value of N can be chosen in such a way (i.e. high enough) that the pilot perturbation $V(X+\delta) - V^{(N)}(X+\delta)$ does not affect the accuracy of the method.

The equation

$$y^{(N)''} = (V^{(N)}(X+\delta) - E)y^{(N)}\,, \quad \delta \in [0,h] \tag{3.36}$$

is then the one whose propagators are actually constructed via CPM. With

$$\bar{V} = V_0, \quad \Delta V(\delta) = \Delta V^{(N)}(\delta) = \sum_{n=1}^{N} V_n h^n P_n^*(\delta/h)\;, \tag{3.37}$$

the integrals (3.30–3.31) can be solved analytically. Each $C_m(\delta)$ is a polynomial and the series (3.27) and (3.28) are finite.

The values V_n in (3.35) are determined using a least-squares procedure. This means that the quantity

$$I = \int_0^h \Big[V(X+\delta) - \sum_{n=0}^{N} V_n h^n P_n^*(\delta/h)\Big]^2 d\delta \tag{3.38}$$

is minimized. The minimum condition gives

$$\frac{\partial I}{\partial V_n} = -2\int_0^h \Big[V(X+\delta) - \sum_{i=0}^{N} V_i h^i P_i^*(\delta/h)\Big] h^n P_n^*(\delta/h) d\delta = 0 \tag{3.39}$$

and consequently

$$\sum_{i=0}^{N} V_i h^i \int_0^h P_i^*(\delta/h) P_n^*(\delta/h) d\delta = \int_0^h V(X+\delta) P_n^*(\delta/h) d\delta. \tag{3.40}$$

By taking into account relation (3.33), the following expressions for $V_0, V_1, \ldots$ result:

$$V_i = \frac{(2i+1)}{h^{i+1}} \int_0^h V(X+\delta) P_i^*(\delta/h)\, d\delta, \quad i = 0, 1, \ldots, N. \tag{3.41}$$

3.1.4 The CPM[N, Q] methods

The formulae in Theorem 3.1 allow us to obtain the analytic form of the corrections in a symbolic software package (Mathematica or Maple). Depending on the number of corrections and the degree of the pilot potential, different CPM versions can be formulated. Ixaru et al. introduced the notation CPM[N, Q] in [60] for a method with N the degree of the pilot $V^{(N)}(x)$ and Q the number of perturbation corrections retained in the algorithm. The simplest version, in which $V(x)$ is approximated by a piecewise constant but no correction is introduced, is thus identified as CPM[0,0]. In [58] also the notation CPM(0) was used for this version. The other versions described in [58] take $N = 2$ as a default value and $Q = 1, 2$. The CPM(0) method was shown to be a method of order two, while CPM[N, Q] in general is of order $2Q + 2$ provided N is sufficiently large.

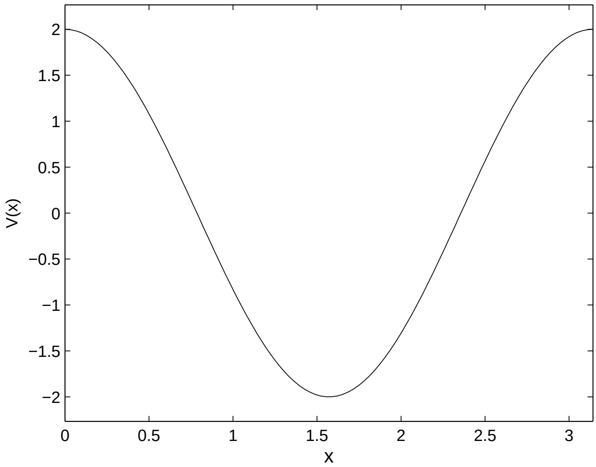


Figure 3.3: The potential function of the Mathieu problem

Example 3.2 Consider the Mathieu problem

$$y''(x) = [2\cos(2x) - E]y(x), \quad y(0) = y(\pi) = 0. \tag{3.42}$$

The potential function $V(x) = 2\cos(2x)$ is shown in figure 3.3. Table 3.1 shows some results for the Mathieu initial value problem with initial conditions in one endpoint of the integration interval: $y(0) = 0, y'(0) = 1$. The solution is propagated towards the other endpoint π for different E values. Since these E values are the exact eigenvalues E_k, $k = \{0, 2, 4, 6, 8, 10\}$ the obtained value for $y(\pi)$ should be zero. The CPM(0), CPM[2,1] and CPM[2,2] were used to propagate the solution on an equidistant mesh with stepsize $h = \pi/16$. It is clear that the accuracy increases with the number of corrections Q. It also seems that the propagation is more accurate for the higher eigenvalues.

Error analysis

A CPM[N, Q] method consists of two stages to be performed at each step. The first consists in the approximation of $V(X + \delta)$ by $V^{(N)}(X + \delta)$. This approximation causes the errors

$$\epsilon_i^{(N)} = \max\left\{|y(x_i) - y^{(N)}(x_i)|, |y'(x_i) - y^{(N)'}(x_i)|\right\}, \quad i = 1, 2, \ldots, n. \tag{3.43}$$

The second stage consists in solving (3.36) by the perturbation technique with Q corrections included. The associated errors are

$$\bar{\epsilon}_i^{[N,Q]} = \max\left\{|y^{(N)}(x_i) - \bar{y}(x_i)|, |y^{(N)'}(x_i) - \bar{y}'(x_i)|\right\}, \quad i = 1, 2, \ldots, n, \tag{3.44}$$

Table 3.1: Propagation of the solution for the Mathieu problem: the value of $y(\pi)$ computed with different CPM[N,Q] versions

k	E_k	CPM(0)	CPM[2,1]	CPM[2,2]
0	-0.11024881699209	0.051358056805	0.000286074277	0.000000143111
2	9.04773925980938	-0.000214352004	0.000019319866	-0.000000000463
4	25.02084082328977	-0.000035972251	0.000004091802	-0.000000000099
6	49.01041824942387	-0.000009344032	0.000001155772	-0.000000000025
8	81.00625032663258	-0.000003599759	0.000000169117	-0.000000000010
10	121.00416676126912	-0.000001808249	-0.000000171429	-0.000000000005

where $\bar{y}(x_i)$ and $\bar{y}'(x_i)$ are the numerical values obtained by propagating the solution along the interval by using CPM[N,Q]. The error of the whole procedure

$$\epsilon_i^{[N,Q]} = \max\left\{|y(x_i) - \bar{y}(x_i)|, |y'(x_i) - \bar{y}'(x_i)|\right\}, \quad i = 1, 2, \ldots, n, \tag{3.45}$$

is bounded by the sum of both errors, i.e.

$$\epsilon_i^{[N,Q]} \leq \bar{\epsilon}_i^{[N,Q]} + \epsilon_i^{(N)}. \tag{3.46}$$

In [60] it was shown that for each CPM[N,Q] a $\bar{h}$ exists such that

Theorem 3.3. *If CPM[N,Q] is applied to propagate the solution on an equidistant partition (mesh) with $h \leq \bar{h}$, then*

- *if the energy E is such that $|Z(h)|$ is small in all intervals, a constant C_N exists such that*
$$\epsilon_i^{[N,Q]} < C_N h^{2N+2}, \quad i = 1, 2, \ldots, n, \tag{3.47}$$
provided $Q \geq \left\lfloor \frac{2}{3}N \right\rfloor + 1$, $N = 1, 2, \ldots$ and $Q = 0$ for $N = 0$. The energy dependence of C_N is stronger and stronger as N increases.

- *if E is such that $Z(h) \ll 0$ in all intervals, an energy independent constant $\bar{C}_N^{\text{as}}$ exists such that*
$$\epsilon_i^{[N,Q]} < \bar{C}_N^{\text{as}} h^N / \sqrt{E}, \quad i = 1, 2, \ldots, n, \tag{3.48}$$
provided $Q \geq 1$ if $N = 1, 2, \ldots$, and $Q = 0$ if $N = 0$.

This theorem suggests that, for one and the same partition, the value of the energy E determines two different behaviours of the error. If E is close enough to $V(x)$, such that $|(Z(h)|$ is small in each interval of the partition, then the method behaves as a method of order $P_0 = 2N + 2$. However, when E is so high that $Z(h)$ is large and negative, the asymptotic order $P_{as} = N$ is valid. The theorem also says that there is a damping of the error when E is increased.

The existence of two distinct orders allows an alternative way of formulating and identifying a CPM version. It is possible to retain in the algorithm only the terms consistent with some input values for P_0 and P_{as}. This leads to a unique N (i.e. $N = P_{\text{as}}$)

but to a sum over incomplete perturbations. Such versions are denoted as CPM$\{P, N\}$ with $P = P_0$. The application of high order schemes involves significant analytic pre-calculation. This used to be a major obstacle but modern symbolic mathematics software made such calculations feasible. In [60] the version CPM$\{$12,10$\}$ was introduced and in [75] the construction of some higher order versions was discussed. These high order CPM$\{P, N\}$ schemes will be discussed in more detail in section 3.4.

3.2 Solving the boundary value problem using CPM

We consider now again the boundary value problem. As seen in chapter 2 a shooting procedure can be used to locate the eigenvalues of this boundary value problem. As integration method for the initial value problems appearing in this shooting procedure we can use a CPM algorithm.

3.2.1 A shooting procedure

The CPM are very well suited for the repeated solution of the initial value problems which appear in the shooting procedure. These initial value problems are solved for a fixed potential V but for different values of E. For a CPM, a mesh can be constructed which only depends on the potential and not on the energy E. This mesh is then computed only once and used in all eigenvalue computations (at least for regular problems). Moreover many data related to this mesh can be computed and stored once for all before the start of the shooting process. This means a big speed advantage for the CPM since the repeatedly asked task of integrating the equation at various values of E is completely separated from the time-consuming process of constructing a mesh and computing the data to be used later on. The construction of the mesh will be discussed in section 3.4.1 for the CPM$\{P, N\}$ schemes.

Algorithm 1 shows the basic shooting procedure in which the CPM propagation algorithm is used to propagate the left-hand and right-hand solutions.

It is clear that some points need to be examined further. In the next sections we discuss the form of the mismatch function and the choice of the matching point. Another refinement that needs to be added to this, concerns the counting of oscillations of the solution so as to home in on a particular eigenvalue.

3.2.2 The mismatch function

The criterion for a trial value for E to be an eigenvalue is that the derivatives y' should match in the matching point x_m, as well as the values. The matching condition is thus

$$\frac{y'_L(x_m)}{y_L(x_m)} = \frac{y'_R(x_m)}{y_R(x_m)}, \tag{3.49}$$

or equivalently

$$\phi(E) = \det \begin{pmatrix} y_L(x_m) & y_R(x_m) \\ y'_L(x_m) & y'_R(x_m) \end{pmatrix} = 0. \tag{3.50}$$

Algorithm 1 The shooting procedure

```
Choose a meshpoint $x_m$ ($0 \le m \le n$) as the matching point.
Set up initial values for $y_L, y'_L$ satisfying the BC at $a$ and initial values for $y_R, y'_R$ satisfying the BC at $b$. Choose a trial value for $E$.
repeat
    for $i = 1$ to $m$ do
        $\begin{bmatrix} y_L(x_i) \\ y'_L(x_i) \end{bmatrix} = \begin{bmatrix} u_i(\delta) & v_i(\delta) \\ u'_i(\delta) & v'_i(\delta) \end{bmatrix} \begin{bmatrix} y_L(x_{i-1}) \\ y'_L(x_{i-1}) \end{bmatrix}$
    end for
    for $i = n$ down to $m + 1$ do
        $\begin{bmatrix} y_R(x_{i-1}) \\ y'_R(x_{i-1}) \end{bmatrix} = \begin{bmatrix} v'_i(\delta) & -v_i(\delta) \\ -u'_i(\delta) & u_i(\delta) \end{bmatrix} \begin{bmatrix} y_R(x_i) \\ y'_R(x_i) \end{bmatrix}$
    end for
    Form a mismatch function $\phi(E)$ by comparing $y_L(x_m), y'_L(x_m)$ with $y_R(x_m), y'_R(x_m)$.
    Adjust $E$ to solve the equation $\phi(E) = 0$.
until $E$ sufficiently accurate
```

The mismatch function $\phi(E)$ is thus a function of the energy that is zero when the trial value of E is an eigenvalue.

If the trial value of E is not found to be an eigenvalue, e.g. when $|\phi(E)|$ is larger than some treshold value close to zero, the procedure is repeated with an adjusted value of E. It is possible to obtain a new E value simply by using one of the standard numerical procedures for finding a zero of a function. In [60] it was suggested to use a Newton-Raphson iteration procedure for the CPM:

$$E_{t+1} = E_t - \frac{\phi(E_t)}{\frac{d\phi}{dE}(E_t)}. \tag{3.51}$$

where

$$\phi(E) = y_L y'_R - y_R y'_L \tag{3.52}$$

and its derivative with respect to E;

$$\phi'(E) = y_{L_E} y'_R + y_L y'_{R_E} - y_{R_E} y'_L - y_R y'_{L_E}. \tag{3.53}$$

From the analytic theory of regular Sturm-Liouville problems, it is known that the zeros of ϕ are simple, hence Newton iterations converge quadratically for starting values near enough to a zero. The CPM algorithm allows a direct evaluation of the first derivatives of the solution with respect to the energy E. On differentiating (2.80) with respect to E one gets

$$\begin{bmatrix} y_E(X+\delta) \\ y'_E(X+\delta) \end{bmatrix} = \begin{bmatrix} u_E(\delta) & v_E(\delta) \\ u'_E(\delta) & v'_E(\delta) \end{bmatrix} \begin{bmatrix} y(X) \\ y'(X) \end{bmatrix} + \begin{bmatrix} u(\delta) & v(\delta) \\ u'(\delta) & v'(\delta) \end{bmatrix} \begin{bmatrix} y_E(X) \\ y'_E(X) \end{bmatrix}. \tag{3.54}$$

Since $[\xi(Z(h))]_E = -\frac{1}{2}\eta_0(Z(h))$, $[\eta_m(Z(h))]_E = -\frac{1}{2}h^2\eta_{m+1}(Z(h))$, $m = 0, 1, \ldots$ (see Eq. (3.14)), the expressions of u_E, v_E, u'_E and v'_E can be obtained upon replacing ξ by $-\frac{1}{2}h^2\eta_0$ and η_m by $-\frac{1}{2}h^2\eta_{m+1}$ in the obtained expressions for u, v, u' and v'. As initial conditions for the propagation of y_{L_E}, y'_{L_E} and y_{R_E}, y'_{R_E} one can use $y_{L_E}(a) = y'_{L_E}(a) = 0$ and $y_{R_E}(b) = y'_{R_E}(b) = 0$.

3.2.3 Choice of the matching point

The position of the matching point can considerably influence the efficiency of the method. The literature agrees that the matching point should generally be in the interior of the interval, away from singular endpoints, that is in the classically allowed region where $E > V(x)$ and not near one or both endpoints where the solutions show exponential behaviour. Therefore we take as matching point the meshpoint closest to the bottom of the potential, that is x_m is the rightmost meshpoint of the meshinterval corresponding to the lowest $\bar{V}$ value.

3.2.4 The Prüfer representation

When we have found a value for E such that $\phi(E) = 0$, we only know that we have found an eigenvalue. But we have no way of knowing if we have found the first, fifth or seventeenth eigenvalue. To obtain the index of the eigenvalue we must have an idea about the number of zeros in the corresponding solution. As discussed in the previous chapter, the Prüfer representation can help us count the number of zeros encountered during the propagation of the solution.

In [61] a procedure is described which allows to calculate the Prüfer angles θ_L and θ_R from shooting data. The procedure is very similar to the procedure used by SLEDGE. The solution of the Schrödinger equation is written in the scaled Prüfer variables ρ and θ in the following way (see section 2.3.2):

$$y(x) = S^{-1/2}\rho\sin\theta, \quad y'(x) = S^{1/2}\rho\cos\theta. \tag{3.55}$$

Both ρ and θ depend on x and E. We take as a global scaling function S:

$$S = \begin{cases} 1, & \text{if } E - \bar{V}_m < 1, \\ \omega_m = \sqrt{E - \bar{V}_m}, & \text{if } E - \bar{V}_m \geq 1, \end{cases} \tag{3.56}$$

where $\bar{V}_m$ is the constant approximation of $V(x)$ on the step in the partition whose rightmost end is the matching point x_m. The choice of this scaling function is based on the observations discussed in section 2.3.2, Eqs. (2.35)-(2.39).

Our purpose is to follow θ (θ_L or θ_R) during propagation. We consider the current interval $[x_{i-1}, x_i]$, $i = 1, 2, \ldots, n$, in which the constant approximation of $V(x)$ is $\bar{V}_i$. Suppose now θ is known in the endpoint x_{i-1} and we want to obtain $\theta(x_i)$. We distinguish two cases.

- $\mathbf{E} > \bar{\mathbf{V}}_\mathbf{i}$. In this (well) case, we take as local scaling factor $S_i = \omega_i = \sqrt{E - \bar{V}_i}$. The Prüfer phase θ_i over the interval $[x_{i-1}, x_i]$ is of the form

$$\theta_i(x) = \omega_i(x - x_{i-1}) + \varphi(x), \tag{3.57}$$

where $\varphi(x)$ is close to the constant value $\varphi(x_{i-1}) = \arctan(\omega_i y(x_{i-1})/y'(x_{i-1}))$. If it is assumed that $\varphi(x)$ remains unchanged over $[x_{i-1}, x_i]$, then the number of zeros of y in (x_{i-1}, x_i) is the number of integers in the interval $(\varphi(x_{i-1})/\pi, (\omega_i(x_i - x_{i-1}) + \varphi(x_{i-1}))/\pi)$, which is the procedure used by SLEDGE. Ixaru suggested in [61] to not just assume that $\varphi(x)$ remains unchanged but to add a correction in the phase. This means, that we assume that $\varphi(x)$ is of the form $\varphi(x_{i-1}) + \Delta\varphi(x)$ and thus write θ_i as

$$\theta_i(x) = \omega_i(x - x_{i-1}) + \varphi(x_{i-1}) + \Delta\varphi(x), \tag{3.58}$$

with $\Delta\varphi(x_{i-1}) = 0$. The value of $\Delta\varphi(x_i)$ is calculated using the available data $y(x_{i-1}), y'(x_{i-1}), y(x_i)$ and $y'(x_i)$. Specifically, we compute

$$\varphi^* = \arctan(\omega_i y(x_i)/y'(x_i)). \tag{3.59}$$

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

$$\Delta\varphi(x_i) = \begin{cases} \varphi^* - \bar{\varphi} + \pi, & \text{if } \varphi^* - \bar{\varphi} < -\pi/2, \\ \varphi^* - \bar{\varphi} - \pi, & \text{if } \varphi^* - \bar{\varphi} > \pi/2, \\ \varphi^* - \bar{\varphi}, & \text{otherwise.} \end{cases} \tag{3.60}$$

Once the values of θ_i at the two ends of the current step are known, the values of θ corresponding to the original global S, are easily obtained by the rescaling procedure already described in section 2.3.3.

- $\mathbf{E} \leq \bar{\mathbf{V}}_\mathbf{i}$. In this (barrier) case the values of θ corresponding to the original (global) S can be obtained directly. In fact

$$\theta(x_{i-1}) = \arctan(Sy(x_{i-1})/y'(x_{i-1})), \tag{3.61}$$

while to get $\theta(x_i)$ the value of $\theta^1 = \arctan(Sy(x_i)/y'(x_i))$ is computed separately and, if $y(x_{i-1})y(x_i) \geq 0$ then the number of zeros of y in (x_{i-1}, x_i) is zero and therefore we take

$$\theta(x_i) = \begin{cases} \theta^1 + \pi, & \text{if } \theta(x_{i-1}) > 0 \text{ and } \theta^1 < 0, \\ \theta^1 - \pi, & \text{if } \theta(x_{i-1}) < 0 \text{ and } \theta^1 > 0, \\ \theta^1, & \text{otherwise,} \end{cases} \tag{3.62}$$

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

$$\theta(x_i) = \begin{cases} \theta^1 + \pi, & \text{if } \theta(x_{i-1})\theta^1 > 0, \\ \theta^1, & \text{otherwise.} \end{cases} \tag{3.63}$$

In each of the two cases the important quantity is the one step increment of θ, $\Delta_i = \theta(x_i) - \theta(x_{i-1})$, since it allows constructing the global Prüfer phase in a simple way. In the shooting procedure, the solution is advanced in two directions, once forwards, from a to x_m and once backwards from b down to x_m. The values of $\theta(x_m)$ obtained from the two directions are given by

$$\theta_L(x_m) = \theta_a + \sum_{i=1}^{i_m} \Delta_i, \quad \theta_R(x_m) = \theta_b - \sum_{i=i_m+1}^{n} \Delta_i, \tag{3.64}$$

where θ_a and θ_b correspond to the values of θ_L and θ_R in a and b as determined by (2.30), i.e. $\theta_a \in [0, \pi)$, $\theta_b \in (0, \pi]$:

$$\theta_a = \begin{cases} \theta(a), & \text{if } \theta(a) \geq 0, \\ \theta(a) + \pi, & \text{if } \theta(a) < 0, \end{cases} \tag{3.65}$$

and

$$\theta_b = \begin{cases} \theta(b), & \text{if } \theta(b) > 0, \\ \theta(b) + \pi, & \text{if } \theta(b) \leq 0, \end{cases} \tag{3.66}$$

The quantity

$$\Delta\theta = \theta_L(x_m) - \theta_R(x_m) \tag{3.67}$$

allows identifying the eigenvalues (see Eq. (2.32)). If $\Delta\theta$ is regarded as a function of E then E_k is that E-value for which $\Delta\theta(E) = k\pi$.

3.2.5 Eigenvalue computation

Algorithm 2 shows the procedure followed to compute the eigenvalues in the user input range $[E_{\min}, E_{\max}]$. It uses the shooting algorithm 1 in combination with the scaled Prüfer variables to prevent any accidental jump over some eigenvalue during the search.

To locate the eigenvalues, the range $[E_{\min}, E_{\max}]$ is scanned for a set of test values of E. In the shooting procedure, for each E the solution of the Schrödinger equation is advanced in two directions, forwards from a to the matching point and backwards from b down to the matching point. The numerical method used to advance the solution is a CPM which produces not only y and y' at each meshpoint but also their derivatives with respect to E: y_E and y'_E. The y_E and y'_E are needed at each side of the matching point in the Newton iteration procedure (3.51) which is used to find the roots of the mismatch function $\phi(E)$ defined by (3.50). This Newton procedure is convergent only if the initial guess for E is sufficiently close to the eigenvalue E_k which has to be located. Therefore the eigenvalue computation consists of two stages. In the first stage (line 3 in algorithm 2), an energy interval $[E_{\text{low}}, E_{\text{up}}]$ is searched for such that each E in this interval is a good starting value for the Newton iteration. The second stage (lines 4-15) is then the Newton iteration process itself: the eigenvalue estimate is iteratively adjusted until the requested accuracy is achieved.

Algorithm 2 Computation of the eigenvalues between $E_{\min}$ and $E_{\max}$

```
Compute the lower and upper limits for the indices of the eigenvalues: $k_{\min} = \lfloor \Delta\theta(E_{\min})/\pi \rfloor$ and $k_{\max} = \lfloor \Delta\theta(E_{\max})/\pi - 1 \rfloor$.
for $k = k_{\min}$ to $k_{\max}$ do
    Find a good initial guess for $E_k$ to start the Newton iteration with.
    repeat
        for $i = 1$ to $m$ do
            $\begin{bmatrix} y^L(x_i) \\ y'^L(x_i) \end{bmatrix} = \begin{bmatrix} u_i(\delta) & v_i(\delta) \\ u_i'(\delta) & v_i'(\delta) \end{bmatrix} \begin{bmatrix} y^L(x_{i-1}) \\ y'^L(x_{i-1}) \end{bmatrix}$
            $\begin{bmatrix} y_E^L(x_i) \\ y'^L_E(x_i) \end{bmatrix} = \begin{bmatrix} u_{i_E}(\delta) & v_{i_E}(\delta) \\ u'_{i_E}(\delta) & v'_{i_E}(\delta) \end{bmatrix} \begin{bmatrix} y^L(x_{i-1}) \\ y'^L(x_{i-1}) \end{bmatrix} + \begin{bmatrix} u_i\delta) & v_i(\delta) \\ u_i'(\delta) & v_i'(\delta) \end{bmatrix} \begin{bmatrix} y_E^L(x_{i-1}) \\ y'^L_E(x_{i-1}) \end{bmatrix}$
        end for
        for $i = n$ down to $m + 1$ do
            $\begin{bmatrix} y^R(x_{i-1}) \\ y'^R(x_{i-1}) \end{bmatrix} = \begin{bmatrix} v_i'(\delta) & -v_i(\delta) \\ -u_i'(\delta) & u_i(\delta) \end{bmatrix} \begin{bmatrix} y^R(x_i) \\ y'^R(x_i) \end{bmatrix}$
            $\begin{bmatrix} y_E^R(x_{i-1}) \\ y'^R_E(x_{i-1}) \end{bmatrix} = \begin{bmatrix} v'_{i_E}(\delta) & -v_{i_E}(\delta) \\ -u'_{i_E}(\delta) & u_{i_E}(\delta) \end{bmatrix} \begin{bmatrix} y^R(x_i) \\ y'^R(x_i) \end{bmatrix}$
                $+ \begin{bmatrix} v_i'(\delta) & -v_i(\delta) \\ -u_i'(\delta) & u_i(\delta) \end{bmatrix} \begin{bmatrix} y_E^R(x_i) \\ y'^R_E(x_i) \end{bmatrix}$
        end for
        Compute the mismatch function $\phi(E)$ and its derivative $\phi'(E)$.
        Adjust $E$ using a Newton iteration: $E \leftarrow E - \phi(E)/\phi'(E)$.
    until $E$ sufficiently accurate, that is $|\phi(E)/\phi'(E)| < tol$.
    $E_k = E$
end for
```

The second stage was already discussed in 3.2.1, now we briefly explain the first stage. Algorithm 3 shows the algorithm which returns a suitable starting value E_c to be used as the initial value in the Newton procedure for locating E_k. First we look for E_{low} and E_{up} such that $E_{\text{low}} \leq E_k \leq E_{\text{up}}$. These E_{low} and E_{up} can be found with the aid of the scaled Prüfer representation discussed above: the quantity $\Delta\theta$ from (3.67) indicates where a certain E value is situated in the energy spectrum. The algorithm uses the function ζ defined as

$$\zeta(E) = \frac{\Delta\theta(E)}{\pi} - k. \tag{3.68}$$

Then we look for sharper lower and upper limits E_{low} and E_{up} and an approximate value E_c for E_k. E_c is calculated by alternative use of linear interpolation and of halving. When $|\zeta(E_{\text{low}})| + |\zeta(E_{\text{up}})| < 0.2$, we assume that the interval $[E_{\text{low}}, E_{\text{up}}]$ around E_k is sufficiently small and E_c is a good initial guess for E_k. The Newton iteration process is thus started with $E = E_c$. To continue the calculation for the next eigenvalue E_{k+1} a good starting value for E_{low} is then $E_{\text{low}} = E_{\text{up}}$.

Algorithm 3 Finding a starting value for the Newton iteration process

```
Find $E_{\text{low}}$ and $E_{\text{up}}$ such that $\zeta(E_{\text{low}}) \leq 0$ and $\zeta(E_{\text{up}}) \geq 0$.
$it = 0$
repeat
    if $it$ even then
        $E_c = \frac{E_{\text{low}} + E_{\text{up}}}{2}$
    else
        $E_c = \frac{\zeta(E_{\text{up}})E_{\text{low}} - \zeta(E_{\text{low}})E_{\text{up}}}{\zeta(E_{\text{up}}) - \zeta(E_{\text{low}})}$
    end if
    if $\zeta(E_{\text{low}})\zeta(E_c) \leq 0$ then
        $E_{\text{up}} = E_c$
    else
        $E_{\text{low}} = E_c$
    end if
    $it = it + 1$
until $|\zeta(E_{\text{low}})| + |\zeta(E_{\text{up}})| < 0.2$ and $E_{\text{low}} < E_c < E_{\text{up}}$
Take $E_c$ as initial approximation for $E_k$ to start the Newton iteration process with
```

3.3 The Sturm-Liouville problem

The CPM are constructed for equations of the Schrödinger form, not for equations of the Sturm-Liouville form. For this reason these methods can be applied to the Sturm-Liouville problems only if the Sturm-Liouville equation can be converted to the Schrödinger form. The conversion is possible and is achieved via the so-called *Liouville's transformation.*

3.3.1 Liouville's transformation

Consider the regular Sturm-Liouville problem to be solved

$$-\frac{d}{dr}\big(p(r)\frac{dz}{dr}\big) + q(r)z = Ew(r)z, \quad r_{\min} < r < r_{\max}, \tag{3.69}$$

where $r_{\min}$ and $r_{\max}$ are finite, functions p, q and w are defined on $[r_{\min}, r_{\max}]$ with p and w strictly positive (it also tacitly assumed that p and w can be differentiated twice on the interval). The boundary conditions are of the form

$$\begin{aligned} &a_0 z(r_{\min}) + b_0 p(r_{\min}) z'(r_{\min}) = 0, \\ &a_1 z(r_{\max}) + b_1 p(r_{\max}) z'(r_{\max}) = 0, \end{aligned} \tag{3.70}$$

where the constants a_0 and b_0 are not both zero and similarly for a_1 and b_1.

A transformation of *dependent variable* is performed of the form

$$z = \sigma(r)y. \tag{3.71}$$

This gives

$$-\frac{d}{dr}\left(p\sigma\frac{dy}{dr}\right) - \frac{d}{dr}\left(py\frac{d\sigma}{dr}\right) + q\sigma y = Ew\sigma y. \tag{3.72}$$

which we multiply by σ to restore the self-adjoint form:

$$-\frac{d}{dr}\left(p\sigma^2\frac{dy}{dr}\right) + \left(-\frac{d}{dr}\left(p\frac{d\sigma}{dr}\right)\sigma + q\sigma^2\right)y = Ew\sigma^2 y. \tag{3.73}$$

Then a transformation of *independent variable* from r to x

$$r = r(x) \tag{3.74}$$

converts (3.73) to the transformed equation

$$-\frac{d}{dx}\left(\frac{p\sigma^2}{\dot{r}}\frac{dy}{dx}\right) + \left(-\frac{d}{dx}\left(\frac{p}{\dot{r}}\frac{d\sigma}{dx}\right)\sigma + q\sigma^2\dot{r}\right)y = Ew\sigma^2\dot{r}y, \tag{3.75}$$

where $\dot{r} = dr/dx$.
When we choose as changes of variable

$$x = \int_{r_{\min}}^{r} \sqrt{w(r')/p(r')}dr' \tag{3.76}$$

and

$$\sigma(r) = (p(r)w(r))^{-1/4}, \tag{3.77}$$

we obtain an equation in the Liouville normal form or Schrödinger form

$$-y''(x) + V(x)y(x) = Ey(x) \tag{3.78}$$

where

$$V(x) = \frac{q}{w} + \sigma\frac{d^2}{dx^2}\left(\frac{1}{\sigma}\right). \tag{3.79}$$

The regular boundary conditions (3.70) are transformed thereby to

$$\begin{aligned} A_0 y(x_{\min}) + B_0 y'(x_{\min}) &= 0, \\ A_1 y(x_{\max}) + B_1 y'(x_{\max}) &= 0, \end{aligned} \tag{3.80}$$

where $B_0 = b_0,\ B_1 = b_1$ and

$$\begin{aligned} A_0 &= a_0\sigma^2(r_{\min}) + b_0 p(r_{\min})\sigma'(r_{\min})\sigma(r_{\min}), \\ A_1 &= a_1\sigma^2(r_{\max}) + b_1 p(r_{\max})\sigma'(r_{\max})\sigma(r_{\max}). \end{aligned} \tag{3.81}$$

To summarize, the original regular Sturm-Liouville problem Eqs. (3.69) and (3.70), is equivalent to the Schrödinger problem Eqs. (3.78) and (3.80) which has the same eigenvalue spectrum. To compute the eigenvalues of a Sturm-Liouville problem, the CPM algorithm is thus applied to the Schrödinger problem which appears after Liouville's transformation.

3.3.2 Implementation of Liouville's transformation

To be able to extend the CPM to the solution of Sturm-Liouville problems, we need an implementation of Liouville's transformation. This means that we need a procedure for computing the quadrature (3.76), which allows us to obtain x for a given r and vice versa. In [61] it was suggested to identify a set of points $r_0^G = r_{\min} < r_1^G < r_2^G < ... < r_{K+1}^G = r_{\max}$, such that the integral

$$q_k = \int_{r_k^G}^{r_{k+1}^G} \sqrt{w(r')/p(r')}dr', \quad k = 0, 1, ...K, \tag{3.82}$$

evaluated by a Gauss formula with twelve points (denoted as Q_k) is correct in all digits available in the double precision arithmetic. More exactly, the interval $[r_k^G, r_{k+1}^G]$ is taken small enough such that $|Q_k^{(1)} + Q_k^{(2)} - Q_k| < \epsilon$, with ϵ a precision threshold representing the double precision arithmetic and $Q_k^{(1)}$ and $Q_k^{(2)}$ the numerical values of the integrals

$$q_k^{(1)} = \int_{r_k^G}^{(r_{k+1}^G + r_k^G)/2} \sqrt{w(r')/p(r')}dr', \quad q_k^{(2)} = \int_{(r_{k+1}^G + r_k^G)/2}^{r_{k+1}^G} \sqrt{w(r')/p(r')}dr', \tag{3.83}$$

evaluated by the same twelve points Gauss method. The value of x_k^G associated with r_k^G is $x_k^G = q_0 + q_1 + ... + q_{k-1}$ and we store the x^G and the r^G in some vectors. When during the computations, the value of x corresponding to some given r is required, then the interval $[r_k^G, r_{k+1}^G]$ which contains the input r is first identified and only the integral from r_k^G up to r is evaluated. When an x value is known and r is required, one first looks for the interval $[x_k^G, x_{k+1}^G]$ and one then applies a Newton iteration procedure to compute r as the root of

$$\int_{r_k^G}^{r} \sqrt{w(r')/p(r')}dr' - x + x_k^G. \tag{3.84}$$

3.4 Higher Order CPM$\{P, N\}$ methods

As mentioned before, there exists a family of CPM algorithms which are identified as CPM$\{P, N\}$. A CPM$\{P, N\}$ includes just enough terms in the perturbation corrections to have an algorithm of order P when $Z(h) = (V_0 - E)h^2 \to 0$ and of order N for $-Z(h) \to +\infty$. The results of Theorem 3.1 allow us to obtain the expressions for the perturbation corrections. However it is clear that a lot of algebraic computations are required and that one has to make use of a powerful symbolic software package to construct CPM$\{P, N\}$ of high order. In [60] the version CPM$\{12, 10\}$ was introduced. This algorithm was later implemented in the Fortran program SLCPM12 [61] which is included in the CPC (Computer Physics Communications) program library [1]. In [75], we introduced a MAPLE code which computes the expressions for the perturbation corrections for an input value of P and N. Using this MAPLE code, versions of higher order CPM$\{14,\ 12\}$, CPM$\{16,\ 14\}$ and CPM$\{18,\ 16\}$ were defined (see also [75]).

The expressions of the propagators $u(h), hu'(h), v(h)/h$ and $v'(h)$ of the CPM$\{P, N\}$ algorithms have the following form:

$$\begin{aligned} u(h) &= \xi(Z) + \sum_{m=1}^{\infty} C_m^{(u)} \eta_m(Z), & (3.85)\\ hu'(h) &= Z\eta_0(Z) + \sum_{m=0}^{\infty} C_m^{(u')} \eta_m(Z), & (3.86)\\ v(h)/h &= \eta_0(Z) + \sum_{m=2}^{\infty} C_m^{(v)} \eta_m(Z), & (3.87)\\ v'(h) &= \xi(Z) + \sum_{m=1}^{\infty} C_m^{(v')} \eta_m(Z). & (3.88) \end{aligned}$$

where the coefficients $C_m^{(u)}, C_m^{(u')}, C_m^{(v)}$ and $C_m^{(v')}$ are expressed in terms of the $\bar{V}_i$ variables, with $\bar{V}_i = V_i h^{i+2}$, $i = 1, 2, \ldots$. In Appendix A.1 (part of) the $C_m^{(u)}$ coefficients are listed for the CPM$\{18, 16\}$ algorithm. The full expressions and the $C_m^{(u')}, C_m^{(v)}$ and $C_m^{(v')}$ can be generated by the MAPLE code included in Appendix B.1. Note that the summations in (3.85)–(3.88) are finite, e.g. for CPM$\{18,\ 16\}$ the coefficients $C_m^{(u)}$ are equal to zero for $m \geq 9$, while for CPM$\{16,\ 14\}$ also $C_8^{(u)}$ is zero which means that in the expression for the CPM$\{16,\ 14\}$ propagator $u(h)$, the coefficient of $\eta_8(Z)$ contains no terms with degree in h smaller or equal to 16.

It is also important to remark that the coefficients $C_m^{(u)}, C_m^{(u')}, C_m^{(v)}$ and $C_m^{(v')}$ are E-independent. As a consequence they have to be computed only once on each step and can be stored at the very beginning of the run. When the solution for a given E is then advanced on successive steps, only the E-dependent ξ and η_m remain to be calculated.

Another important feature of the CPM is that the partition (mesh) of a finite integration interval is formulated from the very beginning of the run and never altered again, no matter how small or how big the energy is. This means that the numerical solution of a Schrödinger or Sturm-Liouville problem consists of two separated stages. In the first stage, if the problem is of Sturm-Liouville form, it is converted to the Schrödinger equation. Then the partition of $[a, b]$ is constructed in terms of a tolerance specified by the user. Also some quantities (i.e. the stepsize h, V_0 and the $C_m^{(\ldots)}$ coefficients) associated to each interval in the partition are generated and stored. These quantities depend only on the potential and will be used repeatedly in the second stage. In the second stage the requested eigenvalues are then calculated. To locate these eigenvalues the shooting procedure is applied. The shooting data is also used to evaluate the Prüfer variable which enables a correct estimation of the eigenvalue index.

The construction of the E-independent partition or mesh will be discussed in the next subsection.

3.4.1 Stepsize selection - the mesh

For a high order CPM$\{P, N\}$ an adaptive stepsize selection algorithm consistent with a user input tolerance tol can be applied. The choice of the stepsize is based on a uniformly distributed local error which requires an estimation for the local error. In [60] and [75] the principle of embedding was used. For a method of higher order — say the method CPM$\{P, N\}$ — a second method of lower order CPM$\{P', N'\}$ (the embedded method) was used for the purpose of error estimation. However we are now able to formulate a different procedure. The MAPLE program allows us to obtain expressions for a higher order version CPM$\{P^*, N^*\}$ and all disregarded contributions which appear in the CPM$\{P^*,\ N^*\}$ algorithm but not in the CPM$\{P,\ N\}$ formulae can be used to estimate the error. The chosen higher order versions are $P^* = 14, N^* = 12$ for CPM$\{12,\ 10\}$, $P^* = 16, N^* = 14$ for CPM{14,12} and $P^* = 18, N^* = 16$ for CPM{16,14}.

Let us focus on CPM{16,14}: All terms in the expressions for $u(h), hu'(h), v(h)/h$ and $v'(h)$ of the CPM{18,16} version which are supplementary to the terms to be used in the CPM{16,14} version can be used to construct an estimation of the error. To start with we take a trial value h for the size of the step originating at X and use a Gauss quadrature formula to calculate $\bar{V}, \bar{V}_1, \bar{V}_2, \ldots, \bar{V}_N$ from (3.41). Here $\bar{V}_i$ is again a short-hand notation for $V_i h^{i+2}$.

Since the η-functions obtain their maximum value in $Z(h) = 0$, we compute ϵ_0 which is defined as

$$\epsilon_0 = \max(|\Delta u(h)|, |\Delta hu'(h)|, |\Delta v(h)/h|, |\Delta v'(h)|), \tag{3.89}$$

at $Z(h) = 0$. $\Delta u(h), \Delta hu'(h), \Delta v(h)/h$ and $\Delta v'(h)$ are the terms in the equations in Appendix A.1 which are additional to the terms of CPM{16,14}. That is, all terms where either (i) the $\bar{V}_i$ have $14 < i \leq 16$ or (ii) where the degree d in h satisfies $16 < d \leq 18$ (whereby the degree of $\bar{V}_i$ in h is $i + 2$). For $\Delta u(h)$, e.g. we have

$$\begin{aligned}
\Delta u(h) = & -\big(\bar{V}_{15}/2\big)\eta_1\,(Z(h)) + \big(119\bar{V}_{15}/2 - \bar{V}_7^2/120\big)\eta_2(Z(h)) \\
& + \big(- 6783\bar{V}_{15}/2 + [2\bar{V}_7^2 + 10\bar{V}_3\bar{V}_{10} + 10\bar{V}_2\bar{V}_{11} + 10\bar{V}_5\bar{V}_8 + 5\bar{V}_3\bar{V}_{11} \\
& \quad + 5\bar{V}_1\bar{V}_{13} + 5\bar{V}_5\bar{V}_9 + 10\bar{V}_4\bar{V}_9 + 10\bar{V}_1\bar{V}_{12} + 10\bar{V}_6\bar{V}_7]/20\big)\eta_3(Z(h)) \\
& + \big(237405\bar{V}_{15}/2 + [-23\bar{V}_7^2 - 282\bar{V}_5\bar{V}_8 - 120\bar{V}_3\bar{V}_{11} - 170\bar{V}_1\bar{V}_{13} \\
& \quad - 78\bar{V}_5\bar{V}_9 - 306\bar{V}_4\bar{V}_9 - 450\bar{V}_1\bar{V}_{12} - 270\bar{V}_6\bar{V}_7 - 342\bar{V}_3\bar{V}_{10} \\
& \quad - 390\bar{V}_2\bar{V}_{11} + 42\bar{V}_6\bar{V}_8 + 6\bar{V}_2\bar{V}_{12} + 20\bar{V}_4\bar{V}_{10}]/8 + [5005\bar{V}_3\bar{V}_4^2 \\
& \quad + 9009\bar{V}_2^2\bar{V}_7 + 15015\bar{V}_1^2\bar{V}_9 + 4095\bar{V}_1\bar{V}_5^2 + 6435\bar{V}_3^2\bar{V}_5 + 810\bar{V}_4^3 \\
& \quad + 9450\bar{V}_2\bar{V}_4\bar{V}_6 + 5400\bar{V}_3\bar{V}_4\bar{V}_5 + 11340\bar{V}_1\bar{V}_5\bar{V}_6 + 3150\bar{V}_2\bar{V}_5^2 \\
& \quad + 4500\bar{V}_3^2\bar{V}_6]/720720\big)\eta_4(Z(h)) - \big(5460315\bar{V}_{15}/2 + \ldots\big)\eta_5(Z(h)) \\
& + \big(81904725\bar{V}_{15}/2 + \ldots\big)\eta_6(Z(h)) - \big(737142525\bar{V}_{15}/2 + \ldots\big)\eta_7(Z(h)) \\
& + \big(3053876175\bar{V}_{15}/2 + \ldots\big)\eta_8(Z(h))
\end{aligned} \tag{3.90}$$

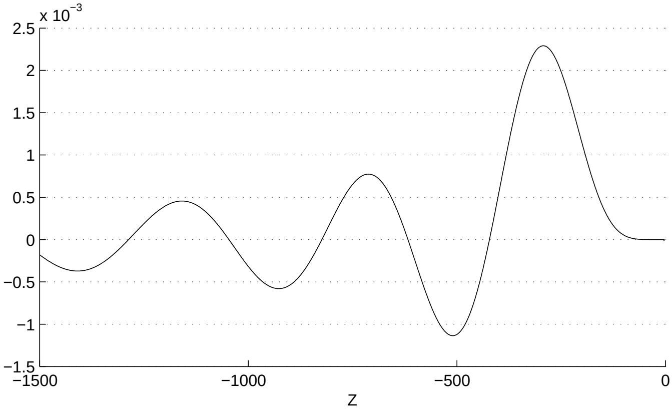


Figure 3.4: The coefficient function $c_{15}(Z)$.

Reordening the terms in this expression, we obtain

$$\Delta u(h) = c_{15}\bar{V}_{15} + c_{7,7}\bar{V}_7^2 + c_{3,10}\bar{V}_3\bar{V}_{10} + ... \tag{3.91}$$

where, e.g.,

$$\begin{aligned} c_{15} = \big[&- \eta_1(Z) + 119\eta_2(Z) - 6783\eta_3(Z) + 237405\eta_4(Z) \\ &- 5460315\eta_5(Z) + 81904725\eta_6(Z) - 737142525\eta_7(Z) \\ &+ 3053876175\eta_8(Z)\big]/2. \end{aligned} \tag{3.92}$$

These $c_{15}, c_{7,7}, \dots$ expressions reach maximum values at different values of $Z(h)$. In the case of $|c_{15}|$,e.g., we have a maximum of approximately 0.0023 at $Z = -293$ (see Figure 3.4). Using these maxima, the following contribution is obtained

$$\epsilon^u_{\rm loc} = 0.0023|\bar{V}_{15}| + 0.000025\bar{V}_7^2 + 0.000018|\bar{V}_3\bar{V}_{10}| + \dots \tag{3.93}$$

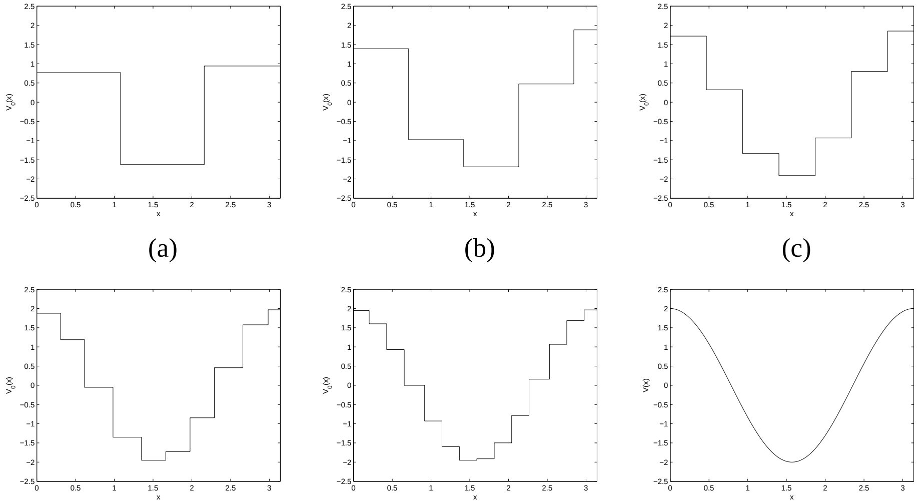
In the same way $\epsilon^{u'}_{\rm loc}$, $\epsilon^v_{\rm loc}$ and $\epsilon^{v'}_{\rm loc}$ are computed and the final error estimate is then

$$\epsilon_{\rm loc} = \max(\epsilon_0, \epsilon^u_{\rm loc}, \epsilon^{u'}_{\rm loc}, \epsilon^v_{\rm loc}, \epsilon^{v'}_{\rm loc}). \tag{3.94}$$

This error estimate is used to construct a new step size:

$$h_{\rm new} = h(tol/\epsilon_{\rm loc})^{1/(P-1)}, \tag{3.95}$$

where tol is the input tolerance. When $|h_{\rm new}/h - 1| > 0.1$ the procedure is repeated with $h = h_{\rm new}$. Otherwise h is accepted to be a good choice for the stepsize and the procedure continues with the stepsize selection for the next interval, which will originate at $X + h$. As first trial value for the stepsize of this new interval one can take h.



(a) (b) (c)

(d) (e) (f)

Figure 3.5: Piecewise constant approximation of the Mathieu potential function obtained with CPM$\{$12,10$\}$ and *tol*= (a) 10^{-6}, (b) 10^{-8}, (c) 10^{-10}, (d) 10^{-12}, (e) 10^{-14} and (f) the exact potential.

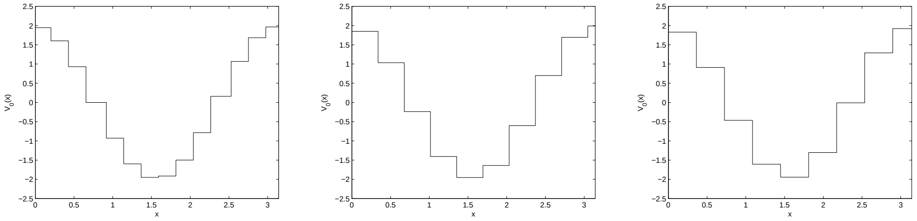
As mentioned above, the $\bar{V}_i$ are determined by applying a Gauss quadrature procedure on

$$\bar{V}_i = (2i+1)h \int_0^h V(X+\delta)P_i^*(\delta/h)d\delta. \tag{3.96}$$

For CPM$\{$12,10$\}$ a ten point Gauss quadrature formula is sufficient for the evaluation of these integrals. The CPM$\{$14,12$\}$ needs at least twelve points, CPM$\{$16,14$\}$ fourteen points and CPM$\{$18,16$\}$ sixteen points, respectively.

3.4.2 Some illustrations

Figure 3.5 shows the piecewise constant approximation constructed by CPM$\{$12,10$\}$ for the Mathieu potential function $V(x) = 2\cos(2x),\ x \in [0, \pi]$ for different values of the input tolerance tol. The number of intervals in the mesh increases with the tolerance: 3 intervals for $tol = 10^{-6}$, 5 for $tol = 10^{-8}$, 7 for $tol = 10^{-10}$, 10 for $tol = 10^{-12}$ and 14 for $tol = 10^{-14}$. Also Figure 3.6 shows the piecewise approximation of the potential but now for different CPM versions. It is clear that for higher order methods the stepsizes are larger: CPM$\{$12,10$\}$ needs 14 steps to reach a 10^{-14} accuracy, while CPM$\{$16,14$\}$ needs only 9. The reason is that more correction terms are included. The same conclusions can be drawn from Figure 3.7. This figure shows the number of steps chosen for the Coffey-Evans equation with $V(x) = -2\beta\cos(2x) + \beta^2 \sin^2 2x$ with $\beta = 20$.



(a) (b) (c)

Figure 3.6: Piecewise constant approximation of the potential function for the Mathieu equation obtained with $tol = 10^{-14}$ and different CPM$\{P, N\}$ versions: (a) CPM{12,10}, (b) CPM{14,12} and (c) CPM{16,14}.

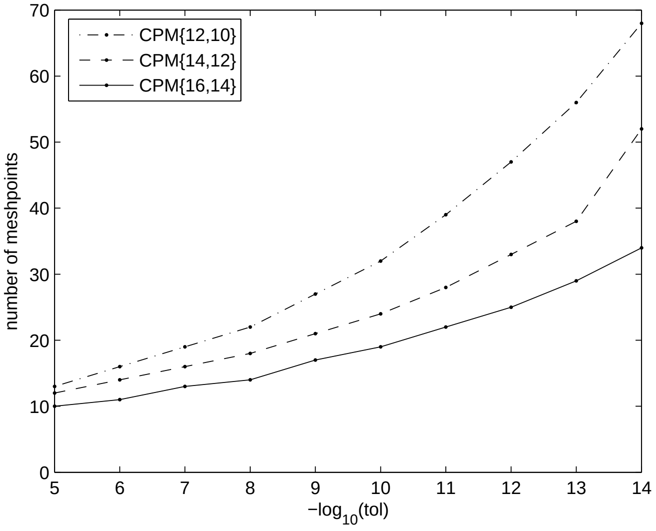


Figure 3.7: Coffey-Evans $\beta = 20$: number of steps for different input tolerances.

Table 3.2: Propagation of the solution for the Mathieu problem : the value of $y(\pi)$ computed with different CPM. Three different E-values are used: $E = E_1, E_{10}, E_{20}$. N is the number of equidistant steps.

k	N	CPM{12,10}	CPM{14,12}	CPM{16,14}
1	1	-0.02318892816899	-0.01456898484001	0.00564390992140
	2	-0.00016711628732	0.00001372532522	0.00000239548106
	3	-0.00000201724621	-0.00000044655324	0.00000000764493
	4	0.00000001824467	-0.00000000398729	0.00000000006233
	6	-0.00000000013490	-0.00000000002371	0.00000000000018
	8	-0.00000000000272	-0.00000000000041	0.00000000000001
10	1	-0.00000173522196	-0.00000047856561	0.00000008111233
	2	0.00000002087816	-0.00000000145525	-0.00000000009680
	3	-0.00000000013744	-0.00000000008640	0.00000000000077
	4	0.00000000094777	0.00000000002487	-0.00000000000025
	6	-0.00000000002253	-0.00000000000012	0.00000000000001
	8	-0.00000000000054	0.00000000000000	0.00000000000000
20	1	-0.00000024712704	-0.00000001236933	-0.00000000147455
	2	-0.00000000003327	-0.00000000000812	0.00000000000160
	3	-0.00000000005096	-0.00000000000613	0.00000000000017
	4	0.00000000000618	0.00000000000001	0.00000000000001
	6	-0.00000000000039	-0.00000000000001	0.00000000000000
	8	-0.00000000000015	0.00000000000000	0.00000000000000

Table 3.2 shows some results for the Mathieu initial value problem with initial conditions in one endpoint of the integration interval: $y(0) = 0, y'(0) = 1$. The solution is propagated towards the other endpoint π for three different E values. Since these three E values are the exact eigenvalues E_k, $k = \{1, 10, 20\}$ the value for $y(\pi)$ is zero. N is the number of steps in the equidistant partition. It is clear that when the number of steps N is increased, the solution obtained in π is more correct. A higher order method needs a smaller N value to reach a given accuracy than a lower order method. Moreover the number of steps used is remarkably small. Note also that the values of $y(\pi)$ go faster to zero for a larger E (or k) value.

The higher order CPM$\{P, N\}$ are implemented in the MATLAB package called MATSLISE (see chapter 7). The package exploits the power of the CPM to calculate the eigenvalues and eigenfunctions of a Schrödinger or Sturm-Liouville problem specified by the user. It therefore uses the shooting procedure discussed in section 3.2. We calculate a set of eigenvalues in MATSLISE for some test problems which appear in the problem set listed in appendix C.

The first test potential we consider is a Woods-Saxon potential [123] of the form

$$V(x) = -50\left(1 - (5t)/[3(1+t)]\right)/(1+t), \quad \text{with } t = e^{x-7}/0.6 \qquad (3.97)$$

over the integration interval $[0, 20]$. For this problem we calculated the first 14 eigenval-

Table 3.3: Calculation of the first 14 eigenvalues of the Woods-Saxon problem with different CPM codes and $tol = 10^{-10}$. $nint$ is the number of steps in the partition, $nfev$ the number of function evaluations and T the CPU time in seconds.

		Errors CPM		
k	E_k	{12,10}	{14,12}	{16,14}
0	-49.45778872808258	$8.6(-11)$	$1.4(-12)$	$1.0(-11)$
2	-46.29075395446608	$1.3(-10)$	$1.4(-10)$	$1.6(-10)$
4	-41.23260777218022	$1.0(-10)$	$2.1(-10)$	$1.6(-11)$
6	-34.67231320569966	$1.4(-10)$	$3.7(-10)$	$2.3(-10)$
8	-26.87344891605987	$1.2(-10)$	$3.8(-11)$	$1.9(-10)$
10	-18.09468828212442	$6.8(-10)$	$1.3(-10)$	$2.4(-10)$
12	-8.67608167073655	$7.3(-10)$	$1.7(-11)$	$3.6(-10)$
$nint$		22	17	16
$nfev$		600	546	528
T		0.9	0.7	0.6

ues. For the second test run we consider again the Coffey-Evans equation with $\beta = 20$. The first 21 eigenvalues are computed. The third test problem is the Mathieu equation for which we calculate the first 51 eigenvalues. As last test problem we consider a Sturm-Liouville problem not in Schrödinger form with $p(x) = 1$, $q(x) = -7x^2 + 0.5x^3 + x^4$, $w(x) = 0.5$ over $[-10, 10]$. We calculate the first 15 eigenvalues for this problem. For each problem the eigenvalues are determined with three CPM versions of different order which are implemented in MATSLISE: CPM$\{12,\ 10\}$, CPM$\{14,\ 12\}$ and CPM$\{16,\ 14\}$. In all cases an accuracy tolerance tol of 10^{-10} is requested.

In Tables 3.3-3.6 we present for each problem a selection of the considered exact eigenvalues E_k and the (absolute value of the) error in the eigenvalues returned by MATSLISE. The 'exact' eigenvalues E_k were obtained with a Fortran code implementing a CPM algorithm in quadruple precision. $nint$ is the number of intervals in the partition, $nfev$ is the number of function evaluations (of the potential function V) and T the CPU time (in seconds).

The data reported in the tables lead to a number of conclusions.

- First of all, one can see that the different CPM versions all produce results within the required accuracy.
- The number of intervals determined by the stepsize selection algorithm decreases with increasing order of the method.
- As a consequence the number of function evaluations also decreases with increasing order. This will however reach somewhere a limit. A higher order method reduces the number of steps in the partition, but in order to keep the accuracy in all intermediate steps the number of nodes in the used Gauss quadrature rules (to compute (3.96)) has to be increased, resulting in a higher number of function evaluations per

Table 3.4: Calculation of the first 21 eigenvalues of the Coffey-Evans problem ($\beta = 20$) with different CPM codes and $tol = 10^{-10}$. $nint$ is the number of steps in the partition, $nfev$ the number of function evaluations and T the CPU time in seconds.

		Errors CPM		
k	E_k	{12,10}	{14,12}	{16,14}
0	0.00000000000000	$5.8(-10)$	$1.8(-10)$	$5.5(-11)$
1	77.91619567714397	$3.8(-10)$	$1.0(-9)$	$2.9(-10)$
2	151.46277834645663	$2.5(-11)$	$8.3(-10)$	$2.0(-10)$
3	151.46322365765863	$2.0(-10)$	$8.6(-10)$	$2.8(-10)$
4	151.46366898835165	$2.5(-11)$	$8.3(-10)$	$2.1(-10)$
5	220.15422983525995	$5.6(-10)$	$1.1(-10)$	$1.1(-9)$
10	380.09491555093168	$4.7(-10)$	$6.0(-10)$	$2.6(-10)$
15	477.71051260907674	$3.7(-10)$	$3.1(-10)$	$9.8(-11)$
20	652.99045708465674	$2.5(-10)$	$1.3(-10)$	$4.5(-10)$
$nint$		32	24	19
$nfev$		540	476	464
T		2.3	1.6	1.4

Table 3.5: Calculation of the first 51 eigenvalues of the Mathieu problem with different CPM codes and $tol = 10^{-10}$. $nint$ is the number of steps in the partition, $nfev$ the number of function evaluations and T the CPU time in seconds.

		Errors CPM		
k	E_k	{12,10}	{14,12}	{16,14}
0	-0.11024881699209	$2.5(-12)$	$4.9(-11)$	$8.4(-12)$
10	121.00416676126912	$1.0(-10)$	$8.2(-11)$	$2.0(-11)$
20	441.00113636549330	$2.1(-11)$	$6.7(-12)$	$2.0(-13)$
30	961.00052083351094	$2.1(-13)$	$3.4(-13)$	$2.1(-13)$
40	1681.00029761908068	$3.7(-13)$	$3.7(-13)$	$1.1(-13)$
50	2601.00019230770122	$1.4(-12)$	$5.8(-13)$	$1.2(-13)$
$nint$		7	5	4
$nfev$		108	84	96
T		0.9	0.5	0.5

Table 3.6: Calculation of the first 15 eigenvalues of the Sturm-Liouville problem with different CPM codes and $tol = 10^{-10}$. $nint$ is the number of steps in the partition, $nfev$ the number of function evaluations and T the CPU time in seconds.

		Errors CPM		
k	E_k	{12,10}	{14,12}	{16,14}
0	-24.51759770716	$8.0(-10)$	$2.3(-10)$	$2.8(-10)$
3	-1.29384368195	$8.4(-10)$	$6.4(-11)$	$1.4(-10)$
6	14.73535195708	$5.8(-10)$	$2.8(-10)$	$7.7(-11)$
9	39.87238796401	$6.0(-10)$	$8.6(-10)$	$3.9(-10)$
12	70.05073428985	$7.9(-10)$	$1.2(-9)$	$2.3(-10)$
$nint$		185	131	109
$nfev$		6672	5920	5856
T		16.1	12.4	11.9

Table 3.7: Calculation of some higher eigenvalues of the Mathieu problem with different CPM codes and $tol = 10^{-10}$.

		Errors CPM		
k	E_k	{12,10}	{14,12}	{16,14}
100	10201.00004901960799	$1.8(-12)$	$1.8(-12)$	$1.8(-12)$
500	251001.00000199203187	$6.8(-11)$	$8.1(-12)$	$8.1(-12)$
1000	1002001.00000049900200	$3.0(-10)$	$9.8(-11)$	$9.8(-11)$
1500	2253001.00000022192632	$1.7(-10)$	$1.7(-10)$	$1.7(-10)$
2000	4004001.00000012487512	$8.2(-10)$	$7.5(-11)$	$7.5(-11)$

interval. This higher order Gauss quadrature rules explain the re-increase of the number of function evaluations with increasing order for the Mathieu test problem.

- The introduction of higher order terms results in most cases in a smaller CPU time. For the Mathieu problem, the higher number of function evaluations means that the CPM$\{16, 14\}$ spends more time determining the partition (i.e., the setting of the step h, the calculation of V_0 and the $C_i^{(\dots)}$ coefficients) than for the CPM$\{14, 12\}$. However, even when the partitioning process requires more time, the total time can still be smaller for a higher order method: the computation of the eigenvalues (the shooting process), which occurs after the partition has been fixed, is faster for the higher order method when there are less intervals. The gain in time in the shooting process can be big enough for a large set of eigenvalues to compensate the loss of time in the partitioning process.

Table 3.7 shows some higher eigenvalues for the Mathieu problem. It is clear that also for these high indices accurate results are obtained.

3.5 Conclusion

This chapter has covered: the CPM class of methods which can be used to efficiently integrate a Schrödinger equation; the shooting method which uses a CPM as propagation method to obtain eigenvalue approximations; the Prüfer form and the choice of a suitable scaling function S, which are used to select a good interval of energy values on which the Newton iteration process can be applied in order to home in on a specific eigenvalue; the extension to Sturm-Liouville problems and the construction of a class of high order methods, identified as the CPM$\{P, N\}$. These high order CPM$\{P, N\}$ methods were shown to have the power of producing very accurate results even for large energies.

Chapter 4

Line Perturbation Methods

In this chapter we consider a class of perturbation methods based on the linear approximation of the (Schrödinger) potential function. Although the solution of the reference equation is closer to the exact solution than if a piecewise constant reference potential would have been used, such an approach will be shown not to be advantageous in practice over the constant approximation technique discussed in the previous chapter.

4.1 A Line Perturbation Method for the Schrödinger equation

The idea of adding perturbation corrections to improve the accuracy of a CPM is already a few decennia old [56, 58]. This is not the case for the Line Perturbation Methods (LPM). Gordon, [41]-[42] was the first to suggest a code based on piecewise line approximation but no perturbation corrections were included. Also the improvements brought to this method along time were mainly related to the computation of the Airy functions which appear in the propagators of the reference equation (see, e.g. [4]) but we are unaware of any attempt of constructing and adding corrections. One reason may be that, though Gordon's papers include a way to compute such corrections, the results produced on this basis would often suffer of heavy loss in accuracy due to near-cancellations of like terms.

In this chapter we will examine the problem of the perturbation corrections for the LPM. We will effectively construct first and second-order corrections. To evaluate them we rely on an approach developed by Ixaru in [58] which is different from that in Gordon's papers. Its results are less exposed to the near-cancellation effect but an extra treatment is still needed.

4.1.1 The reference equation

We again consider the Schrödinger equation

$$\frac{d^2y}{dx^2} + [E - V(x)]y = 0, \quad x \in [a, b]\,. \tag{4.1}$$

The interval $[a, b]$ is divided in a set of subintervals, resulting in a partition with the meshpoints $a = x_0, x_1, x_2, \dots, x_n = b$. Let us focus on the current interval $[x_{i-1}, x_i]$ which we denote $[X, X + h]$, where h is the current stepsize. As for the CPM, we want to obtain expressions for the propagators u and v and their first derivatives u' and v', so that the formulae (with $\delta \in [0, h]$)

$$\begin{bmatrix} y(X+\delta) \\ y'(X+\delta) \end{bmatrix} = \begin{bmatrix} u(\delta) & v(\delta) \\ u'(\delta) & v'(\delta) \end{bmatrix} \begin{bmatrix} y(X) \\ y'(X) \end{bmatrix}, \tag{4.2}$$

and

$$\begin{bmatrix} y(X) \\ y'(X) \end{bmatrix} = \begin{bmatrix} v'(\delta) & -v(\delta) \\ -u'(\delta) & u(\delta) \end{bmatrix} \begin{bmatrix} y(X+\delta) \\ y'(X+\delta) \end{bmatrix} \tag{4.3}$$

can be used to propagate the solution from one end of the interval to the other.

Zeroth order approximations to these propagators are derived from the reference equation. For the LPM, the potential $V(x)$ is first approximated on $[X, X + h]$ by a 'line' function $\bar{V}(x)$ such that the equation $y'' = [\bar{V}(x) - E]y$ can be solved analytically. This means that the reference equation over the interval $[X, X + h]$ is of the form

$$\begin{aligned} y''(\delta) &= (\bar{V}(X+\delta) - E)y(\delta) \\ &= (F_0 + F_1\delta - E)y(\delta) \end{aligned} \tag{4.4}$$

with F_0 and F_1 two constants. The reference propagators $\bar{u}(\delta)$ and $\bar{v}(\delta)$ are two particular solutions of this reference equation which satisfy the initial conditions $\bar{u}(0) = 1, \bar{u}'(0) = 0$ and $\bar{v}(0) = 0, \bar{v}'(0) = 1$.

The reference propagators can be expressed in terms of Airy functions. To see this, we introduce the following change of variable: $F_0 + F_1\delta - E = Cz$, where C will be chosen conveniently. By simple manipulations Eq. (4.4) becomes

$$y''(z) - \frac{C^3}{F_1^2} z y(z) = 0\,. \tag{4.5}$$

When we take $C^3 = F_1^2$, the relation between δ and z is then

$$z = \frac{F_0 + F_1\delta - E}{\sqrt[3]{F_1^2}} \tag{4.6}$$

and Eq. (4.5) reduces to

$$y''(z) - zy(z) = 0. \tag{4.7}$$

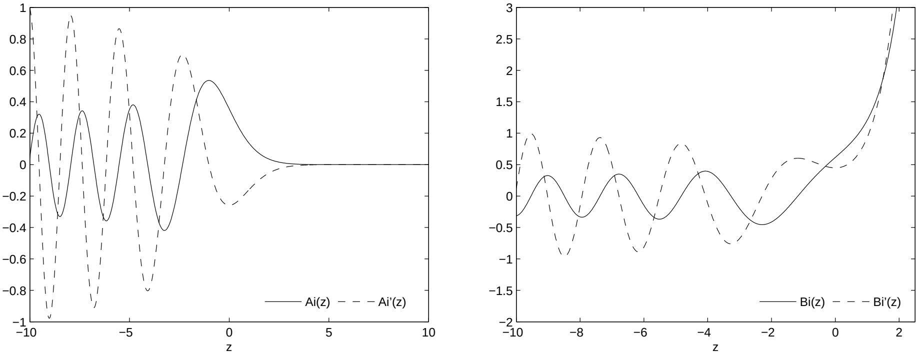


Figure 4.1: The Airy functions. ***(Left)*** $Ai(z)$ and $Ai'(z)$. ***(Right)*** $Bi(z)$ and $Bi'(z)$.

Two linear independent solutions of this equation are the Airy functions Ai and Bi (see [3] or [120]) and a general solution of Eq. (4.7) is a linear combination of the functions Ai and Bi:

$$y(z) = \alpha Ai(z) + \beta Bi(z), \;\; y'(z) = \alpha Ai'(z) + \beta Bi'(z)\,. \tag{4.8}$$

To determine the constants α and β corresponding to the reference propagators $\bar{u}$ and $\bar{v}$, their associated initial conditions are used. By defining $\gamma = F_1/\sqrt[3]{F_1^2} = \sqrt[3]{F_1}$, $z_0 = z(0) = (F_0 - E)/\sqrt[3]{F_1^2}$ and making use of the Wronskian relation $W\{Ai(z),\, Bi(z)\} = Ai(z)Bi'(z) - Ai'(z)Bi(z) = 1/\pi$, we get the expressions for the propagators of the reference equation (4.4):

$$\begin{aligned}
\bar{u}(\delta) &= \pi\,[Bi'(z_0)Ai(z) - Ai'(z_0)Bi(z)] && (4.9)\\
\bar{u}'(\delta) &= \pi\gamma\,[Bi'(z_0)Ai'(z) - Ai'(z_0)Bi'(z)] && (4.10)\\
\bar{v}(\delta) &= \frac{\pi}{\gamma}\,[-Bi(z_0)Ai(z) + Ai(z_0)Bi(z)] && (4.11)\\
\bar{v}'(\delta) &= \pi\,[-Bi(z_0)Ai'(z) + Ai(z_0)Bi'(z)]\,. && (4.12)
\end{aligned}$$

4.1.2 The construction of the perturbation corrections

As pointed out in section 2.3.4, each of the two propagators $u(\delta)$ and $v(\delta)$, denoted generically $p(\delta)$, is written as a perturbation series,

$$p(\delta) = p_0(\delta) + p_1(\delta) + p_2(\delta) + p_3(\delta) + \ldots\,. \tag{4.13}$$

The zeroth-order term $p_0(\delta)$ is the solution of the reference equation (thus $p_0(\delta) = \bar{p}(\delta)$), while the correction $p_q, q = 1, 2, \ldots$ obeys the equation

$$p_q'' = [\bar{V}(X+\delta) - E]p_q + \Delta V(X+\delta)p_{q-1}, \quad p_q(0) = p_q'(0) = 0, \tag{4.14}$$

where $\Delta V(X+\delta) = V(X+\delta) - \bar{V}(X+\delta)$ is the perturbation. We can then construct each correction as a linear combination with δ dependent coefficients of the reference propagators and of their first-order derivatives (see [58]):

$$p_q(\delta) = a_q(\delta)\bar{u}(\delta) + b_q(\delta)\bar{v}(\delta) + c_q(\delta)\bar{u}'(\delta) + d_q(\delta)\bar{v}'(\delta), \tag{4.15}$$

where a_q, b_q, c_q and d_q are functions to be determined. First we evaluate p_q' and p_q'' (with $\bar{F} = \bar{V}(X+\delta) - E$):

$$\begin{aligned} p_q' &= a_q'\bar{u} + a_q\bar{u}' + b_q'\bar{v} + b_q\bar{v}' + c_q'\bar{u}' + c_q\bar{u}'' + d_q'\bar{v}' + d_q\bar{v}'' \\ &= a_q'\bar{u} + (a_q + c_q')\bar{u}' + b_q'\bar{v} + (b_q + d_q')\bar{v}' + c_q\bar{F}\bar{u} + d_q\bar{F}\bar{v} \\ &= (a_q' + c_q\bar{F})\bar{u} + (b_q' + d_q\bar{F})\bar{v} + (a_q + c_q')\bar{u}' + (b_q + d_q')\bar{v}' \end{aligned} \tag{4.16}$$

$$\begin{aligned} p_q'' =& [a_q'' + (2c_q' + a_q)\bar{F} + c_q\bar{F}']\bar{u} + [b_q'' + (2d_q' + b_q)\bar{F} + d_q\bar{F}']\bar{v} \\ &+ [c_q'' + 2a_q' + c_q\bar{F}]\bar{u}' + [d_q'' + 2b_q' + d_q\bar{F}]\bar{v}' \end{aligned} \tag{4.17}$$

and write the right hand side of Eq. (4.14) as

$$\begin{aligned} \bar{F}p_q + \Delta V p_{q-1} = (a_q\bar{F} + \Delta V a_{q-1})\bar{u} + (b_q\bar{F} + \Delta V b_{q-1})\bar{v} \\ + (c_q\bar{F} + \Delta V c_{q-1})\bar{u}' + (d_q\bar{F} + \Delta V d_{q-1})\bar{v}'. \end{aligned} \tag{4.18}$$

Combining Eqs. (4.17) and (4.18) we obtain the following system of differential equations for the coefficients a_q, b_q, c_q, d_q

$$\begin{array}{rcl} a_q'' + 2c_q'(\bar{V} - E) + c_q\bar{V}' &=& \Delta V a_{q-1} \\ b_q'' + 2d_q'(\bar{V} - E) + d_q\bar{V}' &=& \Delta V b_{q-1} \\ c_q'' + 2a_q' &=& \Delta V c_{q-1} \\ d_q'' + 2b_q' &=& \Delta V d_{q-1}. \end{array} \tag{4.19}$$

Taking into account that $\bar{u}(0) = 1, \bar{u}'(0) = 0$ and $\bar{v}(0) = 0, \bar{v}'(0) = 1$, Eqs (4.15) and (4.16) give us the initial conditions

$$a_q(0) + d_q(0) = 0, \;\; a_q'(0) + (\bar{V}(0) - E)c_q(0) + b_q(0) + d_q'(0) = 0 \tag{4.20}$$

for the system (4.19).

For $q = 0$ we have $a_0 = 1, b_0 = c_0 = d_0 = 0$ if $p = u$ and $b_0 = 1, a_0 = c_0 = d_0 = 0$ if $p = v$. For $q \geq 1$, the coefficients of index $q-1$ are introduced in the right-hand side of system (4.19) and then the system is solved with the initial conditions (4.20) to get a_q, b_q, c_q, and d_q. Note that there are only *two* initial conditions for *four* differential equations and hence the solution of (4.19)) is not unique. This allows some flexibility in the determination of a_q, b_q, c_q and d_q.

4.1.3 A pilot reference equation

Since, as for CPM, the perturbation corrections can be evaluated analytically in closed form only when the perturbation ΔV is a polynomial, we make use of the same strategy as in section 3.1.3. This means that we introduce a pilot reference function of the form

$$V^{(N)}(X+\delta)=\sum_{n=0}^{N}V_n h^n P_n^*(\frac{\delta}{h}) \tag{4.21}$$

to finally consider only corrections from the pilot perturbation. It is thus the equation

$$y''(X+\delta)=[V^{(N)}(X+\delta)-E]y(X+\delta),\ \ \delta\in[0,h] \tag{4.22}$$

for which propagators are constructed.

To compute the reference potential $F_0+F_1\delta$ we use a least-squares procedure. This means that

$$\int_0^h [V(X+\delta)-(F_0+F_1\delta)]^2\,d\delta \tag{4.23}$$

has to be minimized. To explicitly evaluate F_0 and F_1 we set to zero the first-order partial derivatives of (4.23) with respect to F_0 and F_1 and obtain the following system of linear equations

$$\begin{cases} hF_0+\dfrac{h^2}{2}F_1=\displaystyle\int_0^h V(X+\delta)d\delta \\ \dfrac{h^2}{2}F_0+\dfrac{h^3}{3}F_1=\displaystyle\int_0^h \delta V(X+\delta)d\delta \end{cases} \tag{4.24}$$

whose solution is given by

$$\begin{cases} F_0=\dfrac{4}{h}\displaystyle\int_0^h V(X+\delta)d\delta-\dfrac{6}{h^2}\int_0^h \delta V(X+\delta)d\delta \\ F_1=-\dfrac{6}{h^2}\displaystyle\int_0^h V(X+\delta)d\delta+\dfrac{12}{h^3}\int_0^h \delta V(X+\delta)d\delta. \end{cases} \tag{4.25}$$

Knowing that (see section 3.1.3)

$$V_0 = \frac{1}{h}\int_0^h V(X+\delta)d\delta, \tag{4.26}$$

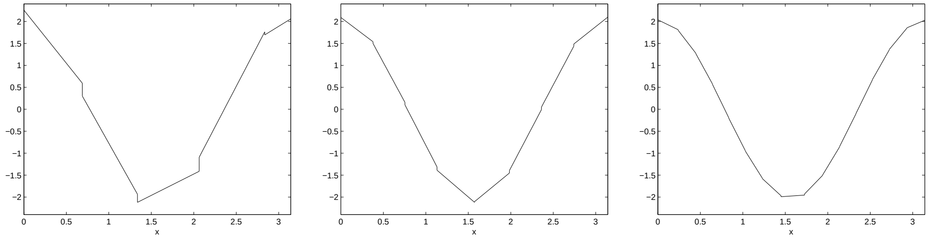
$$V_1 = -\frac{3}{h^2}\int_0^h V(X+\delta)\,d\delta+\frac{6}{h^3}\int_0^h \delta V(X+\delta)\,d\delta \tag{4.27}$$

we have the following form for F_0 and F_1

$$F_0=V_0-V_1h,\quad F_1=2V_1. \tag{4.28}$$

This means that, while for the CPM the reference potential and the perturbation were

$$\bar{V}(X+\delta)=V_0,\quad \Delta V(X+\delta)=\sum_{n=1}^{N}V_n h^n P_n^*(\frac{\delta}{h}), \tag{4.29}$$



(a) (b) (c)

Figure 4.2: Piecewise line approximation of the potential function for the Mathieu equation constructed by the LPM[4,2] method with *tol*= (a) 10^{-6}, (b) 10^{-8} and (c) 10^{-10}.

we now take for the LPM

$$\bar{V}(X+\delta) = V_0 + V_1 h P_1^*(\frac{\delta}{h}), \quad \Delta V(X+\delta) = \sum_{n=2}^{N} V_n h^n P_n^*(\frac{\delta}{h}). \tag{4.30}$$

To identify the different LPM versions, we use the notation LPM[N, Q] where N is the number of Legendre polynomials and $Q > 0$ is the number of perturbations. When $Q = 0$, then the pilot potential is not involved and the method is denoted by LPM(0) (see [58]).

Example 4.1 Figure 4.2 shows some piecewise line approximations of the Mathieu potential function $V(x) = 2\cos(2x),\ x \in [0, \pi]$. The piecewise line approximations are constructed by the LPM[4,2] method for different values of the input tolerance tol. The LPM[4,2] method will be discussed in more detail in the next sections.

4.2 The LPM[4,2] method

In this section we will efffectively construct the first and the second order corrections for the LPM. We assume a pilot potential of fourth degree (i.e. $N = 4$ in Eq. (4.21)); this value will be justified by the error analysis.

4.2.1 Perturbation corrections

The procedure discussed in section 4.1.2 was implemented in the symbolic software package Maple. The Maple program is listed in B.2.1 and produces the expressions of the first and second order corrections for the LPM[4,1] and LPM[4,2] method.

First order corrections

We briefly discuss the procedure followed to construct the first order correction $u_1(\delta)$. This correction is of the form

$$u_1(\delta) = a_1(\delta)\bar{u}(\delta) + b_1(\delta)\bar{v}(\delta) + c_1(\delta)\bar{u}'(\delta) + d_1(\delta)\bar{v}'(\delta), \tag{4.31}$$

where $a_1(\delta)$, $b_1(\delta)$, $c_1(\delta)$ and $d_1(\delta)$ satisfy the system (4.19). This means

$$a_1'' + 2c_1'(F_0 + F_1\delta - E) + c_1 F_1 = \sum_{n=2}^{4} V_n h^n P_n^*(\frac{\delta}{h}) \tag{4.32}$$

$$b_1'' + 2d_1'(F_0 + F_1\delta - E) + d_1 F_1 = 0 \tag{4.33}$$

$$c_1'' + 2a_1' = 0 \tag{4.34}$$

$$d_1'' + 2b_1' = 0. \tag{4.35}$$

with the initial conditions

$$a_1(0) + d_1(0) = 0, \;\; a_1'(0) + (F_0 - E)c_1(0) + b_1(0) + d_1'(0) = 0. \tag{4.36}$$

To solve this system, Eq. (4.34) is differentiated and introduced into Eq. (4.32) to obtain

$$-\frac{1}{2}c_1''' + 2c_1'(F_0 + F_1\delta - E) + c_1 F_1 = \sum_{n=2}^{4} V_n h^n P_n^*(\frac{\delta}{h}). \tag{4.37}$$

A particular solution of this differential equation in c_1 is

$$\begin{aligned} c_1(\delta) = {} & \left[\frac{256}{9}\,\frac{V_4}{F_1}\right] Q^4 + \left[-\frac{64}{63}\,\frac{14\,\delta\,V_4 + 9\,V_3 - 63\,V_4 h}{F_1}\right] Q^3 \\ & + \left[\frac{16}{105}\,\frac{21\,V_2 + 315\,V_4 h^2 + 30\,\delta\,V_3 - 210\,\delta\,V_4 h - 105\,V_3 h + 70\,V_4\delta^2}{F_1}\right] Q^2 \\ & + \Bigg[\frac{4}{315 F_1}\,\big(315\,V_2 h - 270\,\delta^2 V_3 - 1890\,V_4 h^2\delta + 1050\,V_4 h^3 - 126\,V_2\delta + \\ & \quad 630\,V_3 h\delta - 630\,V_3 h^2 - 700\,V_4\delta^3 + 1890\,\delta^2 V_4 h\big) - \frac{800}{9}\,\frac{V_4}{{F_1}^2}\Bigg] Q \\ & + \frac{1}{315 F_1}\,\big(900\,\delta^3 V_3 + 378\,\delta^2 V_2 - 630\,\delta\,V_2 h - 6300\,\delta^3 V_4 h - 2100\,\delta\,V_4 h^3 + \\ & \quad 2450\,V_4\delta^4 + 315\,V_2 h^2 + 1260\,\delta\,V_3 h^2 - 1890\,\delta^2 V_3 h + 5670\,\delta^2 V_4 h^2 - \\ & \quad 315\,V_3 h^3 + 315\,V_4 h^4\big) + \frac{20}{63}\,\frac{98\,\delta\,V_4 + 27\,V_3 - 189\,V_4 h}{{F_1}^2} \end{aligned} \tag{4.38}$$

with $Q = (F_0 - E)/F_1$. We also differentiate Eq. (4.35) and introduce it in Eq. (4.33). A particular solution of the equation

$$-\frac{1}{2}d_1''' + 2d_1'(F_0 + F_1\delta - E) + d_1 F_1 = 0 \tag{4.39}$$

is then $d_1(\delta) = 0$. The particular solutions for c_1 and d_1 are then introduced in Eqs. (4.34) and (4.35) to get the general solutions of a_1 and b_1. The particular solutions for a_1 and b_1 are determined by the two initial conditions (4.36), to get

$$\begin{aligned} a_1(\delta) = &-\left[\frac{32}{3}\,\frac{V_4\delta}{F_1}\right]Q^2 - \left[\frac{2}{105}\,\frac{-700\,V_4\delta^2 + 1260\,\delta\,V_4 h - 180\,\delta\,V_3}{F_1}\right]Q \\ &-\frac{2}{315}\,\frac{\delta\left(189\,V_2 + 2450\,V_4\delta^2 - 945\,V_3 h + 2835\,V_4 h^2 + 675\,\delta\,V_3 - 4725\,\delta\,V_4 h\right)}{F_1} \end{aligned} \tag{4.40}$$

$$\begin{aligned} b_1(\delta) = &-\frac{256}{9}\,V_4 Q^5 - \left[\frac{64}{7}\,(-V_3 + 7\,V_4 h)\right]Q^4 - \left[\frac{16}{5}\,\left(V_2 + 15\,V_4 h^2 - 5\,V_3 h\right)\right]Q^3 \\ &+\left[-4\,V_2 h - \frac{40}{3}\,V_4 h^3 + 8\,V_3 h^2 + \frac{896}{9}\,\frac{V_4}{F_1}\right]Q^2 + \left[V_3 h^3 - V_2 h^2 - V_4 h^4\right. \\ &\left.+\frac{84\,V_4 h - 12\,V_3}{F_1}\right]Q + \frac{6}{5}\,\frac{V_2 + 15\,V_4 h^2 - 5\,V_3 h}{F_1}. \end{aligned} \tag{4.41}$$

By direct differentiation of (4.31), the expression of $u_1'(\delta)$ is obtained. An analogous procedure can be followed to construct $v_1(\delta)$ and $v_1'(\delta)$. The calculations were done using the Maple program. The resulting formulae can be written in the following form:

$$\begin{aligned} u_1(h) &= \sum_{k=0}^{2}\alpha_k Q^k \bar{u}(h) + \sum_{k=0}^{5}\beta_k Q^k \bar{v}(h) + \sum_{k=0}^{4}\gamma_k Q^k \bar{u}'(h) \\ u_1'(h) &= \sum_{k=0}^{5}\delta_k Q^k \bar{u}(h) + \sum_{k=0}^{3}\epsilon_k Q^k \bar{u}'(h) + \sum_{k=0}^{5}\beta_k Q^k \bar{v}'(h) \\ v_1(h) &= \sum_{k=0}^{4}\zeta_k Q^k \bar{u}(h) - \sum_{k=0}^{3}\epsilon_k Q^k \bar{v}(h) + \sum_{k=0}^{4}\gamma_k Q^k \bar{v}'(h) \\ v_1'(h) &= \sum_{k=0}^{5}\delta_k Q^k \bar{v}(h) + \sum_{k=0}^{4}\zeta_k Q^k \bar{u}'(h) - \sum_{k=0}^{2}\alpha_k Q^k \bar{v}'(h) \end{aligned}$$

This form only shows the non-zero terms, e.g. all terms in $Q^k\bar{u}(h), k > 2$ for $u_1(h)$ are zero terms (see (4.40)). The coefficients $\alpha_k, \beta_k, \gamma_k, \delta_k$..., which do not depend on the energy, are calculated only once and stored before the actual propagation. They are:

$$\begin{aligned} \alpha_0 &= -2\,h(189\,V_2 - 270\,V_3 h + 560\,V_4 h^2)/\,(315 F_1) \\ \alpha_1 &= -8\,h\,(28\,V_4 h - 9\,V_3)/(21 F_1) \\ \alpha_2 &= -32\,V_4 h/(3 F_1) \end{aligned}$$

$$\beta_0 = (18\,V_4 h^2 + 6/5\,V_2 - 6\,V_3 h)/F_1$$

$$\begin{aligned}
\beta_1 &= V_3h^3 - V_4h^4 - V_2h^2 + (-12\,V_3 + 84\,V_4h)/F_1\\
\beta_2 &= 8\,V_3h^2 - 40/3\,V_4h^3 - 4\,V_2h + 896\,V_4/(9F_1)\\
\beta_3 &= -16/5\,V_2 - 48\,V_4h^2 + 16\,V_3h\\
\beta_4 &= 64/7\,V_3 - 64\,V_4h\\
\beta_5 &= -256/9\,V_4
\end{aligned}$$

$$\begin{aligned}
\gamma_0 &= (1/5\,V_2h^2 + 1/9\,V_4h^4 - 1/7\,V_3h^3)/F_1 + (-260/9\,V_4h + 60/7\,V_3)\,/{F_1}^2\\
\gamma_1 &= (12/5\,V_2h + 40/9\,V_4h^3 - 24/7\,V_3h^2)/F_1 - 800\,V_4/(9{F_1}^2)\\
\gamma_2 &= (80/3\,V_4h^2 - 80/7\,V_3h + 16/5\,V_2)/F_1\\
\gamma_3 &= (448/9\,V_4h - 64/7\,V_3)/F_1\\
\gamma_4 &= 256\,V_4/(9F_1)
\end{aligned}$$

$$\begin{aligned}
\delta_0 &= 1/9\,V_4h^5 + 1/5\,h^3V_2 - 1/7\,h^4V_3 + (6\,V_3h - 6/5\,V_2 - 302/9\,V_4h^2)/F_1\\
\delta_1 &= -25/7\,V_3h^3 + 41/9\,V_4h^4 + 13/5\,V_2h^2 + (12\,V_3 - 1036/9\,V_4h)\,/F_1\\
\delta_2 &= 280/9\,V_4h^3 + 28/5\,V_2h - 104/7\,V_3h^2 - 896\,V_4/(9F_1)\\
\delta_3 &= 688/9\,V_4h^2 - 144/7\,V_3h + 16/5\,V_2\\
\delta_4 &= 704/9\,V_4h - 64/7\,V_3\\
\delta_5 &= 256/9\,V_4
\end{aligned}$$

$$\begin{aligned}
\epsilon_0 &= (-28/9\,V_4h^3 + 16/7\,V_3h^2 - 4/5\,V_2h)/F_1 + 280\,V_4/(9{F_1}^2)\\
\epsilon_1 &= (32/7\,V_3h - 8/5\,V_2 - 40/3\,V_4h^2)/F_1\\
\epsilon_2 &= (-64/3\,V_4h + 32/7\,V_3)\,/F_1\\
\epsilon_3 &= -128\,V_4/(9F_1)
\end{aligned}$$

$$\begin{aligned}
\zeta_0 &= (-V_4h^4 - V_2h^2 + V_3h^3)/F_1 + (60\,V_4h - 60/7\,V_3)/{F_1}^2\\
\zeta_1 &= (8\,V_3h^2 - 40/3\,V_4h^3 - 4\,V_2h)/F_1 + 800\,V_4/(9{F_1}^2)\\
\zeta_2 &= (-16/5\,V_2 - 48\,V_4h^2 + 16\,V_3h)/F_1\\
\zeta_3 &= (64/7\,V_3 - 64\,V_4h)\,/F_1\\
\zeta_4 &= -256\,V_4/(9F_1)
\end{aligned}$$

Second order corrections

Also the second order corrections are computed solving a system of the form (4.19). The obtained expressions can be written in the form

$$u_2(h) = \sum_{k=0}^{9} \alpha_k^{(u)} Q^k \bar{u}(h) + \sum_{k=0}^{8} \beta_k^{(u)} Q^k \bar{v}(h) + \sum_{k=0}^{7} \gamma_k^{(u)} Q^k \bar{u}'(h) + \sum_{k=0}^{9} \delta_k^{(u)} Q^k \bar{v}'(h)$$

$$\begin{aligned}
u_2'(h) &= \sum_{k=0}^{8}\alpha_k^{(u')}Q^k\bar{u}(h) +\sum_{k=0}^{10}\beta_k^{(u')}Q^k\bar{v}(h) +\sum_{k=0}^{9}\gamma_k^{(u')}Q^k\bar{u}'(h) +\sum_{k=0}^{8}\delta_k^{(u')}Q^k\bar{v}'(h)\\
v_2(h) &= \sum_{k=0}^{7}\alpha_k^{(v)}Q^k\bar{u}(h) +\sum_{k=0}^{9}\beta_k^{(v)}Q^k\bar{v}(h) +\sum_{k=0}^{8}\gamma_k^{(v)}Q^k\bar{u}'(h) +\sum_{k=0}^{7}\delta_k^{(v)}Q^k\bar{v}'(h)\\
v_2'(h) &= \sum_{k=0}^{9}\alpha_k^{(v')}Q^k\bar{u}(h) +\sum_{k=0}^{8}\beta_k^{(v')}Q^k\bar{v}(h) +\sum_{k=0}^{7}\gamma_k^{(v')}Q^k\bar{u}'(h) +\sum_{k=0}^{9}\delta_k^{(v')}Q^k\bar{v}'(h)
\end{aligned}$$

The coefficients $\alpha_k, \beta_k, \gamma_k, \delta_k$ are also calculated and stored before the actual propagation. Their expressions are too long to be listed here. They can be reproduced by using the Maple code in B.2.1.

4.2.2 Error analysis

Let us consider the original equation on the interval $[X, X+h]$:

$$y''(X+\delta) = [V(X+\delta) - E]\, y(X+\delta),\ \ \delta \in [0,h]. \tag{4.42}$$

We restrict the error analysis to the case when the stepsize h is so small that we can rely on the power series representation of the propagators $u(\delta)$ and $v(\delta)$, to accept that the first neglected term in the perturbation expansion is numerically sufficient to measure the error (see again [58]).

It is convenient to assume that the original potential is an infinite series over shifted Legendre polynomials. Then Eq. (4.42) reads

$$y''(X+\delta) = \left(\sum_{n=0}^{\infty} V_n h^n P_n^*(\delta/h) - E\right) y(X+\delta)\,. \tag{4.43}$$

The two independent solutions of Eq. (4.43) u and v (with initial values $u(0)=1, u'(0)=0$ and $v(0)=0, v'(0)=1$) then have the following form

$$p(\delta) = \sum_{s=0}^{\infty} p_s \delta^s \tag{4.44}$$

with p either u or v. Their derivatives can easily be obtained by differentiation of Eq. (4.44):

$$p'(\delta) = \sum_{s=1}^{\infty} s p_s \delta^{s-1}. \tag{4.45}$$

LPM(0)

The zeroth order propagators (or reference propagators) $u_0(=\bar{u})$ and $v_0(=\bar{v})$ are the solutions of the corresponding reference equation

$$y''(X+\delta) = \left[\bar{V}(\delta) - E\right] y(X+\delta) \tag{4.46}$$

where $\bar{V}(\delta) = \sum_{n=0}^{1} V_n h^n P_n^*(\delta/h)$. The zeroth order propagators and their derivatives can be written as

$$p_0(\delta) = \sum_{s=0}^{\infty} p_s^{(0)} \delta^s, \quad p_0'(\delta) = \sum_{s=1}^{\infty} s p_s^{(0)} \delta^{s-1}. \tag{4.47}$$

The error of the LPM(0) method is then determined by the four quantities:

$$\begin{aligned} \Delta u_0(h) = u(h) - u_0(h), &\quad \Delta v_0(h) = v(h) - v_0(h), \\ \Delta u_0'(h) = u'(h) - u_0'(h), &\quad \Delta v_0'(h) = v'(h) - v_0'(h) \end{aligned} \tag{4.48}$$

which are calculated by subtracting Eq. (4.47) from Eq. (4.44) and Eq. (4.45). We have used Maple to get:

$$\begin{aligned} \Delta u_0(h) &= [V_3 E - V_3 V_0 + V_2 V_1]\, h^7/210 + O(h^8) \\ \Delta v_0(h) &= -V_2\, h^5/30 + O(h^7) \\ \Delta u_0'(h) &= [-V_2 E + V_2 V_0]\, h^5/30 + O(h^7) \\ \Delta v_0'(h) &= [-V_3 E + V_3 V_0 - V_2 V_1]\, h^7/210 + O(h^8). \end{aligned} \tag{4.49}$$

The smallest power of h in these errors is five. The LPM(0) is thus a *fourth*-order method, as was already shown in [58]. We notice that the determination of V_0 and V_1 in terms of shifted Legendre polynomials is essential to obtain this order. Any other determination will lead to lower orders. For example, if V_0 and V_1 are chosen to represent the tangent to $V(x)$ at the midpoint $(x_{i-1} + x_i)/2$, as in [41], the order of the method is two. For a proof of this see [55].

LPM[4,1]

The first order correction p_1 satisfies the equation

$$p_1'' = (\bar{V}(\delta) - E)p_1 + \Delta V(\delta) p_0, \quad p_1(0) = p_1'(0) = 0 \tag{4.50}$$

where $\Delta V(\delta) = \sum_{n=2}^{N} V_n h^n P_n^*(\delta/h)$ and $N = 4$. Also this first order correction can be expressed as a power series:

$$p_1(\delta) = \sum_{s=0}^{\infty} p_s^{(1)} \delta^s, \quad p_1'(\delta) = \sum_{s=1}^{\infty} s p_s^{(1)} \delta^{s-1}. \tag{4.51}$$

The error of the LPM[4,1] method is then determined by

$$\begin{aligned} \Delta u_1(h) = u(h) - (u_0(h) + u_1(h)), &\quad \Delta v_1(h) = v(h) - (v_0(h) + v_1(h)), \\ \Delta u_1'(h) = u'(h) - (u_0'(h) + u_1'(h)), &\quad \Delta v_1'(h) = v'(h) - (v_0'(h) + v_1'(h)). \end{aligned} \tag{4.52}$$

Using Maple, we obtain

$$\begin{aligned}
\Delta u_1(h) &= -V_2^2 h^8/420 + O(h^9) \\
\Delta v_1(h) &= [V_4 V_2/2970 - V_1 V_5/8316 - V_3^2/4620 - \\
&\quad (V_0 - E)V_2^2/415800]\, h^{11} + O(h^{13}) \\
\Delta u_1'(h) &= -V_2^2 h^7/210 + O(h^9) \\
\Delta v_1'(h) &= -V_2^2 h^8/420 + O(h^9).
\end{aligned} \tag{4.53}$$

We can conclude that the LPM[4,1] method is of *sixth* order. As a matter of fact, all versions LPM[N,1] with $N \geq 3$ will give the same order, such that for the versions with $Q = 1$ taking $N = 3$ in the pilot potential is sufficient.

LPM[4,2]

The error for the LPM[4,2] version is obtained in the same way, with the result:

$$\begin{aligned}
\Delta u_2(h) &= [-V_5 V_2/3465 - V_5(V_0 - E)^2/20790]\, h^{11} + O(h^{12}) \\
\Delta v_2(h) &= -V_1 V_5/8316\, h^{11} + O(h^{13}) \\
\Delta u_2'(h) &= [V_5 V_3/1386 + V_5 V_1(V_0 - E)/5940 - V_2^3/6930]\, h^{11} + O(h^{13}) \\
\Delta v_2'(h) &= [V_5(V_0 - E)^2/20790 + V_2 V_5/3465]\, h^{11} + O(h^{12}).
\end{aligned}$$

The LPM[4,2] method is thus a method of order *ten* and the same holds for all versions LPM[N,2] with $N \geq 4$. This is actually the reason why we have adopted $N = 4$ in the pilot potential. More general, our investigations have shown that the order of LPM[N,Q], $Q = 1,\, 2,\, 3,\, \ldots$ is $4Q + 2$ for any $N \geq Q + 2$.

4.2.3 Near-cancellation effects

Some precaution is necessary when computing the propagators in terms of the Airy functions. Near-cancellation of like-terms may appear, causing a severe decrease in accuracy.

Looking at Table 4.1, it is clear that for large arguments it is not a good idea to calculate the values of Ai, Bi, Ai' and Bi' at z and z_0 separately, and introduce them in (4.9)-(4.12). For large positive arguments the Airy function Ai and its derivative Ai' will eventually underflow, while Bi and Bi' will overflow. Also for (very) large negative arguments the evaluation of the Airy functions may be numerically inaccurate. These inaccuracies in the reference propagators are then propagated into the first and second order corrections, where they become even worse. Especially when the potential is nearly flat ($F_1 \to 0$), the powers of Q appearing in the first and second order corrections become very large and near-cancellation of like terms causes heavy loss of accuracy.

The near-cancellation of like-terms will force us to distinguish two regimes for computation in terms of $z(h) = [F_0 - E + F_1 h]/\sqrt[3]{F_1^2}$ (denoted hereinafter simply z) and $z_0 = z(0)$, with distinct formulae in each regime. The analytic formulae for the reference propagators and the corrections discussed in section 4.2.1 are used for small values of z and z_0 while for big values of z and z_0 asymptotic formulae are introduced. These asymptotic expressions will be discussed in section 4.3.2.

Table 4.1: The Airy functions at large arguments, obtained in Maple.

Z	$Ai(Z)$	$Bi(Z)$	$Ai'(Z)$	$Bi'(Z)$
0	0.36	0.61	-0.26	0.45
10	0.11(-9)	0.46(9)	-0.35(-9)	0.14(10)
25	0.81(-37)	0.39(36)	-.41(-36)	0.20(37)
50	0.46(-103)	0.49(102)	-0.32(-102)	0.35(103)
75	0.84(-189)	0.22(188)	-0.73(-188)	0.19(189)
100	0.26(-290)	0.60(289)	-0.26(-289)	0.60(290)
150	0.10(-532)	0.13(532)	-0.12(-531)	0.16(533)

4.3 Some technical issues

In this section we concentrate on some technical issues concerning the LPM. In 4.3.1 we consider the computation of the Airy functions appearing in the expressions of the zeroth order propagators. Alternative formulae, based on some asymptotic representations are discussed in section 4.3.2. These asymptotic representations also form the basis for asymptotic formulae for the first and second order corrections. The formulae are obtained by a Maple code presented in B.2.2. In a last subsection (4.3.3) we propose a procedure for choosing the stepsize in terms of the preset accuracy.

4.3.1 Computation of the Airy functions

The standard way of expressing the zeroth order propagators is through the Airy functions of arguments z and z_0; see the formulae (4.9)-(4.12). There is a rich literature on the computation of these functions. For real arguments we cite [29], [48], [93], [98], [133], while for complex arguments we mention [8], [35], [40]. More information about the calculation of Airy functions can be found in the recent book of Vallée and Soares [120].

Only the codes with real arguments are potentially important for us and we have compared them on a set of test cases. The accuracies were quite similar, but the NAG subroutines [93] proved to be somewhat faster and have been finally adopted for the Fortran implementation of the LPM[4,2] method. For the MATLAB implementation we used the MATLAB build-in Airy functions which are based on the work of Amos [7, 8].

4.3.2 Asymptotic formulae

The accuracy in the computation of the Airy functions appearing in formulae (4.9)-(4.12) depends on the range of the arguments z and z_0. In particular severe accuracy losses are observed when z and z_0 have big (negative or positive) values. The experimental investigations have then lead us to introduce an asymptotic range which collects the situations when $z,\, z_0 \leq -2$ or $z,\, z_0 \geq 2$. Asymptotic expansions for the propagators will be used on this range while the standard representation through Airy functions is used otherwise.

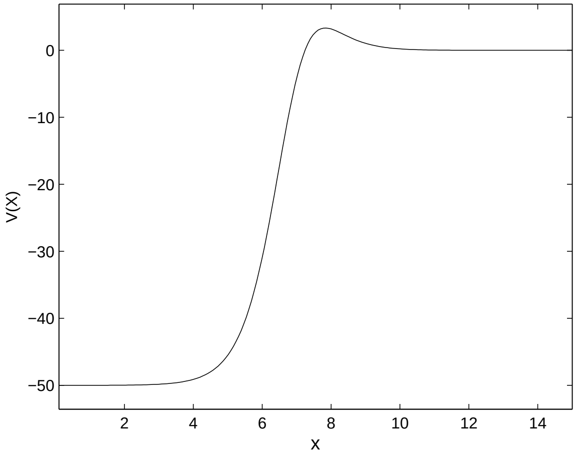


Figure 4.3: The Woods-Saxon potential in the interval $[0, 15]$

Example 4.2 As a rule big values (in modulus) for z and z_0 appear when the slope F_1 of the potential is small and $F_0 - E$ is big. We illustrate this on the Woods-Saxon potential defined by

$$V(x) = v_0 w(x) \left(1 - \frac{1 - w(x)}{a_0}\right) \tag{4.54}$$

with $w(x) = \{1 + \exp[(x - x_0)/a_0]\}^{-1}$, $v_0 = -50$, $x_0 = 7$, $a_0 = 0.6$, $x \in [0,\ x_f = 15]$ - see Figure 4.3. We used the MATLAB implementation of the LPM[4,2] method with a user input tolerance $tol = 10^{-10}$ to construct the Figures 4.4. In the upper figure the variation with x of F_1 and of $F_0 - E = V_0 - V_1 h - E$ is shown (in the middle of each meshinterval) for three test values of E. The choice of these values is rather free but we have chosen three eigenvalues: $E_0 \approx -49.458$, $E_7 \approx -30.912$ and $E_{13} \approx -3.908$. Since the stepsizes h are rather small, F_0 is in essence the potential $V(x)$ shifted by E. We see that $F_1(x)$ is small when x is in the vicinity of the endpoints and that it has a root somewhere around 7.75, i.e. at the maximum of the potential function. If these curves are compared with the x dependence of $z(h) = [F_0 - E + F_1 h]/\sqrt[3]{F_1^2}$, shown in the lower figure, we see that small F_1 and big $F_0 - E$ indeed lead to z in the asymptotic zone. Moreover, once z is in the asymptotic range the same holds for z_0 because the difference $|z - z_0| = |F_1|^{1/3} h$ is small.

Asymptotic expressions for the reference propagators

Since big values for z and z_0 appear when F_1 is small and $G_0 = F_0 - E$ is big, it is appropriate to expand the zeroth order propagators in powers of F_1 and $1/(G_0)$ in order to remove the near-cancellation effects. As a matter of fact, when $F_1 \to 0$ such formulae

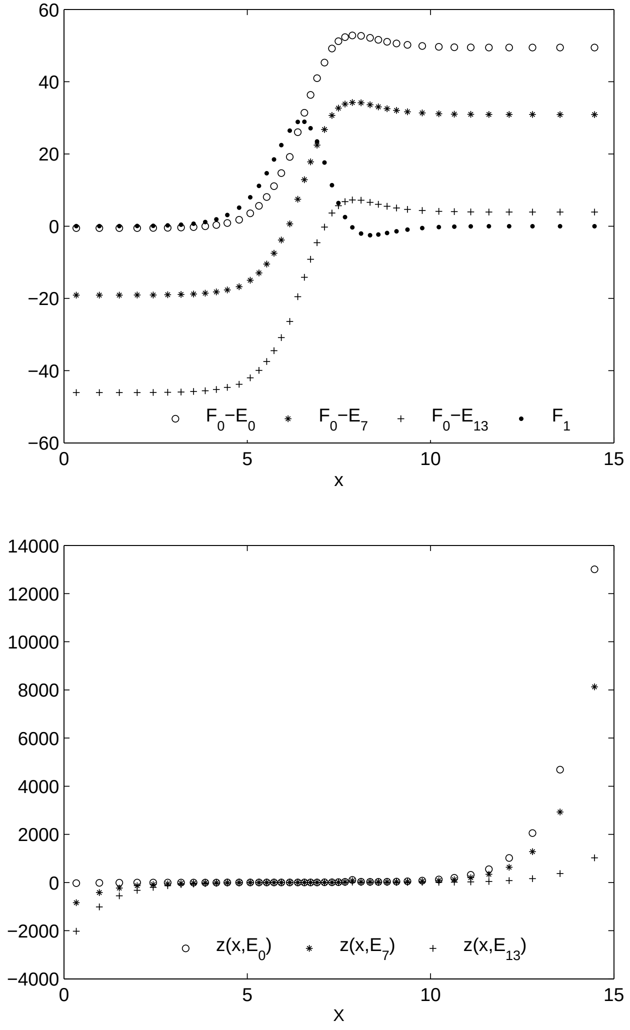


Figure 4.4: Woods-Saxon potential: the parameters $F_1(x)$ and $F_0(x) - E$ for $E = E_0, E_7, E_{13}$ in terms of $x \in [0, 15]$ and the evolution of $z(x, E) = [F_1(x)h + F_0(x) - E]/\sqrt[3]{F_1(x)^2}$ for $E = E_0, E_7, E_{13}$, $tol = 10^{-10}$ and $x \in [0, 15]$.

must tend to the expressions of the zeroth order propagators for the CPM.

We have used the known asymptotic expansions of the Airy functions (see [3], chapter 10.4). Define

$$c_0 = 1, \;\; d_0 = 1 \tag{4.55}$$

$$c_k = \frac{(6k-5)(6k-1)}{72k} c_{k-1}, \;\; d_k = -\frac{6k+1}{6k-1} c_k \tag{4.56}$$

and

$$\zeta = (2/3)Z^{3/2}. \tag{4.57}$$

The asymptotic expansions of the Airy functions for Z large negative are then of the form

$$\begin{aligned} Ai(Z) = \pi^{-1/2} Z^{-1/4} \big[\sin(\zeta + \frac{\pi}{4}) \sum_{k=0}^{\infty} (-1)^k c_{2k} \zeta^{-2k} - \\ \cos(\zeta + \frac{\pi}{4}) \sum_{k=0}^{\infty} (-1)^k c_{2k+1} \zeta^{-2k-1} \big] \end{aligned} \tag{4.58}$$

$$\begin{aligned} Ai'(Z) = -\pi^{-1/2} Z^{1/4} \big[\cos(\zeta + \frac{\pi}{4}) \sum_{k=0}^{\infty} (-1)^k d_{2k} \zeta^{-2k} + \\ \sin(\zeta + \frac{\pi}{4}) \sum_{k=0}^{\infty} (-1)^k d_{2k+1} \zeta^{-2k-1} \big] \end{aligned} \tag{4.59}$$

$$\begin{aligned} Bi(Z) = \pi^{-1/2} Z^{-1/4} \big[\cos(\zeta + \frac{\pi}{4}) \sum_{k=0}^{\infty} (-1)^k c_{2k} \zeta^{-2k} + \\ \sin(\zeta + \frac{\pi}{4}) \sum_{k=0}^{\infty} (-1)^k c_{2k+1} \zeta^{-2k-1} \big] \end{aligned} \tag{4.60}$$

$$\begin{aligned} Bi'(Z) = \pi^{-1/2} Z^{1/4} \big[\sin(\zeta + \frac{\pi}{4}) \sum_{k=0}^{\infty} (-1)^k d_{2k} \zeta^{-2k} - \\ \cos(\zeta + \frac{\pi}{4}) \sum_{k=0}^{\infty} (-1)^k d_{2k+1} \zeta^{-2k-1} \big] \end{aligned} \tag{4.61}$$

and the asymptotic expansions of the Airy functions for Z large positive are given by

$$Ai(Z) = (1/2)\pi^{-1/2} Z^{-1/4} \exp(-\zeta) \sum_{k=0}^{\infty} (-1)^k c_k \zeta^{-k} \tag{4.62}$$

$$Ai'(Z) = -(1/2)\pi^{-1/2} Z^{1/4} \exp(-\zeta) \sum_{k=0}^{\infty} (-1)^k d_k \zeta^{-k} \tag{4.63}$$

$$Bi(Z) = \pi^{-1/2} Z^{-1/4} \exp(\zeta) \sum_{k=0}^{\infty} c_k \zeta^{-k} \tag{4.64}$$

$$Bi'(Z) = \pi^{-1/2} Z^{1/4} \exp(\zeta) \sum_{k=0}^{\infty} d_k \zeta^{-i} \tag{4.65}$$

These asymptotic expressions are introduced in the standard representation of the propagators and the result is organized in the mentioned form (in powers of F_1 and $1/(G_0)$). The Maple code listed in B.2.2 has been used for this purpose. We give below the resulting asymptotic formulae.

With the notations

$$C(h) = \cos\left(\sqrt{-G_0}h\right), \;\; S(h) = -\sqrt{-G_0}\sin\left(\sqrt{-G_0}h\right).$$

when both z_0 and z are *large negative* and

$$C(h) = \cosh\left(\sqrt{G_0}h\right), \;\; S(h) = \sqrt{G_0}\sinh\left(\sqrt{G_0}h\right)$$

when both z_0 and z are *large positive* we have

$$\begin{aligned} \bar{u}(h) \;\; = \;\; & C(h) + \Big[- G_0 h C(h) + \big(G_0 h^2 + 1\big) S(h)\Big] F_1/(4{G_0}^2) \\ & + \Big[\big(3\,h^4 {G_0}^2 + 21\,h^2 G_0\big)\,C(h) + \big(-10\,h^3 G_0 - 21\,h\big)\,S(h)\Big] {F_1}^2/(96{G_0}^3) \\ & + \Big[\big(h^6 G_0^3 + 28\,h^4 G_0^2 + 105\,h^2 G_0 + 105\big)\,S(h) + \big(- 7\,h^5 {G_0}^3 - 70\,h^3 {G_0}^2 \\ & -105\,h G_0\big) C(h)\Big] {F_1}^3/(384{G_0}^5) + \Big[\big(3\,h^8 {G_0}^4 + 238\,h^6 {G_0}^3 + 3255\,h^4 {G_0}^2 \\ & +12285\,h^2 G_0\big) C(h) + \big(- 36\,h^7 G_0^3 - 1050\,h^5 G_0^2 - 7350\,h^3 G_0 \\ & -12285\,h\big) S(h)\Big] {F_1}^4/(18432\,{G_0}^6) \end{aligned} \tag{4.66}$$

$$\begin{aligned} \bar{v}(h) \;\; = \;\; & S(h)/G_0 + \Big[G_0 h^2 C(h) - h S(h)\Big] F_1/(4{G_0}^2) + \Big[\big(- 10\,h^3 {G_0}^2 \\ & -15\,h G_0\big) C(h) + \big(3\,h^4 G_0^2 + 15\,h^2 G_0 + 15\big) S(h)\Big] {F_1}^2/(96\,{G_0}^4) \\ & + \Big[\big(h^6 {G_0}^3 + 25\,h^4 {G_0}^2 + 105\,h^2 G_0\big) C(h) + \big(- 60\,h^3 G_0 - 7\,h^5 G_0^2 \\ & -105\,h\big) S(h)\Big] {F_1}^3/(384\,{G_0}^5) + \Big[\big(- 36\,h^7 {G_0}^4 - 966\,h^5 {G_0}^3 - 6930\,h^3 {G_0}^2 \\ & -10395\,h G_0\big) C(h) + \big(3\,h^8 G_0^4 + 226\,h^6 G_0^3 + 3045\,h^4 G_0^2 + 10395\,h^2 G_0 \\ & +10395\big) S(h)\Big] {F_1}^4/(18432\,{G_0}^7) \end{aligned} \tag{4.67}$$

$$\begin{aligned} \bar{u}'(h) \;\; = \;\; & S(h) + \Big[G_0 h^2 C(h) + h S(h)\Big] F_1/(4\,G_0) + \Big[\big(2\,{G_0}^2 h^3 + 21\,G_0 h\big) C(h) \\ & + \big(+ 3\,{G_0}^2 h^4 - 9\,G_0 h^2 - 21\big) S(h)\Big] {F_1}^2/(96\,{G_0}^3) + \Big[\big({G_0}^3 h^6 - 7\,{G_0}^2 h^4 \end{aligned}$$

$$\begin{aligned}
&-105\,G_0h^2\big)C(h)+\big(-{G_0}^2h^5+42\,G_0h^3+105\,h\big)S(h)\Big]{F_1}^3/(384\,{G_0}^4)\\
&+\Big[\big(-12\,{G_0}^4h^7+378\,{G_0}^3h^5+5670\,{G_0}^2h^3+12285\,G_0h\big)C(h)\\
&+\big(3\,{G_0}^4h^8-14\,{G_0}^3h^6-1995\,{G_0}^2h^4-9765\,G_0h^2-12285\big)S(h)\Big]{F_1}^4\\
&/(18432\,{G_0}^6)
\end{aligned} \tag{4.68}$$

$$\begin{aligned}
\bar{v}'(h) \;=\;& C(h)+\Big[\,\big(G_0h^2-1\big)\,S(h)+G_0hC(h)\Big]F_1/(4\,{G_0}^2)\\
&+\Big[\,\big(3\,{G_0}^2h^4-15\,G_0h^2\big)\,C(h)+\big(2\,G_0h^3+15\,h\big)\,S(h)\Big]{F_1}^2/(96\,{G_0}^3)\\
&+\Big[\,\big(-{G_0}^3h^5+40\,{G_0}^2h^3+105\,G_0h\big)\,C(h)+\big(-10\,{G_0}^2h^4-75\,G_0h^2\\
&+{G_0}^3h^6-105\big)S(h)\Big]{F_1}^3/(384\,{G_0}^5)+\Big[\big(-1785\,{G_0}^2h^4-10395\,G_0h^2\\
&-26\,{G_0}^3h^6+3\,{G_0}^4h^8\big)C(h)+\big(5250\,G_0h^3+390\,{G_0}^2h^5-12\,{G_0}^3h^7\\
&+10395\,h\big)S(h)\Big]{F_1}^4/(18432\,{G_0}^6).
\end{aligned} \tag{4.69}$$

It is obvious that for a flat potential ($F_1 \to 0$) these formulae reduce to the CPM zeroth order propagators (see 3.1.1), as expected (note that the CPM basic function $\xi(Z(h))$ is the same as $C(h)$ and that $h\eta_0(Z)$ corresponds to $S(h)/G_0$). Extensive experimental tests with values of G_0 and F_1 which lead to z and z_0 in the asymptotic range have shown that these truncated series are sufficient to produce the zeroth order propagators with an accuracy of 16 digits.

Asymptotic expressions for first and second-order corrections

Asymptotic expressions for the first and second order correction are obtained by substituting the asymptotic formulae for the reference propagators (4.66)-(4.69) in the analytic expressions of the first and second order correction discussed in section 4.2.1. With $Q = (F_0 - E)/F_1$, the resulting asymptotic expansions of the first and second order corrections are then of the following form :

$$u_1(h) = \sum_{k=0}^{\infty} \sigma_k^{(u)}\, Q^{-k}, \;\; v_1(h) = \sum_{k=0}^{\infty} \sigma_k^{(v)}\, Q^{-k}$$

$$u_1'(h) = \sum_{k=0}^{\infty} \sigma_k^{(u')}\, Q^{-k}, \;\; v_1'(h) = \sum_{k=0}^{\infty} \sigma_k^{(v')}\, Q^{-k}$$

$$u_2(h) = \sum_{k=0}^{\infty} \rho_k^{(u)}\, Q^{-k}, \;\; v_2(h) = \sum_{k=0}^{\infty} \rho_k^{(v)}\, Q^{-k}$$

$$u_2'(h) = \sum_{k=0}^{\infty} \rho_k^{(u')}\, Q^{-k}, \;\; v_2'(h) = \sum_{k=0}^{\infty} \rho_k^{(v')}\, Q^{-k}$$

where the coefficients σ and ρ do not depend on F_1. In our implementation the series were truncated at $k = 4$ for the first order corrections and at $k = 3$ for the (smaller) second order corrections. These truncations ensure values which are accurate enough for double precision calculations.

The expressions of the coefficients are too long to be listed in full. We give only the expressions of the first three $\sigma^{(u)}$ coefficients to offer an idea on how they look like. The other σ and ρ coefficients can be derived using the Maple code in the Appendix B.2.2. With $T_0 = 1/(F_0 - E)$ the first three coefficients are:

$$\begin{aligned}
\sigma_0^{(u)} &= 15/2\, V_3 S\,(h)\, {T_0}^3 + \big(3\, h^2 S\,(h) - 15/2\, h C\,(h)\big)\, V_3 {T_0}^2 - 1/2\, V_3 h^3 C\,(h)\, T_0 \\
\\
\sigma_1^{(u)} &= -\,2205/8\, V_4 S\,(h)\, {T_0}^4 + \Big[-\,15/4\, V_2 S\,(h) - 45/4\, V_3 h S\,(h) \\
&\quad + \big(-975/8\, S\,(h)\, h^2 + 2205/8\, h C\,(h)\big)\, V_4\Big] {T_0}^3 + \Big[\big(-\,39/8\, h^3 S\,(h) \\
&\quad + 45/4\, h^2 C\,(h)\,\big) V_3 + \big(-33/8\, h^4 S\,(h) + 30\, h^3 C\,(h)\big)\, V_4 \\
&\quad + \big(15/4\, h C\,(h) - 3/2\, S\,(h)\, h^2\big)\, V_2\Big] {T_0}^2 + \Big[\big(-\,1/8\, h^5 S\,(h) \\
&\quad + 9/8\, h^4 C\,(h)\,\big) V_3 + 1/4\, h^3 V_2 C\,(h) + 1/4\, V_4 h^5 C\,(h)\,\Big] T_0 \\
\\
\sigma_2^{(u)} &= \Big[7245/64\, V_3 S\,(h) + 21735/32\, V_4 h S\,(h)\,\Big] {T_0}^4 + \Big[\big(1995/32\, S\,(h)\, h^2 \\
&\quad -\, 7245/64\, h C\,(h)\,\big) V_3 + \big(4935/16\, h^3 S\,(h) - 21735/32\, h^2 C\,(h)\big)\, V_4 \\
&\quad + 231/32\, V_2 h S\,(h)\,\Big] {T_0}^3 + \Big[\,\big(115/16\, h^4 S\,(h) - 1575/64\, h^3 C\,(h)\big)\, V_3 \\
&\quad + \big(435/32\, h^5 S\,(h) - 2625/32\, h^4 C\,(h)\big)\, V_4 + \big(99/32\, h^3 S\,(h) \\
&\quad -\, 231/32\, h^2 C\,(h)\,\big) V_2\Big] {T_0}^2 + \Big[\,\big(-3/2\, h^5 C\,(h) + 5/24\, h^6 S\,(h)\big)\, V_3 \\
&\quad + \big(-43/32\, h^6 C\,(h) + 1/16\, h^7 S\,(h)\big)\, V_4 + \big(11/160\, h^5 S\,(h) \\
&\quad -\, 11/16\, h^4 C\,(h)\,\big) V_2\Big] T_0 - \frac{1}{64}\, V_3 h^7 C\,(h)\,.
\end{aligned}$$

4.3.3 Stepsize selection

As for the CPM, the problem of constructing a rule for the stepsize adjustment in terms of the preset error is not easy. This is because, in contrast with most of the numerical methods for the Schrödinger equation, these methods usually achieve high accuracy at very coarse partitions, with steps too big for the error evaluation in terms of its leading term only. A number of extra terms of higher order must be added for a reasonable evaluation.

LPM[4,2] is of order ten and then the collection in the local error of all the terms proportional to h^{11} up to no less than h^{13} or h^{14} is sufficient. By applying Maple we have obtained the following error formulae for the four propagators where $Z = V_0 - E$:

$$\begin{aligned}
\Delta u \;=\;& -1/540540\, h^{13} V_5 Z^3 + \big[-1/20790\, h^{11} V_5 + 1/1081080\, h^{14} V_5 V_1\big] Z^2 \\
&+\big[-1/54054\, h^{13} V_1 V_6 + 41/1081080\, h^{14} V_3 V_5 + 1/180180\, h^{14} V_2 V_6 \\
&+23/75675600\, h^{14} {V_2}^3 + 1/41580\, h^{12} V_5 V_1 - 1/135135\, h^{13} V_5 V_2\big] Z \\
&-1/3465\, h^{11} V_5 V_2 + 1/5148\, h^{14} V_6 V_4 + 1/2772\, h^{12} V_5 V_3 + 1/138600\, h^{12} {V_2}^3 \\
&+29/1801800\, h^{14} {V_2}^2 V_4 + 19/1081080\, h^{13} V_5 {V_1}^2 + 1/154440\, h^{14} V_6 {V_1}^2 \\
&+5/36036\, h^{13} V_5 V_4 - 5/36036\, h^{13} V_6 V_3 - 1/103950\, h^{14} V_2 {V_3}^2 \\
&-1/1081080\, h^{13} {V_2}^2 V_3 - 1/5148\, h^{14} {V_5}^2 - 17/540540\, h^{14} V_2 V_5 V_1
\end{aligned}$$

$$\begin{aligned}
\Delta u' \;=\;& 1/270270\, h^{13} V_6 Z^3 + 1/360360\, h^{13} V_5 V_1 Z^2 + \big[-1/150150\, h^{13} {V_2}^3 \\
&+1/16380\, h^{13} V_2 V_6 + 1/5940\, h^{11} V_5 V_1 + 3/20020\, h^{13} V_3 V_5\big] Z \\
&-1/6930\, h^{11} {V_2}^3 - 43/270270\, h^{13} V_2 V_5 V_1 + 1/1386\, h^{11} V_5 V_3 \\
&+23/270270\, h^{13} V_4 {V_2}^2 - 1/13860\, h^{13} V_2 {V_3}^2 + 1/2574\, h^{13} V_6 V_4 \\
&-1/2574\, h^{13} {V_5}^2 + 1/77220\, h^{13} V_6 {V_1}^2
\end{aligned}$$

$$\begin{aligned}
\Delta v \;=\;& -1/270270\, h^{13} Z^2 V_6 - 1/1081080\, h^{13} V_1 V_5 Z - 1/8316\, h^{11} V_1 V_5 \\
&+1/1351350\, h^{13} {V_2}^3 - 1/20020\, h^{13} V_2 V_6 + 1/6006\, h^{13} V_3 V_5
\end{aligned}$$

$$\begin{aligned}
\Delta v' \;=\;& 1/540540\, h^{13} V_5 Z^3 + \big[1/20790\, h^{11} V_5 + 1/1081080\, h^{14} V_1 V_5\big] Z^2 \\
&+\big[1/54054\, h^{13} V_1 V_6 + 1/180180\, h^{14} V_2 V_6 + 1/135135\, h^{13} V_2 V_5 \\
&+41/1081080\, h^{14} V_3 V_5 + 1/41580\, h^{12} V_1 V_5 + 23/75675600\, h^{14} {V_2}^3\big] Z \\
&+1/3465\, h^{11} V_2 V_5 - 1/103950\, h^{14} V_2 {V_3}^2 + 1/138600\, h^{12} {V_2}^3 \\
&+1/2772\, h^{12} V_3 V_5 + 1/154440\, h^{14} {V_1}^2 V_6 + 1/1081080\, h^{13} V_3 {V_2}^2 \\
&+1/5148\, h^{14} V_6 V_4 + 5/36036\, h^{13} V_3 V_6 - 5/36036\, h^{13} V_4 V_5 \\
&-19/1081080\, h^{13} {V_1}^2 V_5 - 1/5148\, h^{14} {V_5}^2 + 29/1801800\, h^{14} V_4 {V_2}^2 \\
&-17/540540\, h^{14} V_1 V_2 V_5
\end{aligned}$$

To evaluate the size of the step originating at X we take some trial h value and calculate

$$\Delta_{\text{loc}} = \max\{|\Delta u(h)|, |h\Delta u'(h)|, |\Delta v(h)/h|, |\Delta v'(h)|\} \tag{4.70}$$

at some value of Z chosen such that the above deviations reach their maximum. Actually, we observed that a reasonable choice is the following : let

$$Z_m = \max\{|V_0 - E_{\text{min}}|, |V_0 - E_{\text{max}}|\} \tag{4.71}$$

where $E_{\min}$ and $E_{\max}$ are the lower and the upper bound, respectively, of the eigenvalue spectrum, fixed by input. If $Z_m \leq \pi^2/h^2$ then $Z = Z_m$, otherwise $Z = \pi^2/h^2$.
The deviation $\Delta u'$ is multiplied by h and the deviation Δv is divided by h in order to ensure a uniform dimension of the compared quantities.

We define a new h as

$$h_{\text{new}} = h(tol/\Delta_{\text{loc}})^{1/10} \tag{4.72}$$

and examine $H = |h_{\text{new}}/h - 1|$. If $H > 0.1$ the procedure is repeated with $h = h_{\text{new}}$. If $H \leq 0.1$, h is accepted and the procedure continues to compute the stepsize of the next interval, which originates at $X + h$.

4.4 Eigenvalue computation

As for the CPM, a shooting procedure can be formulated to compute the eigenvalues. As described in section 2.3.1, this means that the Schrödinger equation is integrated forwards and backwards from the two boundary points and the eigenvalues are found from the matching condition

$$y_L(x_m)y_R'(x_m) - y_R(x_m)y_L'(x_m) = 0. \tag{4.73}$$

The roots of this mismatch function can be found using an iterative procedure as the bisection method, or more preferably the iterative procedure suggested by Blatt in [25]. The LPM[4,2] method was implemented together with the shooting procedure in Fortran. The program includes not only the analytic expressions of the perturbation corrections of section 4.2.1 but also the asymptotic expressions described in section 4.3.2. In addition, a MATLAB version of the LPM[4,2] method was developed, which can be used to compute the eigenvalues of a regular Schrodinger problem. This MATLAB version also includes a Prüfer procedure, similar as the one used for the CPM in section 3.2.4, which allows to compute the index of an eigenvalue. The Fortran code applied on the Woods-Saxon problem as well as the MATLAB program are available at [2].

4.5 Some illustrations

As first test potential we again consider the Woods-Saxon potential defined by

$$V(x) = v_0 w(x)\left(1 - \frac{1-w(x)}{a_0}\right) \tag{4.74}$$

with $w(x) = \{1 + \exp[(x - x_0)/a_0]\}^{-1}, v_0 = -50, x_0 = 7, a_0 = 0.6, x \in [0, x_f = 15]$. We computed the eigenvalues $E_0, \ldots, E_{13}$, that is we considered $E \in (-50, 0)$ with the boundary conditions

$$a_0 y(0) + b_0 y'(0) = 0 \tag{4.75}$$

$$a_1 y(x_f) + b_1 y'(x_f) = 0, \tag{4.76}$$

Table 4.2: Woods-Saxon potential: errors ΔE_k at several equidistant steps: versions Gordon (upper entries) and LPM(0) (lower entries).

k	E_k	$h = 1/4$	$h = 1/8$	$h = 1/16$
0	-49.45778872808258	$9.91(-4)$	$2.46(-4)$	$6.15(-5)$
		$5.08(-6)$	$3.16(-7)$	$1.98(-8)$
1	-48.14843042000636	$2.91(-3)$	$7.23(-4)$	$1.81(-4)$
		$1.29(-5)$	$8.07(-7)$	$5.04(-8)$
2	-46.29075395446608	$5.22(-3)$	$1.30(-3)$	$3.24(-4)$
		$1.83(-5)$	$1.14(-6)$	$7.13(-8)$
3	-43.96831843181423	$7.69(-3)$	$1.92(-3)$	$4.79(-4)$
		$1.81(-5)$	$1.13(-6)$	$7.05(-8)$
4	-41.23260777218022	$1.01(-2)$	$2.53(-3)$	$6.31(-4)$
		$1.04(-5)$	$6.47(-7)$	$4.04(-8)$
5	-38.12278509672792	$1.23(-2)$	$3.08(-3)$	$7.71(-4)$
		$-5.30(-6)$	$-3.36(-7)$	$-2.11(-8)$
6	-34.67231320569966	$1.41(-2)$	$3.54(-3)$	$8.85(-4)$
		$-2.78(-5)$	$-1.76(-6)$	$-1.10(-7)$
7	-30.91224748790885	$1.53(-2)$	$3.85(-3)$	$9.62(-4)$
		$-5.44(-5)$	$-3.43(-6)$	$-2.15(-7)$
8	-26.87344891605987	$1.57(-2)$	$3.95(-3)$	$9.90(-4)$
		$-8.00(-5)$	$-5.06(-6)$	$-3.17(-7)$
9	-22.58860225769321	$1.51(-2)$	$3.81(-3)$	$9.53(-4)$
		$-9.83(-5)$	$-6.23(-6)$	$-3.91(-7)$
10	-18.09468828212442	$1.33(-2)$	$3.35(-3)$	$8.38(-4)$
		$-1.01(-4)$	$-6.43(-6)$	$-4.03(-7)$
11	-13.43686904025008	$9.98(-3)$	$2.51(-3)$	$6.28(-4)$
		$-7.97(-5)$	$-5.09(-6)$	$-3.20(-7)$
12	-8.67608167073655	$4.84(-3)$	$1.21(-3)$	$3.02(-4)$
		$-2.67(-5)$	$-1.75(-6)$	$-1.11(-7)$
13	-3.90823248120623	$-2.54(-3)$	$-6.59(-4)$	$-1.66(-4)$
		$5.88(-5)$	$3.65(-6)$	$2.28(-7)$

where $a_0 = 1, b_0 = 0, a_1 = \sqrt{V(x_f) - E}, b_1 = 1$. Note that the condition at x_f is slightly different from the standard Sturm-Liouville formulation since at least one coefficient is not a constant but energy dependent.

Some numerical results are presented in the Tables 4.2 and 4.3. Four different methods were used as propagation methods in a shooting procedure to compute the eigenvalues: Gordon's original method (that is LPM(0) with the local linear approximation $\bar{V}(X+\delta) = V(X+h/2) + (\delta - h/2)V'(X+h/2)$, $\delta \in [0, h]$), LPM(0) (Table 4.2), LPM[4,1] and LPM[4,2] (Table 4.3). The calculations were done in Fortran using double precision arithmetic. We give the errors in the eigenvalues for different (uniform) stepsizes ($h = 1/4, 1/8, 1/16$) and different numbers of perturbation corrections. For each eigenvalue E_k the error is defined as $\Delta E_k = E_k^{exact} - E_k^{calc}(h)$.

One can see the substantial gain in accuracy produced when introducing more per-

Table 4.3: Woods-Saxon potential: errors ΔE_k at several equidistant steps: versions LPM$[4,1]$ and LPM$[4,2]$.

	LPM[4,1]			LPM[4,2]	
k	$h = 1/4$	$h = 1/8$	$h = 1/16$	$h = 1/4$	$h = 1/8$
0	$-6.16(-9)$	$-9.38(-11)$	$-1.46(-12)$	$-2.13(-13)$	$< 1.0(-14)$
1	$-3.58(-8)$	$-5.46(-10)$	$8.46(-12)$	$-2.21(-12)$	$2.1(-14)$
2	$-1.05(-7)$	$-1.60(-9)$	$-2.48(-11)$	$-8.57(-12)$	$< 1.0(-14)$
3	$-2.21(-7)$	$-3.35(-9)$	$-5.20(-11)$	$-2.23(-11)$	$-2.1(-14)$
4	$-3.76(-7)$	$-5.67(-9)$	$-8.78(-11)$	$-4.45(-11)$	$-5.0(-14)$
5	$-5.48(-7)$	$-8.23(-9)$	$-1.27(-10)$	$-7.08(-11)$	$-8.0(-14)$
6	$-7.08(-7)$	$-1.06(-8)$	$-1.63(-10)$	$-9.34(-11)$	$-1.1(-13)$
7	$-8.24(-7)$	$-1.21(-8)$	$-1.86(-10)$	$-1.05(-10)$	$-1.2(-13)$
8	$-8.72(-7)$	$-1.27(-8)$	$-1.93(-10)$	$-9.44(-11)$	$-1.1(-13)$
9	$-8.52(-7)$	$-1.21(-8)$	$-1.82(-10)$	$-5.89(-11)$	$-8.5(-14)$
10	$-7.99(-7)$	$-1.10(-8)$	$-1.64(-10)$	$-3.10(-11)$	$-6.7(-14)$
11	$-7.94(-7)$	$-1.08(-8)$	$-1.59(-10)$	$7.76(-12)$	$-3.4(-14)$
12	$-9.63(-7)$	$-1.34(-8)$	$-1.98(-10)$	$1.41(-10)$	$3.2(-14)$
13	$-1.46(-6)$	$-2.13(-8)$	$-3.20(-10)$	$3.15(-10)$	$1.9(-13)$

turbation corrections. The data at different steps also confirm the prediction of the error analysis that Gordon's method is of order 2, LPM(0) is of order 4 and LPM[4,1] of order 6 (see an illustration in Figure 4.5). A full confirmation of the order is impossible for LPM[4,2] because with this version we practically get 14 exact decimal digits already at $h = 1/8$ such that the error at $h = 1/16$ would be beyond the limit accessible in double precision. This is why the column corresponding to $h = 1/16$ is no more included in Table 4.3 for LPM[4,2].

The experimental evidence for the order, based on the errors at $2h$ and h, is reliable only when h is sufficiently small and the data are not altered significantly by round-off errors. For the low-order versions both conditions hold for $h = 1/8$ or $1/16$, but for the version LPM[4,2] of order 10 the accuracy in the eigenvalues is often inside the round-off limits for double precision calculations, especially at the two ends of the spectrum. For this reason only data from the middle part of the spectrum have been presented on Figure 4.5. Concerning the Gordon method, the low order (two) is a consequence of the linear approximation of the potential by a Taylor series instead of the best fit polynomial approximation, i.e. by shifted Legendre polynomials.

We used the Fortran implementation of the LPM[4,2] method to solve three eigenvalue problems for the Schrödinger equation. Now the automatic stepsize selection is applied. The three problems are: (i) the Woods-Saxon potential, (ii) the Paine potential $V(x) = 1/(x{+}0.1)^2$, $x \in [0, \pi]$, with the boundary conditions $y(0) = y(\pi) = 0$, (iii) the Mathieu potential $V(x) = 2\cos(2x)$, $x \in [0, \pi]$ with the boundary conditions $y(0) = y(\pi) = 0$. For problems (ii) and (iii) only a selection of eigenvalues was investigated.

To check for the validity of the rule for the stepsize adjustment we carried out computations at three values of the tolerance viz. $tol = 10^{-10}$, 10^{-12} and 10^{-14}. To get an

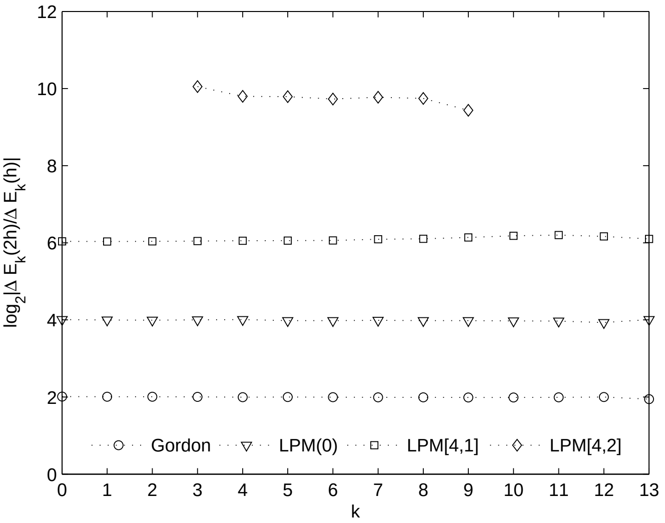


Figure 4.5: Woods-Saxon potential: experimental evidence for the method order: variation with k of $\log_2 |\Delta E_k(2h)/\Delta E_k(h)|$ at $h = 1/8$.

Table 4.4: Woods-Saxon potential: absolute errors ΔE_k at different input tolerances. $nint$ is the number of steps.

k	$tol = 10^{-10}$	$tol = 10^{-12}$	$tol = 10^{-14}$
0	$-3.1(-12)$	$4.0(-14)$	$7(-14)$
1	$-2.4(-11)$	$-5.3(-13)$	$2(-14)$
2	$-8.6(-11)$	$-1.9(-12)$	$-3(-14)$
3	$-1.7(-10)$	$-4.0(-12)$	$-5(-14)$
4	$-2.1(-10)$	$-4.9(-12)$	$-6(-14)$
5	$-1.5(-10)$	$-3.6(-12)$	$-5(-14)$
6	$-1.5(-10)$	$-3.1(-12)$	$-6(-14)$
7	$-1.8(-10)$	$-4.3(-12)$	$-8(-14)$
8	$-1.5(-10)$	$-5.0(-12)$	$-8(-14)$
9	$6.4(-12)$	$-3.8(-12)$	$-8(-14)$
10	$5.0(-11)$	$-2.7(-12)$	$-8(-14)$
11	$-4.1(-12)$	$-3.9(-12)$	$-8(-14)$
12	$-3.7(-12)$	$3.3(-12)$	$-1(-14)$
13	$8.6(-11)$	$2.8(-12)$	$8(-14)$
$nint$	42	64	96

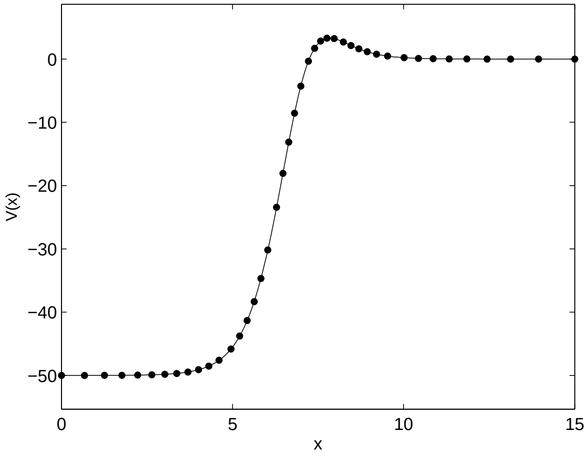


Figure 4.6: The Woods-Saxon potential: the dots represent the values at the meshpoints of the partition consistent with a tolerance of 10^{-10}.

Table 4.5: Paine potential: absolute errors ΔE_k at different input tolerances. $nint$ is the number of steps.

k	E_k	$tol = 10^{-10}$	$tol = 10^{-12}$	$tol = 10^{-14}$
0	1.519865821099347	2.7(−13)	2.3(−13)	−7.8(−14)
10	123.4977068009282	−9.9(−10)	−4.6(−12)	−7.3(−12)
20	443.8529598351504	5.4(−10)	2.0(−11)	−2.0(−13)
30	963.9644462621100	−1.5(−08)	4.5(−12)	1.0(−12)
40	1684.012014337853	2.1(−08)	−1.1(−10)	< 1(−12)
50	2604.036332024594	−1.7(−09)	1.2(−10)	< 1(−12)
$nint$		23	40	67

idea on the distribution of the meshpoints resulting from the stepsize adjustment, we give in Figure 4.6 the partition sample corresponding to $tol = 10^{-10}$ for the Woods-Saxon potential.

The absolute errors $\Delta E_k = E_k^{exact} - E_k^{calc}$ are collected on Tables 4.4-4.6.

A first remark is that the data from problems (ii) and (iii) involve sets of eigenvalues with uncommonly large values but no systematic deterioration of the accuracy is observed as k is increased. As a rule the results around some mid-lying k, $k = 30$ for these problems, are the least accurate but when k is further increased the accuracy tends to improve again. This is a general behaviour with the piecewise perturbation methods. We also see that, as expected, the maximal error along the spectrum at $tol = 10^{-10}$ is approximately by two orders of magnitude bigger than the one at 10^{-12} and the same holds if the data at

Table 4.6: Mathieu potential: absolute errors ΔE_k at different input tolerances. $nint$ is the number of steps.

k	E_k	$tol = 10^{-10}$	$tol = 10^{-12}$	$tol = 10^{-14}$
0	-0.1102488169920971	$-1.4(-11)$	$-3.0(-14)$	$5.8(-14)$
10	121.0041667612691	$-2.5(-10)$	$-6.0(-13)$	$< 1(-13)$
20	441.0011363654933	$9.6(-10)$	$-2.2(-11)$	$< 1(-13)$
30	961.0005208335109	$-2.3(-08)$	$-1.1(-10)$	$-2(-13)$
40	1681.000297619081	$3.5(-09)$	$1.0(-10)$	$< 1(-12)$
50	2601.000192307701	$-3.4(-10)$	$1.0(-10)$	$-1(-12)$
$nint$		15	28	52

$tol = 10^{-12}$ and 10^{-14} are compared but here we must be aware that the latter data are often within the roundoff limit.
It is perhaps worth noticing that one may expect that the order of magnitude of the maximal error in the eigenvalues must be equal to tol but there is no solid basis for such an expectation. This is because tol is used to control the error in the wave function, not in the eigenenergy. The two are certainly related somehow but they may be very different in magnitude. Expressed in other words, although for the three problems considered here the order of magnitude of the maximal error in the eigenvalues happens to agree with tol, this is not a general property. The only behaviour which has to be normally expected is that the ratio of the maximal error at two tolerances is close to the ratio of the tolerances.

A final set of tests was aimed at comparing the LPM[4,2] with a CPM. We compared the LPM Fortran implementation with the Fortran implementation of the CPM{12,10} method (the SLCPM12 package [61]). The order of the CPM{12,10} (twelve) is close to that of the LPM[4,2] (ten). The number of steps consistent with similar accuracies was slightly in the favour of the CPM version, which is normal because the order is higher. A major difference was detected for the computational effort. We observed that the CPU time / step for the LPM[4,2] is about fifteen times bigger than for the CPM{12,10}. Again, this was not a surprise: the zeroth order propagators of the CPM are the Fortran functions $\sin$, $\cos$ or $\sinh$, $\cosh$ while for the LPM they are Airy functions, whose computation requires adequate software. Also the formulae of the CPM for perturbation corrections are much shorter and easier to compute than those of the LPM.

4.6 Conclusion

In this chapter, we investigated the old problem of improving the accuracy of the LPM for the Schrödinger equation by adding perturbation corrections to the algorithm. We effectively constructed the first and the second order corrections. We also performed the error analysis to predict that the introduction of successive corrections substantially enhances the order of the method from four, for the zeroth order version, to six and ten when the first and the second order corrections are included. In order to remove the effect of the accuracy loss due to near-cancellation of like-terms when evaluating the perturbation corrections we constructed alternative asymptotic formulae using a Maple code.

Numerical tests confirmed that the LPM versions share the general property of the piecewise perturbation methods of producing eigenvalues with uniform accuracy over large sets for the index k. However, the LPM approach does not seem to be more convenient in practice than CPM since the evaluation of the Airy functions on a computer is not only more difficult but also much more time consuming than the evaluation of the trigonometric or hyperbolic functions required by the CPM algorithm. The right way of using them in practice consists perhaps in activating them only in the subintervals where the potential exhibits a strong variation.

Chapter 5

Solving systems of coupled Schrödinger equations

The successful CPM$\{P,N\}$ methods for the one-dimensional time-independent Schrödinger problem are generalized to the coupled channel case. The derivation of the formulae is discussed and a Maple program code is presented which allows us to determine the analytic expressions of the perturbation corrections needed to construct methods of different orders. As for the one-dimensional problem, we apply the generalized CPM in a shooting procedure to compute the eigenvalues of the matrix boundary value problem. A generalization of the Prüfer method for scalar Sturm-Liouville problems makes the whole procedure more robust and allows us to specify the required eigenvalue by its index.

5.1 Introduction

A coupled channel Schrödinger equation is a system of linear ordinary differential equations of the second order obtained after separating the so-called scattering (or radial) coordinate from the rest of variables in the multidimensional Schrödinger equation describing the motion of an atomic or molecular system (see [47]). Such a time-independent Schrödinger equation may be written as

$$H\Psi = \epsilon\Psi \tag{5.1}$$

with the Hamiltonian

$$H(R,\Omega) = -\frac{\hbar^2}{2\mu}\frac{d^2}{dR^2} + V(R,\Omega), \tag{5.2}$$

where ϵ is the energy of the system, μ is the appropriate reduced mass, R is the 'radial' coordinate describing the separation of two particles, Ω is the set of 'angular' coordinates which describe the 'internal motion' of a system, and $V(R,\Omega)$ is the interaction potential.

Problems of this type frequently arise from the interactions of pairs of molecules, but also occur in electronic structure theory and nuclear physics.

There are some different approaches to find the eigenvalues and eigenfunctions of Hamiltonians as (5.2). In the so-called *coupled channel* approach, the total wave function $\Psi(R,\Omega)$ for the kth state is expanded over the complete orthonormal set of the basis angular functions $\{\Phi_j(\Omega)\}_{j=1}^{\infty}$:

$$\Psi_k(R,\Omega) = \sum_{j=1}^{\infty} \psi_{jk}(R)\Phi_j(\Omega). \tag{5.3}$$

The wavefunction in each *channel* j is described by a radial *channel function* $\psi_{jk}(R)$. The expansion (5.3) is substituted into the total Schrödinger equation, and the result is projected onto a basis function $\Phi_i(\Omega)$. Taking advantage of the orthonormality of the $\Phi_j(\Omega)$, we obtain a differential equation for the channel function $\psi_{ik}(R)$,

$$\frac{d^2\psi_{ik}}{dR^2} = \sum_j [Q_{ij}(R) - E\delta_{ij}]\psi_{jk}(R), \tag{5.4}$$

where δ_{ij} is the Kronecker delta, E is the energy scaled by $2\mu/\hbar^2$ (that is $E = (2\mu/\hbar^2)\epsilon$), and

$$Q_{ij}(R) = \frac{2\mu}{\hbar^2}\int \Phi_i^*(\Omega)V(R,\Omega)\Phi_j(\Omega)d\Omega. \tag{5.5}$$

Here, * denotes the complex conjugate. A similar equation arises for each channel, and the different equations are coupled by the off-diagonal terms $Q_{ij}(R)$ with $i \neq j$.

Having chosen M angular basis functions as an adequate approximate representation of Ψ we can truncate the infinite sum in Eq. (5.3). This approximation is known as the *close-coupling* approximation: the name indicates that only channels that are 'close' to one another in some sense are retained. The Schrödinger equation (5.1) then reduces to a system of M coupled differential equations which can be written in matrix form as

$$\frac{d^2\psi}{dR^2} = [\mathbf{Q}(R) - E\mathbf{I}]\,\psi(R), \tag{5.6}$$

where $\psi(R)$ is a column vector of order M with elements $\psi_{jk}(R)$, $\mathbf{I}$ is the $M \times M$ unit matrix, and $\mathbf{Q}$ is the symmetric $M \times M$ potential matrix with elements $Q_{ij}(R)$. The particular choice of the basis functions $\Phi_j(\Omega)$ and the resulting form of the coupling matrices $Q_{ij}(R)$ depend on the physical problem being considered.

There are various approaches to the solution of the coupled equations (5.6) (see a.o. [5, 6, 47, 81, 113]). In the more early work approximate schemes were used which attempt to reduce the coupled equations to a set of one-dimensional problems (e.g. in [81]). A more modern approach is to propagate the solutions numerically, without reducing them to a one-dimensional form. A large number of numerical methods have been suggested for carrying out the propagation. However when bound state boundary conditions are applied, acceptable solutions of the coupled equations exist only when E is an eigenvalue of the Hamiltonian and additional techniques are needed to locate the eigenvalues. Early methods for doing this were developed by Gordon [41] and Johnson [65].

It was already decribed in [58] by Ixaru that a piecewise perturbation method can be constructed for a system of differential equations. In [59] a CPM-based method was formulated for systems of coupled Schrödinger equations and implemented by the Fortran program LILIX (available under the identifier ADQF_v1_0 in the CPC library [1]). However using the CPM formulation implemented by LILIX it is very difficult to construct high order correction terms and only a limited number of correction terms was calculated and included in the LILIX program. Using the symbolic software package Maple we are now able to determine the analytic expressions of additional perturbation corrections and to formulate the natural extension of the CPM$\{P, N\}$ methods to the coupled channel case. These extensions preserve the important features of the one-dimensional CPM$\{P, N\}$ method such as the uniform accuracy with respect to the energy E and the big step widths. The generalized CPM$\{P, N\}$ as discussed in this chapter, include some additional improvements over the LILIX method. One such improvement is that more information associated to the partition is calculated at the very beginning of the procedure and stored (i.e. the **C**-matrices, see further). When the CPM method is used in a shooting method for the generation of the eigenvalues, this improvement will certainly speed up the eigenvalue search.

5.2 Generalized CPM$\{P, N\}$ methods

5.2.1 Brief description of the procedure

Consider the initial value problem for the coupled channel Schrödinger equation with n channels:

$$\mathbf{y}'' = (\mathbf{V}(x) - E\mathbf{I})\,\mathbf{y}, \quad x \in [a, b] \tag{5.7}$$

where **I** is the $n \times n$ unity matrix. When the domain ends a and b are complex numbers, $x \in [a, b]$ means that x is placed along the line segment joining a and b in the complex plane. The $n \times n$ matrix $\mathbf{V}(x)$ is assumed symmetric, i.e. $V_{ij}(x) = V_{ji}(x), i, j = 1, 2, ...n$ and it is also assumed that each component of this matrix is a well behaved function of the argument x. $\mathbf{y}$ is a set of $nsol$ column vectors with n components and $nsol \leq n$ represents the number of different (generally linear independent) solutions needed.

A partition of $[a, b]$ is introduced, with the mesh points $x_0 = a, x_1, x_2, ..., x_{\mathrm{nstep}} = b$. Let $I = [X, X + h]$ be the current one step interval of this partition. A transfer matrix is constructed, which allows a blockwise propagation of the solution $\mathbf{y}$ and of its first derivative $\mathbf{y}'$:

$$\begin{bmatrix} \mathbf{y}(X+h) \\ \mathbf{y}'(X+h) \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{y}(X) \\ \mathbf{y}'(X) \end{bmatrix}. \tag{5.8}$$

To construct $\mathbf{T}$ we use two particular solutions of the equation

$$\mathbf{p}'' = (\mathbf{V}(X + \delta) - E\mathbf{I})\,\mathbf{p}, \quad \delta \in [0, h]. \tag{5.9}$$

Specifically, if $\mathbf{u}(\delta)$ and $\mathbf{v}(\delta)$ are the $n \times n$ solutions corresponding to the initial conditions $\mathbf{p}(0) = \mathbf{I}$, $\mathbf{p}'(0) = \mathbf{0}$ and $\mathbf{p}(0) = \mathbf{0}$, $\mathbf{p}'(0) = \mathbf{I}$, respectively ($\mathbf{0}$ is the n by n zero matrix) then $\mathbf{T}$ has the form

$$\mathbf{T} = \begin{bmatrix} \mathbf{u}(\delta) & \mathbf{v}(\delta) \\ \mathbf{u}'(\delta) & \mathbf{v}'(\delta) \end{bmatrix}. \tag{5.10}$$

To determine $\mathbf{u}$ and $\mathbf{v}$ the potential matrix is approximated by a truncated series over the shifted Legendre polynomials $P_n^*(\delta/h)$. The used parametrization is

$$\mathbf{V}(X+\delta) = \sum_{m=0}^{N} \mathbf{V}_m h^m P_m^*(\delta/h) \tag{5.11}$$

where the matrix weights are calculated by quadrature ($\bar{\mathbf{V}}_m = \mathbf{V}_m h^{m+2}, m = 1, 2, ...$),

$$\begin{aligned} \mathbf{V}_0 &= \frac{1}{h}\int_0^h \mathbf{V}(X+\delta)d\delta, \\ \bar{\mathbf{V}}_m &= (2m+1)h\int_0^h \mathbf{V}(X+\delta)P_m^*(\delta/h)d\delta, \quad m = 1, 2, 3, \dots . \end{aligned} \tag{5.12}$$

The symmetric matrix $\mathbf{V}_0$ is then diagonalized and let $\mathbf{D}$ be the diagonalization matrix. In the $\mathbf{D}$ representation Eq. (5.9) becomes

$$\mathbf{p}^{\mathbf{D}''} = \left(\sum_{m=0}^{N} \mathbf{V}_m^{\mathbf{D}} h^m P_m^*(\delta/h) - E\mathbf{I} \right) \mathbf{p}^{\mathbf{D}}, \quad \delta \in [0, h] \tag{5.13}$$

and this is solved for $\mathbf{u}^{\mathbf{D}}$ and $\mathbf{v}^{\mathbf{D}}$; the initial conditions are the same as in the original representation. The perturbation procedure is used, in which the diagonal matrix $\mathbf{V}_{\text{o}}^{\mathbf{D}}$ is the reference potential and

$$\Delta\mathbf{V} = \sum_{m=1}^{N} \mathbf{V}_m^{\mathbf{D}} h^m P_m^*(\delta/h) \tag{5.14}$$

is the perturbation with $\mathbf{V}_m^{\mathbf{D}}$ symmetric matrices. The perturbation corrections can be determined analytically up to any order (see 5.2.2). Once the values at h of the $\mathbf{u}^{\mathbf{D}}$, $\mathbf{v}^{\mathbf{D}}$ matrices and of their derivatives have been evaluated, they are reconverted to the original representation to obtain the desired $\mathbf{T}$.

It is also possible to write the algorithm to advance the derivatives with respect to E of $\mathbf{y}$ and $\mathbf{y}'$. These derivatives are propagated by the partial derivative with respect to E of (5.8), that is

$$\begin{bmatrix} \mathbf{y_E}(X+h) \\ \mathbf{y_E}'(X+h) \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{y_E}(X) \\ \mathbf{y_E}'(X) \end{bmatrix} + \mathbf{T_E} \begin{bmatrix} \mathbf{y}(x) \\ \mathbf{y}'(x) \end{bmatrix}. \tag{5.15}$$

For the evaluation of the elements of $\mathbf{T_E}$ the analytic expressions of the partial derivatives of $\mathbf{u}^{\mathbf{D}}, \mathbf{u}^{\mathbf{D}'}, \mathbf{v}^{\mathbf{D}}, \mathbf{v}^{\mathbf{D}'}$ with respect to E are computed and reconverted to the original representation. As in the one-dimensional case, the knowledge of the first derivative with respect to E allows implementing a Newton-Raphson procedure for the localization of the eigenvalues of the boundary value problem associated to the coupled channel Schrödinger equation.

5.2.2 Construction of the perturbation corrections

We now describe the procedure used to construct the correction terms. First, the matrices of functions $\mathbf{u}^{\mathbf{D}}$ and $\mathbf{v}^{\mathbf{D}}$ are written as the perturbation series:

$$\mathbf{u}^{\mathbf{D}}(\delta) = \mathbf{u}_0(\delta) + \mathbf{u}_1(\delta) + \mathbf{u}_2(\delta) + \mathbf{u}_3(\delta) + \dots \tag{5.16}$$

$$\mathbf{v}^{\mathbf{D}}(\delta) = \mathbf{v}_0(\delta) + \mathbf{v}_1(\delta) + \mathbf{v}_2(\delta) + \mathbf{v}_3(\delta) + \dots \tag{5.17}$$

where $\mathbf{u}_0(\delta)$ and $\mathbf{v}_0(\delta)$ are the solutions of

$$\mathbf{p}_0'' = \left(\mathbf{V}_{\mathrm{o}}^{\mathbf{D}} - E\right)\mathbf{p}_0 \tag{5.18}$$

with $\mathbf{p}_0(0) = \mathbf{I}, \mathbf{p}_0'(0) = \mathbf{0}$ for $\mathbf{u}_0$ and $\mathbf{p}_0(0) = \mathbf{0}, \mathbf{p}_0'(0) = \mathbf{I}$ for $\mathbf{v}_0$. The $n \times n$ 'correction' matrices of functions $\mathbf{u}_q$ and $\mathbf{v}_q$ ($q = 1, 2, ...$) are the solutions of the systems

$$\mathbf{u}_q'' = (\mathbf{V}_{\mathrm{o}}^{\mathbf{D}} - E\mathbf{I})\mathbf{u}_q + \Delta\mathbf{V}(\delta)\mathbf{u}_{q-1} \tag{5.19}$$

$$\mathbf{v}_q'' = (\mathbf{V}_{\mathrm{o}}^{\mathbf{D}} - E\mathbf{I})\mathbf{v}_q + \Delta\mathbf{V}(\delta)\mathbf{v}_{q-1} \tag{5.20}$$

with vanishing initial conditions;

$$\mathbf{u}_q(0) = \mathbf{v}_q(0) = \mathbf{u}'_q(0) = \mathbf{v}'_q(0) = \mathbf{0}. \tag{5.21}$$

As for the one-dimensional CPM we will express the corrections in terms of the functions $\xi(Z), \eta_0(Z), \eta_1(Z), ...$ Note that when working with complex numbers, the complex extension [59, 63] of these functions can be used:

$$\xi(Z) = \cos(iZ^{1/2}) \tag{5.22}$$

and

$$\eta_0(Z) = \begin{cases} -i\sin(iZ^{1/2})/Z^{1/2} & \text{if } Z \neq 0\,, \\ 1 & \text{if } Z = 0\,. \end{cases} \tag{5.23}$$

With $Z_i(\delta) = (V_{0_{ii}}^D - E)\delta^2$, the zeroth order propagators $\mathbf{u}_0(\delta)$ and $\mathbf{v}_0(\delta)$ are diagonal matrices, defined as follows:

$$\mathbf{u}_0 = \mathbf{v}_0' = \boldsymbol{\xi}(\mathbf{Z}) \tag{5.24}$$

$$\delta\mathbf{u}_0' = \mathbf{Z}(\delta)\boldsymbol{\eta}_0(\mathbf{Z}) \tag{5.25}$$

$$\mathbf{v}_0 = \delta\boldsymbol{\eta}_0(\mathbf{Z}) \tag{5.26}$$

where

$$\mathbf{Z}(\delta) = (\mathbf{V}_{\mathrm{o}}^{\mathbf{D}} - E\mathbf{I})\delta^2 \tag{5.27}$$

and $\boldsymbol{\xi}(\mathbf{Z})$, $\boldsymbol{\eta}_m(\mathbf{Z})$ two $n \times n$ diagonal matrices of functions

$$\boldsymbol{\xi}(\mathbf{Z}) = \begin{bmatrix} \xi(Z_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \xi(Z_n) \end{bmatrix}, \tag{5.28}$$

$$\boldsymbol{\eta}_m(\mathbf{Z}) = \begin{bmatrix} \eta_m(Z_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \eta_m(Z_n) \end{bmatrix}. \tag{5.29}$$

The following iteration procedure exists to construct the corrections.

Correction $\mathbf{p}_{q-1}$ ($\mathbf{p} = \mathbf{u}, \mathbf{v}$) is assumed to be known and of such a form that the product $\Delta\mathbf{V}(\delta)\mathbf{p}_{q-1}$ reads

$$\Delta\mathbf{V}(\delta)\mathbf{p}_{q-1}(\delta) = \mathbf{Q}(\delta)\boldsymbol{\xi}(\mathbf{Z}) + \sum_{m=0}^{+\infty} \delta^{2m+1}\mathbf{R}_m(\delta)\boldsymbol{\eta}_m(\mathbf{Z}). \tag{5.30}$$

Then $\mathbf{p}_q(\delta)$ and $\mathbf{p}'{}_q(\delta)$ are of the form

$$\mathbf{p}_q(\delta) = \sum_{m=0}^{+\infty} \delta^{2m+1}\mathbf{C}_m(\delta)\boldsymbol{\eta}_m(\mathbf{Z}), \tag{5.31}$$

$$\mathbf{p}'_q(\delta) = \mathbf{C}_0(\delta)\boldsymbol{\xi}(\mathbf{Z}) + \sum_{m=0}^{+\infty} \delta^{2m+1}\left(\frac{d\mathbf{C}_m(\delta)}{d\delta} + \delta\mathbf{C}_{m+1}(\delta)\right)\boldsymbol{\eta}_m(\mathbf{Z}). \tag{5.32}$$

All $\mathbf{C}_m$ matrices are given by quadrature. To show this, we first differentiate each element of $\mathbf{p}_q(\delta)$ twice with respect to δ and form $\mathbf{P} = \mathbf{p}''_q(\delta) - (\mathbf{V}_{\text{o}}^{\mathbf{D}} - E\mathbf{I})\mathbf{p}_q$. One gets

$$\begin{aligned}\mathbf{P} \;=\; & 2\frac{d\mathbf{C}_0}{d\delta}\boldsymbol{\xi}(\mathbf{Z}) + \delta\left(\frac{d^2\mathbf{C}_0}{d\delta^2} + 2\delta\frac{d\mathbf{C}_1}{d\delta} + 2\mathbf{C}_1 + [\mathbf{C}_0, \mathbf{V}_{\text{o}}^{\mathbf{D}}]\right)\boldsymbol{\eta}_0(\mathbf{Z}) + \dots \\ & + \delta^{2m+1}\left(\frac{d^2\mathbf{C}_m}{d\delta^2} + 2\delta\frac{d\mathbf{C}_{m+1}}{d\delta} + 2(m+1)\mathbf{C}_{m+1} + [\mathbf{C}_m, \mathbf{V}_{\text{o}}^{\mathbf{D}}]\right)\boldsymbol{\eta}_m(\mathbf{Z}) \\ & + \dots\end{aligned} \tag{5.33}$$

where $[\mathbf{C}_m, \mathbf{V}_{\text{o}}^{\mathbf{D}}]$ is the commutator of the matrices $\mathbf{C}_m$ and $\mathbf{V}_{\text{o}}^{\mathbf{D}}$. Upon identifying the coefficients in (5.30) and (5.33) we get

$$2\frac{d\mathbf{C}_0}{d\delta} = \mathbf{Q}(\delta) \tag{5.34}$$

and

$$\frac{d^2\mathbf{C}_m}{d\delta^2} + 2\delta\frac{d\mathbf{C}_{m+1}}{d\delta} + 2(m+1)\mathbf{C}_{m+1} + [\mathbf{C}_m, \mathbf{V}_{\text{o}}^{\mathbf{D}}] = \mathbf{R}_m \tag{5.35}$$

or

$$\frac{d^2\mathbf{C}_{m-1}}{d\delta^2}\delta^{m-1} + 2\frac{d\delta^m\mathbf{C}_m}{d\delta} + [\mathbf{C}_{m-1}, \mathbf{V}_{\text{o}}^{\mathbf{D}}]\delta^{m-1} = \mathbf{R}_{m-1}\delta^{m-1}. \tag{5.36}$$

$\mathbf{C}_0$ is then given by

$$\mathbf{C}_0(\delta) = \frac{1}{2}\int_0^\delta \mathbf{Q}(\delta_1)d\delta_1 \tag{5.37}$$

and $\mathbf{C}_m$, $m = 1, 2, 3, ...$ results as

$$\mathbf{C}_m(\delta) = \frac{1}{2}\delta^{-m} \int_0^{\delta} \delta_1^{m-1} \left(\mathbf{R}_{m-1}(\delta_1) - \frac{d^2\mathbf{C}_{m-1}(\delta_1)}{d\delta_1^2} - [\mathbf{C}_{m-1}(\delta_1), \mathbf{V}_0^{\mathbf{D}}] \right) d\delta_1. \tag{5.38}$$

To calculate successive corrections for $\mathbf{u}$, the starting functions in $\Delta\mathbf{V}(\delta)\mathbf{p}_0(\delta)$ are $\mathbf{Q}(\delta) = \Delta\mathbf{V}$ and $\mathbf{R_0}(\delta) = \mathbf{R}_1(\delta) = ... = \mathbf{0}$. For $\mathbf{v}$ the starting functions are $\mathbf{Q}(\delta) = \mathbf{0}$, $\mathbf{R}_0(\delta) = \Delta\mathbf{V}(\delta)$, $\mathbf{R}_1(\delta) = \mathbf{R}_2(\delta) = ... = \mathbf{0}$.

In Appendix A.2 we give the expressions of $\mathbf{u^D}(h)$, $\mathbf{u^D}'(h)$, $\mathbf{v^D}(h)$ and $\mathbf{v^D}'(h)$ obtained by Maple with a sufficient number of terms to generate the CPM{10,8} algorithm of maximum order ten at low energies and order eight at the asymptotic regime. This means that the terms in (A.5)-(A.8) are collected on the basis that only contributions proportional to $h^p, p \leq 10$ are retained. For those also interested in the full expressions of $\mathbf{u^D}(h)$, $\mathbf{u^D}'(h)$, $\mathbf{v^D}(h)$ and $\mathbf{v^D}'(h)$ for other CPM$\{P, N\}$ algorithms we give in appendix B.3 the source of the Maple program by which these expressions can be generated. The E-independent coefficient matrices $\mathbf{C}^{(u)}$, $\mathbf{C}^{(u')}$, $\mathbf{C}^{(v)}$ and $\mathbf{C}^{(v')}$ in (A.5)-(A.8) are computed only once on each step and are stored. When the solution for a given E is advanced on successive steps, only the E dependent ξ and η_m remain to be calculated. This is an important difference with the LILIX method [59] where the correction terms are constructed during propagation, i.e. in the LILIX package only the $\mathbf{V}$-matrices are calculated and stored prior to the propagation.

The expressions of the coefficient matrices in (A.5)-(A.8) contain many commutators of two matrices. Note that in the one-dimensional case all these commutators are zero and the same expressions are obtained as for the CPM$\{P, N\}$ methods described for the one-dimensional problem.

5.2.3 Stepsize selection

We want to construct a partition with nonequal steps whose widths are consistent with a preset tolerance tol. A procedure is used which is very analogously to the stepsize selection for the one-dimensional case discussed in [60, 75]. The evaluation of the step lengths in terms of only the leading term of the one-step error is usually unsatisfactory. Several terms must be used instead. The terms generated by the Maple code in Appendix A.2 do not allow expressing the error for the considered CPM$\{P, N\}$ but they allow it for weaker versions CPM$\{P', N'\}$. Let us focus on the CPM{10,8} method: all terms in the expressions for $\mathbf{u^D}(h), h\mathbf{u^D}'(h)$, $\mathbf{v^D}(h)/h$ and $\mathbf{v^D}'(h)$ which are supplementary to the terms to be used in the weaker CPM{8,4}-version are used to construct an estimation of the error. To start with, we take a trial value h for the size of the step originating at X and use a Gauss quadrature formula to calculate the matrices $\bar{\mathbf{V}}_0, \bar{\mathbf{V}}_1, ..., \bar{\mathbf{V}}_8$, directly by

$$\bar{\mathbf{V}}_i = (2i+1)h \int_0^h \mathbf{V}(X+\delta) P_i^*(\delta/h) d\delta, \qquad i = 0, 1, \ldots 8. \tag{5.39}$$

It is sufficient to take eight points in the quadrature formula. After diagonalization of the $\mathbf{V}_0$ matrix, we obtain the matrices in the $\mathbf{D}$ representation: $\bar{\mathbf{V}}_0^{\mathbf{D}}$, $\bar{\mathbf{V}}_1^{\mathbf{D}}$,...,$\bar{\mathbf{V}}_8^{\mathbf{D}}$. Since the $\eta(Z(h))$-functions obtain their maximum value in $Z(h) = 0$, we compute ϵ which is defined as

$$\epsilon = \max\big(|\Delta u(h)|, |\Delta u'(h)h|, |\Delta v(h)/h|, |\Delta v'(h)|\big),$$

at $Z(h) = 0$. Herein $\Delta u(h)$, $\Delta u'(h)$, $\Delta v(h)$ and $\Delta v'(h)$ are determined by the terms in the equations in appendix B (or generated by the Maple code), which are additional to the terms of CPM$\{8,4\}$. That is, all terms where either (i) the $\bar{\mathbf{V}}_i^{\mathbf{D}}$'s have $N' = 4 < i \leq N = 8$ or (ii) where the degree d in h satisfies $P' = 8 < d \leq P = 10$ (whereby the degree of $\bar{\mathbf{V}}_i$ in h is $i+2$). For $\Delta u(h)$ e.g. we have (where the upper label $\mathbf{D}$ is suppressed)

$$\begin{aligned}
\Delta u(h) =& \\
&\max\Big(-(\bar{\mathbf{V}}_5+\bar{\mathbf{V}}_7)/2+[\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_2]/280\Big)\,\eta_1(Z(h))\\
&+\max\Big((14\bar{\mathbf{V}}_5+27\bar{\mathbf{V}}_7)/2-\bar{\mathbf{V}}_3^2/56+[\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_2]/280+[3\bar{\mathbf{V}}_5-3\bar{\mathbf{V}}_6,\bar{\mathbf{V}}_0]/24\\
&\qquad-\big[[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_1],\bar{\mathbf{V}}_1\big]/1680-\big[[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_2\big]/3360+\big[[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_2],\bar{\mathbf{V}}_0\big]/480\\
&\qquad-\big[[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_3\big]/1680+\big[[\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_1\big]/1120\Big)\,\eta_2(Z(h))\\
&+\max\Big((-63\bar{\mathbf{V}}_5-297\bar{\mathbf{V}}_7)/2+\bar{\mathbf{V}}_3^2/14+\{\bar{\mathbf{V}}_1,+2\bar{\mathbf{V}}_4+\bar{\mathbf{V}}_5\}/8+\{\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_3\}/4\\
&\qquad+4[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_3]/35+\big(13[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_2]\bar{\mathbf{V}}_1+\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_2]\big)/3360\\
&\qquad+[-15\bar{\mathbf{V}}_5+18\bar{\mathbf{V}}_6,\bar{\mathbf{V}}_0]/8-\big[[+5\bar{\mathbf{V}}_3-5\bar{\mathbf{V}}_4,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big]/160\\
&\qquad-\big(6\bar{\mathbf{V}}_2[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]+3[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1+3\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0]-4[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]\bar{\mathbf{V}}_2\big)/480\\
&\qquad+\big(4\bar{\mathbf{V}}_2[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0]+3[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0]\bar{\mathbf{V}}_2\big)/560+\big[[[3\bar{\mathbf{V}}_1-\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big]/1920\\
&\qquad-\big(41[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]\bar{\mathbf{V}}_3+29\bar{\mathbf{V}}_3[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]+9\,\big[\bar{\mathbf{V}}_1,[\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_0]\big]\,\big)/3360\\
&\qquad+\Big[\big[[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big],\bar{\mathbf{V}}_1\Big]/1920+[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]^2/1152\Big)\,\eta_3(Z(h))\\
&+\max\Big(1287\bar{\mathbf{V}}_7/2-15\bar{\mathbf{V}}_3^2/56+\bar{\mathbf{V}}_1^3/48-\{\bar{\mathbf{V}}_1,21\bar{\mathbf{V}}_4+9\bar{\mathbf{V}}_5\}/8\\
&\qquad+\{\bar{\mathbf{V}}_2,-15\bar{\mathbf{V}}_3+3\bar{\mathbf{V}}_4\}/8+\bar{\mathbf{V}}_1\{\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_2\}/80+3[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_3]/56+[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_4]/8\\
&\qquad-(5\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_2]+2[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_2]\bar{\mathbf{V}}_1)/560+[72\bar{\mathbf{V}}_5-99\bar{\mathbf{V}}_6,\bar{\mathbf{V}}_0]/8\\
&\qquad+\Big(2\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_0]+6[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]\bar{\mathbf{V}}_3+\bar{\mathbf{V}}_3[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]+5[\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1\Big)/112\\
&\qquad+\big[[\bar{\mathbf{V}}_3-2\bar{\mathbf{V}}_4,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big]/8+\big[[[2\bar{\mathbf{V}}_1-3\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big]/480\\
&\qquad-\big[[\bar{\mathbf{V}}_2\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big]/10-\Big(4[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]\bar{\mathbf{V}}_2+6\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0]-3[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1\Big)/80\\
&\qquad+\Big(9[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0]\bar{\mathbf{V}}_2+12\bar{\mathbf{V}}_2[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0]\Big)/140+\big[[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big]\,\bar{\mathbf{V}}_1/480\\
&\qquad+[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]^2/240\Big)\,\eta_4(Z(h))
\end{aligned}$$

where $\max(\mathbf{M})$ is the absolute value of the matrix element of $\mathbf{M}$ with the largest absolute value and the notations $[\mathbf{A},\mathbf{B}]$ and $\{\mathbf{A},\mathbf{B}\}$ are used to denote the commutator and anticommutator of the matrices $\mathbf{A}$ and $\mathbf{B}$. The error estimate ϵ at $Z(h) = 0$ is then used to construct a new stepsize:

$$h_{\text{new}} = h(tol/\epsilon)^{1/9}. \tag{5.40}$$

When $|h_{\text{new}}/h - 1| > 0.1$ the procedure is repeated with $h = h_{\text{new}}$. Otherwise h is accepted to be a good choice for the stepsize and the procedure starts computing the stepsize of the next interval which will originate at $X + h$.

A very important property of the CPM is that their errors are bounded vs. the energy E. This is proved in [60] for the single Schrödinger equation but, since the set of reasons invoked in that proof remains the same for systems, the mentioned property continues to apply in this case. The implication is that, once the partition has been fixed, the error will be more or less the same (i.e. irrespective of the value of E) and for this reason the partition should be generated only once and never modified again during the session. Also important is that there is no theoretical upper bound for the stepsize, which is a useful feature when treating long range potentials.

5.2.4 Stabilizing transformations

In many applications, the computation of n linear independent vector solutions is required (e.g. to compute the eigenvalues of the associated boundary value problem, see further). The procedure described above requires that the wavefunction matrix and its first order derivative be propagated explicitly. However there is one well known difficulty in the theory of close-coupled equations. The propagation of the wavefunction into the so-called *classically forbidden region* (where $V(x) > E$) is numerically unstable. It is due to the fact that the exponentially growing component y_j of the wavefunction in the most strongly closed ($V_{jj}(x) > E$) channel soon dominates the entire wavefunction matrix and destroys the required linear independence of the solutions. One way to avoid the difficulty is to use a so-called *invariant imbedding* method, in which the propagated quantity is not the wave function matrix $\mathbf{Y}(x)$ but rather its logarithmic derivative $\mathbf{Y}'(x)\mathbf{Y}(x)^{-1}$ (see e.g. [47, 65]). Another approach to overcome the difficulty is to apply certain stabilizing transformations during propagation. Gordon e.g. (see [41]) minimized the undesired exponentially increasing functions by a “triangularization” method. In [59] Ixaru describes a stabilizing procedure based on the LU decomposition for the propagation by CPM. After some propagation steps this regularization procedure can be applied to re-establish the linear independence of the columns in the wavefunction matrix.

Consider e.g. the LU decomposition of the 2 by 2 matrix with large elements

$$\mathbf{Y} = 10^{16}\begin{bmatrix} 0.1 & 2 \\ 0.03 & 0.5 \end{bmatrix}. \tag{5.41}$$

This means that we compute $\mathbf{L}$, $\mathbf{U}$ and $\mathbf{P}$ such that $\mathbf{Y} = \mathbf{PLU}$, where $\mathbf{L}$ is a lower triangular matrix with a unit diagonal and $\mathbf{U}$ is an upper triangular matrix. $\mathbf{P}$ is the permutation matrix such that $\mathbf{LU} = \mathbf{P}^{-1}\mathbf{Y}$. The LU decomposition for matrix $\mathbf{Y}$ defined above results in the following matrices

$$\mathbf{L} = \begin{bmatrix} 1 & 0 \\ 0.3 & 1 \end{bmatrix}, \quad \mathbf{U} = 10^{16} \begin{bmatrix} 0.1 & 2 \\ 0 & -0.1 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{5.42}$$

Thus the exponential behaviour is collected in the $\mathbf{U}$ matrix, while $\mathbf{L}$ and $\mathbf{P}$ contain small elements. This observation was used by Ixaru to develop a procedure to avoid exponentially increasing values in the wavefunction matrix. To explain Ixaru's procedure, we consider the forward propagation of the solution between the two meshpoints X_0 and X_M and select the points $X_1, X_2, ..., X_p$ among the meshpoints on $[X_0, X_M]$, where the regularization should be performed. The n vector solutions at X_1 form a $n \times n$ matrix $\mathbf{Y}(X_1)$. The LU decomposition of this matrix is performed, i.e. $\mathbf{Y}(X_1) = \mathbf{P_1 L_1 U_1}$.We then use the new, renormalized $n \times n$ matrices $\mathbf{Y}^{\text{new}}(X_1) = \mathbf{P_1 L_1}$, $\mathbf{Y'}^{\text{new}}(X_1) = \mathbf{Y}'(X_1)\mathbf{U_1^{-1}}$, $\mathbf{Y_E}^{\text{new}}(X_1) = \mathbf{Y_E}(X_1)\mathbf{U_1^{-1}}$ and $\mathbf{Y_E'}^{\text{new}}(X_1) = \mathbf{Y_E'}(X_1)\mathbf{U_1^{-1}}$ as the initial conditions for the propagation from X_1 up to X_2. The resultant $\mathbf{Y}(X_2)$, $\mathbf{Y}'(X_2)$, $\mathbf{Y_E}(X_2)$ and $\mathbf{Y_E'}(X_2)$ are replaced by $\mathbf{Y}^{\text{new}}(X_2) = \mathbf{P_2 L_2}$, $\mathbf{Y'}^{\text{new}}(X_2) = \mathbf{Y}'(X_2)\mathbf{U_2^{-1}}$, $\mathbf{Y_E}^{\text{new}}(X_2) = \mathbf{Y_E}(X_2)\mathbf{U_2^{-1}}$ and $\mathbf{Y_E'}^{\text{new}}(X_2) = \mathbf{Y_E'}(X_2)\mathbf{U_2^{-1}}$ for the further propagation and so on.

Clearly, the solution obtained in this way, is no longer consistent with the initial conditions imposed at X_0. The original solution at any mesh point between X_t and X_{t+1} is recovered if each of the four matrices representing the renormalized solution is postmultiplied by the n by n matrix $\mathbf{U}_t\mathbf{U}_{t-1}...\mathbf{U}_2\mathbf{U}_1$. Note however that in a shooting process (see further) the renormalized solution obtained in the matching point can be used to construct the mismatch function.

5.2.5 Some experiments

The test system reads

$$\begin{bmatrix} y_1'' \\ y_2'' \\ y_3'' \end{bmatrix} = \begin{bmatrix} 3-2x-E & -x & 1+x \\ -x & -1-2x-E & 1-x \\ 1+x & 1-x & 1-2x-E \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \tag{5.43}$$

and is solved on $x \in [0, 10]$ for $E = 0$ with the initial conditions

$$y_1(0) = y_2(0) = y_3'(0) = 1, \quad y_1'(0) = 2, \quad y_3(0) = y_2'(0) = 0. \tag{5.44}$$

The exact solution is

$$y_1(x) = (1+x)e^x, \quad y_2(x) = (1-x)e^x, \quad y_3(x) = xe^x. \tag{5.45}$$

The experiment exists in the forward propagation of the solution from $x = 0$ to $x = 10$. Table 5.1 shows the relative errors

$$\Delta\tilde{y}_i(10) = \left| \frac{\tilde{y}_i(10) - y_i(10)}{y_i(10)} \right| \tag{5.46}$$

Table 5.1: Relative errors in $\tilde{\mathbf{y}}(10)$ computed with three different CPM-versions: (a) the LILIX method, (b) CPM{8,6} and (c) CPM{10,8}.

tol		$nstep$	T	$\Delta\tilde{y}_1(10)$	$\Delta\tilde{y}_2(10)$	$\Delta\tilde{y}_3(10)$
10^{-8}	(a)	77	1.43	$5.33(-8)$	$5.32(-8)$	$5.33(-8)$
	(b)	90	1.51	$1.77(-10)$	$1.75(-11)$	$1.05(-10)$
	(c)	59	1.23	$3.86(-11)$	$8.18(-12)$	$2.49(-11)$
10^{-10}	(a)	165	2.67	$5.70(-10)$	$5.74(-10)$	$2.50(-11)$
	(b)	174	2.81	$9.20(-13)$	$3.05(-13)$	$6.44(-13)$
	(c)	97	1.96	$2.55(-13)$	$4.29(-14)$	$1.60(-13)$
10^{-12}	(a)	355	5.26	$5.88(-12)$	$5.92(-12)$	$5.90(-12)$
	(b)	357	5.43	$1.61(-14)$	$1.76(-14)$	$1.45(-14)$
	(c)	162	2.93	$3.72(-15)$	$2.50(-15)$	$5.50(-15)$

in the computed solution $\tilde{\mathbf{y}}(10)$ obtained with the MATLAB implementation of three different methods: (a) the LILIX method [59], (b) the CPM{8,6} method and (c) the CPM{10,8} method. The experiment was repeated for different values of the accuracy tolerance tol. $nstep$ represents the number of intervals in the partition constructed by the method and T is the CPU time (in seconds) needed to obtain the results (in MATLAB). The data reported in the table enable some conclusions:

- The two CPM$\{P, N\}$ methods produce more accurate results than the LILIX method; LILIX should have a higher number of intervals in its partition to reach the same accuracy as the other two methods.
- As for the one-dimensional formulation of the CPM$\{P, N\}$ methods, the number of intervals decreases with increasing order.
- As a consequence of the lower number of intervals, CPM{10,8} is the fastest method. The CPM{8,6} method is somewhat slower than LILIX, but, as already mentioned, the results of this CPM{8,6} method are more precise.

So the CPM{10,8} method seems to be the best choice, even though the CPM{10,8} method needs a higher number of matrix multiplications (in the calculation of the $\mathbf{C}_m$ matrices in (A.5)-(A.8)) *per interval* than the CPM{8,6} method and the LILIX method to construct its correction terms. Note however that the number of matrix multiplications actually performed can be reduced substantially by computing each matrix product, which occurs in the computation of the $\mathbf{C}_m$ matrices, only once. In addition, we can remark that a commutator $[\bar{\mathbf{V}}_i, \bar{\mathbf{V}}_j]$ or an anticommutator $\{\bar{\mathbf{V}}_i, \bar{\mathbf{V}}_j\}$ needs only one matrix multiplication since $\bar{\mathbf{V}}_j\bar{\mathbf{V}}_i = (\bar{\mathbf{V}}_i\bar{\mathbf{V}}_j)^{\mathrm{T}}$ for $\bar{\mathbf{V}}_i$ and $\bar{\mathbf{V}}_j$ symmetric matrices. Moreover in MATLAB, matrix multiplications are performed relatively rapidly and as a consequence the matrix multiplications take only a small part of the total time. Also important is that the matrix multiplications appearing in the CPM$\{P, N\}$ algorithm must be performed only once (before the actual propagation), while for the LILIX method the matrix multiplications occur during propagation. This is particularly important for the solution of the boundary value problem where the solution is advanced for several trial values of E in a shooting procedure (see further). This means that even when the time needed to construct the partition and to calculate the data associated to it, is higher for a certain CPM$\{P, N\}$ method than for the LILIX method, the CPM$\{P, N\}$ method can be expected to be faster when it is used to calculate a sufficiently large batch of eigenvalues.

A minor drawback of the higher order CPM$\{P, N\}$ methods is that they require more memory resources. The reason is that more information is stored prior to propagation, in other words, in order to gain some speed in the propagation process some memory had to be sacrificed. However we believe that the higher memory load forms no problem for modern computer capacities, at least not for the n values which occur in practice.

5.3 Solving the boundary value problem

5.3.1 Problem definitions

We consider the numerical solution of the *regular* boundary value problem of the form

$$\mathbf{y}''(x) = [\mathbf{V}(x) - E\mathbf{I}]\,\mathbf{y}(x), \quad x \in [a, b]. \tag{5.47}$$

If there are n channels, $\mathbf{y}(x)$ is a column vector of order n, $\mathbf{I}$ is the $n \times n$ unity matrix and $\mathbf{V}(x)$ is a symmetric $n \times n$ matrix. For a regular problem the endpoints of the integration interval a and b are finite and the functions in the $\mathbf{V}(x)$ matrix lie in $L^1[a, b]$. In the endpoints a and b regular boundary conditions are applied. Acceptable solutions of the coupled equations exist only when E is an *eigenvalue* of the system. The regular boundary conditions are of the following form (with $\mathbf{0}$ the zero vector):

$$\begin{aligned} \mathbf{A_0}\mathbf{y}(a) + \mathbf{B_0}\mathbf{y}'(a) &= \mathbf{0} \\ \mathbf{A_1}\mathbf{y}(b) + \mathbf{B_1}\mathbf{y}'(b) &= \mathbf{0} \end{aligned} \tag{5.48}$$

where $\mathbf{A_0}, \mathbf{B_0}, \mathbf{A_1}, \mathbf{B_1}$ are real n by n matrices satisfying the so-called *conjointness* conditions (see [43] or [85])

$$\begin{aligned} \mathbf{A_0^T}\mathbf{B_0} - \mathbf{B_0^T}\mathbf{A_0} &= \mathbf{0} \\ \mathbf{A_1^T}\mathbf{B_1} - \mathbf{B_1^T}\mathbf{A_1} &= \mathbf{0}, \end{aligned} \tag{5.49}$$

and the rank conditions

$$\text{rank}(\mathbf{A_0}|\mathbf{B_0}) = n, \quad \text{rank}(\mathbf{A_1}|\mathbf{B_1}) = n. \tag{5.50}$$

Here $(\mathbf{A_0}|\mathbf{B_0})$ denotes the $n \times 2n$ matrix whose first n columns are the columns of $\mathbf{A_0}$ and whose $(n+1)^{\text{st}}$ to $2n^{\text{th}}$ columns are the columns of $\mathbf{B_0}$.

For the regular problem there are infinitely many eigenvalues. The eigenvalues are real, there are countable many of them and they can be ordered to satisfy

$$-\infty < E_0 \leq E_1 \leq E_2 \leq \ldots, \text{ with } E_k \to \infty \text{ as } k \to \infty.$$

Only for the scalar case $n = 1$, it is guaranteed that all the eigenvalues are simple and distinct. For $n > 1$ however, any of the eigenvalues may have a *multiplicity* as great as n (see [34]).

The objective of this section is to show how these eigenvalues can be found using a CPM$\{P, N\}$ method. A CPM$\{P, N\}$ method is expected to be well suited to use as the propagation method in a shooting procedure: since the partition is E-independent, many information associated to this partition has to be computed only once and can be stored before the actual propagation. First we describe the shooting method more elaborately. Next it is shown how the shooting procedure can be improved using the theory of Atkinson [14] which extends the Prüfer theory to the vector case.

5.3.2 A shooting procedure

For a system of n coupled equations, a shooting procedure can be used which is largely inspired from the method outlined in section 3.2 for the one-dimensional problem. One way to locate the eigenvalues is to look for E such that the determinant

$$\phi(E) = \begin{vmatrix} \mathbf{Y}_L & \mathbf{Y}_R \\ \mathbf{Y}'_L & \mathbf{Y}'_R \end{vmatrix} \tag{5.51}$$

is zero in the interior matching point $x_m \in (a, b)$ (see [41, 47]). $\mathbf{Y}_L$ and $\mathbf{Y}_R$ represent the left- and right-hand matrix solutions. A *matrix solution* $\mathbf{Y}$ of (5.47) is a matrix each of whose columns is a solution such that

$$\mathbf{Y}''(x) = [\mathbf{V}(x) - E\mathbf{I}]\,\mathbf{Y}(x). \tag{5.52}$$

The left $n\times n$ *fundamental solution* $\mathbf{Y}_L(x)$ with columns satisfying the left hand boundary conditions, is found by taking the initial values $\mathbf{Y}_L(a) = \mathbf{B_0}, \mathbf{Y}'_L(a) = -\mathbf{A_0}$. Then any solution satisfying these boundary conditions is of the form $\mathbf{Y}_L(x)\mathbf{c}$ where $\mathbf{c}$ is a constant vector. Similarly we can find a fundamental solution $\mathbf{Y}_R(x)$ with $\mathbf{Y}_R(b) = \mathbf{B_1}, \mathbf{Y}'_R(b) = -\mathbf{A_1}$. So the basis for our numerical method is to integrate the fundamental solutions from the ends to some matching point x_m, evaluate $\phi(E)$ and take this as the *mismatch* (also called miss-distance in [105] and [85]).

Example 5.1 As a first test problem we use

$$\mathbf{y}''(x) = \begin{bmatrix} 3x/2 - E & -x/2 \\ -x/2 & 3x/2 - E \end{bmatrix} \mathbf{y}(x), \quad x \in [0, 1] \tag{5.53}$$

with boundary condition matrices

$$\mathbf{A_0} = \mathbf{B_0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \ \mathbf{A_1} = \mathbf{B_1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{5.54}$$

Table 5.2 lists the first 16 exact eigenvalues (rounded to 12 decimals) as they are mentioned in [22]. Figure 5.1 shows the mismatch function $\phi(E)$ for this test problem. It is clear that the function $\phi(E)$ is zero for E equal to an eigenvalue.

Table 5.2: First 16 exact eigenvalues of test problem (5.53)

k	E_k	k	E_k
0	10.368507161836	8	247.24018932857
1	10.865215710533	9	247.74042723263
2	39.978744789883	10	355.805814598764
3	40.479726088439	11	356.305983077456
4	89.326634542478	12	484.110657395956
5	89.827219332229	13	484.610782623713
6	158.41378981431	14	632.154713876864
7	158.91414800462	15	632.654810465433

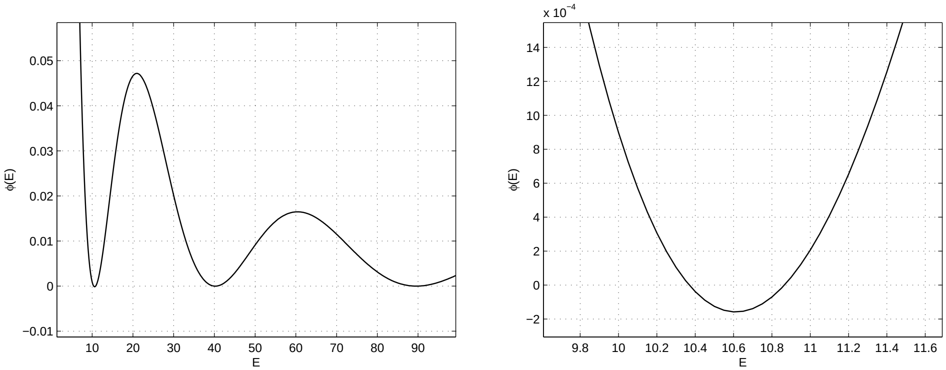


Figure 5.1: Mismatch function $\phi(E)$ for test problem (5.53).

Example 5.2 The second test problem is of the form

$$\mathbf{y}''(x) = [\mathbf{V}(x) - E\mathbf{I}]\,\mathbf{y}(x), \quad x \in [0.1, 1], \quad \mathbf{y}(0.1) = \mathbf{y}(1) = \mathbf{0}, \tag{5.55}$$

where $\mathbf{V}(x)$ is the 4×4 matrix:

$$\mathbf{V}_{ij} = \frac{1}{\max(i,j)} \cos(x) + \frac{\delta_{i,j}}{x^i}. \tag{5.56}$$

This problem was discussed by Marletta as a test example for the SL12F-code [86]. This Fortran code solves eigenvalue problems for linear Hamiltonian systems and is available in the Netlib repository. The mismatch function for test problem 2 is shown in Figure 5.2. The first few eigenvalues of the problem are $\{14.94180054, 17.04349658, 21.38042053, 26.92073133, 51.82570724, 55.80351609, \ldots\}$.

There are some problems associated with the approach outlined above. One problem is that the function $\phi(E)$ does not change sign as E passes through an eigenvalue E_k of even multiplicity.

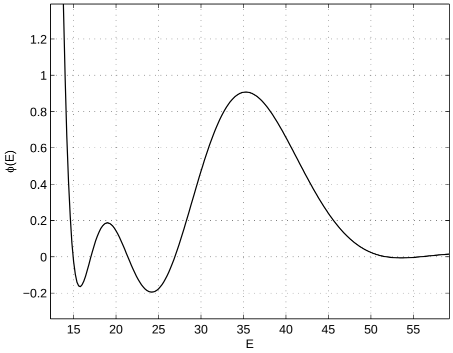


Figure 5.2: Mismatch function $\phi(E)$ for test problem (5.55)-(5.56).

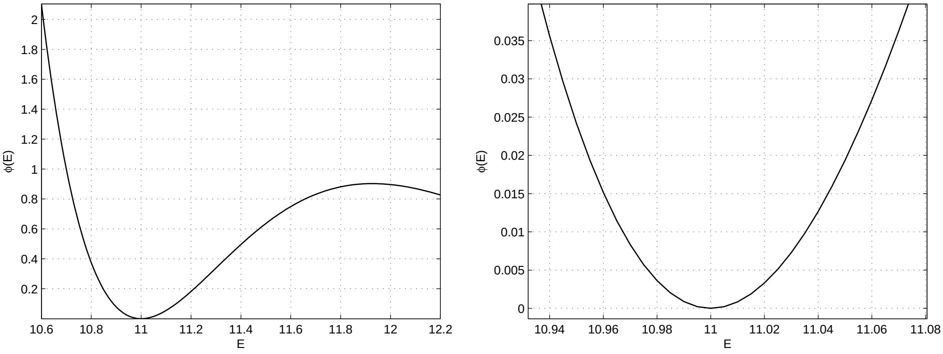


Figure 5.3: Mismatch function $\phi(E)$ for problem (5.57) around $E_2 = 11$.

Example 5.3 Consider the Schrödinger system where

$$\mathbf{V}(x) = \begin{bmatrix} x^2 & 0 \\ 0 & x^2 \end{bmatrix} \tag{5.57}$$

over the interval $[0, 10]$. Each of the eigenvalues $E_k = 3, 7, 11, ...$ has then multiplicity equal to $n = 2$. Figure 5.3 shows the $\phi(E)$ function around the eigenvalue $E_2 = 11$. The $\phi(E)$ function is zero at $E = 11$ but does not change sign, making it very difficult to locate $E_k = 11$.

Another problem is that the function $\phi(E)$ does not give any way of determining the index of the eigenvalue once it has been found. Thus we have no way of knowing which eigenvalue we found when $\phi(E) = 0$. For the scalar case ($n = 1$), this problem was avoided by the Prüfer method (see sections 2.3.2 and 3.2.4). Using the Prüfer form we were able to index the eigenvalues and to approximate the kth eigenvalue without consideration of other eigenvalues. We would like to use an analogous procedure for a matrix Schrödinger eigenvalue problem.

Atkinson [14] developed a Prüfer-like method for the matrix Sturm-Liouville problem. Marletta [85] used Atkinson's theory to construct an integer-valued function $M(E)$ with jumps at each eigenvalue. If E is an eigenvalue with multiplicity m, then $M(E+) - M(E-) = m$, with $M(E+)$ and $M(E-)$ the right and left limit of $M(E)$. This $M(E)$ function allows us to define the spectral function

$$N(E) = \text{The number of eigenvalues of (5.47) that are less than } E.$$

If we can calculate this function from shooting data, we are able to determine whether a trial value of E is "near" the eigenvalue E_k we are looking for, and whether it is too high or too low. If $E' < E''$ are two values such that $N(E') \leq k$ and $N(E'') \geq k+1$, then the kth eigenvalue E_k lies in the interval $E' \leq E_k < E''$. Once an interval $[E', E'']$ has been found which contains just one eigenvalue and is "sufficiently small" (see further in 5.3.5), a Newton iteration process can be applied on the mismatch function $\phi(E)$, $($with $E' \leq E \leq E'')$ to obtain E_k.

5.3.3 The Atkinson-Prüfer method

Before considering the general problem and its complexities, we will reconsider briefly the classical Prüfer method and describe how this classical method can be used to construct the function $N(E)$ for the simplest case: a scalar equation.

The classical Prüfer method

Consider the one-dimensional equation

$$y''(x) = [V(x) - E]\, y(x), \quad x \in (a, b), \tag{5.58}$$

with boundary conditions

$$a_0 y(a) + b_0 y'(a) = 0, \tag{5.59}$$

$$a_1 y(a) + b_1 y'(a) = 0. \tag{5.60}$$

We introduce coordinates in the phase plane:

$$u = y, \quad w = y'. \tag{5.61}$$

In a shooting process, the equation (5.58) is integrated from left to right, with initial values $u(a) = b_0, w(a) = -a_0$, to obtain a left solution u_L, w_L; and integrated from right to left, with initial values $u(b) = b_1, w(b) = -a_1$, to obtain a right solution u_R, w_R.

The main idea of the Prüfer method is to introduce polar coordinates (ρ, θ) in the phase plane:

$$u = \rho \sin\theta, \quad w = \rho\cos\theta. \tag{5.62}$$

The phase angle θ is defined (modulo π) by the equation

$$\tan\theta = \frac{u}{w}. \tag{5.63}$$

Using (5.58) and (5.61)-(5.62) it can be shown that $\theta = \theta(x)$ satisfies a differential equation of first order (see section 2.3.2):

$$\theta' = \cos^2\theta - [V(x) - E]\sin^2\theta, \quad a < x < b. \tag{5.64}$$

Equation (5.64) has a left-solution $\theta_L(x)$, with $\theta_L(a) = \theta_0(a)$; and a right-solution $\theta_R(x)$, with $\theta_R(b) = \theta_0(b)$, where the initial conditions $\theta_0(a)$ and $\theta_0(b)$ are defined by

$$\tan\theta_0(a) = \left(-\frac{b_0}{a_0}\right), \quad 0 \le \theta_0(a) < \pi, \tag{5.65}$$

$$\tan\theta_0(b) = \left(-\frac{b_1}{a_1}\right), \quad 0 < \theta_0(b) \le \pi. \tag{5.66}$$

From equation (5.64) we see that if $\theta(x_i) = m\pi$ (where m is an integer), then $\theta'(x_i) = 1 > 0$. This shows that $\theta_L(x)$ increases through multiples of π as x increases, this means that θ_L can never be decreasing in a point $x = x_i$ when x_i is a multiple of π. Similarly $\theta_R(x)$ decreases through multiples of π as x decreases (see e.g. Figure 2.4). Since $u = 0$ just when θ is a multiple of π, the number of zeros of u (or y) on (a, x_m) is then the number of multiples of π (strictly) between $\theta_L(a)$ and $\theta_L(x_m)$. Analogously the number of zeros of u on (x_m, b) is the number of multiples of π through which θ_R decreases going from b to x_m. Knowing that the index k of an eigenvalue equals the number of zeros of the associated eigenfunction $u(x)$ on the open interval (a, b), we can use these results to formulate the function $N(E)$.

The functions θ_L, θ_R depend on E. So we can write explicitly $\theta_L(x, E)$ and $\theta_R(x, E)$. To define a formula for $N(E)$, let

$$\theta_L(x_m, E) - \theta_R(x_m, E) = n(x_m, E)\pi + \omega(x_m, E) \tag{5.67}$$

where $n(x_m, E)$ is an integer and

$$0 \le \omega(x_m, E) < \pi.$$

$N(E)$ can then be defined as follows:

$$N(E) = n(x_m, E) + 1$$

or

$$N(E) = \frac{1}{\pi}\left[\theta_L(x_m, E) - \theta_R(x_m, E) - \omega(x_m, E)\right] + 1. \tag{5.68}$$

The function $N(E)$ is a piecewise constant with jumps at the eigenvalues: $N(E+) - N(E-) = 1$.

The Atkinson-Prüfer Method for matrix problems

We again consider the vector Schrödinger equation (5.47). As for the scalar case, we introduce:

$$\mathbf{U} = \mathbf{Y}, \quad \mathbf{W} = \mathbf{Y}'.$$

We integrate the equation (5.47) from the left and right endpoints towards a chosen point $x_m \in [a,b]$. Let $\mathbf{U_L}, \mathbf{W_L}$ be the matrix solution of (5.47) with initial conditions $\mathbf{U}(a) = \mathbf{B}_0, \mathbf{W}(a) = -\mathbf{A}_0$, and $\mathbf{U_R}, \mathbf{W_R}$ the solution with initial values $\mathbf{U}(b) = \mathbf{B_1}, \mathbf{W}(b) = -\mathbf{A_1}$. Although it is possible to define an inverse-tangent function for matrix variables, the result is not really useful. The difficulty is that the corresponding sine and cosine functions do not have the desirable derivative properties of their scalar counterparts, introducing difficulties in the formulation of a first order differential equation. Therefore Atkinson used another mapping from complex analysis to map the real line onto some bounded curve in the complex plane.

The matrix function $\mathbf{\Theta}$ is defined:

$$\mathbf{\Theta}(x,E) = [\mathbf{W}(x) + i\mathbf{U}(x)]\,[\mathbf{W}(x) - i\mathbf{U}(x)]^{-1}\,. \tag{5.69}$$

This matrix $\mathbf{\Theta}$ and its phase angles were introduced into oscillation theory by Atkinson [14] and Reid [110]. The conjointness property (5.49) and rank conditions (5.50) ensure that $\mathbf{\Theta}$ exists and is unitary.

Also here the differential equation (5.47) may be reformulated in terms of a nonlinear first-order differential equation for $\mathbf{\Theta}$:

$$\mathbf{\Theta}' = i\mathbf{\Theta}\mathbf{\Omega}, \quad a < x < b, \tag{5.70}$$

where $\mathbf{\Omega}$ is the Hermitian matrix given by

$$2\mathbf{\Omega} = (\mathbf{\Theta} + \mathbf{I})^{+}(\mathbf{\Theta} + \mathbf{I}) - (\mathbf{\Theta} - \mathbf{I})^{+}(\mathbf{V} - E\mathbf{I})(\mathbf{\Theta} - \mathbf{I}), \tag{5.71}$$

with $^{+}$ as the conjugate transpose (or hermitian conjugate) symbol.

Let now $\mathbf{\Theta}_L$ and $\mathbf{\Theta}_R$ be the unitary matrices obtained from $\mathbf{U_L}, \mathbf{W_L}$ and $\mathbf{U_R}, \mathbf{W_R}$ by formula (5.69). Because these $\mathbf{\Theta}$ matrices are unitary, their eigenvalues all lie on the unit circle. The eigenvalues of $\mathbf{\Theta}_L$ and $\mathbf{\Theta}_R$ are denoted by $\{\exp(i\phi_j^L(x)) \mid j = 1, \ldots, n\}$ and $\{\exp(i\phi_j^R(x)) \mid j = 1, \ldots, n\}$ respectively. The so-called *phase angles* $\phi_j^L(x), \phi_j^R(x)$ are uniquely determined continuous functions when the following conditions are imposed

$$\begin{aligned} \phi_1^L(x) \le \phi_2^L(x) \le \cdots \le \phi_n^L(x) \le \phi_1^L(x) + 2\pi, \\ \phi_1^R(x) \le \phi_2^R(x) \le \cdots \le \phi_n^R(x) \le \phi_1^R(x) + 2\pi, \\ 0 \le \phi_j^L(a) < 2\pi, \quad 0 < \phi_j^R(b) \le 2\pi. \end{aligned} \tag{5.72}$$

From [14] and [85] we know that the phase-angles ϕ_j^L of the matrix $\mathbf{\Theta}_L$ increase (and never decrease) through multiples of 2π with increasing x. Similarly the ϕ_j^R decrease through multiples of 2π as x decreases. Notice the correspondence between $\phi_j/2$ and the scalar θ appearing in the classical Prüfer method.

At the chosen point $x_m \in [a,b]$ let the eigenvalues of $\boldsymbol{\Theta}_R^+(x_m)\boldsymbol{\Theta}_L(x_m)$ be $\{\exp(i\omega_j) \mid j=1,\dots,n\}$, where the ω_j are normalized by the condition

$$0 \le \omega_j < 2\pi. \tag{5.73}$$

We can now give the formula for the function $M(E)$ which was defined by Marletta in [84] and [85]:

$$M(E) = \frac{1}{2\pi}\left\{\sum_{j=1}^{n}\phi_j^L(x_m) - \sum_{j=1}^{n}\phi_j^R(x_m) - \sum_{j=1}^{n}\omega_j(x_m)\right\}. \tag{5.74}$$

This formula can be used to define the function $N(E)$ (see [43]), which is the number of eigenvalues of (5.47) that are less than E:

$$N(E) = M(E) + n. \tag{5.75}$$

In [85] the following notations were used to represent the three quantities appearing in Eq. (5.74):

$$\operatorname{argdet}\boldsymbol{\Theta}_L(x_m,E) = \sum_{j=1}^{n}\phi_j^L(x_m), \tag{5.76}$$

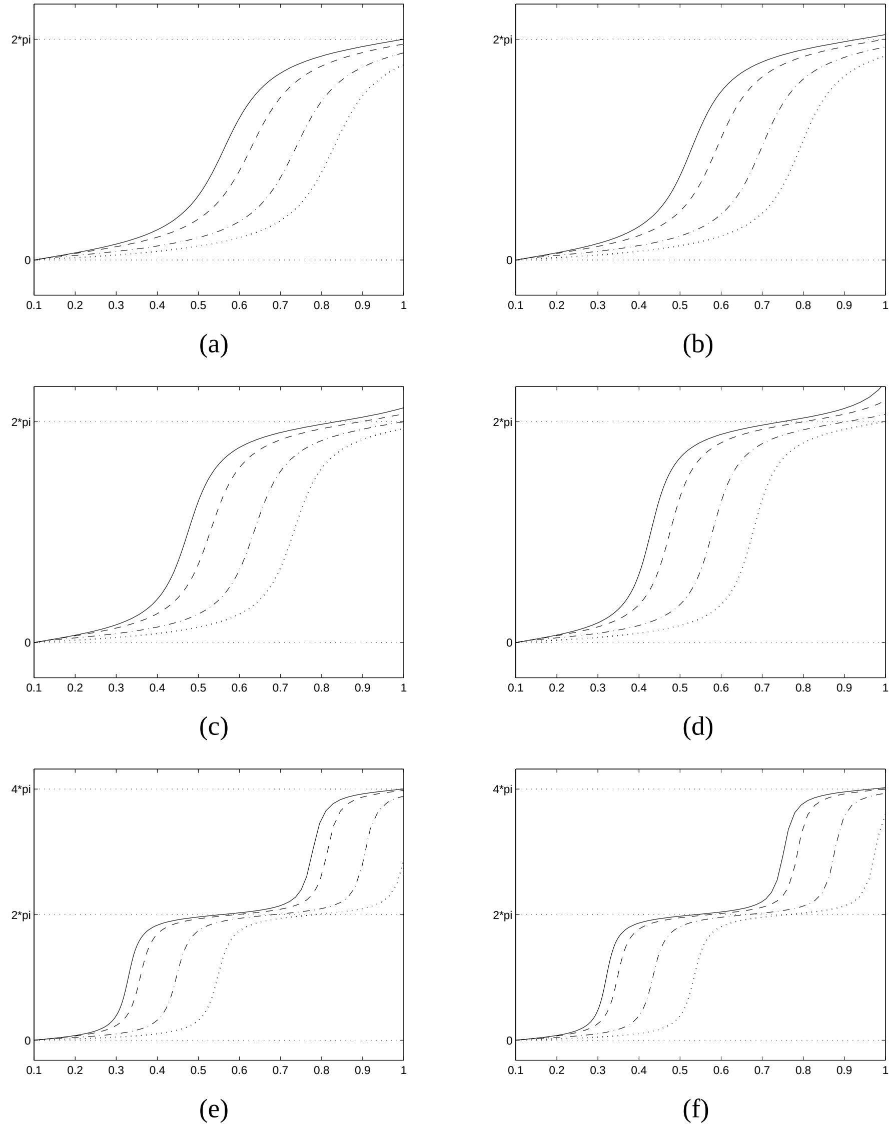
$$\operatorname{argdet}\boldsymbol{\Theta}_R(x_m,E) = \sum_{j=1}^{n}\phi_j^R(x_m), \tag{5.77}$$

$$\overline{\operatorname{argdet}}\boldsymbol{\Theta}_{LR}(x_m,E) = \sum_{j=1}^{n}\omega_j(x_m). \tag{5.78}$$

The overbar on $\overline{\operatorname{argdet}}\ \boldsymbol{\Theta}_{LR}(x_m,E)$ indicates that the angles are normalized to lie in the interval $[0,2\pi)$.

Example 5.4 Figure 5.4 shows the phase angles $\phi_j^L(x), j=1,\dots,4$ for the first 6 eigenvalues of the second test problem given by (5.55)-(5.56) (see Example 5.2). We just took $x_m = b = 1$, that is we propagated the left-hand solution from a up to b. It is clear that the $\phi_j^L(x)$ increase through multiples of 2π. For the first eigenvalue E_0, there is one phase angle reaching 2π, for the second eigenvalue E_1 there are two phase angles passing through 2π, for the third eigenvalues E_2 one can see three phase angles passing through 2π and so on.

Example 5.5 Figure 5.5 illustrates the construction of Marletta's M function for the first test problem (5.53) given in Example 5.1. Since the problem is symmetric the two phase angles coincide, that is $\phi_1^L(x) = \phi_2^L(x)$ and $\phi_1^R(x) = \phi_2^R(x)$. In the matching point $x_m = 0.24$, we have $\operatorname{argdet}\boldsymbol{\Theta}_L\ (x_m) \approx 4.4\pi$, $\operatorname{argdet}\boldsymbol{\Theta}_R(x_m) \approx -14.1\pi$ and $\overline{\operatorname{argdet}}\boldsymbol{\Theta}_{LR} \approx 2.5\pi$. With the formula (5.74) we compute $M(350) = [4.4\pi + 14.1\pi - 2.5\pi]/(2\pi) = 8$, and thus the number of eigenvalues less than 350 is then given by $N(350) = 10$. Table 5.2 indeed shows that $E = 350$ is larger than E_9 but smaller than E_{10}.



(a) (b) (c) (d) (e) (f)

Figure 5.4: The phase angles $\phi_j^L(x), j = 1, \ldots, 4$ for the first 6 eigenvalues of the second test problem (5.55)-(5.56). In all cases $x_m = b = 1$.

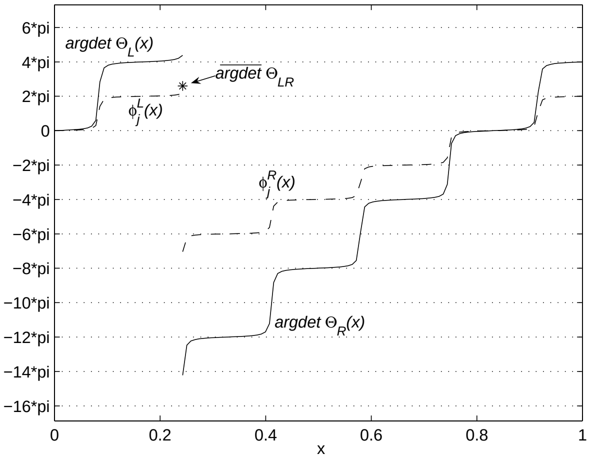


Figure 5.5: The Atkinson-Prüfer functions for the first test problem (5.53) with $E = 350$, $x_m = 0.24$. The two phase angles coincide, that is $\phi_1^L(x) = \phi_2^L(x)$ and $\phi_1^R(x) = \phi_2^R(x)$. $\operatorname{argdet}\mathbf{\Theta}_L(x) = \sum_{j=1}^n \phi_j^L(x)$, $\operatorname{argdet}\mathbf{\Theta}_R(x) = \sum_{j=1}^n \phi_j^R(x)$ and $\overline{\operatorname{argdet}}\mathbf{\Theta}_{LR} = \sum_{j=1}^n \omega_j(x_m)$.

5.3.4 Computing Marletta's $M(E)$ function

In order to compute $M(E)$ for a certain E-value, we need to be able to compute the quantities $\operatorname{argdet}\mathbf{\Theta}_L(x_m, E)$, $\operatorname{argdet}\mathbf{\Theta}_R(x_m, E) = \sum_{j=1}^n \phi_j^R(x_m)$ and $\overline{\operatorname{argdet}}\mathbf{\Theta}_{LR}(x_m, E) = \sum_{j=1}^n \omega_j(x_m)$. Note that we need only to know $\mathbf{\Theta}_L(x_m, E)$ and $\mathbf{\Theta}_R(x_m, E)$ to calculate $\overline{\operatorname{argdet}}\mathbf{\Theta}_{LR}(x_m, E)$, since the angles ω_j are normalized to lie in the interval $[0, 2\pi)$. $\mathbf{\Theta}_L(x_m, E)$ (or $\mathbf{\Theta}_R(x_m, E)$) is easily obtained by substituting the matrix solution $\mathbf{U}_L(x_m), \mathbf{W}_L(x_m)$ (or $\mathbf{U}_R(x_m), \mathbf{W}_R(x_m)$) in (5.69).

The calculation of $\operatorname{argdet}\mathbf{\Theta}_L(x_m, E)$ or $\operatorname{argdet}\mathbf{\Theta}_R(x_m, E)$ is more difficult. We have to integrate $\mathbf{\Theta}_L$ (actually we integrate the original differential system (5.47) and form the $\mathbf{\Theta}$ matrices from the appropriate matrix solutions of (5.47)), from $x = a$ to $x = x_m$, and $\mathbf{\Theta}_R$ from $x = b$ to $x = x_m$. During the integration we have to follow $\operatorname{argdet}\mathbf{\Theta}_L$ and $\operatorname{argdet}\mathbf{\Theta}_R$ continuously and count the number of multiples of 2π in each. In [85], Marletta describes a method based on constant coefficient approximation to compute $\operatorname{argdet}\mathbf{\Theta}_L$ and $\operatorname{argdet}\mathbf{\Theta}_R$ for the general Sturm-Liouville problem. The algorithm we will describe here is based on this method of Marletta, but adapted to the use of a CPM as propagation method for the solution of the original differential equation (5.47).

A CPM is used to propagate the solution $\mathbf{U}, \mathbf{W}$ over the mesh

$$a = x_0 < x_1 < \cdots < x_{\text{nstep}} = b.$$

argdet$\mathbf{\Theta}_L(a)$ and argdet$\mathbf{\Theta}_R(b)$ are unambiguously known from (5.72). Consider now the propagation across an interval $[x_{t-1}, x_t]$. Suppose argdet$\mathbf{\Theta}$ (argdet$\mathbf{\Theta}_L$ or argdet$\mathbf{\Theta}_R$) is known in one endpoint of the interval x_{begin} and we want to obtain argdet$\mathbf{\Theta}$ in the other endpoint, called x_{end}. This means that for the left propagation $x_{\text{begin}} = x_{t-1}$ and $x_{\text{end}} = x_t$, while for the right propagation $x_{\text{begin}} = x_t$ and $x_{\text{end}} = x_{t-1}$.

In order to compute argdet$\mathbf{\Theta}$ correctly, and not just modulo 2π, we must count the number of times that some phase-angle ϕ_j, $j = 1, \dots, n$ of $\mathbf{\Theta}$ passes through a multiple of 2π as x moves from x_{begin} to x_{end}. We try to do this by decoupling the system into n scalar ones to which the simple Prüfer method can be applied. This means that we try to obtain a problem in diagonal form. Again, we consider the diagonalization process discussed in section 5.2.1. Since

$$\mathbf{D}^{-1}\mathbf{\Theta}\mathbf{D} = (\mathbf{W}^{\mathbf{D}} + i\mathbf{U}^{\mathbf{D}})(\mathbf{W}^{\mathbf{D}} - i\mathbf{U}^{\mathbf{D}})^{-1} = \mathbf{\Theta}^{\mathbf{D}} \tag{5.79}$$

we know that the eigenvalues of $\mathbf{\Theta}$ are precisely the same as those of $\mathbf{\Theta}^{\mathbf{D}}$. So we may forget about $\mathbf{\Theta}$ and think only in terms of $\mathbf{\Theta}^{\mathbf{D}}$.

As seen in section 5.2.2, the zeroth order propagation of the matrices $\mathbf{U}^{\mathbf{D}}$ and $\mathbf{W}^{\mathbf{D}}$ can be written as

$$\begin{aligned} \mathbf{U}^{\mathbf{D}}(x) &\approx \boldsymbol{\xi}(\mathbf{Z})\mathbf{U}^{\mathbf{D}}(x_{\text{begin}}) + \delta\boldsymbol{\eta}_0(\mathbf{Z})\mathbf{W}^{\mathbf{D}}(x_{\text{begin}}), \\ \mathbf{W}^{\mathbf{D}}(x) &\approx (\mathbf{Z}/\delta)\boldsymbol{\eta}_0(\mathbf{Z})\mathbf{U}^{\mathbf{D}}(x_{\text{begin}}) + \boldsymbol{\xi}(\mathbf{Z})\mathbf{W}^{\mathbf{D}}(x_{\text{begin}}), \end{aligned} \tag{5.80}$$

with $\mathbf{Z} = (\mathbf{V}_\mathbf{0}^{\mathbf{D}} - E\mathbf{I})\delta^2$ and $\delta = x - x_{\text{begin}}$. Note that here δ is positive for the forward propagation and negative for the backward propagation.

In order to compute argdet$\mathbf{\Theta}^{\mathbf{D}}(x)$ from argdet$\mathbf{\Theta}^{\mathbf{D}}(x_{\text{begin}})$ we consider the auxiliary matrix $\mathbf{\Phi}(x)$ given by

$$\mathbf{\Phi}(x) = (\boldsymbol{\xi} - i\delta\boldsymbol{\eta}_0)^{-1}(\boldsymbol{\xi} + i\delta\boldsymbol{\eta}_0)\,\mathbf{\Theta}^{\mathbf{D}}(x_{\text{begin}}). \tag{5.81}$$

This is a product of unitary matrices and is therefore unitary. Let now the eigenvalues of $\mathbf{\Theta}^{\mathbf{D}}$ be $\exp(i\phi_j)$ and those of $\mathbf{\Phi}$ be $\exp(i\psi_j)$, and suppose that

$$\phi_j = 2\pi n_j + \beta_j, \quad \psi_j = 2\pi m_j + \alpha_j,$$

where n_j and m_j are integers and α_j and β_j lie in $[0, 2\pi)$. Then we can write

$$\text{argdet}\mathbf{\Theta}^{\mathbf{D}} = \text{argdet}\mathbf{\Phi} + \sum_{j=1}^{n}(\beta_j - \alpha_j) + 2\pi\sum_{j=1}^{n}(n_j - m_j). \tag{5.82}$$

The α_j and β_j are easily computed directly from $\mathbf{\Phi}$ and $\mathbf{\Theta}^{\mathbf{D}}$, because the number of multiples of 2π in the α_j and β_j is unambiguous. $\mathbf{\Theta}^{\mathbf{D}}$ is calculated from $\mathbf{U}^{\mathbf{D}}$ and $\mathbf{W}^{\mathbf{D}}$ using (5.79) and $\mathbf{\Phi}$ is obtained using (5.81).

The quantity argdet$\mathbf{\Phi}$ can be calculated using the following identity

$$\text{argdet}\mathbf{\Phi} = \text{argdet}\mathbf{\Theta}^{\mathbf{D}}(x_{\text{begin}}) + \text{argdet}(\boldsymbol{\xi} - i\delta\boldsymbol{\eta}_0)^{-1}(\boldsymbol{\xi} + i\delta\boldsymbol{\eta}_0). \tag{5.83}$$

The term $\operatorname{argdet}(\boldsymbol{\xi} - i\delta\boldsymbol{\eta}_0)^{-1}(\boldsymbol{\xi} + i\delta\boldsymbol{\eta}_0)$ is computed by applying a Prüfer transformation to each diagonal term in turn: for each j let y_j be the solution of the initial value problem

$$-y_j'' + d_j y_j = 0, \quad y_j(x_{\text{begin}}) = 0,\ y_j'(x_{\text{begin}}) = 1,$$

where the d_j are the elements of the diagonal matrix $\mathbf{V_0^D} - E\mathbf{I}$. The Prüfer transformation

$$y_j = \rho_j \sin\theta_j, \quad y_j' = \rho_j \cos\theta_j.$$

is applied. Then θ_j satisfies the initial value problem

$$\theta_j' = \cos^2\theta_j - d_j \sin^2\theta_j, \quad \theta_j(x_{\text{begin}}) = 0.$$

This may be solved exactly in terms of elementary functions as follows

$$\theta_j(x) = \text{atan2}\left(\delta\eta_0(Z_j(\delta)), \xi(Z_j(\delta))\right)$$

with atan2 the four-quadrant inverse tangent. Then we may compute

$$\operatorname{argdet}(\boldsymbol{\xi} - i\delta\boldsymbol{\eta}_0)^{-1}(\boldsymbol{\xi} + i\delta\boldsymbol{\eta}_0) = 2\sum_{j=1}^{n}\theta_j. \tag{5.84}$$

Returning to (5.82), the only unknown quantity is

$$\sum_{j=1}^{n} n_j - \sum_{j=1}^{n} m_j.$$

In [85], it was shown that when only zeroth order propagators are used, this quantity is zero, provided the interval is taken small enough such that the (zeroth order) propagation is exact. Here larger intervals are used, on which the propagation by the CPM is exact. As a consequence it is no longer guaranteed that $\sum_{j=1}^{n} n_j - \sum_{j=1}^{n} m_j$ is zero. However when we want the CPM propagation to be exact, the chosen intervals are small enough to reach already a reasonable approximation of the solution by the zeroth order propagation. The differences between the phases calculated by the zeroth order propagation and the ones obtained with the CPM will then generally be much smaller than 2π and in almost all cases $\sum_{j=1}^{n} n_j - \sum_{j=1}^{n} m_j$ is zero. Only some special care is necessary when one of the phases is close to a multiple of 2π. In this case we compute argdetΘ over the two halves of the current interval $[x_{\text{begin}}, x_{\text{end}}]$. The accumulation of the results over the two subintervals should lead to the same change in multiples of 2π as over the whole interval.

In the assumption that the intervals are small enough to have $\sum_{j=1}^{n} n_j - \sum_{j=1}^{n} m_j = 0$, we use the formula

$$\operatorname{argdet}\boldsymbol{\Theta}^{\mathbf{D}}(x) = \operatorname{argdet}\boldsymbol{\Theta}^{\mathbf{D}}(x_{\text{begin}}) + \operatorname{argdet}(\boldsymbol{\xi} - i\delta\boldsymbol{\eta}_0)^{-1}(\boldsymbol{\xi} + i\delta\boldsymbol{\eta}_0) + \sum_{j=1}^{n}(\beta_j - \alpha_j)$$

to propagate $\operatorname{argdet}\boldsymbol{\Theta}^{\mathbf{D}}$. And since $\operatorname{argdet}\boldsymbol{\Theta} = \operatorname{argdet}\boldsymbol{\Theta}^{\mathbf{D}}$, this result allows us to keep track of the number of multiples of 2π in argdet$\boldsymbol{\Theta}$ as we integrate across an interval.

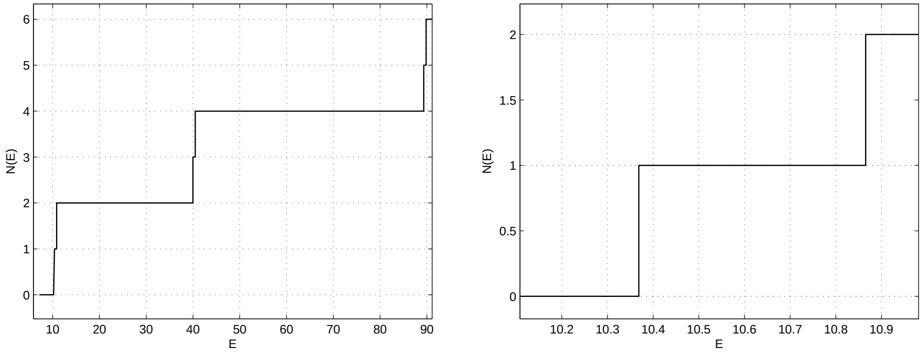


Figure 5.6: The function $N(E)$ for the first test problem (5.53).

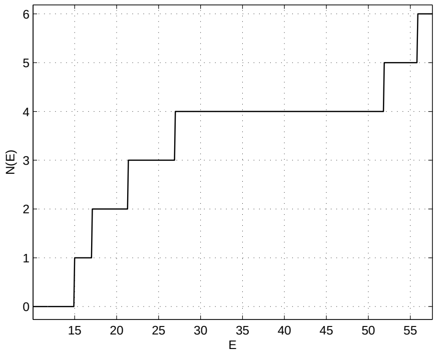


Figure 5.7: The function $N(E)$ for the second test problem (5.55)-(5.56).

Example 5.6 Figures 5.6 and 5.7 show the $N(E)$ function (computed by the algorithm described above) for the test problems given in Examples 5.1-5.2. As we increase the E value, we see a "jump" in the index value as E passes through an eigenvalue of the Schrödinger problem. The size of the jump indicates the multiplicity of the eigenvalue.

5.3.5 Eigenvalue computation

Algorithm

Our objective is now to describe how we can use $N(E)$ to compute the eigenvalue E_k.

Of course, first a mesh must be constructed for our CPM. We again create a mesh which has stepsizes consistent with a user specified tolerance. As already mentioned, this mesh has to be generated only once (since it is independent of E) at the very beginning of the run and can then be used for all eigenvalue calculations. For this reason our algorithm is particularly suited for calculating large sets of eigenvalues.

Over the generated mesh, for different trial values of E a left solution will be computed on $[a, x_m]$ and a right solution on $[x_m, b]$ to obtain the values of $\mathbf{Y}, \mathbf{Y}'$ at each side of the matching point x_m. This matching point was fixed at the very beginning of the computation and is the meshpoint which is the closest to the deepest bottom of the potential functions $\mathbf{V}_{ij}$ $(1 \leq i \leq n,\, 1 \leq j \leq i)$. The data obtained at each side of x_m are combined to calculate the mismatch function $\phi(E)$ using Eq. (5.51). Since the CPM algorithm allows a direct evaluation of the first derivative of $\mathbf{Y}$ and $\mathbf{Y}'$ with respect to E, the derivative of $\phi(E)$ with respect to E can be computed. This means that a Newton iteration procedure can be used to locate the roots of $\phi(E)$:

$$E_{\text{new}} = E - \frac{\phi(E)}{\phi'(E)}. \tag{5.85}$$

The Newton iteration procedure is convergent only if the initial guess for E is sufficiently close to the eigenvalue E_k which has to be located. Therefore, the procedure consists of two stages. In the first stage, a good initial guess for the Newton iteration is searched for. The second stage consists in effectively iterating until the requested accuracy is achieved.

In the first stage we look for an interval $[E_{\text{low}}, E_{\text{up}}]$ containing just one eigenvalue (E_k), using the function $N(E)$. Both $E_{\text{new}}(E_{\text{low}})$ (from (5.85)) and $E_{\text{new}}(E_{\text{up}})$ must be inside $[E_{\text{low}}, E_{\text{up}}]$. Once acceptable values for E_{low} and E_{up} are found, the Newton iteration (stage 2) is started with $E = (E_{\text{low}} + E_{\text{up}})/2$ as initial guess. To continue the calculation for the next eigenvalue E_{k+1}, a good starting value is $E_{\text{low}} = E_{\text{up}}$.

The error in the eigenvalue approximations can be estimated by calculating for each eigenvalue an associated ‘reference’ eigenvalue. The estimation of the error in a certain calculated eigenvalue (the so-called basic eigenvalue) is then the difference between the basic eigenvalue and the more accurate reference eigenvalue. One way to obtain a reference eigenvalue is to use an additional mesh with finer stepsizes. We constructed the additional ‘reference’ mesh by halving each interval of the original ‘basic’ mesh.

The search for the basic eigenvalues is first done (on the basic mesh) and only in this case the first stage of the search is activated. The search for the reference eigenvalues involves only the Newton iteration (stage 2) which starts with the basic eigenvalue as initial guess. Since the difference of the two eigenvalues is usually very small, only a small number of extra iterations is necessary to calculate a reference eigenvalue.

MATLAB code

The algorithm described in the previous sections is implemented in a MATLAB package. The package was developed in MATLAB version 7.1 and can be downloaded from [2]. The package includes some examples showing how the different methods can be called in order to compute some eigenvalues. Here, we briefly discuss the MATLAB code needed to solve the first test problem. The following commands define the system of Schrödinger equations and initialize the CPM:

```
a=0;                             % Integration interval
b=1;
n=2;                             % Dimension of the problem
V={'3*x/2','-x/2','3*x/2'};  % Potential matrix
s=scs(V,a,b,n);                  % constructs the system
cp=cpm10_8(s,1e-10);             % constructs the partitions
```

First the problem is specified by its endpoints a and b and the potential matrix function $\mathbf{V}(x)$. This potential matrix is symmetric and only the elements in the lower triangle have to be specified. The constructor of the class `scs` is called, which constructs an object respresenting the system of coupled Schrödinger equations. This object is then passed to another method which implements the CPM. Here we used `cpm10_8` which implements the CPM$\{10,8\}$ algorithm, but also `cpm8_6` is included. The second argument of `cpm10_8` is a positive constant representing the accuracy requested in the results. An object is returned containing information on the partition.

The method `get_eigenvalues` can be used to calculate a batch of eigenvalues

```
E = get_eigenvalues(cp,pmin,pmax,indices)
```

where `cp` is an instance of the classes `cpm10_8` or `cpm8_6`. If `indices` is true, the eigenvalues E_k with index k between `pmin` and `pmax` are calculated. If `indices` is false the eigenvalues in the energy-range `[pmin,pmax]` are computed. The method returns a structure `E`, in which all information related to the calculated eigenvalues is stored. `E.eigenvalues` contains the eigenvalues in ascending order, while the associated indices are collected in `E.indices` and `E.errors` holds the estimated errors. The field `E.success` is false when the CPM was not able to obtain any data. `E.status` is a vector of status-flags. When `E.status(k)`> 0, there were some difficulties detected during the calculation of the kth eigenvalue or its index. When `E.status(k)` is equal to one, the input tolerance tol was too large to ensure a correct evaluation of the $N(E)$ function (see section 5.3.4). A status flag larger than zero suggests that the input tolerance should be decreased.

When we want to calculate the eigenvalues with indices from 0 to 10, we use the following command:

```
E = get_eigenvalues(cp,0,10,true)
```

while

```
E = get_eigenvalues(cp,50,100,false)
```

returns the eigenvalues between 50 and 100.

Table 5.3: Some eigenvalues of the first test problem calculated with the CPM$\{10,8\}$ in a shooting procedure. k is the eigenvalue index and $nint$ is the number of intervals in the basic partition.

k	$tol = 10^{-6}$	$tol = 10^{-8}$	$tol = 10^{-10}$
0	10.3685071614415	10.3685071618259	10.3685071618362
1	10.8652157041883	10.8652157103674	10.8652157105314
2	39.9787447902932	39.9787447898867	39.9787447898833
3	40.4797260951163	40.4797260884903	40.4797260884383
4	89.3266345425364	89.3266345424888	89.3266345424788
5	89.8272193331295	89.8272193323913	89.8272193322298
10	355.8058145966053	355.8058145987640	355.8058145987645
15	632.6548104654120	632.6548104654298	632.6548104654330
$nint$	2	3	4

Some results

We take again the first test problem (5.53). Table 5.3 shows some results we obtained with the CPM$\{10,8\}$ algorithm for different values of the input tolerance. The last line of the table contains the number of intervals $nint$ in the basic partition. Table 5.4 shows the ratio of the true error to the estimated error for the two CPM: CPM$\{8,6\}$ and CPM$\{10,8\}$. This 'goodness' ratio has always values smaller or very close to one, which illustrates the adequacy of our error estimation.

We also included some results for the second test problem (5.55). Table 5.5 contains the first eigenvalues of the problem calculated to a high accuracy. These eigenvalues correspond to the roots of the mismatch function shown in Figure 5.2. Table 5.6 shows some results for a higher eigenvalue index. The second column shows the 'exact' eigenvalues which were obtained using the algorithm with an input tolerance 10^{-12}. The eigenvalue approximations calculated with an input tolerance 10^{-6} are listed in the third column. The fourth column contains the estimated errors ΔE for these eigenvalues, i.e. the differences between the reference eigenvalues and the basic eigenvalues. Again one can see the accuracy of the error estimates.

All calculations were done using the CPM$\{10,8\}$ method, however the same accuracy can be reached using the CPM$\{8,6\}$ method. In most cases, the CPM$\{8,6\}$ method needs some more time than the CPM$\{10,8\}$ algorithm. The reason is that the CPM$\{8,6\}$ method needs more meshpoints in its partition than the CPM$\{10,8\}$ method.

5.4 Conclusion

In this chapter we discussed the extension of the CPM to systems of coupled Schrödinger equations. As for the one-dimensional problem, a Maple program was developed to compute the expressions of the perturbation corrections for the class of CPM$\{P, N\}$ methods. A CPM$\{10,8\}$ algorithm was presented.

Table 5.4: Ratio $\frac{|\text{actual error}|}{|\text{error estimate}|}$ for the first test problem ($tol = 10^{-8}$). k is the eigenvalue index and $nint$ is the number of intervals in the basic partition.

k	CPM{8,6}	CPM{10,8}
0	1.000	1.001
1	0.993	1.001
2	1.000	0.998
3	0.998	0.997
4	1.002	1.002
5	1.001	1.000
10	0.997	0.887
15	0.999	0.995
$nint$	6	3

Table 5.5: The first 6 eigenvalues of the second test problem (5.55)-(5.56) calculated with the CPM{10,8} in a shooting procedure. ΔE is the error estimate, k is the eigenvalue index and $nint$ is the number of intervals in the basic partition.

k	$tol = 10^{-12}$	ΔE
0	14.94180054416473	$3.6(-15)$
1	17.04349658304373	$3.8(-15)$
2	21.38042052885422	$1.8(-14)$
3	26.92073133400956	$6.0(-15)$
4	51.82570724029870	$4.3(-14)$
5	55.80351609486795	$2.8(-14)$
$nint$	81	

Table 5.6: Some higher eigenvalues of the second test problem (5.55)-(5.56): the exact eigenvalues, the calculated eigenvalues for $tol = 10^{-6}$ and the corresponding error estimates ΔE. k is the eigenvalue index and $nint$ is the number of intervals in the basic partition.

k	true eigenvalue	$tol = 10^{-6}$	ΔE
201	31702.815244147	31702.8152435166	$-6.3(-7)$
202	31747.557394158	31747.5573941581	$4.0(-10)$
203	32069.904602246	32069.9046041610	$1.9(-6)$
204	32950.776323037	32950.7763247290	$1.7(-6)$
$nint$	81	15	

We also discussed the computation of the eigenvalue problem for systems of regular equations. Using a CPM in a shooting procedure, eigenvalues are calculated accurately. However such a shooting method does not determine the index of the eigenvalue. To solve this problem, the algorithm has been supplemented by Atkinson's matrix generalization of the Prüfer transformation.

Chapter 6

Singular problems

Until now the main focus was on regular Sturm-Liouville and Schrödinger problems defined on a finite integration interval for which the CPM were shown to be very efficient. However many problems are defined on an infinite integration interval, i.e. $a = -\infty$ and/or $b = +\infty$. Other Sturm-Liouville problems have singular endpoints, that is at least one of p^{-1}, q, w is not integrable in any neighbourhood of the endpoint a or b. Both problems defined on an infinite integration interval and problems with singular endpoints require a special numerical treatment.

6.1 A singular Sturm-Liouville problem

A *singular* problem is, of course, one that is not regular. More precisely, a classical singular Sturm-Liouville problem is one that is defined by the Sturm-Liouville differential equation

$$-\frac{d}{dx}\left[p(x)\frac{dy(x)}{dx}\right] + q(x)y(x) = Ew(x)y(x), \tag{6.1}$$

on a finite or infinite interval (a, b) where p, w and q are piecewise continuous with p and w strictly positive and one or both of a, b is a singular endpoint. The endpoint (say) $x = b$ is singular if one or more of

$$\int \left|\frac{1}{p}\right| dx, \quad \int |q|\, dx, \quad \int |w|\, dx \tag{6.2}$$

diverges at $x = b$, and regular if they all converge. The above allows the endpoint $a = -\infty$ or the endpoint $b = +\infty$ to be regular, but from the computational viewpoint some special treatment will be needed to deal with the infinite integration interval.

The theory of singular Sturm-Liouville problems is more complicated than for regular Sturm-Liouville problems and gives rise to a whole range of difficult numerical tasks, such as (i) the classification of the endpoints a, b as limit-circle or limit-point and oscillatory or nonoscillatory; (ii) automatically finding appropriate (approximating) boundary conditions in the endpoints; (iii) determining how many eigenvalues there are, if this is finite; (iv) finding resonances (quasi eigenvalues) within the continuous spectrum (see e.g. [106] and [38]).

Since the theory for singular problems can be very intricate, we only attempt to describe briefly some of the main points. More concrete we will briefly discuss in section 6.2 the different types of singular endpoints and the form of the associated eigenvalue spectra. For the full theory, we can refer to the classical works of Weyl [129], Kodaira [69], Titchmarsh [119], or Dunford and Schwartz [33] and for a numerical viewpoint to [105].

For singular problems and problems defined on an infinite integration interval an *interval truncation* procedure must be adopted. For instance assume that $x = b$ is a singular or infinite endpoint, and $x = a$ is regular and finite (the case of two singular or infinite points is a simple extension of this case). We then choose some $b^* < b$ and solve a truncated problem on $[a, b^*]$ to obtain our results. The choice of b^* will generaly depend on the index k of the eigenvalue sought, and it will also be necessary to impose some sort of artificial boundary condition at $x = b^*$. We will discuss a truncation procedure for problems with infinite endpoints in section 6.3, while in section 6.4 we consider the treatment of a specific class of singularities.

6.2 Classification of singular endpoints

The most important properties of a singular endpoint are the Weyl-Kodaira *limit-point*, *limit-circle* classification, which is independent of E; and whether it is *oscillatory* or *nonoscillatory*, which may depend on E. To avoid making all statements twice (once at each end), we will often use the letter e as a generic endpoint, i.e. either $e = a$ or $e = b$.

6.2.1 Limit-point and limit-circle endpoints

The primary classification of a singular endpoint is the classical one of Weyl [129], Kodaira [69] and Titchmarsh [119] as follows.

The endpoint e is *limit-circle* (LC) if e is singular and all solutions $y(x)$ of the Sturm-Liouville differential equation are *square-integrable* (L^2) at e with respect to the weight-function w, i.e., for some E any solution $y(., E)$ of the differential equation (6.1) satisfies

$$\int_{e}^{e+\epsilon} |y(x, E)|^2 w(x) dx < +\infty. \tag{6.3}$$

Otherwise the equation is called *limit-point* (LP) at $x = e$, that is for some E there exists only one nonzero solution $y(., E)$ (up to a scalar factor) of the differential equation which is square-integrable at the endpoint e. This LC/LP classification is independent of E (see [126]). In the LP case no boundary condition is required at $x = e$ to get a well-posed

Sturm-Liouville problem: the condition (6.3) is sufficient (see [88]). For the LC case however, a boundary condition is needed.

As mentioned in [88], for both LP and LC, if the problem concerned has a kth eigenvalue, then imposing the boundary condition $y(e^*) = 0$ at the truncation point e^* yields a regular problem whose kth eigenvalue converges to the kth eigenvalue of the original problem as $e^* \to e$.

6.2.2 Oscillatory and nonoscillatory behaviour

A second classification is into *nonoscillatory* behaviour, for which some solution and hence every solution has only finitely many zeros in some neighbourhood of the endpoint e; and *oscillatory*, for which every solution has infinitely many zeros near e. For LC endpoints this classification is independent of E; for LP endpoints it may be E-dependent.

For each singular endpoint $x = e$ of Eq. (6.1) one and only one of the following cases occurs:

- O: Eq. (6.1) is oscillatory at $x = e$ for all real E.
- N: Eq. (6.1) is nonoscillatory at $x = e$ for all real E.
- N/O: There exists a real number Λ such that (6.1) is nonoscillatory at $x = e$ for all $E \in (-\infty, \Lambda)$ and oscillatory at $x = e$ for $E \in (\Lambda, +\infty)$. The cutoff value Λ may be oscillatory or nonoscillatory.

Independently of square-integrability, if for a given real E solutions of the Sturm-Liouville differential equation are nonoscillatory at e, then there is a unique (up to a constant multiple) 'small' solution $y_p(x)$, called *principal* or *subdominant* , such that if z is any solution linearly independent of y_p, we have

$$y_p(x)/z(x) \to 0 \text{ as } x \to e. \tag{6.4}$$

For this and other related results on principal solutions see [45]. For both LPN (limit-point nonoscillatory) and LCN (limit-circle nonoscillatory), the 'small' principal solution is the most numerically stable. In the LCN case there is one special boundary condition, the *Friedrichs boundary condition,* which selects the principal solution for any E. This is the boundary condition that is relevant in almost all physical applications. For a discussion of the Friedrichs boundary condition, see [94].

6.2.3 Classifying the spectrum

Whereas the spectrum of a regular Sturm-Liouville problem always consists of a sequence of isolated, simple eigenvalues tending monotonely to $+\infty$, that of a singular problem is a closed infinite subset of the real line which can show a wide variety of 'shapes'. The most common cases are:

(a) The eigenvalues form an infinite sequence bounded below with $+\infty$ the only accumulation point, as for a regular Sturm-Liouville problem;

(b) As case (a), but a sequence unbounded in both directions, i.e. tending to both $-\infty$ and $+\infty$;

(c) An infinite sequence of eigenvalues, bounded below, with one finite accumulation point E_c such that all $E \geq E_c$ are in the continuous spectrum σ_c;

(d) A finite, possibly empty, sequence all less than E_c such that all $E \geq E_c$ are in σ_c.

Example 6.1 An example of a singular problem having both discrete spectrum (eigenvalues) and continuous spectrum (σ_c) is the hydrogen atom equation

$$y'' = (-1/x + 2/x^2 - E)y, \quad a = 0,\ b = +\infty. \tag{6.5}$$

This problem has a discrete spectrum with exact eigenvalues of the form $E_k = -1/(2k+4)^2$, $k = 0, 1, \ldots$ and a continuous spectrum $\sigma_c = (0, +\infty)$.

The limit-circle/limit-point (LC/LP) and nonoscillatory/oscillatory (N/O) classification of endpoints tells us a lot about the spectrum. There are five possible combinations at an endpoint:

- **LCN**: LC and nonoscillatory for all real E.
- **LCO**: LC and oscillatory for all real E.
- **LPN**: LP and nonoscillatory for all real E.
- **LPN/O**: LP and nonoscillatory for E less than some critical Λ, oscillatory for $E > \Lambda$. The changeover point Λ is the infimum of the *continuous spectrum*.
- **LPO**: LP and oscillatory for all real E.

This includes cases of regular endpoints since we may regard a regular endpoint as an example of an LCN endpoint. There are close connections between the oscillatory/ nonoscillatory behaviour of solutions, the LP/LC endpoint classifications, and the qualitative properties of the spectrum, particularly, the location of discrete and continuous spectra, and boundedness below of the spectrum.

A problem with no LCO or LPO endpoint has a spectrum bounded below, and the eigenvalues can be counted from the lowest one E_0 upward to form an increasing sequence (E_k) (possiblily finite or empty), the integer k, the eigenvalue index, being the number of zeros of the associated eigenfunction in (a, b). This case is thus the closest to a regular problem.

By contrast a problem having one LCO endpoint and one LCN or regular endpoint will have a discrete spectrum with an infinite decreasing sequence extending to $-\infty$, as well as an increasing sequence tending to $+\infty$. Then all the eigenfunctions have infinitely many zeros clustering at the LCO endpoint, so the eigenvalues can no longer be labeled by the number of zeros in (a, b) as in the nonoscillatory case.

When one of the endpoints is LPN/O there is a finite or infinite set, possibly empty, of eigenvalues $E_0 < E_1 < \cdots < E_k$, bounded above by a continuous spectrum.

For a LPO endpoint, for all real E the solutions oscillate infinitely often and are not square-integrable. The spectrum is then the whole real line and there are no eigenvalues.

There are also some other possibilities, such as bands of continuous spectrum, separated by gaps. Such situations can occur when the coefficient functions are oscillating on an infinite interval.

Our MATSLISE package (see chapter 7) only solves problems with no oscillatory (LCO or LPO) endpoints, that is problems with a spectrum bounded below (and possibly bounded above by a continuous spectrum) where the kth eigenfunction has precisely k zeros in the interval (a, b). The numerical problems in the oscillatory case are particularly difficult. In this case every eigenfunction has an infinite number of zeros in any neighbourhood of the endpoint and specialized techniques are required (see [18]).

6.2.4 The automatic classification of Sturm-Liouville problems

As already mentioned, Sturm-Liouville problems can be classified as regular or singular, limit point or limit circle, oscillatory or nonoscillatory. From the classical Sturm-Liouville solvers only SLEDGE [101] has an automatic endpoint classification algorithm build-in. The algorithm (by S. Pruess, C.T. Fulton and Y. Xie) is based on a number of evaluations of the coefficient functions near the endpoints and the returned classification information can be used to improve the determination of eigenvalues and eigenfunctions.

6.3 Problems defined on an infinite integration interval

In the previous chapters we devised the CPM and LPM algorithms for a Schrödinger problem defined on a finite integration interval. In this section we now discuss the development of a truncation algorithm for Schrödinger problems defined on an infinite integration interval. We adopt the technique developed in [57] for anharmonic oscillators, based on a WKB-approximation (named after Wentzel[128] - Kramers[71] - Brillouin[27]) and apply it on a larger class of potentials.

We also show that a separate technique, with better results, can be introduced for the potentials with a Coulomb-like tail. The explicit use of the asymptotic form of a Coulomb equation leads us to a smaller cutoff value (truncation point) and more precise boundary conditions.

We will use the truncation algorithms in combination with a CPM, but the procedures discussed in this section can equally well be applied for the LPM.

6.3.1 Truncation of an infinite integration interval

Introduction

Let us first illustrate things by considering an example. Suppose the problem consists in computing the first eigenvalues of the hydrogen problem

$$y''(x) = \left(\frac{l(l+1)}{x^2} - \frac{1}{x} - E \right) y(x), \quad x \in (0, +\infty), \tag{6.6}$$

giving exact eigenvalues $E_k = -1/(2k+4)^2$, $k = 0, 1, \ldots$ when $l = 1$. We simply impose the regular boundary conditions $y(\epsilon) = 0 = y(b^*)$ where ϵ is a small and b^* is a large x-value, and solve the resulting problem on $[\epsilon, b^*]$. Table 6.1 shows the obtained

Table 6.1: The eigenvalues of the truncated hydrogen problem on $[0.0001, 1000]$, computed with CPM{16,14} and $tol = 10^{-12}$.

k	$\tilde{E}_k$	$\lvert E_k - \tilde{E}_k \rvert$	k	$\tilde{E}_k$	$\lvert E_k - \tilde{E}_k \rvert$
0	-0.062500000000	$4.41(-15)$	11	-0.001479288720	$1.22(-9)$
1	-0.027777777778	$1.14(-15)$	12	-0.001275339874	$1.70(-7)$
2	-0.015625000000	$1.30(-16)$	13	-0.001106144110	$4.97(-6)$
3	-0.010000000000	$3.94(-16)$	14	-0.000939152963	$3.74(-5)$
4	-0.006944444444	$8.02(-16)$	15	-0.000744066802	$1.21(-4)$
5	-0.005102040816	$1.29(-15)$	16	-0.000515946529	$2.56(-4)$
6	-0.003906250000	$1.50(-15)$	17	-0.000257573592	$4.35(-4)$
7	-0.003086419753	$1.52(-15)$	18	0.000028739013	$6.54(-4)$
8	-0.002500000000	$1.45(-15)$	19	0.000341439723	$9.08(-4)$
9	-0.002066115702	$9.13(-16)$	20	0.000679441392	$1.20(-3)$
10	-0.001736111109	$1.75(-12)$	21	0.001041942107	$1.51(-3)$

eigenvalue approximations $\tilde{E}_k$ for the truncated problem with $\epsilon = 0.0001$ and $b^* = 1000$. The CPM{16,14} method is used with a user input tolerance $tol = 10^{-12}$. The first eigenvalues from the truncated interval $\tilde{E}_k$ obviously agree with those from the infinite interval E_k. However, when k is further increased the agreement gradually deteriorates. In particular, all eigenvalues with $k \geq 18$ are even positive. An increase with k of the value of b^* is then needed to preserve a certain level of accuracy.

The need to increase the value of b^* with the index k is also illustrated in Figure 6.1, where the first eigenfunctions of the hydrogen equation are shown. The horizontal dotted lines represent the energy levels of the (exact) eigenvalues and the associated wavefunctions (eigenfunctions) are shown on the same level. The wavefunction is oscillating in the region where the eigenvalue E is larger than the potential function (i.e. $E > V(x)$) but it decreases exponentially in the so-called classically forbidden region ($E < V(x)$). The value of the truncation point b^* for a certain value of E should thus be taken far enough into the classically forbidden region to be able to impose $y(b^*) = 0$ as boundary condition without loss of accuracy in the eigenvalue calculations.

Selection of the cutoff value

The need for finding a rule for an accurate updating of the cutoff value b^* in terms of E is not restricted to the case of Coulomb-like potentials, as in the hydrogen example. The problem was already considered in [57] by Ixaru for oscillators. He proposed an algorithm based on the WKB-approximation. We will show now that this algorithm is also applicable to other potential forms. We will describe the procedure for the case where b is infinite. An infinite endpoint a can then be treated in the same way.

Let us fix the value of E and let x_t be the corresponding outer turning point, that is the rightmost point where $E = V(x)$. We also assume that $V(x) > E$ for all $x > x_t$. As we are interested in the physically acceptable wavefunction we normally impose the

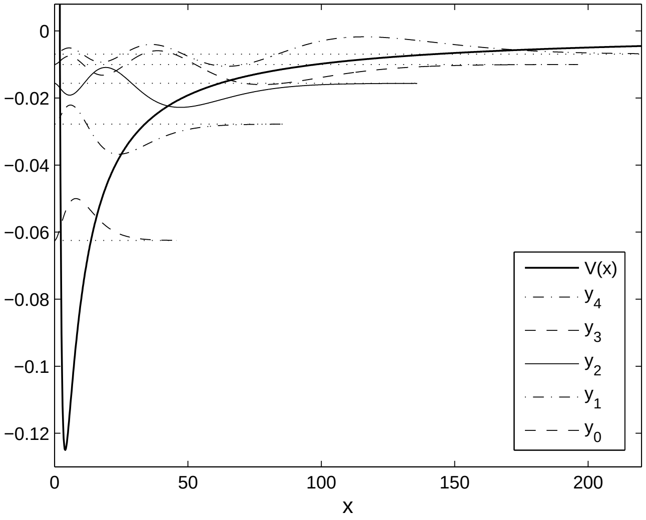


Figure 6.1: The first eigenvalues, eigenfunctions and potential of the hydrogen equation

asymptotic condition $\lim_{x\to\infty} y(x) = 0$. The Schrödinger equation has two linearly independent solutions and for $x > x_t$ these are well described by the WKB approximation, which means in essence that

$$y_{\pm}(x) \sim \exp[\pm w(x_t, x)] \tag{6.7}$$

where

$$w(x_1, x_2) = \int_{x_1}^{x_2} z(t)dt \quad \text{with} \quad z(x) = [V(x) - E]^{1/2}. \tag{6.8}$$

As a matter of fact, the genuine WKB formulae contain also a factor $[V(x)-E]^{-1/4}$ but its consideration would only complicate the derivation without altering the main conclusion. The form (6.7)-(6.8) allows using simple relations like

$$y_{\pm}(x) = 1/y_{\mp}(x) \tag{6.9}$$

and

$$y'_{\pm}(x) = \pm z(x)y_{\pm}(x) \tag{6.10}$$

in deriving the formulae below.

The problem is to determine b^* as the leftmost point such that the cut does not affect the accuracy. In applications initial conditions which mimic the physical condition will be imposed at this point (these are $y(b^*) = 0$ and $y'(b^*) = A$) and the numerical solution will be propagated backwards. The value of the constant $A \neq 0$ is arbitrary because the equation is homogenous. The general solution is a linear combination of the two linearly independent solutions,

$$y(x) = \alpha_+ y_+(x) + \alpha_- y_-(x)\,, \tag{6.11}$$

and similarly for its first derivative,

$$y'(x) = \alpha_+ y'_+(x) + \alpha_- y'_-(x)\,. \tag{6.12}$$

On imposing the stated initial conditions and applying the relations (6.9) and (6.10) we get

$$\alpha_+ = \frac{Ay_-(b^*)}{2z(b^*)}\,,\ \alpha_- = -\frac{Ay_+(b^*)}{2z(b^*)}\,,$$

and then

$$y(x) = \frac{Ay_+(b^*)y_-(x)}{2z(b^*)}(S(x\,,b^*)-1)\,, \tag{6.13}$$

$$y'(x) = \frac{Az(x)y_+(b^*)y_-(x)}{2z(b^*)}(S(x\,,b^*)+1)\,, \tag{6.14}$$

where

$$S(x,\,b^*) = [y_+(x)y_-(b^*)]^2 = \exp[-2w(x,\,b^*)]. \tag{6.15}$$

We see that $S(x,\,b^*)$ depends on the distance between x and b^*, to decrease fastly when x moves to the left, for fixed b^*, or when b^* moves to the right, for fixed x. The influence of the position of b^* on the value of the logarithmic derivative at some x, that is

$$\frac{y'(x)}{y(x)} = z(x)\frac{S(x\,,b^*)+1}{S(x\,,b^*)-1}\,,$$

depends on how $S(x,\,b^*)$, which measures the influence of the propagation from b^* to x, compares with the other term (1 or -1) to which it has to be added. For double precision arithmetic, $S(x,\,b^*)$ is no longer 'seen' if it is smaller than 10^{-16}. It follows that no gain has to be expected if b^* is chosen bigger than the value which ensures simply that $w(x,\,b^*) \approx 18$ (equivalent to $S(x,\,b^*) = \exp[-2w(x,\,b^*)] \approx 10^{-16}$). Taking for x the turning point, the condition for the determination of the suitable b^* is

$$w(x_t,\,b^*) = \int_{x_t}^{b^*} [V(x)-E]^{1/2}dx \geq 18 \tag{6.16}$$

and this will be used in all runs.

What this condition (6.16) actually says is that the area enclosed by the energy level and the potential function should be sufficiently large. This is illustrated by Figure 6.2: when the potential function $V(x)$ increases rapidly the cutoff point b^* can be close to the turning point x_t; when the potential increases less rapidly the cutoff point must be chosen further away from the turning point.

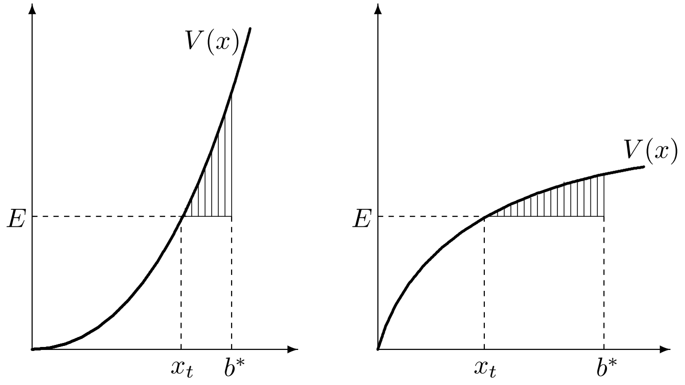


Figure 6.2: The selection of the cutoff value using the WKB-condition (6.16).

Eigenvalue computation

Suppose that we want to compute the energy spectrum between E_{min} and E_{max}. First a suitable cutoff b^* is computed for $E = E_{\text{max}}$, i.e. we look for the leftmost point b^* such that $\int_{x_t}^{b^*} [V(x) - E_{\text{max}}]^{1/2} dx \geq 18$. The integral is approximated by repeated application of the trapezoidal rule which is accurate enough for such an estimation. As for a regular problem, we then generate a partition over the interval $[a, b^*]$ and calculate the quantities associated to this partition. One of these quantities is the array $\bar{V}$ which contains for each step the constant reference potential used by the CPM algorithm. For each trial value $\bar{E}$ of E in $[E_{\text{min}}, E_{\text{max}}]$ appearing in the shooting procedure, a truncation point $\bar{b} \in [a, b^*]$ can then be found. The integral used to deduce a value for $\bar{b}$ can now easily be approximated using $\bar{V}$ in the quadrature formula (so no new function evaluations of $V(x)$ are necessary):

$$\int_{x_t}^{\bar{b}} [V(x) - \bar{E}]^{1/2} dx \approx \sum_{i=i_t}^{i_{\bar{b}}} h_i [\bar{V}_i - \bar{E}]^{1/2} \qquad (6.17)$$

where i_t is the index of the interval containing the outer turning point. h_i and $\bar{V}_i$ are the stepsize and constant reference potential of the i^{th} step. The solution is then propagated in the shooting procedure from a up to the matching point x_m and from $\bar{b}$ (where the solution is assumed to be zero, i.e. the right boundary condition is taken as $y(\bar{b}) = 0$) down to x_m to generate the mismatch. A new trial value $\bar{E}$ is calculated in terms of this mismatch and the procedure is repeated as many times as necessary to obtain the eigenvalue within the tolerance specified by the user.

Table 6.2: The first eigenvalues of the harmonic oscillator calculated with CPM$\{16,14\}$ and $tol = 10^{-12}$.

k	$\bar{b}$	E_k
0	5.7306	0.999999999998
1	6.1815	2.999999999999
2	6.6459	4.999999999999
3	6.6459	6.999999999998
4	7.0940	8.999999999997
5	7.5422	10.999999999996
6	7.9903	12.999999999995
7	7.9903	14.999999999994
8	8.4360	16.999999999999
9	8.4360	18.999999999995
10	8.4360	20.999999999995

Table 6.3: Some higher eigenvalues of the harmonic oscillator calculated with CPM$\{16,14\}$ and $tol = 10^{-12}$.

k	$\bar{b}$	E_k
100	17.4590	201.000000000003
500	35.0838	1000.999999999998
1000	50.1788	2001.000000000012

Numerical illustrations

Consider the harmonic oscillator defined over an infinite integration interval:

$$y'' = \left(x^2 - E\right) y, \quad x \in (-\infty, +\infty) \tag{6.18}$$

with exact eigenvalues given by $E_k = 2k + 1$, $k = 0, 1, \ldots$. We solved this problem with the CPM$\{16,14\}$ method included in MATSLISE (with input tolerance $tol = 10^{-12}$). Table 6.2 shows the first eigenvalues obtained in MATSLISE. When asking for the calculation of the first 11 eigenvalues, the value obtained for b^* is 8.4360 (and $a^* = -b^*$ due to the symmetry of the problem). The partition is then constructed over the truncated integration interval $[a^*, b^*]$. However during the shooting process, the solution is only propagated on different $[\bar{a}, \bar{b}]$ intervals with $\bar{a}$ and $\bar{b}$ two meshpoints in the partition and $\bar{b} \leq b^*$, $\bar{a} = -\bar{b}$. The second column shows the value of $\bar{b}$ obtained for each eigenvalue. Figure 6.3 shows the eleven lowest-energy eigenfunctions of the harmonic oscillator problem. Again the horizontal dotted lines represent the energy levels of the (exact) eigenvalues and the associated wavefunctions (eigenfunctions) are shown on the same level. It is clear that the chosen $\bar{b}$ values are all situated in the classically forbidden region and that an eigenfunction can be assumed to be zero in its corresponding $\bar{b}$ value. Table 6.3 shows that also for higher eigenvalues good cutoff values $\bar{b}$ are obtained.

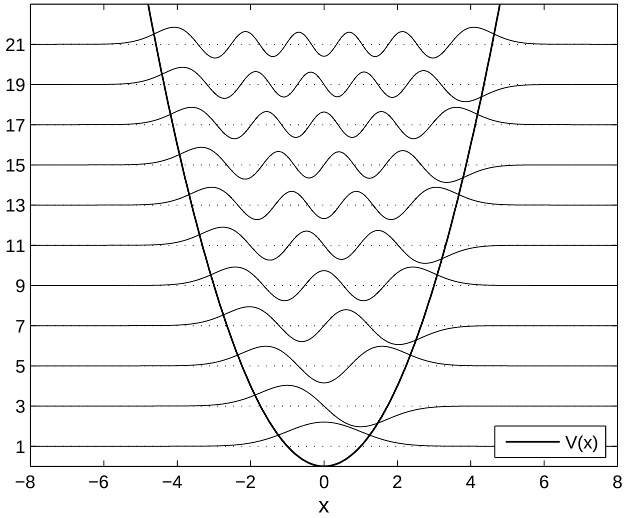


Figure 6.3: The first eleven eigenfunctions of the harmonic oscillator.

Table 6.4: Some eigenvalues of the hydrogen problem calculated with CPM$\{16,14\}$ and $tol = 10^{-12}$.

k	$\bar{b}$	E_k
0	1.05E2	-0.0625000000000
10	1.27E3	-0.0017361111111
100	5.32E4	-0.0000240292195
1000	4.25E6	-0.0000000249003

Let us also reconsider the hydrogen problem. Table 6.4 lists the eigenvalue approximations and truncation points for some different k values. All digits shown in the eigenvalue approximations are exact. The value of $\bar{b}$ increases rapidly with k.

6.3.2 Adapted boundary conditions for Coulomb-like potentials at large distance

As shown in the previous section, a WKB-approach can be used to obtain good choices for the cutoff values of the integration interval. These cutoff points are chosen large enough such that the solution in these points may be assumed to be zero. However the algorithm can be improved for problems with a potential which behaves as a Coulomb potential in the asymptotic range, that is for large x values. Using an approximation of the asymptotic Coulomb function more precise boundary conditions can be constructed which allows us to take smaller cutoff values.

The Coulomb equation in the asymptotic region

We consider the Coulomb equation of the form

$$y'' + \left(E + \frac{Z}{x} - \frac{l(l+1)}{x^2}\right) y = 0, \quad x > 0. \tag{6.19}$$

where Z is a constant. This Coulomb equation is an example of a radial Schrödinger equation, as we will see in section 6.4.

With the change of variable

$$x = \frac{r}{2\sqrt{-E}}, \tag{6.20}$$

and dividing by $-4E$ Eq. (6.19) becomes:

$$\frac{d^2w}{dr^2} + \left(-\frac{1}{4} + \frac{Z}{2\sqrt{-E}r} - \frac{l(l+1)}{r^2}\right) w = 0, \quad w(r) = y(x(r)). \tag{6.21}$$

When we take

$$\mu = l + \frac{1}{2}, \;\; \kappa = \frac{Z}{2\sqrt{-E}} \tag{6.22}$$

we obtain the Whittaker differential equation [3]

$$\frac{d^2w}{dr^2} + \left(-\frac{1}{4} + \frac{\kappa}{r} + \frac{\frac{1}{4} - \mu^2}{r^2}\right) w = 0. \tag{6.23}$$

The solution of this equation can be expressed in terms of the second confluent hypergeometric function (see Eq. (13.1.33) in [3])

$$W_{\kappa,\mu}(r) = e^{-\frac{r}{2}} r^{\mu+\frac{1}{2}} U(\frac{1}{2} + \mu - \kappa, 1 + 2\mu, r), \tag{6.24}$$

where, according to Eq. (13.5.2) in [3], $U(a,b,r)$ can be written for large r as

$$U(a,b,r) = r^{-a} \sum_n \frac{(a)_n (1+a-b)_n}{n!} (-r)^{-n} \tag{6.25}$$

with

$$\begin{aligned}(a)_0 &= 1,\\ (a)_n &= a(a+1)\ldots(a+n-1).\end{aligned}$$

This means that for large r, we can write Eq. (6.24) as

$$W_{\kappa,\mu}(r) = e^{-\frac{r}{2}} r^{\kappa} \sum_n c_n r^{-n} \tag{6.26}$$

where the coefficients c_n are defined by

$$\begin{aligned} c_0 &= 1, \\ (n+1)c_{n+1} &= -(l+1-\kappa+n)(-l-\kappa+n)c_n. \end{aligned} \tag{6.27}$$

The first derivative (with respect to the original Coulomb-variable x) is then

$$\frac{dW_{\kappa,\mu}}{dx} = 2\sqrt{-E}\,\frac{dW_{\kappa,\mu}}{dr} \tag{6.28}$$

with (for large r)

$$\frac{dW_{\kappa,\mu}}{dr} = -\frac{1}{2}W_{\kappa,\mu}(r) + \frac{\kappa}{r}W_{\kappa,\mu}(r) - e^{-\frac{r}{2}}r^{\kappa}\sum_n nc_n r^{-n-1}. \tag{6.29}$$

We have the additional result that the derivative of $W_{\kappa,\mu}$ and $W'_{\kappa,\mu}$ with respect to E can be expressed as

$$\frac{dW_{\kappa,\mu}}{dE} = -\frac{x}{\sqrt{-E}}\frac{dW_{\kappa,\mu}(r)}{dr} \tag{6.30}$$

and

$$\begin{aligned} \frac{d}{dE}\frac{dW_{\kappa,\mu}}{dx} &= -\frac{x}{\sqrt{-E}}\frac{d}{dr}\frac{dW_{\kappa,\mu}(r)}{dx} \\ &= -2x\frac{d^2W_{\kappa,\mu}(r)}{dr^2} \\ &= 2x\left(-\frac{1}{4}+\frac{\kappa}{r}+\frac{\frac{1}{4}-\mu^2}{r^2}\right)W_{\kappa,\mu}(r). \end{aligned} \tag{6.31}$$

These derivatives with respect to the energy are needed in the Newton iteration procedure to obtain the eigenvalue approximations as the roots of the shooting mismatch function. This means that we need the derivatives with respect to the energy to evaluate $\phi'(E_t)$ in the Newton formula

$$E_{\mathrm{t}+1} = E_t - \phi(E_t)/\phi'(E_t) \tag{6.32}$$

where $\phi(E) = y_L y'_R - y_R y'_L$ is the mismatch function (see section 3.2.2).

Adapted boundary conditions for problems with a Coulomb-like potential in the asymptotic region

The formulae derived above can be used to improve the truncation algorithm for problems which behave as a Coulomb problem in the asymptotic region. Instead of assuming that the solution is zero in the truncation point $\bar{b}$ we can use Eq. (6.26) and Eqs. (6.28),(6.29) to obtain more precise values for $y(\bar{b})$ and $y'(\bar{b})$.

We also want to mention that as the eigenvalue index k increases the eigenvalues of a Coulomb problem come close to zero. As a result, κ increases very rapidly with increasing k and the factor r^{κ} in Eq. (6.26) and Eq. (6.29) can give rise to some overflow problems. However in the shooting procedure the values of the left-hand and right-hand solutions can be normalized arbitrarily. This means that we only need to know the ratio $W'_{\kappa,\mu}/W_{\kappa,\mu}$ in $\bar{b}$. So we construct the expressions for the scaled wavefunctions $\bar{W}_{\kappa,\mu}$ such that $\bar{W}_{\kappa,\mu} = 1$, i.e.:

$$\bar{W}_{\kappa,\mu}(x) = 1, \tag{6.33}$$

$$\frac{d\bar{W}_{\kappa,\mu}}{dx}(r) = 2\sqrt{-E}\,\Omega(r), \tag{6.34}$$

$$\frac{d\bar{W}_{\kappa,\mu}}{dE}(r) = -\frac{x}{\sqrt{-E}}\,\Omega(r), \tag{6.35}$$

$$\frac{d}{dE}\frac{d\bar{W}_{\kappa,\mu}}{dx}(r) = 2x\left(-\frac{1}{4} + \frac{\kappa}{r} + \frac{\frac{1}{4} - \mu^2}{r^2}\right), \tag{6.36}$$

with

$$\Omega(r) = -\frac{1}{2} + \frac{\kappa}{r} - \frac{\sum_n nc_n r^{-n-1}}{\sum_n c_n r^{-n}}. \tag{6.37}$$

These last expressions are used to compute the boundary conditions in the cutoff point $\bar{b}$ for problems which behave like a Coulomb problem around $\bar{b}$. This allows us to cut off the interval even at points where the solution is not yet zero (or close to zero).

Levin's summation algorithm

The accurate summation of the asymptotic series $\sum_n c_n r^{-n}$ and $\sum_n nc_n r^{-n-1}$ in (6.37) gives rise to some problems. The evaluation of such series by direct summation is very difficult in so much that the required number of terms can become very large and uncontrolled numerical instabilities may occur. Some special approach should be used instead and we have taken advantage of the convergence acceleration and summation procedures. Generally, given a sequence s_n, which can be the sequence of the partials sums of a series, these procedures consist in introducing a transformation of the sequence which enables us to obtain either a more rapidly convergent sequence, if the sequence has a limit, or an approximation of its sum (anti-limit), if the sequence is divergent but deduced from an asymptotic series. There are a large number of such methods (see e.g. [26]). We have used the Levin algorithm [80], the best suited for summing the asymptotic series. We will describe in short the main ideas of this Levin procedure.

If we want to construct a transformation which is able to accelerate the convergence of an infinite series

$$\sum_{n=0}^{\infty} a_n z^{-n}, \tag{6.38}$$

we are confronted with the practical problem that the information contained in a finite string of partial sums $s_0, s_1, \ldots, s_m$ has to be extracted and utilized in a way which is more efficient than the conventional approach of adding up one term after the other. We assume that for all $n \in \mathbb{N}$ a sequence element s_n can be partitioned into the limit s and the remainder r_n

$$s_n = s + r_n. \tag{6.39}$$

This essentially means that we have to find a way of eliminating the remainder r_n and determining the limit s at least approximately by exploiting the information stored in the finite sequence $s_0, s_1, \ldots, s_m$ of the partial sums.

The idea of Levin's transformation is to find a sequence ω_n which gives the leading behaviour of the remainder r_n, so that

$$\frac{s_n - s}{\omega_n} = \frac{r_n}{\omega_n} \to c,\ n \to \infty. \tag{6.40}$$

One chooses

$$\frac{s_n - s}{\omega_n} \approx c_0 + \frac{c_1}{(n+\beta)} + \frac{c_2}{(n+\beta)^2} + \cdots + \frac{c_t}{(n+\beta)^t} \tag{6.41}$$

where β is some nonzero constant. This is a Poincaré-type expansion which is equivalent to

$$s_n \approx s + \omega_n \left(c_0 + \frac{c_1}{(n+\beta)} + \frac{c_2}{(n+\beta)^2} + \cdots + \frac{c_t}{(n+\beta)^t} \right). \tag{6.42}$$

By inserting values of s_n and ω_n into (6.42) we obtain a system of equations which can be solved for s giving an approximate value for it. For this method a recursive scheme also exists (see [127]).

Levin suggested some simple remainder estimates ω_n which can be computed from at most two terms a_n of the series (6.38) to be transformed. With the help of these remainder estimates the following variants of Levin's sequence transformation result:

- $\omega_n = (\beta + n)a_n$ (u transformation)
- $\omega_n = a_n$ (t transformation)
- $\omega_n = a_{n+1}$ (d transformation)
- $\omega_n = (a_n a_{n+1})/(a_n - a_{n+1})$ (v transformation).

We used the d transformation, the most appropriate for our series.

Numerical illustrations

Again we reconsider the hydrogen problem. Table 6.5 shows the same experiment as in Table 6.4, only now we used adapted boundary conditions in $\bar{b}$. These $\bar{b}$ must be sufficiently large so that the asymptotic expansion of the Coulomb equation is valid. Here we selected the values of $\bar{b}$ using the condition $\int_{x_t}^{\bar{b}} [V(x) - \bar{E}]^{1/2} dx \approx \sum_{i=i_t}^{i_{\bar{b}}} h_i [\bar{V}_i - \bar{E}]^{1/2} \geq 2$ (instead of 18 as before). Compared to the results shown in Table 6.4, this means that the $\bar{b}$ can be taken smaller now to reach the same accuracy in the eigenvalue approximations. The condition for $\bar{b}$ is now related only to the Coulomb behaviour and not to the decrease of the solution according to the WKB approximation. For the hydrogen

Table 6.5: Some eigenvalues of the hydrogen problem calculated with CPM$\{16,14\}$ and $tol = 10^{-12}$. Now the adapted boundary condition in $\bar{b}$ is used.

k	$\bar{b}$	E_k
0	6.63E1	-0.0625000000000
10	8.32E2	-0.0017361111111
100	4.60E4	-0.0000240292195
1000	4.11E6	-0.0000000249003

Table 6.6: The first eigenvalues of problem (6.43) calculated with CPM$\{16,14\}$, $tol = 10^{-12}$ and adapted boundary conditions in $\bar{b}_a$.

k	$\bar{b}$	$\bar{b}_a$	E_k
0	110.80	18.35	-0.061681846633
1	137.13	42.56	-0.027498099943
2	203.53	88.74	-0.015501561691
3	243.83	110.80	-0.009935496851
4	340.03	168.07	-0.006906701382

problem, the value of 2 for the above integral was found to be sufficient to ensure an accurate evaluation of the asymptotic series with a reasonable number of terms.

Table 6.6 shows some results for another test potential with a Coulomb-type decay at $x = \infty$:

$$y''(x) = \left(\frac{l(l+1)}{x^2} + \frac{-1 + 5e^{-2x}}{x} - E \right) y(x), \quad x \in (0, +\infty), \tag{6.43}$$

where we take $l = 1$ (see [123]). The eigenvalues were calculated using the adapted boundary conditions in $\bar{b}_a$. For this problem the Coulomb-type decay is reached already for (approximately) $x > 18$, that is the term $5e^{-2x}$ can be neglected for $x > 18$. Therefore we can use a $\bar{b}_a$ value which is very close to the turning point. To obtain the results in Table 6.6, we used the right endpoint of the interval containing the turning point as cutoff value $\bar{b}_a$. The $\bar{b}$ values are the cutoff values which would have been used without adapted boundary conditions, thus the cutoff values selected by the WKB-condition. All figures shown in the eigenvalue approximations are exact.

The procedure can also be used to obtain boundary conditions for a Woods-Saxon problem

$$y''(x) = \left(\frac{l(l+1)}{x^2} - 50\left[1 - 5t/(3(1+t))\right]/(1+t) - E \right) y(x), \quad x \in (0, +\infty), \tag{6.44}$$

with $t = e^{(x-7)/0.6}$, since for large x values the term $50\left[1 - 5t/(3(1+t))\right]/(1+t)$ disappears and only the centrifugal term $l(l+1)/x^2$ remains. When $l = 2$, there are 13

Table 6.7: The eigenvalues of the Woods-Saxon problem ($l = 2$) calculated with CPM{16,14}, $tol = 10^{-12}$ and adapted boundary conditions in $\bar{b} = 15$.

k	E_k
0	-48.34948105212
2	-44.12153737732
4	-38.25342653968
6	-31.02682092177
8	-22.68904151018
10	-13.52230335295
12	-3.97249143284

eigenvalues in the discrete spectrum (see [123]). Table 6.7 shows the approximations obtained for these eigenvalues using the adapted boundary conditions in $\bar{b} = 15$. Again all displayed figures are exact. For this problem the region with the Coulomb-like behaviour starts relatively far from the different turning points, because the central term (with exponentials) is negligible only at rather large distance ($x > 30$). But since the area delimited by the potential and the energy increases quickly after the outer turning point, the solution decays rapidly (according to the WKB approximation) and we can take the cutoff at more reduced distance, where the central part is only a few orders of magnitude smaller than the other terms.

The hydrogen problem, as well as problems (6.43) and (6.44) are radial Schrödinger equations with a potential which exhibits singularities of the form x^{-2} and x^{-1} near the origin. In the next section, we discuss the procedure which was applied to deal with such singularities.

6.4 Solution near the origin for radial Schrödinger equations

An important class of Schrödinger equations is formed by the *radial* Schrödinger equations. The radial Schrödinger equation can be written as

$$y'' = \left(\frac{l(l+1)}{x^2} + V(x) - E\right) y, \quad x > 0, \tag{6.45}$$

where x represents the distance from a spherically symmetric nucleus, and l is a constant arising out of the method of separation of variables applied to the three-dimensional Schrödinger equation. Physically, l is called an orbital rotational quantum number and the term $l(l+1)/x^2$ is the *centrifugal* component of the *effective potential* $l(l+1)/x^2 + V(x)$. The *underlying potential* $V(x)$ tends to a limit, the *dissociation energy*, at infinity.

For the radial Schrödinger problem a specific problem occurs: the potential is singular at the origin and therefore on a short interval around the origin a specially tuned implementation is used, with $l(l+1)/x^2$ as reference potential and the rest of the potential seen as a perturbation. This technique was already described in [58] and [112] for the more general case of a system of coupled channel Schrödinger equations. In [112], Rizea described the PERSYS Fortran code which is included in the CPC library. A different technique, based on $l(l+1)/x^2$+constant as reference potential and valid for a single equation, is proposed in [62]. Here we apply the algorithm from [112] to the one-dimensional Schrödinger problem. This algorithm allows us to obtain the value of the solution in $\epsilon \neq 0$, where ϵ is small enough such that the centrifugal term is numerically dominating with respect to the other terms of the potential. The solution and its derivatives in this ϵ then form the starting values for the integration (using the CPM) on the interval $[\epsilon, \bar{b}]$.

6.4.1 Algorithm

A wide variety of physical problems has a potential which can be put in the form

$$V(x) = \frac{l(l+1)}{x^2} + \frac{S(x)}{x} + R(x), \tag{6.46}$$

where $R(x)$ is the non-singular part. We assume that the functions $S(x)$ and $R(x)$ can be approximated by a second degree polynomial over the interval $[0, \epsilon]$:

$$S(x) = S_0 + S_1 x + S_2 x^2, \quad R(x) = R_0 + R_1 x + R_2 x^2, \tag{6.47}$$

where $S_0, S_1, S_2, R_0, R_1, R_2$ are constants. This means that the potential is approximated by

$$\frac{l(l+1)}{x^2} + \frac{V_{-1}}{x} + V_0 + V_1 x + V_2 x^2 \tag{6.48}$$

where

$$V_{-1} = S_0,\ V_0 = S_1 + R_0,\ V_1 = S_2 + R_1,\ V_2 = R_2. \tag{6.49}$$

With the operator $L = \frac{d^2}{dx^2} - \frac{l(l+1)}{x^2}$ and $\Delta V(x) = \frac{V_{-1}}{x} + V_0 + V_1 x + V_2 x^2 - E$, we can write the radial Schrödinger equation as

$$Ly(x) = \Delta V(x) y(x) \tag{6.50}$$

where the dominant term in x^{-2} is retained in the left-hand side while the other terms are collected in the right-hand side. This suggests a perturbative approach. This means that we introduce a parameter λ and consider the equation

$$L\bar{y}(x) = \lambda \Delta V(x) \bar{y}(x). \tag{6.51}$$

The solution $\bar{y}$ depends on the parameter λ. Upon expanding $\bar{y}(x;\lambda)$ in powers of λ,

$$\bar{y}(x;\lambda) = y_0(x) + \lambda y_1(x) + \lambda^2 y_2(x) + \dots, \tag{6.52}$$

the coefficients $y_q(x)$ are found to satisfy the recurrence relations

$$Ly_0(x) = 0, \quad Ly_{q+1}(x) = \Delta V(x) y_q(x), \;\; q = 0, 1, \ldots . \tag{6.53}$$

For $\lambda = 1$, the expansion (6.52) gives the solution of Eq. (6.50):

$$y(x) = y_0(x) + y_1(x) + y_2(x) + \ldots . \tag{6.54}$$

For the zeroth order solution one takes

$$y_0(x) = x^{l+1} \tag{6.55}$$

since x^{l+1} is the regular solution of the equation $Ly_0(x) = 0$. The first perturbation $y_1(x)$ is then obtained using the recurrence relation in (6.53),

$$Ly_1(x) = \Delta V(x) y_0(x) = \Delta V(x) x^{l+1} \tag{6.56}$$

which can also be written as

$$Ly_1(x) = \sum_{p=1}^{4} A_1^p x^{l+p-1} \tag{6.57}$$

where $A_1^1 = V_{-1}$, $A_1^2 = \bar{V}_0 = V_0 - E$, $A_1^3 = V_1$ and $A_1^4 = V_2$. The first perturbation $y_1(x)$ is then of the form $y_1(x) = \sum_{p=1}^{4} z_p(x)$ where $z_p(x)$ is the solution of the equation

$$Lz_p(x) = A_1^p x^{l+p-1}. \tag{6.58}$$

In general, $z_p(x) = s_0(x) + s_p(x)$ where $s_0(x)$ is the regular solution of the homogeneous equation and $s_p(x)$ is a particular solution of the non-homogeneous equation. $s_0(x)$ is already included in the final solution (it is $y_0(x)$). It only remains to compute the particular solution. Suppose z_p is of the following form

$$z_p(x) = B_1^p x^k, \tag{6.59}$$

then Eq. (6.58) gives

$$[k(k-1) - l(l+1)] B_1^p x^{k-2} = A_1^p x^{l+p-1}. \tag{6.60}$$

By identification, it results that $k = l + p + 1$ and

$$B_1^p = \frac{A_1^p}{p(1 + p + 2l)}. \tag{6.61}$$

The first order solution y_1 is then introduced in the right-hand side of (6.53) to compute y_2 and so on. Each q^{th} perturbation is then obtained from an equation of the form

$$Ly_q(x) = \sum_{p=q}^{4q} A_q^p x^{l+p-1} \tag{6.62}$$

with

$$\begin{aligned}
A_q^q &= V_{-1}B_{q-1}^{q-1},\\
A_q^{q+1} &= V_{-1}B_{q-1}^{q} + \bar{V}_0 B_{q-1}^{q-1}\\
A_q^{q+2} &= V_{-1}B_{q-1}^{q+1} + \bar{V}_0 B_{q-1}^{q} + V_1 B_{q-1}^{q-1}\\
A_q^{q+3} &= V_{-1}B_{q-1}^{q+2} + \bar{V}_0 B_{q-1}^{q+1} + V_1 B_{q-1}^{q} + V_2 B_{q-1}^{q-1}\\
\ldots &= \ldots\\
A_q^{4q-3} &= V_{-1}B_{q-1}^{4(q-1)} + \bar{V}_0 B_{q-1}^{4q-5} + V_1 B_{q-1}^{4q-6} + V_2 B_{q-1}^{4q-7}\\
A_q^{4q-2} &= \bar{V}_0 B_{q-1}^{4(q-1)} + V_1 B_{q-1}^{4q-5} + V_2 B_{q-1}^{4q-6}\\
A_q^{4q-1} &= V_1 B_{q-1}^{4(q-1)} + V_2 B_{q-1}^{4q-5}\\
A_q^{4q} &= V_2 B_{q-1}^{4(q-1)}.
\end{aligned} \tag{6.63}$$

We can write the perturbation as $y_q(x) = \sum_{p=q}^{4q} z_p(x)$ where each $z_p(x)$ is of the form

$$z_p(x) = B_q^p x^{l+p+1}, \tag{6.64}$$

with

$$B_q^p = \frac{A_q^p}{p(1+p+2l)}. \tag{6.65}$$

As shown in [112] the calculation of the perturbations $y_1, y_2, \ldots$ leads to values which typically decrease in magnitude. In our implementation, we add (iteratively) as many perturbation corrections y_q as necessary to reach a certain preset accuracy.

6.4.2 Fitting of the potential

The algorithm requires an initial approximation of the potential functions $S(x)$ and $R(x)$ of Eq. (6.46) by second degree polynomials. An approximation by shifted Legendre polynomials is used, known to give the best fit in the least square sense. We approximate the function $S(x)$ (the same can be done for $R(x)$) by

$$S(x) \approx \sum_{j=0}^{2} c_j P_j^*(x) \tag{6.66}$$

where the P_j^* are the shifted Legendre polynomials (see [3]):

$$P_0^*(t) = 1, \;\; P_1^*(t) = 2t - 1, \;\; P_2^*(t) = 6t^2 - 6t + 1. \tag{6.67}$$

Using a least squares procedure, it can be shown that the coefficients c_j should be chosen as follows:

$$c_j = \frac{2j+1}{\epsilon} \int_0^{\epsilon} S(r) P^*(r/\epsilon) dr. \tag{6.68}$$

A four point Gauss quadrature rule was used to evaluate these integrals. The approximant of $S(x)$ can then be expressed as a polynomial in x:

$$S(x) \approx S_0 + S_1 x + S_2 x^2 \tag{6.69}$$

with $S_0 = c_0 - c_1 + c_2$, $S_1 = 2c_1/\epsilon - 6c_2/\epsilon$ and $S_2 = 6c_2/\epsilon^2$. The quality of the solution of the original equation depends of course on how good is the approximation of the S and R functions by polynomials of second degree. A value for ϵ should be chosen which is small enough such that the fitting by a parabola is sufficiently accurate and there is a strong domination of the reference potential (proportional to x^{-2}).

It is also important to mention that the coefficients S_0, S_1, S_2 and R_0, R_1, R_2 do not depend on the energy E and have thus to be computed only once at the beginning of the whole procedure.

6.5 Other singularities: numerical treatment

The algorithm considered in section 6.4 only deals with a specific type of singularities. For other types of singularities it is possible to develop similar procedures which compute the solution in a truncation point close to the singular endpoint. The user can also apply the interval truncation manually, the user chooses a sequence of regular endpoints converging to the singular one and applies the CPM to each of the regular problems. Of course, this process can also be done automatically. An algorithm can be constructed which selects an initial point $b^* < b$ (in the assumption that b is the singular endpoint). The partition $a = x_0 < x_1 < \cdots < x_N = b^*$ is then first constructed on the interval $[a, b^*]$ and the eigenvalue approximation is computed over this partition. An additional meshpoint $x_{N+1} \in (b^*, b)$ can then be added to the partition and a new eigenvalue approximation is computed. This process can be repeated until a number of successive eigenvalue approximations seem to agree within the requested accuracy. In each iteration, the shooting algorithm for the next eigenvalue approximation is started using the previous approximation, so that the process gets faster as the truncated endpoint comes closer to b.

6.6 Conclusion

In this chapter we considered the treatment of some singular problems. In particular, we discussed an interval truncation algorithm for problems defined on an infinite integration interval. This truncation procedure is based on the WKB-approximation of the wavefunction and selects a cutoff point which is large enough such that the solution may be assumed to be zero there. For a Schrödinger problem with a potential which behaves as a Coulomb potential at large distance, we described a more accurate procedure to compute the value of the solution in the cutoff point.

For the important class of radial Schrödinger equations, we discussed an algorithm which can be used in a small region around the origin. This algorithm deals with the singularity of the problem in the origin and computes the value of the solution in a small value ϵ. This value of the solution in ϵ then forms the boundary condition in ϵ for the truncated problem defined over $[\epsilon, b^*]$ which can be solved by one of the CPM.

Chapter 7

The MATSLISE package

MATSLISE is a MATLAB package collecting the CPM and LPM codes for (one-dimensional) Sturm-Liouville and Schrödinger equations. In this chapter we discuss the structure and use of this package. Given the interest of researchers in various fields (quantum chemistry, quantum physics, ...) in this type of software, a user-friendly graphical user interface has been built on top of the package. This graphical user interface allows one to enter the input in a straightforward manner, to control certain parameters interactively and to present the results graphically.

7.1 The MATLAB language

MATLAB is a software tool and programming environment that has become common place among scientists and engineers. Working in MATLAB has however advantages and drawbacks. With the purpose of a tool for research and education, the advantages are briefly

- MATLAB is a wide-spread, standardized programming environment with a big number of built-in functionalities. Many useful mathematical functions and graphical features are integrated with the language.
- Programming in MATLAB is easy.
- The multitude of MATLAB toolboxes allows programmers to choose from a large number of prewritten functions to accomplish tedious or hard tasks.
- MATLAB runs on many platforms and operating systems.
- A MATLAB package like MATSLISE requires no installation or compilation (when MATLAB is installed).

Whereas the drawback are mainly two points:

- MATLAB is commercial and costly.

- Although operations on matrices are very fast, the overall speed of MATLAB can be poor. Since MATLAB is an interpreted (i.e. not pre-compiled) language, it can be slow compared to other compiled languages (Fortran, C++).

7.2 The MATSLISE package

MATSLISE contains an implementation of the CPM algorithms CPM$\{12,10\}$, CPM$\{14,12\}$, CPM$\{16,14\}$ and CPM$\{18,16\}$ discussed in section 3.4 and of the LPM code LPM[4,2] (see section 4.2). These PPM algorithms are used as propagation methods in a shooting procedure in order to compute eigenvalue approximations (as described in sections 3.2 and 4.4). A Prüfer representation is applied which makes the search of the eigenvalues more efficient. Also the truncation algorithms discussed in chapter 6 for problems defined on an infinite integration interval are included in MATSLISE as well as the procedure discussed in section 6.4 which deals with the singularity in the origin of a radial Schrödinger problem.

As mentioned in chapter 6, it is not possible to solve problems with oscillatory endpoints using MATSLISE. This means that MATSLISE only solves problems which have a spectrum bounded below where the kth eigenfunction has precisely k zeros in the interval (a, b). The spectrum can be bounded above by a continuous spectrum, this means that one of the endpoints may be LPN/O. MATSLISE includes (part of) the SLEDGE automatic classification algorithm [101] to determine the form of the eigenvalue spectrum, i.e. to know the number of eigenvalues in the discrete spectrum or to detect if there is a continuous spectrum and where it starts. This information returned by the classification algorithm is used by MATSLISE to return an error message e.g. when the user asks to solve a problem with an oscillatory endpoint or when the users wants to compute eigenvalues in an energy-range situated in the continuous spectrum.

The MATSLISE package is available for download at [2]. The file MATSLISE2006.zip contains the latest version of the MATSLISE package. Unzipping the file creates a directory `MATSLISE` with three subdirectories:

- `GUI` : collects all *.fig files and *.m files which produce the graphical user interface (GUI). The GUI can be considered as the top layer of the package: the methods from the GUI call the (public) methods from the `source`-directory. A subdirectory of the `GUI` directory is `predefined_problems`. This subdirectory contains several problems (saved as *.mat files) which are predefined in the GUI. Many of these problems are included in SLTSTPAK [108] or the Pruess-Fulton test set [102] (see also Appendix C).
- `examples` : holds some example MATLAB M-files, demonstrating the use of the different MATSLISE (command line) functions.
- `source` : contains the actual source-code. This directory collects some classes: a number of classes representing the different types of problems: `schrod`, `slp`, `distorted_coulomb`; some classes implementing the actual PPM algorithms: `cpm12_10`, `cpm14_12`, `cpm16_14`, `cpm18_16`, `lpm10` and some auxiliary

classes such as e.g. `transformed_slp` used by Liouville’s transformation.

These three directories need to be added to the MATLAB search path in order to run MATSLISE. The MATLAB `addpath` command can be used for this purpose, e.g. the commands

```
addpath([cd ’/source/’]);
addpath([cd ’/GUI/’]);
addpath([cd ’/examples/’]);
```

add the three directories to the search path when the current directory is the MATSLISE directory.

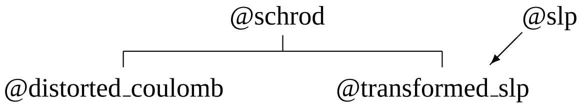
The current version of MATSLISE requires the symbolic toolbox of MATLAB. This symbolic toolbox allows us e.g. to calculate the derivatives of the coefficient functions of a Sturm-Liouville equation in the implementation of Liouville’s transformation.

The numerical solution of a Sturm-Liouville or Schrödinger problem by MATSLISE is subdivided in three stages. In the first stage the partition is constructed, this partition is passed into the second stage where the eigenvalues are calculated. In the third stage it is possible to calculate the eigenfunctions of some of the eigenvalues. Each of these stages has its own methods (or functions) and information is passed from one stage to the other by the input arguments of these methods.

7.2.1 Stage 1: Construction of the partition / Liouville’s transformation

Specification of the problem

First we have to define the problem to be solved. The following scheme gives an overview of the classes used in the representation of the equations:



A Schrödinger problem is represented by an object of the class `schrod`. The PPM algorithms are applied on objects of this type. To be able to apply the PPM algorithm to a Sturm-Liouville problem (i.e. an object of the type `slp`) Liouville’s transformation is used to transform the `slp` object to a `transformed_slp` object. The `transformed_slp` class is a child class of `schrod`: a `transformed_slp` object contains a `schrod` object, namely the Schrödinger problem obtained after Liouville’s transformation, but also some additional fields (such as e.g. the x^G and r^G vectors, see section 3.3.2). Another child class of `schrod` is `distorted_coulomb`. The problems of the `distorted_coulomb` type are radial Schrödinger problems for which the improved truncation algorithm discussed in section 6.3.2 is applied when the problem is defined on an infinite integration interval. Also the treatment of the singularity in the origin seen in section 6.4 is applied for these `distorted_coulomb` problems.

Using the constructor of the class `schrod`

```
 sch = schrod(V,a,b,a0,b0,a1,b1)
 sch = schrod(V,a,b,a0,b0,a1,b1,var)
```

a Schrödinger object is created. The first argument `V` is a string representing the potential function V. The double precision constants `a`, `b`, `a0`, `b0`, `a1` and `b1` specify the integration interval and the boundary conditions. It is allowed to enter `-inf` or `inf` for the two input parameters `a` and `b`. Good truncated endpoints are then determined automatically for every eigenvalue, using the truncation algorithm discussed in chapter 6. If there are less than eight input arguments, then the independent variable is supposed to be x, otherwise the independent variable is the character or string in `var` and `V` is then $V($`var`$)$.

In a very analogous manner a Sturm-Liouville equation is specified:

```
sl = slp(p,q,w,a,b,a0,b0,a1,b1)
sl = slp(p,q,w,a,b,a0,b0,a1,b1,var)
```

where `p`, `q` and `w` are strings representing the coefficient functions $p(x)$, $q(x)$ and $w(x)$ (or $p($`var`$)$, $q($`var`$)$ and $w($`var`$)$) and the double precision numbers `a`, `b`, `a0`, `b0`, `a1`, `b1` are the endpoints of the integration interval and the coefficients of the boundary conditions. Again `inf`-values are allowed for `a` and `b`.

An object representing an equation with a *distorted Coulomb potential*

$$\frac{l(l+1)}{x^2} + \frac{S(x)}{x} + R(x)$$

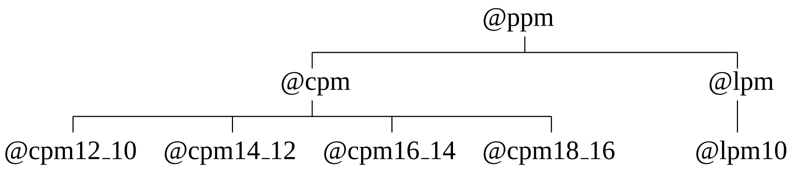
is produced by

```
d = distorted_coulomb(l,S,R,xmax)
d = distorted_coulomb(l,S,R,xmax,var)
```

where `S` and `R` are two strings (representing $S(x)$ and $R(x)$) and the orbital quantum number `l` is a double. `xmax` is the endpoint of the integration interval $(0,$ `xmax`$)$, in many examples `xmax=inf`.

Initialization of the CPM or LPM

In addition to the classes for the equations themselves, a number of classes which implement the actual PPM algorithms, were developed :



These classes collect the methods which construct the partition and calculate the eigenvalues and eigenfunctions. The different PPM methods are instances of the parent class `ppm`. This class collects the information and methods which is used by both the CPM algorithms and the LPM algorithm. The class `cpm` contains all methods and properties which are shared among the various CPM algorithms. The child classes `cpm12_10`, `cpm14_12`, `cpm16_14` and `cpm18_16` only contain the information which is specific for a certain order. In this way a new method of a given order can be added easily. Similarly we have a `lpm10` class, which implements the LPM[4,2] algorithm of order 10. This `lpm10` class is a child class of `lpm`.

The user calls a constructor of a child class e.g.:

```
cp = cpm16_14(slp/schrod/distorted_coulomb,tol)
```

with `tol` a positive constant representing the accuracy requested in the results. This constructor of the child class then calls the constructor of `cpm`. The `cpm` constructor is never called by the user directly. The constructors start the calculation of the partition but when the first input argument is a `slp` object some additional work is done first. The Sturm-Liouville equation is converted to the Schrödinger form: that is the `slp` object is transformed into a `transformed_slp`-object. As already mentioned, a `transformed_slp`-object is a `schrod`-object but contains some more information which is necessary e.g. to obtain the eigenfunctions of the original Sturm-Liouville equation.

The partition of $[a, b]$ (or $(\epsilon, x_{\max}]$ for distorted Coulomb problems where ϵ is set by an empirical formula) is constructed in terms of the tolerance `tol` and some potential dependent expressions (e.g. $C_m^{(u)}$, $C_m^{(u')}$, $C_m^{(v)}$, $C_m^{(v')}$ from Eqs. (3.85)-(3.88) and V_0 from Eq. (3.37) for the CPM), which will be used repeatedly in the second and third stage, are calculated in each step of the partition and stored. Additionally the execution of this stage furnishes the value of the matching point x_m.

It is important to point out that the partition is dictated only by the behaviour of $V(x)$; the value of E is not involved. So the construction of the partition (stage 1) happens completely in advance of and separate from the calculation of the eigenvalues and eigenfunctions (stages 2 and 3). At least this is true for a problem with a finite integration interval. For an infinite integration interval however, the partition is not constructed in the first stage, but in the second stage (see further). The E-independence of the partition is thus lost for an infinite problem: as seen in chapter 6 a higher eigenvalue needs a larger (truncated) integration interval than a lower eigenvalue. This means that in the calculation of the eigenvalues (stage 2) a lengthening of the partition interval may be necessary, which makes the eigenvalue-search for an infinite problem somewhat slower.

7.2.2 Stage 2: Eigenvalue computation

In this stage the eigenvalues, in a range fixed by the user, are calculated. The user starts the calculation by calling a method from the `ppm` class:

```
E = get_eigenvalues(pp_child,pmin,pmax)
```

```
E = get_eigenvalues(pp_child,pmin,pmax,indices)
```

where `pp_child` is an instance of one of the child classes of the `cpm` class or `lpm` class. If `indices` is `true`, the eigenvalues E_k $(k = 0, 1, ...)$ with indices k between `pmin` and `pmax` are calculated; if `indices` is `false` or omitted the eigenvalues in the range [`pmin`,`pmax`] are calculated. The method returns a structure `E`, in which all information related to the calculated eigenvalue(s) is stored. The fields `E.eigenvalues`, `E.indices` and `E.errors` are three vectors. `E.eigenvalues` contains the calculated eigenvalues in ascending order. The associated indices are collected in `E.indices` and `E.errors` holds an estimation of the error for each eigenvalue. The field `E.success` is false when the PPM was not able to obtain the data due to an error, e.g. when there is no eigenvalue in the interval [`pmin`,`pmax`]. In some cases a warning is generated in the method `get_eigenvalues`: e.g. when two eigenvalues are so close that double precision accuracy is not sufficient to differentiate between them adequately.

The estimation of the error in `E.errors` is the difference between the calculated eigenvalue (the so-called basic eigenvalue) and a reference eigenvalue computed on the same partition but with a higher order method. To make things clear, let us assume that the CPM$\{$16,14$\}$ is used. First the CPM$\{$16,14$\}$ algorithm is applied to construct the partition and to calculate the basic eigenvalue, and then the reference eigenvalue is computed using the higher order CPM$\{$18,16$\}$ (but on the CPM$\{$16,14$\}$ partition). The difference between the basic eigenvalue and the (more accurate) reference eigenvalue forms the estimation of the error in the basic eigenvalue. The Newton iteration process in the shooting procedure for the reference eigenvalue starts with the basic eigenvalue. Since the difference between the two eigenvalues is usually small, only a small number of Newton iterations is necessary and thus the calculation of the reference eigenvalue requires an extra effort which is almost negligible. Analogously, the error in a CPM$\{$12,10$\}$ (or CPM$\{$14,12$\}$) eigenvalue is estimated using the CPM$\{$14,12$\}$ (or CPM$\{$16,14$\}$ resp.). Table 7.1 shows the ratio of the true error to the estimated error (using the CPM$\{$16,14$\}$–CPM$\{$18,16$\}$ combination) for the harmonic oscillator

$$y'' = \left(x^2 - E\right) y, \quad x \in (-\infty, \infty), \tag{7.1}$$

for which the correct eigenvalues are known: $E_k = 2k + 1,\ k = 0, 1, ...$, and for the hydrogen atom equation

$$y'' = \left(-\frac{1}{x} + \frac{2}{x^2} - E\right) y, \quad x > 0 \tag{7.2}$$

with known eigenvalues $E_k = -1/(2k + 4)^2,\ k = 0, 1, ...$ This 'goodness' ratio always has values smaller or very close to one, which illustrates the adequacy of our error estimation. When the first eigenvalues of the hydrogen atom equation are calculated with tolerance $\leq 10^{-12}$, the obtained eigenvalues are (nearly) as accurate as the machine precision (10^{-16}). This means that the 'actual error' will be very close to zero (or even zero) which explains the smaller values in the last column of Table 7.1.

Table 7.1: Ratio $\frac{|\text{actual error}|}{|\text{error estimate}|}$ for the harmonic oscillator and hydrogen equation using CPM{16,14} at different input tolerances.

	Harmonic Oscillator			Hydrogen		
k	10^{-8}	10^{-10}	10^{-12}	10^{-8}	10^{-10}	10^{-12}
0	1.016	1.006	1.034	0.987	0.994	0.931
1	0.971	0.929	0.692	0.978	0.707	0.021
2	0.990	0.981	0.945	0.981	1.048	0.036
3	0.980	0.984	0.871	0.974	0.834	0.206
4	0.991	0.990	1.054	0.975	1.100	0.442
5	1.000	0.996	0.909	0.974	1.016	0.959
6	1.020	0.999	0.995	0.975	0.987	0.901
7	0.989	0.993	0.995	0.961	0.992	0.975
8	0.988	0.995	0.884	0.968	0.991	1.012
9	1.006	0.998	1.054	0.998	0.989	0.971
10	1.001	0.997	0.995	0.977	0.989	0.994

Since we have no higher order LPM available, we use a slighty different procedure to compute error estimates for the LPM[4,2] algorithm (something analogously is applied for CPM{18,16}). This procedure was also used by Ixaru to compute error estimates in the SLCPM12 Fortran package [61]. A second partition is constructed (called the reference partition by Ixaru) by halving each interval of the ‘basic’ partition. The reference eigenvalue is then computed on the reference partition and the error estimate is again formed by the difference between the basic and reference eigenvalue.

For a problem with a infinite integration interval, an extra output argument can be returned:

```
[E,pp_child] = get_eigenvalues(pp_child,pmin,pmax)
[E,pp_child] = get_eigenvalues(pp_child,pmin,pmax,indices)
```

The returned object `pp_child` then contains information on the constructed partition. This partition was constructed on a truncated integration interval which is large enough for the largest requested eigenvalue (see section 6.3.1). All eigenvalue calculations were done on (parts of) this partition.

Table 7.2 displays the times needed by CPM{16,14} (on a 2.4GHz PC) to calculate some eigenvalues of the Mathieu equation

$$y'' = (2\cos(2x) - E)\,y, \quad 0 < x < \pi, \tag{7.3}$$

and of the hydrogen atom equation (7.2). The Mathieu problem is a regular problem with a finite integration interval. This means that the shooting procedure for each eigenvalue is performed on one and the same partition which was already constructed in the first stage. This explains why the time increases only very slowly with the eigenvalue index. For the hydrogen atom equation however, the (truncated) integration interval grows with the

Table 7.2: Time (s) to compute some eigenvalues with CPM{16,14} at different input tolerances.

	Mathieu			Hydrogen		
k	10^{-8}	10^{-10}	10^{-12}	10^{-8}	10^{-10}	10^{-12}
0	0.06	0.09	0.11	0.57	0.60	0.63
10	0.09	0.09	0.13	0.86	0.89	0.92
100	0.09	0.11	0.13	1.37	1.51	1.75
1000	0.11	0.13	0.14	3.44	3.94	4.78
10000	0.13	0.16	0.17	13.8	16.7	21.3

eigenvalue index and as a consequence the calculation of a higher eigenvalue requires a larger amount of time. The truncation algorithm is also partly the reason why the computation of E_0 for the hydrogen atom equation needs more time than for the Mathieu equation. Another reason is the larger amount of intervals needed in the partition constructed for the hydrogen atom equation.

7.2.3 Stage 3: Eigenfunction computation

Another method in the ppm class which is visible to the user, is the method which allows the calculation of the eigenfunction associated with a certain eigenvalue e:

```
V = get_eigenfunction(pp_child,e)
V = get_eigenfunction(pp_child,e,n)
V = get_eigenfunction(pp_child,e,ap,bp,n)
```

where again pp_child is an instance of one of the child classes of cpm or lpm. This pp_child is the returned object from one of the cpm or lpm constructors or from the get_eigenvalues method when the problem is infinite. To obtain a good approximation for the eigenfunction the eigenvalue e must be sufficiently accurate (e.g. by choosing a small value for tol in stage 1). If n is omitted, then the eigenfunction is evaluated only in the meshpoints of the partition. In most cases the number of meshpoints in the partition is too small to have a good idea of the shape of the eigenfunction. Therefore the eigenfunction can be evaluated in more points by choosing a sufficiently high value for n: the interval [ap,bp] is then taken and partitioned in n intervals of equal size. On these intervals the additional potential-dependent expressions are calculated and the propagation matrix algorithm (2.80) is applied to produce the eigenfunction. Only the part of the eigenfunction corresponding with the n+1 points in [ap,bp] is returned. When the input-arguments ap and bp are omitted, then the whole interval $[a, b]$ is considered. The structure V has three fields: the three vectors V.x, V.y and V.yprime of which the meaning is clear.

7.2.4 The Coffey-Evans example

As an illustration we compute the eigenvalues of an example from [105], the Coffey-Evans equation

$$-y'' + (-2\beta \cos 2x + \beta^2 \sin^2 2x)y = Ey \tag{7.4}$$

with

$$y(-\pi/2) = y(\pi/2) = 0. \tag{7.5}$$

For our first test run, we assume that the parameter β is equal to 20. The commands used to solve this problem can also be found in `example8.m` in the `examples` directory. The example can be runned by entering `example8` in the MATLAB command window. Here we discuss each of these commands. First we specify our Coffey-Evans problem:

```
r=schrod('-2*20*cos(2*x)+20^2*sin(2*x)^2',...
        -pi/2,pi/2,1,0,1,0);
```

The `schrod` constructor is used, the first argument is a string representing the potential function, the second and third argument are the two endpoints of the integration interval and the last arguments specify the boundary conditions.

We want to solve the problem with the CPM{16,14} method. We initialize this method by the following command

```
cp=cpm16_14(r,1e-10);
```

We have chosen the input tolerance to be 10^{-10}. The object `cp` now contains information on the partition. Using the command

```
plot_partition(cp);
```

a plot of this partition and the meshpoints is made (Figure 7.1).

We ask for the first 11 eigenvalues

```
E=get_eigenvalues(cp,0,10,true);
```

The computed eigenvalues and the error estimates can then be accessed through the fields `E.eigenvalues` and `E.errors`. The lines

```
disp(sprintf('k \t  E_k \t\t\t\t  estimated error'))
for i=1:length(E.eigenvalues)
    disp([num2str(E.indices(i),'%4.0f\t') '   ' ...
        num2str(E.eigenvalues(i),'%16.12f\t') '   '...
        num2str(E.errors(i),'%+5.2e')])
end
```

produce the following output

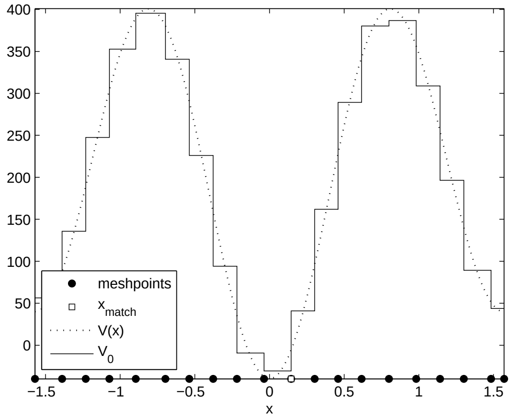


Figure 7.1: Plot of the partition generated by MATSLISE for the Coffey-Evans potential with $\beta = 20$.

```
k   E_k                  estimated error
0   -0.000000002121     -2.05e-009
1   77.916195679902     +2.78e-009
2   151.462778348882    +2.40e-009
3   151.463223659596    +1.98e-009
4   151.463668990776    +2.40e-009
5   220.154229836780    +1.44e-009
6   283.094814694590    -8.70e-010
7   283.250743741632    -1.55e-009
8   283.408735402515    -9.90e-010
9   339.370665648747    -3.76e-009
10  380.094915551000    -1.04e-009
```

The eigenfunction associated to the fifth eigenvalue (i.e. $y_4(x)$), evaluated in 100 points is given by

```
V=get_eigenfunction(cp,E.eigenvalues(5),100);
```

A simple plot command

```
plot(V.x,V.y);
xlabel('x')
ylabel('y')
```

produces Figure 7.2.

We can then compute some other eigenvalues, e.g. the eigenvalues between 1000 and 1500:

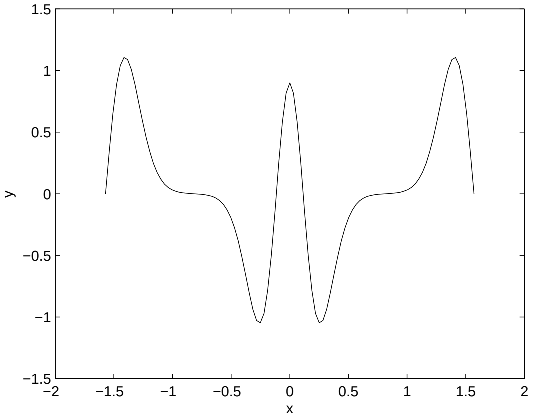


Figure 7.2: Eigenfunction y_4 for the Coffey-Evans potential for $\beta = 30$.

```
E=get_eigenvalues(cp,1000,1500,false);
```

which gives as result

```
k     E_k                     estimated error
28    1047.204086283367    -2.83e-010
29    1105.794050195401    +1.28e-010
30    1166.423692498202    -6.14e-010
31    1229.087995655108    +2.09e-009
32    1293.782722437993    +4.12e-010
33    1360.504272201038    +1.78e-009
34    1429.249567674530    -6.13e-010
```

Note that these last eigenvalue computations still use the same partition, i.e. still the same `cp`-object is passed to the `get_eigenvalues` method.

Even though the theory guarantees that for the separated boundary conditions (7.5), there can be no multiple eigenvalues, the triple well of the Coffey-Evans potential produces triples of eigenvalues which can be made arbitrarily close by deepening the well. The potential looks like Figure 7.3. The parameter β, typically in the range 0 to 50, controls the depth of the well. When we thus take $\beta = 50$, the eigenvalues in the triplets will be even closer than for $\beta = 20$. The clustering of the eigenvalues causes difficulty for all library codes (see [105]) and computation is often expensive. The MATSLISE commands are

```
r=schrod('-2*50*cos(2*x)+50ˆ2*sin(2*x)ˆ2',-pi/2,pi/2,1,0,1,0);
cp=cpm16_14(r,1e-14);
E=get_eigenvalues(cp,0,10,true);
```

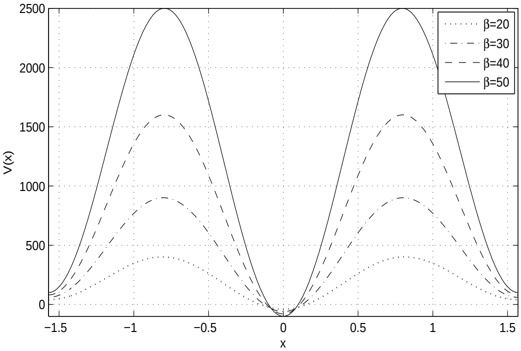


Figure 7.3: Coffey-Evans potential for different values of β.

k	Estimate of E	Time(s)
0	-0.000000000002	0.5
1	197.968726516499	0.3
2	391.808191489040	1.0
3	391.808191489045	1.6
4	391.808191489061	2.0
5	581.377109231564	0.5
6	766.516827285497	1.5
7	766.516827285506	1.1
8	766.516827285516	1.6
9	947.047491585820	0.3
10	1122.762920067867	0.7

Table 7.3: Eigenvalues of the Coffey-Evans Equation ($\beta = 50$), computed in MATSLISE.

When the tight tolerance of 10^{-14} is requested, the CPM$\{$16,14$\}$ code in MATSLISE delivers the output in Table 7.3 for the first 11 eigenvalues. While the code had to work harder on some of the triplets, it was able to return what appear to be reasonable answers.

Numerically, clustering of the eigenvalues causes the eigenfunctions to be very ill-conditioned (the so-called ‘flea on the elephant’ effect, see [105]). This makes it very difficult for a Sturm-Liouville code to compute eigenfunctions of the Coffey-Evans equation with a larger β value. This difficulty can be (partly) avoided by using half-range reduction (see further).

7.3 The graphical user interface

The methods mentioned in the previous section, can all be called from the MATLAB command line. But in order to increase the ease of use and to hide the technical issues from the user, a graphical user interface (GUI) has been built on top of the classes in MATSLISE. It slows the computations down somewhat; but on the other hand, it facilitates giving input, it gives the user more control and graphical features are built in.

The GUI-version of MATSLISE uses the CPM{16,14} method to calculate the (basic) eigenvalue and the higher order CPM{18,16} is used on the same CPM{16,14} partition to obtain a reference eigenvalue. As seen in section 7.2.2, this reference eigenvalue enables an accurate error estimation. The (slower) CPM{12,10}, CPM{14,12} and LPM[4,2] are not used in the GUI but they can still be invoked from the command line using the constructors `cpm12_10`, `cpm14_12` and `lpm10`.

The root directory of MATSLISE contains two files: `matslise.m` and `matslise_help.m`. By entering `matslise` at the command line, the GUI is opened. `matslise_help` opens the corresponding Help-files.

7.3.1 The problem specification window

In Figure 7.4 the input window is shown for the Coffey-Evans equation. Similar windows can be opened for Sturm-Liouville equations or for problems with a distorted Coulomb potential. The Coffey-Evans problem is one of the predefined problems which is included within MATSLISE in the directory `predefined_problems`. To see a list of the other problems included in this directory, type `showPredefinedProblems` at the command line. The list of predefined problems is also shown in Appendix C. For more details on the different inputfields and buttons of the problem specification windows we refer to the MATSLISE help files.

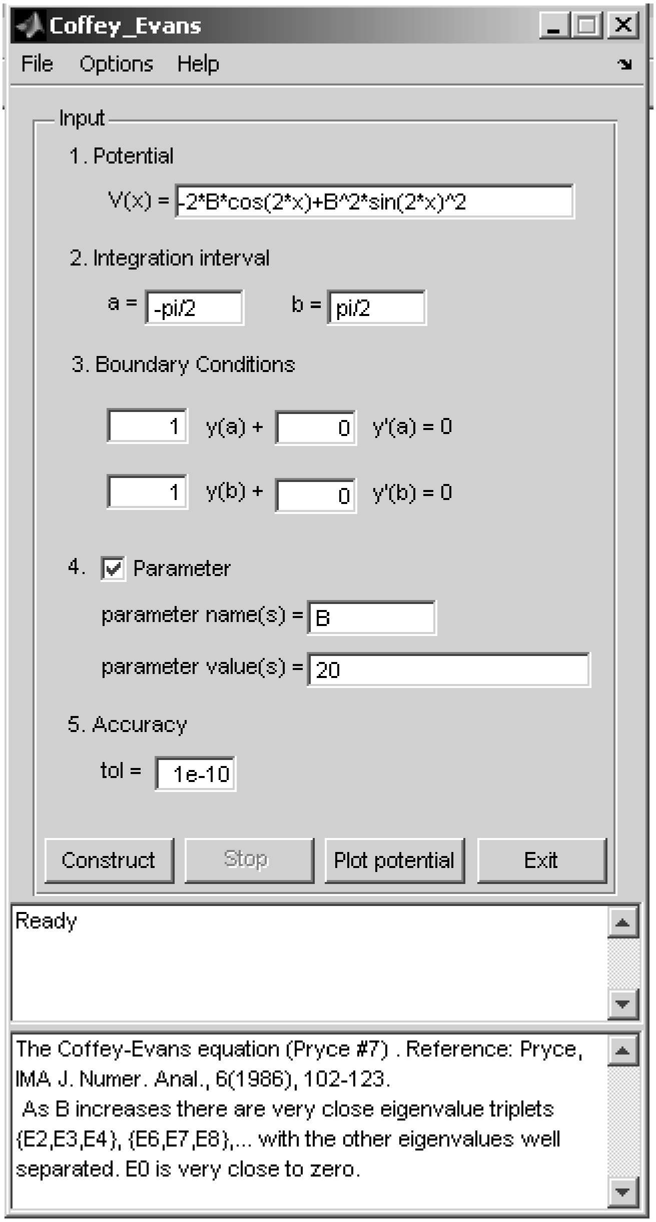
After the input has been entered, the "Construct"-button starts the calculations of stage 1. That is the constructors of the classes `schrod` and `cpm16_14` are called. A second window is opened where the user is able to obtain the eigenvalues of the problem he/she specified: the eigenvalues window.

7.3.2 The eigenvalues window

Figure 7.5 shows the eigenvalues window for the Coffey-Evans test problem. In this window the user specifies which eigenvalues he/she wants to calculate. Several batches of eigenvalues can be calculated one after the other without revisiting the problem specification window.

7.3.3 Computation and visualization of the eigenfunctions

The eigenfunctions associated to the selected eigenvalues are calculated by pressing the "Eigenfunction"-button in the eigenvalues window, which opens a new window. When only one eigenvalue is selected the eigenfunction window is opened; when more than



1. Potential

V(x) = -2*B*cos(2*x)+B^2*sin(2*x)^2

2. Integration interval

a = -pi/2   b = pi/2

3. Boundary Conditions

1 y(a) + 0 y'(a) = 0

1 y(b) + 0 y'(b) = 0

4. ☑ Parameter

parameter name(s) = B

parameter value(s) = 20

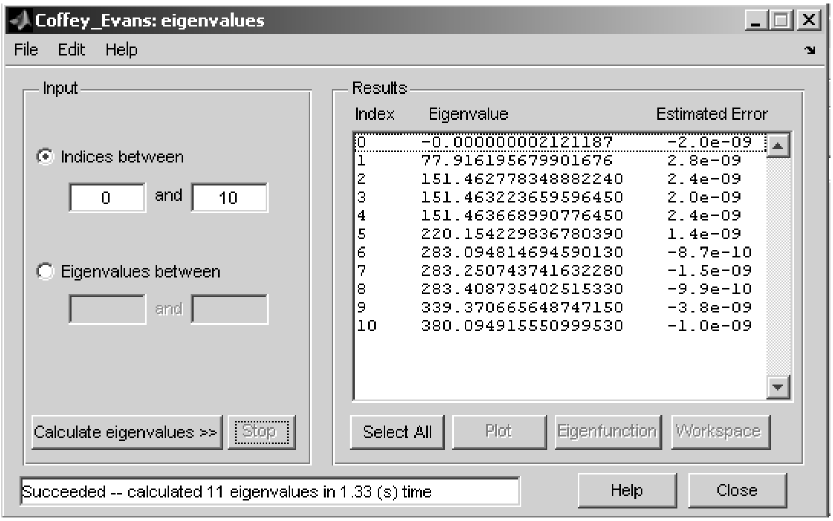
5. Accuracy

tol = 1e-10

Ready

The Coffey-Evans equation (Pryce #7) . Reference: Pryce, IMA J. Numer. Anal., 6(1986), 102-123.
As B increases there are very close eigenvalue triplets {E2,E3,E4}, {E6,E7,E8},... with the other eigenvalues well separated. E0 is very close to zero.

Figure 7.4: Example of the Schrödinger problem specification window of the MATSLISE GUI



```
Index   Eigenvalue                  Estimated Error
0       -0.000000002121187          -2.0e-09
1       77.916195679901676          2.8e-09
2       151.462778348882240         2.4e-09
3       151.463223659596450         2.0e-09
4       151.463668990776450         2.4e-09
5       220.154229836780390         1.4e-09
6       283.094814694590130         -8.7e-10
7       283.250743741632280         -1.5e-09
8       283.408735402515330         -9.9e-10
9       339.370665648747150         -3.8e-09
10      380.094915550999530         -1.0e-09
```

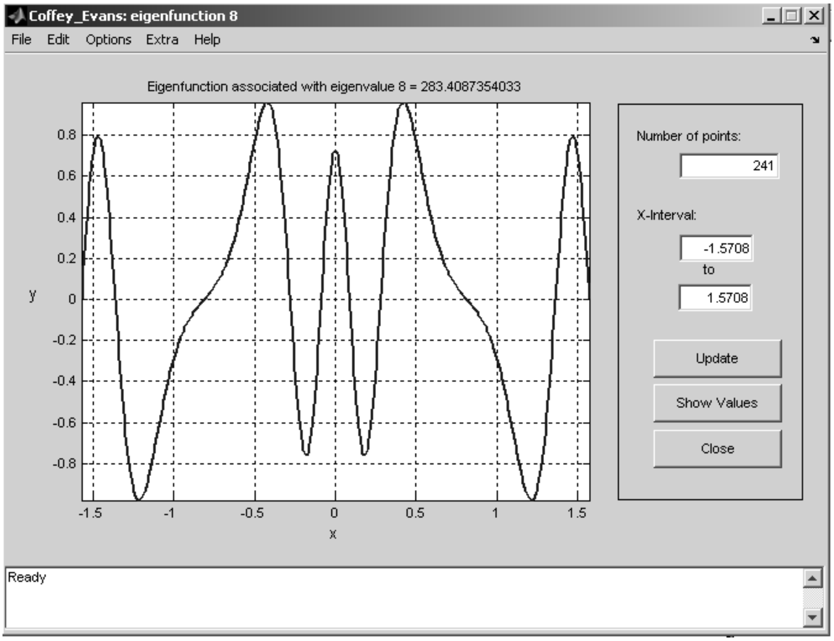
Succeeded -- calculated 11 eigenvalues in 1.33 (s) time

Figure 7.5: Example of the eigenvalues window of the MATSLISE GUI.

one eigenvalue is selected another window is opened: the eigenfunctions window. Figure 7.6 shows an example of the eigenfunction window, while Figure 7.7 shows the eigenfunctions window. For more information on the different options and features avaible in these windows we again refer to the MATSLISE help files. With the "Show with potential"-button in the eigenfunctions window e.g. it possible to plot the eigenfunctions together with the potential function and generate plots like Figures 6.1 and 6.3 discussed in chapter 6. Another interesting button is the "Test orthonormality"-button, which allows to test the correctness of the eigenfunctions via an orthogonality-check. This check applies the trapezoidal rule on each interval between two points where the eigenfunction was evaluated, in order to compute a crude approximation of $\int y_k(x)y_l(x)dx$ (or $\int y_k(x)y_l(x)w(x)dx$ for a Sturm-Liouville problem). When two eigenfunctions seem to be not orthogonal, the eigenfunctions cannot be correct. This inadequacy can be caused by two reasons:

- The eigenvalue is not accurate enough to compute the associated eigenfunction correctly. Decreasing the input tolerance might help in this case.

- Very close eigenvalues occur and as mentioned in [105] clustering causes the eigenfunctions to be very ill-conditioned. Therefore the user will be warned and asked to be cautious when close eigenvalues are detected. For symmetric double well problems, half-range reduction (see further) may make the problem more tractable: check the option "Half-range reduction" in the Options menu of the problem specification window.



Coffey_Evans: eigenfunction 8

Eigenfunction associated with eigenvalue 8 = 283.4087354033

Number of points: 241

X-Interval: -1.5708 to 1.5708

Ready

Figure 7.6: Example of the eigenfunction window of the MATSLISE GUI.

7.3.4 Half-range reduction

A Schrödinger problem (or Sturm-Liouville problem) is symmetric when the problem is defined on the interval $-b$ to b, where b may be $+\infty$, the potential function is even and the boundary conditions are symmetric, this means in the regular case that $a_0 = a_1$ and $b_0 = -b_1$. For a symmetric problem the eigenfunctions belonging to eigenvalue E_k, $(k = 0, 1, ...)$ are even or odd functions according as k is even or odd. The eigenvalues can be obtained by solving the given equation, but on the interval $[0, b]$, with the given boundary condition at b and with (i) $y'(0) = 0$ to get the even eigenvalues, (ii) $y(0) = 0$ to get the odd eigenvalues. The eigenvalues $E_0, E_2, E_4, \ldots$ of the full-range problem are then the eigenvalues of the half-range problem with boundary condition (i) in $x = 0$, while $E_1, E_3, E_5, \ldots$ are the eigenvalues of the half-range problem with the boundary condition (ii). The normalized eigenfunctions of the full-range problem are reconstructed from those of the half-range problem by extending in the appropriate way and dividing by $\sqrt{2}$. For symmetric double well problems, this reduction may make the difference between a highly ill-conditioned problem and a perfectly straightforward one.

An example of a symmetric double well problem with close eigenvalues is the Coffey-Evans equation with a large β value. Another example is the close-eigenvalues problem

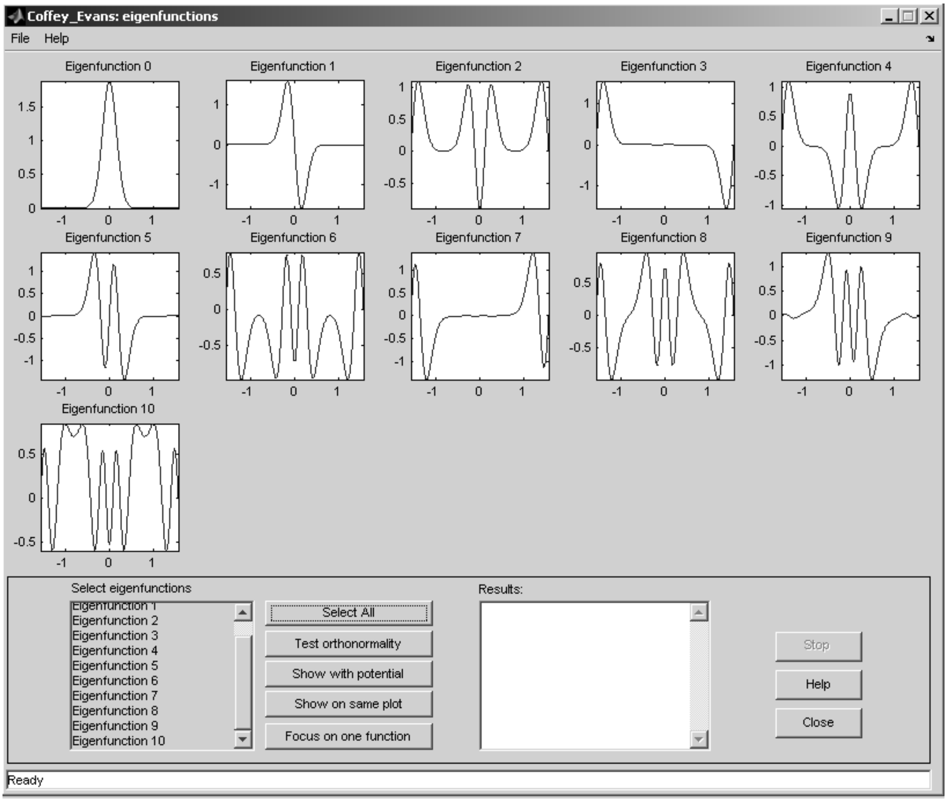
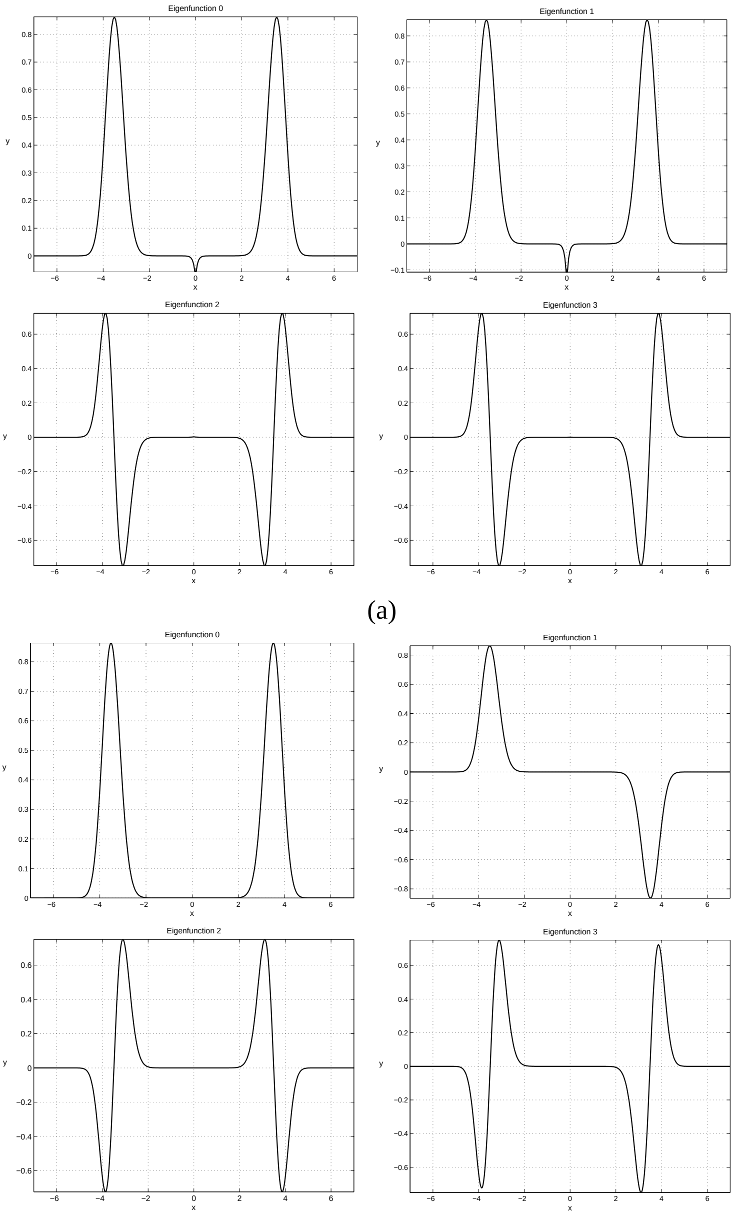


Figure 7.7: Example of the eigenfunctions window of the MATSLISE GUI.

which is also included in the test set of SLTSTPAK ([108]).

$$V(x) = x^4 - 25x^2, \;\; x \in (-\infty, \infty), \; y(-\infty) = y(\infty) = 0 \tag{7.6}$$

The lowest eigenvalues of this problem occur in very close pairs. The clustering of eigenvalues causes difficulty in the computation of the eigenfunctions for all Sturm-Liouville codes. Figure 7.8(a) shows the first four eigenfunctions of the close eigenvalues problem, calculated in the GUI-version of MATSLISE with input tolerance $tol = 10^{-12}$ but without half-range reduction. In this case MATSLISE returns a message which warns the user that the eigenfunctions are very ill-conditioned. And indeed, it is easy to see that the eigenfunctions are not correct: the number of roots of the eigenfunction should correspond with the eigenvalue index. Also the ortogonality-check included in MATSLISE (started by clicking the “Test orthonormality”-button) indicates that the computed eigenfunctions are not correct. Figure 7.8(b) shows the same eigenfunctions but now calculated with half-range reduction. It is clear that these last eigenfunctions are the correct ones.



(a)

(b)

Figure 7.8: The eigenfunctions of the close-eigenvalues problem (7.6), ***(a)*** calculated on the full-range, ***(b)*** on the half-range.

7.3.5 Using parameters

It is possible to use a parameter (or parameters) in the specification of the problem. These parameters can occur in the definition of the coefficients, and/or the range, and/or the boundary conditions. The parameter-name(s) and -value(s) are defined by checking the ‘Parameter’-box and filling in the two corresponding fields in the problem specification window (Figure 7.4). The parameter can then be used in the other inputfields. This is used to facilitate the input process or to replace rather lengthy subexpressions in the potential function by a parameter, but also to study the behaviour of the eigenvalue(s) or solution when the parameter changes. The directory `predefined_problems` contains some examples of the use of parameters in the problem specification. One such an example is the problem in `parameter_example3.mat`. It is the Coffey-Evans equation

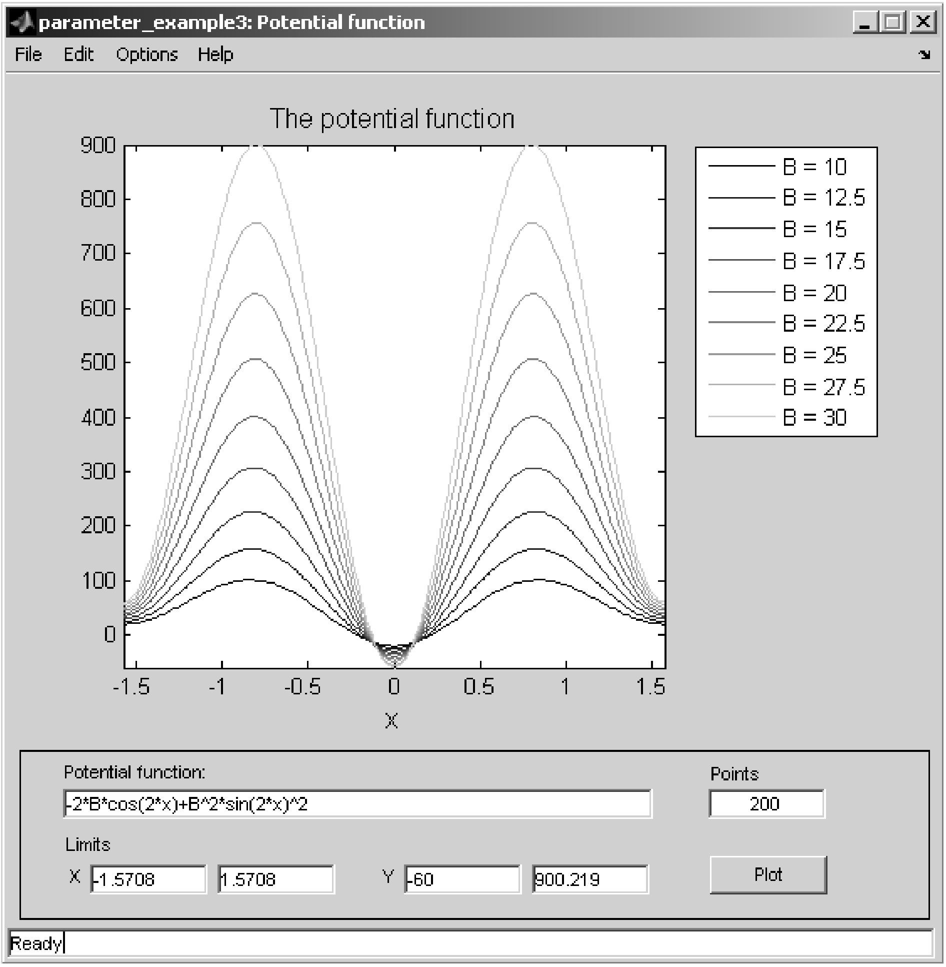
$$V(x) = -2\beta\cos(2x) + \beta^2\sin(2x)^2, \quad y(-\pi/2) = y(\pi/2) = 0, \tag{7.7}$$

with the parameter β running through `10:2.5:30`, that is the values in $[10, 12.5, \ldots, 30]$. The potential $V(x)$ changes with the parameter values as in Figure 7.9. Note that the problem is symmetric and half-range reduction can (should) be applied.

We take $\mathrm{tol} = 10^{-12}$ and solve the problem using the GUI. MATSLISE automatically generates a `cpm16_14`-object for each parameter value and all these `cpm16_14`-objects are used when the eigenvalues are calculated. The eigenvalues of the different problems are easily compared by plotting them: Figure 7.10 contains the plot of the first 21 eigenvalues of Eq. (7.7). The lower eigenvalues are clustered in groups of three with an isolated eigenvalue between clusters. Increasing β makes more clusters appear and makes each one tighter. After calculating the eigenvalues, the eigenfunction corresponding to a certain eigenvalue index can be computed for each parameter value. In Figure 7.11 the result is shown for eigenfunction y_2 of Eq. (7.7).

7.4 Conclusion

In this chapter we discussed the MATSLISE package. MATSLISE is a graphical MATLAB software package for the interactive numerical study of one-dimensional (regular) Sturm-Liouville problems and Schrödinger equations and radial Schrödinger equations with a distorted Coulomb potential. It allows the accurate computation of the eigenvalues and the visualization of the corresponding eigenfunctions. This is realized by making use of the power of the high order PPM algorithms discussed in chapter 3 and 4. We looked at the different MATSLISE functions and demonstrated the use of the graphical user interface, which was built on top of the package in order to increase the accessibility.



Potential function: -2*B*cos(2*x)+B^2*sin(2*x)^2

Points 200

Limits

X -1.5708 1.5708

Y -60 900.219

Ready

Figure 7.9: Example of the potential window of the MATSLISE GUI.

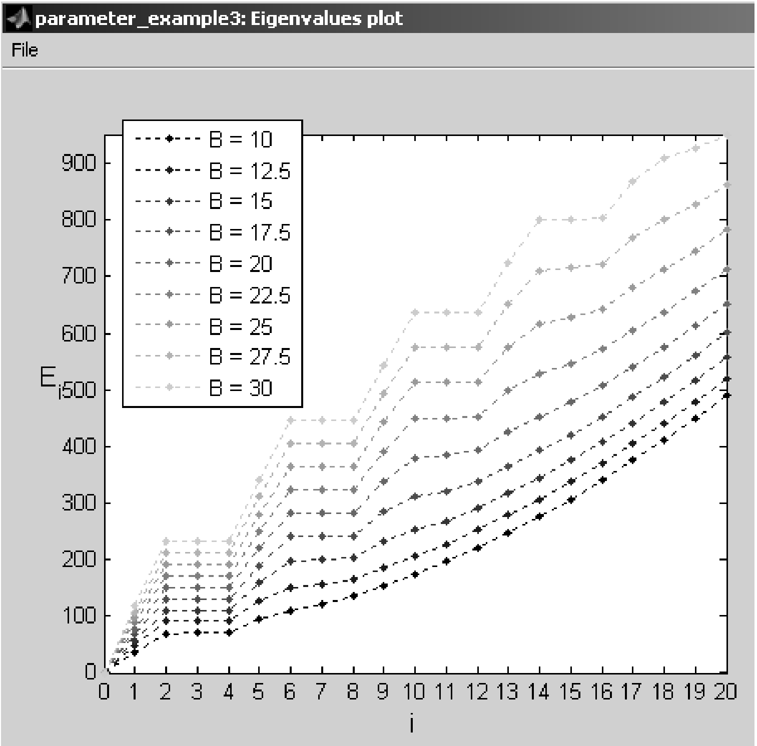
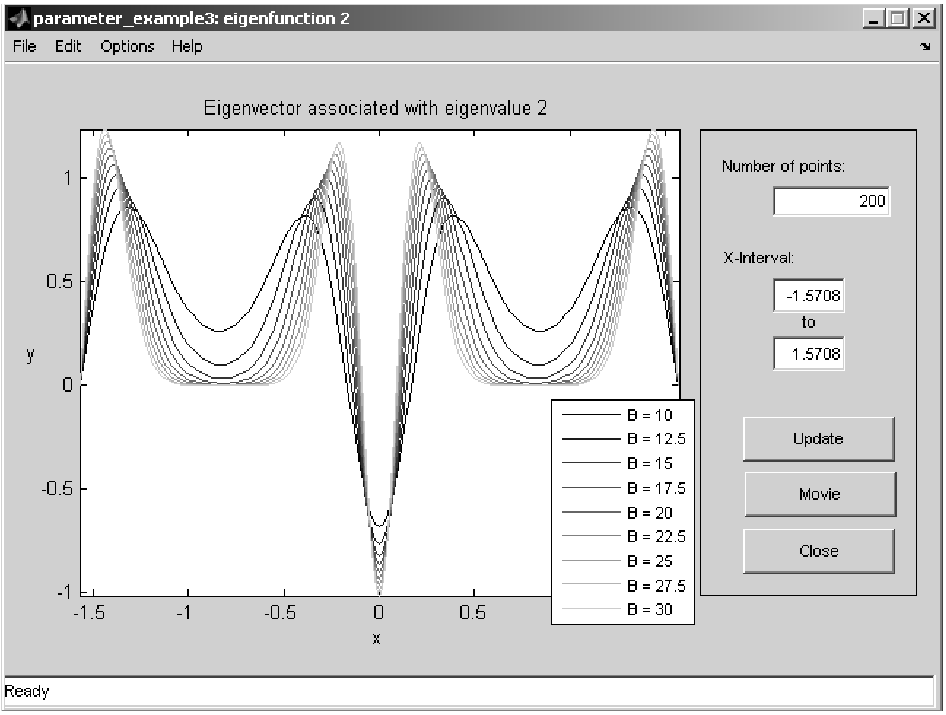


Figure 7.10: Example of a plot generated by MATSLISE: the eigenvalues of a problem defined with parameters.



Number of points: 200

X-Interval: -1.5708 to 1.5708

Figure 7.11: Example of the MATSLISE eigenfunction window for a problem with parameters.

Chapter 8

Conclusions

In this chapter, the conclusions from this dissertation are summarized, and an overview of the main contributions is given.

8.1 Summary

In this thesis, we mainly concentrated on a specific class of methods specially devised for the numerical solution of a Sturm-Liouville problem or a Schrödinger problem. This class of methods is formed by the family of Piecewise Perturbation Methods (PPM). The main idea behind these methods is the perturbation approximation. This means that the orginal differential equation is replaced piecewisely by another differential equation (the so-called reference differential equation) which can be solved exactly. The perturbation theory is used to construct some correction terms which are added to the (known) solution of the reference differential equation in order to approximate the (unknown) solution of the orginal equation. The accuracy in the solution increases with the number of correction terms included.

In this thesis we considered two subclasses of the PPM: the Constant Perturbation Methods (CPM) and the Line Perturbation Methods (LPM). Both CPM and LPM are constructed for a Sturm-Liouville problem in Schrödinger form. By applying Liouville's transformation any (regular) Sturm-Liouville equation can be transformed to such a Schrödinger form. A CPM approximates the potential function of the Schrödinger problem by a piecewise constant, while for the LPM piecewise lines are used.

Both CPM and LPM are well suited to be used as the integration method in a shooting procedure to solve the boundary value problem. We applied a procedure based on the Prüfer transformation to estimate the index associated to an eigenvalue. This allows us to compute a specific eigenvalue without consideration of other eigenvalues. For the LPM, we constructed some alternative asymptotic formulae for the perturbation corrections which are less affected by the accuracy loss due to the near-cancellation of like-terms.

Also for systems of coupled Schrödinger equations a CPM algorithm can be formulated. As for the one-dimensional problem, the generalized CPM is used in a shooting procedure to compute the eigenvalues of the associated boundary value problem. The shooting algorithm has been supplemented by Atkinson's matrix generalization of the Prüfer transformation.

We also devoted a chapter to the solution of some singular problems. We mainly concentrated on a truncation algorithm to cut off the integration interval of problems with infinite endpoints. Also an algorithm was described which deals with the singularity in the origin of some radial Schrödinger equations.

The different PPM algorithms were implemented in the MATLAB environment. The one-dimensional algorithms e.g. are collected in the MATSLISE software package. MATSLISE allows the computation and visualization of the eigenvalues and eigenfunctions of a Sturm-Liouville or Schrödinger problem.

8.2 Contributions

The main novelties and contributions presented in this thesis are the following. In chapter 3, we presented a Maple program which automates the construction of the perturbation corrections for the so-called CPM$\{P, N\}$ methods. This made it possible for us to formulate the higher order CPM$\{14{,}12\}$, CPM$\{16{,}14\}$ and CPM$\{18{,}16\}$. In chapter 4 we collected some new results, in particular the obtained formulae for the first and second order correction of a LPM algorithm of order 10 are new. Also some asymptotic formulae for these corrections are presented. The use of these asymptotic expressions avoids loss in accuracy due to near-cancellations of like-terms during computation. In chapter 5 we discussed our extension of the CPM algorithm to systems of coupled Schrödinger equations. We developed a shooting procedure which uses this generalized CPM to compute the eigenvalues of the boundary value problem. We improved the calculation of the eigenvalues by adapting an algorithm by Marletta to the CPM case. In chapter 6, our contribution exists in the development of the truncation algorithms for problems defined on an infinite integration interval. We implemented the different CPM and LPM algorithms in MATLAB. The MATSLISE package, discussed in chapter 7, collects the one-dimensional algorithms and allows the computation of the eigenvalues and eigenfunctions of a problem specified by the user.

Chapter 9

Nederlandse samenvatting

Een groot aantal belangrijke fysische processen, zowel uit de klassieke fysica als uit de kwantumfysica, worden beschreven aan de hand van een Sturm-Liouville vergelijking. Een dergelijke Sturm-Liouville vergelijking is een lineaire tweede orde differentiaalvergelijking met als algemene vorm

$$-\frac{d}{dx}\left[p(x)\frac{dy(x)}{dx}\right]+q(x)y(x)=Ew(x)y(x),\quad \text{met } x \text{ in } [a,b]. \tag{9.1}$$

Wanneer randvoorwaarden worden opgelegd in de eindpunten a en b van het integratie-interval $[a,b]$, bestaat een niet-triviale oplossing $y(x)$ enkel voor bepaalde waarden van de parameter E. Deze E-waarden en de bijhorende oplossingen $y(x)$ worden de *eigenwaarden*, respectievelijk *eigenfuncties* van het Sturm-Liouville probleem genoemd.

Een specifieke klasse van Sturm-Liouville problemen wordt gevormd door de Schrödinger problemen. De Schrödinger vergelijking is de fundamentele vergelijking in de kwantummechanica en kan in de volgende vorm geschreven worden:

$$y''(x)=[V(x)-E]y(x)\quad \text{met } x \text{ in } [a,b]. \tag{9.2}$$

Het oplossen van een Sturm-Liouville of Schrödinger probleem bestaat erin de koppels (E_k,y_k) te bepalen, waarbij E_k de k-de eigenwaarde is en y_k de eigenfunctie die erbij hoort. Het natuurlijk getal k ($k=0,1,2,...$) wordt de *index* van de eigenwaarde E_k genoemd. Een Sturm-Liouville probleem is over het algemeen niet analytisch oplosbaar en computationeel efficiënte benaderingsmethoden zijn dan ook van groot belang.

In de literatuur werden reeds verschillende numerieke methoden beschreven voor de berekening van de eigenwaarden en eigenfuncties van Sturm-Liouville randwaardeproblemen. Deze technieken omvatten onder andere de eindige differentiemethoden (waar de optredende afgeleiden worden gediscretiseerd), variationele methoden en ‘shooting’ methoden. Maar de nauwkeurigheid van zowel de variationele als eindige differentiemethoden neemt sterk af bij stijgende index van de eigenwaarde. Dit is een gevolg van het toenemende oscillatorisch gedrag van de eigenfunctie bij een stijgende eigenwaarde-index. Hetzelfde geldt ook voor veel van de shooting methoden wanneer ze rechtstreeks op het Sturm-Liouville probleem worden toegepast. Een ander minpunt van veel methoden is dat eerst alle voorafgaande eigenwaarden moeten berekend worden vooraleer met de berekening van een specifieke eigenwaarde E_k kan gestart worden.

Voor het Sturm-Liouville probleem kunnen de moeilijkheden opgelost worden door gebruik te maken van de Prüfer transformatie. Deze Prüfer transformatie laat toe om een shooting methode op te stellen waar een specifieke eigenwaarde met een bepaalde index wordt berekend zonder dat er kennis nodig is van de voorafgaande (lagere) eigenwaarden. Ook maakt de Prüfer gebaseerde shooting methode het mogelijk om een redelijke nauwkeurigheid te bekomen voor eigenwaarden met een hogere index.

Wanneer de Prüfer gebaseerde shooting methoden echter gebruik maken van standaard codes voor beginwaardeproblemen (bvb. een Runge-Kutta code), kunnen problemen met stijfheid ontstaan, waardoor zeer kleine stapgroottes genomen worden. Deze problemen verdwijnen wanneer de shooting procedure wordt gecombineerd met coëfficiënt-approximatie. Hierbij worden de coëfficiëntfuncties $p(x)$, $q(x)$ en $w(x)$ stuksgewijs vervangen door polynomen van een lage graad zodat de resulterende vergelijking analytisch kan opgelost worden. Pruess en Fulton gebruikten een stuksgewijs constante benadering van de coëfficiëntfuncties in hun gekende SLEDGE code. Ixaru beschreef de basisprincipes voor een andere klasse van methoden gebaseerd op coëfficiëntapproximatie, de zogenaamde PPM (Eng.: Piecewise Perturbation Methods). Aan deze PPM wordt een perturbatietechniek toegevoegd om de oplossing van het benaderende probleem bijkomend te corrigeren. Hoe hoger het aantal perturbaties, hoe nauwkeuriger de bekomen oplossing. Dit laat toe om methoden met hogere ordes te definiëren.

Het is vooral de klasse van PPM die het onderwerp van deze thesis vormt. In hoofdstuk 3 concentreren we ons op een belangrijke subklasse van de PPM, namelijk de CPM (Constant Perturbation Methods) die speciaal werd ontworpen voor de Schrödinger vergelijking. Deze CPM steunen op de benadering van de optredende potentiaalfunctie $V(x)$ door een stuksgewijs *constante* referentiefunctie om een oplossing van orde nul te vormen. Deze CPM hebben het belangrijke voordeel dat de toename van de fout bij stijgende index k verdwijnt, dit in contrast met de traditionele methoden waar de fout stijgt met de energie E. Als een direct gevolg, kan een energie-onafhankelijke verdeling van het integratie-interval geconstrueerd worden met ongewoon grote stapgroottes waarover de berekeningen snel kunnen gebeuren. Het construeren van een dergelijk CPM-algoritme werd echter in het verleden erg moeilijk bevonden doordat een groot aantal analytische berekeningen nodig zijn voor het opstellen van de correctietermen. Door het symbolisch softwarepakket Maple te gebruiken, slaagden we erin om deze barrière grotendeels te overbruggen. Verschillende hogere orde CPM-versies werden ontwikkeld en toegepast op het Sturm-Liouville en Schrödinger probleem. Meer concreet behandelen we de extensie van de zogenaamde CPM{12,10} methode, geïntroduceerd door Ixaru et. al. en gebruikt in de Fortran SLCPM12 code, naar de hogere orde methodes CPM{14,12}, CPM{16,14} en CPM{18,16}. Een aantal numerieke tests toont aan dat de hogere orde methoden nog efficiënter zijn dan het CPM{12,10} algoritme: het aantal subintervallen in de partitie kan sterk gereduceerd worden waardoor de CPU-tijd, die nodig is om een aantal eigenwaarden binnen een vooraf opgelegde nauwkeurigheid te bepalen, kleiner is.

De CPM representeren slechts één mogelijke manier om een PPM te implementeren. Een andere mogelijkheid bestaat erin om methoden te construeren die gebaseerd zijn op een stuksgewijs *lineaire* coëfficiëntapproximatie in plaats van een stuksgewijs constante coëfficiëntapproximatie. Deze methoden duiden we aan als de LPM (Line Perturbation Methods). We verfijnden het LPM algoritme en bespreken in hoofdstuk 4 de constructie van de eerste en tweede orde correctieterm. Deze correcties laten ons toe om een methode van orde 10 te definiëren. Numerieke testen bevestigen dat ook deze LPM-versie voldoet aan de belangrijke eigenschap van de PPM-methoden: de eigenwaarden kunnen benaderd worden met een uniforme nauwkeurigheid over één en dezelfde partitie van het integratie-interval. Het spreekt vanzelf dat lijnsegmenten betere stuksgewijze benaderingen geven van een functie dan constantes. Toch tonen we aan dat in de praktijk de LPM-versie vaak minder geschikt is dan een CPM van dezelfde orde. De reden is dat de propagatie van de oplossing over een subinterval meer tijd vraagt voor de LPM. Dit komt doordat de exacte oplossing van het benaderende referentieprobleem wordt uitgedrukt in Airy functies. De evaluatie van deze Airy functies is complexer dan de evaluatie van de goniometrische en hyperbolische functies die optreden in het algoritme van de CPM. Het is dan ook aangewezen een LPM enkel te gebruiken wanneer de potentiaal sterk varieert. In alle andere gevallen hoort een CPM de voorkeur te krijgen.

In hoofdstuk 5 bestuderen we de ontwikkeling van CPM voor stelsels van meerdere Schrödinger vergelijkingen. De behoefte aan goede numerieke methodes voor het oplossen van stelsels van gekoppelde Schrödinger vergelijkingen duikt op in vele kwantummechanische berekeningen, zowel uit de nucleaire als moleculaire fysica. We construeerden een aantal CPM-gebaseerde methodes van hoge orde door een zeer analoge procedure te gebruiken als bij het één-dimensionale geval. Opnieuw werden de berekeningen voor het opstellen van de correctietermen uitgevoerd in het symbolisch softwarepakket Maple. De geconstrueerde algoritmes zijn erg stabiel en vertonen (net zoals de CPM voor een één-dimensionaal probleem) een uniforme nauwkeurigheid ten opzichte van de energie. Er wordt aangetoond dat de meer-dimensionale CPM in feite een veralgemening is van de één-dimensionale versie: de bekomen formules reduceren tot de formules van de één-dimensionale CPM wanneer het aantal vergelijkingen gelijk is aan één. De meer-dimensionale CPM kan dan gebruikt worden in een shooting procedure om de eigenwaarden te bepalen van een stelsel van gekoppelde vergelijkingen.

Hoofdstuk 6 handelt over singuliere Sturm-Liouville problemen. Een singulier probleem ontstaat wanneer minstens één van de coëfficiënten p^{-1}, q, w niet integreerbaar is tot aan een eindpunt van het integratie-interval of als één (of beide) eindpunt(en) oneindig is. Deze singuliere problemen brengen een aantal specifieke moeilijkheden met zich mee. We bespreken een interval-truncatieprocedure voor problemen gedefinieerd over een oneindig integratie-interval. Daarnaast beschouwen we ook de belangrijke klasse van radiale Schrödinger vergelijkingen waarvoor een verbeterd truncatie-algoritme wordt voorgesteld. Ook bespreken we een algoritme dat rekening houdt met het singulier karakter van een radiaal Schrödinger probleem in de oorsprong $x = 0$. Dit algoritme dient dan toegepast te worden in een klein interval rond de oorsprong.

In hoofdstuk 7 tenslotte, wordt wat meer uitleg gegeven over de structuur en het gebruik van het softwarepakket MATSLISE. MATSLISE is een MATLAB-pakket dat alle CPM en LPM algoritmes voor het één-dimensionale Sturm-Liouville en Schrödinger probleem implementeert. Ook de truncatie-algoritmes voor problemen gedefinieerd over een oneindig integratie-interval, zijn geïncludeerd in MATSLISE, net als de procedure die de singulariteit in de oorsprong van een radiaal Schrödinger probleem afhandelt. Om de gebruiksvriendelijkheid van het pakket te verhogen werd een grafische gebruikersinterface toegevoegd. Aan de hand van een voorbeeld illustreren we de mogelijkheden van deze gebruikersinterface.

De onderzoeksresultaten besproken in deze doctoraatsthesis werden gepubliceerd in gespecialiseerde wetenschappelijke tijdschriften, zie [73–79].

Appendix A

CPM Coefficients

A.1 One-dimensional CPM$\{P,\ N\}$

The four elements of the propagation matrix at $\delta = h$ are (see section 3.4)

$$\begin{aligned}
u(h) &= \xi(Z) + \sum_{m=1}^{\infty} C_m^{(u)} \eta_m(Z), && \text{(A.1)}\\
hu'(h) &= Z\eta_0(Z) + \sum_{m=0}^{\infty} C_m^{(u')} \eta_m(Z), && \text{(A.2)}\\
v(h)/h &= \eta_0(Z) + \sum_{m=2}^{\infty} C_m^{(v)} \eta_m(Z), && \text{(A.3)}\\
v'(h) &= \xi(Z) + \sum_{m=1}^{\infty} C_m^{(v')} \eta_m(Z), && \text{(A.4)}
\end{aligned}$$

where the C coefficients only depend upon the perturbation while the energy dependence is absorbed entirely in the $Z = (V - E)\, h^2$-dependent functions ξ and η_m. Below we give a list of the C coefficients for the CPM{18,16} method. For brevity reasons, the ellipsis symbol was used in some coefficients. The full expressions can be reproduced by using the Maple code given in section B.1. The $V_i, i = 1, 2, \ldots, 16$ are defined by expression (3.41) and $\bar{V}_i = V_i h^{i+2}, i = 1, 2, \ldots, 16$.

$$\begin{aligned}
C_1^{(u)} = \quad & -[\bar{V}_1 + \bar{V}_3 + \bar{V}_5 + \bar{V}_7 + \bar{V}_9 + \bar{V}_{11} + \bar{V}_{13} + \bar{V}_{15}]/2 + \mathcal{O}(h^{19}),\\
C_2^{(u)} = \quad & [5\,\bar{V}_3 + 14\,\bar{V}_5 + 27\,\bar{V}_7 + 44\,\bar{V}_9 + 65\,\bar{V}_{11} + 90\,\bar{V}_{13} + 119\,\bar{V}_{15}]/2\\
& -[15015\,\bar{V}_1^2 + 9009\,\bar{V}_2^2 + 6435\,\bar{V}_3^2 + 5005\,\bar{V}_4^2 + 4095\,\bar{V}_5^2 + 3465\,\bar{V}_6^2\\
& +3003\,\bar{V}_7^2]/360360 + \mathcal{O}(h^{19}),\\
C_3^{(u)} = \quad & [-63\,\bar{V}_5 - 297\,\bar{V}_7 - 858\,\bar{V}_9 - 1950\,\bar{V}_{11} - 3825\,\bar{V}_{13} - 6783\,\bar{V}_{15}]/2\\
& +[-9009\,\bar{V}_2^2 + 8580\,\bar{V}_3^2 - 5005\,\bar{V}_4^2 + 10920\,\bar{V}_5^2 - 3465\,\bar{V}_6^2 + 12012\,\bar{V}_7^2
\end{aligned}$$

$$
\begin{aligned}
&+60060\,\bar{V}_1\bar{V}_2+30030\,\bar{V}_1\bar{V}_3+60060\,\bar{V}_1\bar{V}_4+30030\,\bar{V}_1\bar{V}_5+60060\,\bar{V}_1\bar{V}_6\\
&+30030\,\bar{V}_1\bar{V}_7+60060\,\bar{V}_1\bar{V}_8+30030\,\bar{V}_1\bar{V}_9+60060\,\bar{V}_1\bar{V}_{10}+30030\,\bar{V}_1\bar{V}_{11}\\
&+60060\,\bar{V}_1\,\bar{V}_{12}+30030\,\bar{V}_1\bar{V}_{13}+60060\,\bar{V}_2\,\bar{V}_3+60060\,\bar{V}_2\bar{V}_5+60060\,\bar{V}_2\bar{V}_7\\
&+60060\,\bar{V}_2\bar{V}_9+60060\,\bar{V}_2\bar{V}_{11}+60060\,\bar{V}_3\bar{V}_4+30030\,\bar{V}_3\bar{V}_5+60060\,\bar{V}_3\bar{V}_6\\
&+30030\,\bar{V}_3\bar{V}_7+60060\,\bar{V}_3\bar{V}_8+30030\,\bar{V}_3\bar{V}_9+60060\,\bar{V}_3\bar{V}_{10}+30030\,\bar{V}_3\bar{V}_{11}\\
&+60060\,\bar{V}_4\bar{V}_5+60060\,\bar{V}_4\bar{V}_7+60060\,\bar{V}_4\bar{V}_9+60060\,\bar{V}_5\bar{V}_6+30030\,\bar{V}_5\bar{V}_7\\
&+60060\,\bar{V}_5\bar{V}_8+30030\,\bar{V}_5\bar{V}_9+60060\,\bar{V}_6\bar{V}_7]/120120+\mathcal{O}(h^{19}),\\
C_4^{(u)} = \quad &[1287\,\bar{V}_7+8580\,\bar{V}_9+33150\,\bar{V}_{11}+96900\,\bar{V}_{13}+237405\,\bar{V}_{15}]/2\\
&+[-126126\,\bar{V}_1\bar{V}_4-54054\,\bar{V}_1\bar{V}_5-324324\,\bar{V}_1\bar{V}_6-132132\,\bar{V}_1\,\bar{V}_7\\
&-594594\,\bar{V}_1\bar{V}_8-234234\,\bar{V}_1\,\bar{V}_9-936936\,\bar{V}_1\bar{V}_{10}-360360\,\bar{V}_1\bar{V}_{11}\\
&-1351350\,\bar{V}_1\bar{V}_{12}-510510\,\bar{V}_1\bar{V}_{13}-90090\,\bar{V}_2\bar{V}_3+18018\,\bar{V}_2\bar{V}_4\\
&-252252\,\bar{V}_2\bar{V}_5+18018\,\bar{V}_2\bar{V}_6-486486\,\bar{V}_2\bar{V}_7+18018\,\bar{V}_2\bar{V}_8-792792\,\bar{V}_2\bar{V}_9\\
&+18018\,\bar{V}_2\bar{V}_{10}-1171170\,\bar{V}_2\,\bar{V}_{11}+18018\,\bar{V}_2\bar{V}_{12}-6435\,\bar{V}_3^2-216216\,\bar{V}_3\bar{V}_4\\
&-54054\,\bar{V}_3\,\bar{V}_5-414414\,\bar{V}_3\bar{V}_6-132132\,\bar{V}_3\bar{V}_7-684684\,\bar{V}_3\bar{V}_8-234234\,\bar{V}_3\bar{V}_9\\
&-1027026\,\bar{V}_3\bar{V}_{10}-360360\,\bar{V}_3\bar{V}_{11}+25025\,\bar{V}_4^2-378378\,\bar{V}_4\bar{V}_5\\
&+60060\,\bar{V}_4\bar{V}_6-612612\,\bar{V}_4\bar{V}_7+60060\,\bar{V}_4\bar{V}_8-918918\,\bar{V}_4\bar{V}_9+60060\,\bar{V}_4\bar{V}_{10}\\
&-31122\,\bar{V}_5^2-576576\,\bar{V}_5\bar{V}_6-132132\,\bar{V}_5\bar{V}_7-846846\,\bar{V}_5\bar{V}_8-234234\,\bar{V}_5\bar{V}_9\\
&+59598\,\bar{V}_6^2-810810\,\bar{V}_6\bar{V}_7+126126\,\bar{V}_6\bar{V}_8-69069\,\bar{V}_7^2]/24024\\
&+[15015\,\bar{V}_1^3+18018\,\bar{V}_1^2\bar{V}_2+15015\,\bar{V}_1^2\bar{V}_3+15015\,\bar{V}_1^2\,\bar{V}_5+15015\,\bar{V}_1^2\bar{V}_7\\
&+15015\,\bar{V}_1^2\bar{V}_9+9009\,\bar{V}_1\bar{V}_2^2+23166\,\bar{V}_1\bar{V}_2\bar{V}_3+6435\,\bar{V}_1\bar{V}_3^2\\
&+17160\,\bar{V}_1\bar{V}_3\bar{V}_4+5005\,\bar{V}_1\bar{V}_4^2+13650\,\bar{V}_1\,\bar{V}_4\bar{V}_5+4095\,\bar{V}_1\bar{V}_5^2\\
&+11340\,\bar{V}_1\bar{V}_5\bar{V}_6+2574\,\bar{V}_2^3+9009\,\bar{V}_2^2\bar{V}_3+7722\,\bar{V}_2^2\,\bar{V}_4+9009\,\bar{V}_2^2\bar{V}_5\\
&+9009\,\bar{V}_2^2\bar{V}_7+5148\,\bar{V}_2\bar{V}_3^2+11700\,\bar{V}_2\bar{V}_3\bar{V}_5+3900\,\bar{V}_2\bar{V}_4^2\\
&+9450\,\bar{V}_2\bar{V}_4\bar{V}_6+3150\,\bar{V}_2\bar{V}_5^2+6435\,\bar{V}_3^3+3510\,\bar{V}_3^2\bar{V}_4+6435\,\bar{V}_3^2\,\bar{V}_5\\
&+4500\,\bar{V}_3^2\bar{V}_6+5005\,\bar{V}_3\bar{V}_4^2+5400\,\bar{V}_3\bar{V}_4\bar{V}_5+810\,\bar{V}_4^3]/720720+\mathcal{O}(h^{19}),\\
C_5^{(u)} = \quad &[-36465\,\bar{V}_9-314925\,\bar{V}_{11}-1526175\,\bar{V}_{13}-5460315\,\bar{V}_{15}]/2\\
&+\ldots+\mathcal{O}(h^{19}),\\
C_6^{(u)} = \quad &[1322685\,\bar{V}_{11}+14040810\,\bar{V}_{13}+81904725\,\bar{V}_{15}]/2+\ldots+\mathcal{O}(h^{19}),\\
C_7^{(u)} = \quad &-[58503375\,\bar{V}_{13}+737142525\,\bar{V}_{15}]/2+\ldots+\mathcal{O}(h^{19}),\\
C_8^{(u)} = \quad &3053876175\,\bar{V}_{15}/2+\ldots+\mathcal{O}(h^{19}),\\
C_m^{(u)} = \quad &0+\mathcal{O}(h^{t(m)}) \quad \text{with} \quad t(m)\geq 19, \quad \forall m\geq 9.
\end{aligned}
$$

$$
\begin{aligned}
C_0^{(u')} = \quad &[\bar{V}_2+\bar{V}_4+\bar{V}_6+\bar{V}_8+\bar{V}_{10}+\bar{V}_{12}+\bar{V}_{14}+\bar{V}_{16}]/2+\mathcal{O}(h^{20}),\\
C_1^{(u')} = \quad &-[3\bar{V}_2+10\bar{V}_4+21\bar{V}_6+36\bar{V}_8+55\bar{V}_{10}+78\bar{V}_{12}+105\bar{V}_{14}+136\bar{V}_{16}]/2\\
&-[15015\bar{V}_1^2+9009\bar{V}_2^2+6435\bar{V}_3^2+5005\bar{V}_4^2+4095\bar{V}_5^2+3465\bar{V}_6^2\\
&+3003\bar{V}_7^2]/360360+\mathcal{O}(h^{20}),
\end{aligned}
$$

$$\begin{aligned}
C_2^{(u')} =\ & = [35\bar{V}_4 + 189\bar{V}_6 + 594\bar{V}_8 + 1430\bar{V}_{10} + 2925\bar{V}_{12} + 5355\bar{V}_{14} + 4522\bar{V}_{16}]/2 \\
& -[105105\bar{V}_1^2 + 180180\bar{V}_1\bar{V}_3 + 180180\bar{V}_1\bar{V}_5 + 180180\bar{V}_1\bar{V}_7 + 180180\bar{V}_1\bar{V}_9 \\
& +180180\bar{V}_1\bar{V}_{11} + 180180\bar{V}_1\bar{V}_{13} + 96525\bar{V}_3^2 + 180180\bar{V}_3\bar{V}_5 + 180180\bar{V}_3\bar{V}_7 \\
& +180180\bar{V}_3\bar{V}_9 + 180180\bar{V}_3\bar{V}_{11} + 94185\bar{V}_5^2 + 180180\bar{V}_5\bar{V}_7 + 180180\bar{V}_5\bar{V}_9 \\
& +93093\bar{V}_7^2 + 54054\bar{V}_2^2 + 90090\bar{V}_2\bar{V}_4 + 90090\bar{V}_2\bar{V}_6 + 90090\bar{V}_2\bar{V}_8 \\
& +90090\bar{V}_2\bar{V}_{10} + 90090\bar{V}_2\bar{V}_{12} + 50050\bar{V}_4^2 + 90090\bar{V}_4\bar{V}_6 + 90090\bar{V}_4\bar{V}_8 \\
& +90090\bar{V}_4\bar{V}_{10} + 48510\bar{V}_6^2 + 90090\bar{V}_6\bar{V}_8]/360360 + \mathcal{O}(h^{20}), \\
C_3^{(u')} =\ & [-693\bar{V}_6 - 5148\bar{V}_8 - 21450\bar{V}_{10} - 66300\bar{V}_{12} - 169575\bar{V}_{14} - 379848\bar{V}_{16}]/2 \\
& +[4144140\bar{V}_5\bar{V}_7 + 480480\bar{V}_1\bar{V}_3 + 3963960\bar{V}_6\bar{V}_8 + 2849847\bar{V}_7^2 \\
& +1291290\bar{V}_1\bar{V}_5 + 1482390\bar{V}_5^2 + 5675670\bar{V}_5\bar{V}_9 + 700700\bar{V}_4^2 + 2072070\bar{V}_4\bar{V}_6 \\
& +2972970\bar{V}_4\bar{V}_8 + 3483480\bar{V}_2\bar{V}_{10} + 2342340\bar{V}_2\bar{V}_8 + 1441440\bar{V}_2\bar{V}_6 \\
& +780780\bar{V}_2\bar{V}_4 + 8138130\bar{V}_1\bar{V}_{13} + 5885880\bar{V}_1\bar{V}_{11} + 3993990\bar{V}_1\bar{V}_9 \\
& +2462460\bar{V}_1\bar{V}_7 + 6486480\bar{V}_3\bar{V}_{11} + 4594590\bar{V}_3\bar{V}_9 + 3063060\bar{V}_3\bar{V}_7 \\
& +1891890\bar{V}_3\bar{V}_5 + 4864860\bar{V}_2\bar{V}_{12} + 4114110\bar{V}_4\bar{V}_{10} + 171171\bar{V}_2^2 + 534105\bar{V}_3^2 \\
& +1528065\bar{V}_6^2]/120120 + [3003\bar{V}_1^2\bar{V}_2 - 15015\bar{V}_1^2\bar{V}_4 - 15015\bar{V}_1^2\bar{V}_6 \\
& -15015\bar{V}_1^2\bar{V}_8 - 15015\bar{V}_1^2\bar{V}_{10} + 23166\bar{V}_1\bar{V}_2\bar{V}_3 + 17160\bar{V}_1\bar{V}_3\bar{V}_4 \\
& +13650\bar{V}_1\bar{V}_4\bar{V}_5 + 11340\bar{V}_1\bar{V}_5\bar{V}_6 - 6435\bar{V}_2^3 - 1287\bar{V}_2^2\bar{V}_4 - 9009\bar{V}_2^2\bar{V}_6 \\
& -9009\bar{V}_2^2\bar{V}_8 - 1287\bar{V}_2\bar{V}_3^2 + 11700\bar{V}_2\bar{V}_3\bar{V}_5 - 1105\bar{V}_2\bar{V}_4^2 + 9450\bar{V}_2\bar{V}_4\bar{V}_6 \\
& -945\bar{V}_2\bar{V}_5^2 - 2925\bar{V}_3^2\bar{V}_4 - 1935\bar{V}_3^2\bar{V}_6 + 5400\bar{V}_3\bar{V}_4\bar{V}_5 - 4195\bar{V}_4^3]/720720 \\
& +\mathcal{O}(h^{20}), \\
C_4^{(u')} =\ & [19305\bar{V}_8 + 182325\bar{V}_{10} + 944775\bar{V}_{12} + 3561075\bar{V}_{14} + 10920630\bar{V}_{16}]/2 \\
& + \ldots + \mathcal{O}(h^{20}), \\
C_5^{(u')} =\ & [-692835\bar{V}_{10} - 7936110\bar{V}_{12} - 49142835\bar{V}_{14} - 218412600\bar{V}_{16}]/2 \\
& + \ldots + \mathcal{O}(h^{20}), \\
C_6^{(u')} =\ & [30421755\bar{V}_{12} + 409523625\bar{V}_{14} + 2948570100\bar{V}_{16}]/2 + \ldots + \mathcal{O}(h^{20}), \\
C_7^{(u')} =\ & [-1579591125\bar{V}_{14} - 24431009400\bar{V}_{16}]/2 + \ldots + \mathcal{O}(h^{20}), \\
C_8^{(u')} =\ & 94670161425\bar{V}_{16}/2 + \ldots + \mathcal{O}(h^{20}), \\
C_m^{(u')} =\ & 0 + \mathcal{O}(h^{t(m)}) \quad \text{with} \quad t(m) \geq 20, \quad \forall m \geq 9.
\end{aligned}$$

$$\begin{aligned}
C_2^{(v)} =\ & -[\bar{V}_2 + \bar{V}_4 + \bar{V}_6 + \bar{V}_8 + \bar{V}_{10} + \bar{V}_{12} + \bar{V}_{14}]/2 + \mathcal{O}(h^{18}), \\
C_3^{(v)} =\ & [7\bar{V}_4 + 18\bar{V}_6 + 33\bar{V}_8 + 52\bar{V}_{10} + 75\bar{V}_{12} + 102\bar{V}_{14}]/2 \\
& -[15015\bar{V}_1^2 + 9009\bar{V}_2^2 + 6435\bar{V}_3^2 + 5005\bar{V}_4^2 + 4095\bar{V}_5^2 + 3465\bar{V}_6^2]/360360 \\
& +\mathcal{O}(h^{18}), \\
C_4^{(v)} =\ & -[99\bar{V}_6 + 429\bar{V}_8 + 1170\bar{V}_{10} + 2550\bar{V}_{12} + 4845\bar{V}_{14}]/2 \\
& +[60060\bar{V}_1\bar{V}_3 + 60060\bar{V}_1\bar{V}_5 + 60060\bar{V}_1\bar{V}_7 + 60060\bar{V}_1\bar{V}_9 + 60060\bar{V}_1\bar{V}_{11} \\
& +27027\bar{V}_2^2 + 90090\bar{V}_2\bar{V}_4 + 90090\bar{V}_2\bar{V}_6 + 90090\bar{V}_2\bar{V}_8 + 90090\bar{V}_2\bar{V}_{10}
\end{aligned}$$

$$\begin{aligned}
& +17160\bar{V}_3^2+60060\bar{V}_3\bar{V}_5+60060\bar{V}_3\bar{V}_7+60060\bar{V}_3\bar{V}_9+35035\bar{V}_4^2\\
& +90090\bar{V}_4\bar{V}_6+90090\bar{V}_4\bar{V}_8+21840\bar{V}_5^2+60060\bar{V}_5\bar{V}_7+38115\bar{V}_6^2]/120120\\
& +\mathcal{O}(h^{18}),\\
C_5^{(v)} =\;& [2145\bar{V}_8+13260\bar{V}_{10}+48450\bar{V}_{12}+135660\bar{V}_{14}]/2+\ldots+\mathcal{O}(h^{18}),\\
C_6^{(v)} =\;& -[62985\bar{V}_{10}+508725\bar{V}_{12}+2340135\bar{V}_{14}]/2+\ldots+\mathcal{O}(h^{18}),\\
C_7^{(v)} =\;& [2340135\bar{V}_{12}+23401350\bar{V}_{14}]/2+\ldots+\mathcal{O}(h^{18}),\\
C_8^{(v)} =\;& -105306075\bar{V}_{14}/2+\ldots+\mathcal{O}(h^{18}),\\
C_m^{(v)} =\;& 0+\mathcal{O}(h^{t(m)}) \quad \text{with} \quad t(m)\geq 18, \quad \forall m\geq 9.
\end{aligned}$$

$$\begin{aligned}
C_1^{(v')} =\;& [\bar{V}_1+\bar{V}_3+\bar{V}_5+\bar{V}_7+\bar{V}_9+\bar{V}_{11}+\bar{V}_{13}+\bar{V}_{15}]/2+\mathcal{O}(h^{19}),\\
C_2^{(v')} =\;& -[5\bar{V}_3+14\bar{V}_5+27\bar{V}_7+44\bar{V}_9+65\bar{V}_{11}+90\bar{V}_{13}+119\bar{V}_{15}]/2\\
& -[15015\bar{V}_1^2+9009\bar{V}_2^2+6435\bar{V}_3^2+5005\bar{V}_4^2+4095\bar{V}_5^2+3465\bar{V}_6^2\\
& +3003\bar{V}_7^2]/360360+\mathcal{O}(h^{19}),\\
C_3^{(v')} =\;& [63\bar{V}_5+297\bar{V}_7+858\bar{V}_9+1950\bar{V}_{11}+3825\bar{V}_{13}+6783\bar{V}_{15}]/2\\
& +[-9009\bar{V}_2^2+8580\bar{V}_3^2-5005\bar{V}_4^2+10920\bar{V}_5^2-3465\bar{V}_6^2+12012\bar{V}_7^2\\
& +30030\bar{V}_1\bar{V}_3+30030\bar{V}_1\bar{V}_5+30030\bar{V}_1\bar{V}_7+30030\bar{V}_1\bar{V}_9+30030\bar{V}_1\bar{V}_{11}\\
& +30030\bar{V}_1\bar{V}_{13}-60060\bar{V}_1\bar{V}_2-60060\bar{V}_1\bar{V}_4-60060\bar{V}_1\bar{V}_6-60060\bar{V}_1\bar{V}_8\\
& -60060\bar{V}_1\bar{V}_{10}-60060\bar{V}_1\bar{V}_{12}+30030\bar{V}_3\bar{V}_5+30030\bar{V}_3\bar{V}_7+30030\bar{V}_3\bar{V}_9\\
& +30030\bar{V}_3\bar{V}_{11}-60060\bar{V}_3\bar{V}_2-60060\bar{V}_3\bar{V}_4-60060\bar{V}_3\bar{V}_6\bar{V}_3\bar{V}_8-60060\bar{V}_3\bar{V}_{10}\\
& +30030\bar{V}_5\bar{V}_7+30030\bar{V}_5\bar{V}_9-60060\bar{V}_5\bar{V}_2-60060\bar{V}_5\bar{V}_4-60060\bar{V}_5\bar{V}_6\\
& -60060\bar{V}_5\bar{V}_8-60060\bar{V}_7\bar{V}_2-60060\bar{V}_7\bar{V}_4-60060\bar{V}_7\bar{V}_6-60060\bar{V}_9\bar{V}_2\\
& -60060\bar{V}_9\bar{V}_4-60060\bar{V}_{11}\bar{V}_2]/120120+\mathcal{O}(h^{19}),\\
C_4^{(v')} =\;& -[1287\bar{V}_7+8580\bar{V}_9+33150\bar{V}_{11}+96900\bar{V}_{13}+237405\bar{V}_{15}]/2\\
& +[-54054\bar{V}_1\bar{V}_5-132132\bar{V}_1\bar{V}_7-234234\bar{V}_1\bar{V}_9-360360\bar{V}_1\bar{V}_{11}\\
& -510510\bar{V}_1\bar{V}_{13}+126126\bar{V}_1\bar{V}_4+324324\bar{V}_1\bar{V}_6+594594\bar{V}_1\bar{V}_8+936936\bar{V}_1\bar{V}_{10}\\
& +1351350\bar{V}_1\bar{V}_{12}-6435\bar{V}_3^2-54054\bar{V}_3\bar{V}_5-132132\bar{V}_3\bar{V}_7-234234\bar{V}_3\bar{V}_9\\
& -360360\bar{V}_3\bar{V}_{11}+90090\bar{V}_3\bar{V}_2+216216\bar{V}_3\bar{V}_4+414414\bar{V}_3\bar{V}_6+684684\bar{V}_3\bar{V}_8\\
& +1027026\bar{V}_3\bar{V}_{10}-31122\bar{V}_5^2-132132\bar{V}_5\bar{V}_7-234234\bar{V}_5\bar{V}_9+252252\bar{V}_5\bar{V}_2\\
& +378378\bar{V}_5\bar{V}_4+576576\bar{V}_5\bar{V}_6+846846\bar{V}_5\bar{V}_8-69069\bar{V}_7^2+486486\bar{V}_7\bar{V}_2\\
& +612612\bar{V}_7\bar{V}_4+810810\bar{V}_7\bar{V}_6+792792\bar{V}_9\bar{V}_2+918918\bar{V}_9\bar{V}_4+1171170\bar{V}_{11}\bar{V}_2\\
& +18018\bar{V}_2\bar{V}_4+18018\bar{V}_2\bar{V}_6+18018\bar{V}_2\bar{V}_8+18018\bar{V}_2\bar{V}_{10}+18018\bar{V}_2\bar{V}_{12}\\
& +25025\bar{V}_4^2+60060\bar{V}_4\bar{V}_6+60060\bar{V}_4\bar{V}_8+60060\bar{V}_4\bar{V}_{10}+59598\bar{V}_6^2\\
& +126126\bar{V}_6\bar{V}_8]/24024+[5400\bar{V}_3\bar{V}_4\bar{V}_5+13650\bar{V}_1\bar{V}_4\bar{V}_5+11340\bar{V}_1\bar{V}_5\bar{V}_6\\
& +17160\bar{V}_1\bar{V}_3\bar{V}_4+3150\bar{V}_2\bar{V}_5^2+9450\bar{V}_2\bar{V}_4\bar{V}_6+23166\bar{V}_1\bar{V}_2\bar{V}_3+11700\bar{V}_2\bar{V}_3\bar{V}_5\\
& -15015\bar{V}_1^2\bar{V}_5+18018\bar{V}_1^2\bar{V}_2+4500\bar{V}_3^2\bar{V}_6+7722\bar{V}_2^2\bar{V}_4-9009\bar{V}_1\bar{V}_2^2\\
& -6435\bar{V}_1\bar{V}_3^2-15015\bar{V}_1^2\bar{V}_7+3900\bar{V}_2\bar{V}_4^2+5148\bar{V}_2\bar{V}_3^2+3510\bar{V}_3^2\bar{V}_4+2574\bar{V}_2^3
\end{aligned}$$

$$\begin{aligned}
& +810\bar{V}_4^3 - 15015\bar{V}_1^2\bar{V}_3 - 15015\bar{V}_1^2\bar{V}_9 - 5005\bar{V}_1\bar{V}_4^2 - 4095\bar{V}_1\bar{V}_5^2 - 9009\bar{V}_2^2\bar{V}_3 \\
& -9009\bar{V}_2^2\bar{V}_5 - 9009\bar{V}_2^2\bar{V}_7 - 6435\bar{V}_3^2\bar{V}_5 - 5005\bar{V}_3\bar{V}_4^2 - 15015\bar{V}_1^3 \\
& -6435\bar{V}_3^3]/720720 + \mathcal{O}(h^{19}), \\
C_5^{(v')} = {} & [36465\bar{V}_9 + 314925\bar{V}_{11} + 1526175\bar{V}_{13} + 5460315\bar{V}_{15}]/2 + \ldots + \mathcal{O}(h^{19}), \\
C_6^{(v')} = {} & -[1322685\bar{V}_{11} + 14040810\bar{V}_{13} + 81904725\bar{V}_{15}]/2 + \ldots + \mathcal{O}(h^{19}), \\
C_7^{(v')} = {} & [58503375\bar{V}_{13} + 737142525\bar{V}_{15}]/2 + \ldots + \mathcal{O}(h^{19}), \\
C_8^{(v')} = {} & -3053876175\bar{V}_{15}/2 + \ldots + \mathcal{O}(h^{19}), \\
C_m^{(v')} = {} & 0 + \mathcal{O}(h^{t(m)}) \quad \text{with} \quad t(m) \geq 19, \quad \forall m \geq 9.
\end{aligned}$$

A.2 Generalized CPM$\{P,\ N\}$

The propagators at $\delta = h$ have the following form :

$$\mathbf{u}^{\mathbf{D}}(h) = \boldsymbol{\xi}(\mathbf{Z}) + \sum_{m=1}^{\infty} \mathbf{C}_m^{(u)} \boldsymbol{\eta}_m(\mathbf{Z}), \tag{A.5}$$

$$h\mathbf{u}^{\mathbf{D}\prime}(h) = \mathbf{Z}\boldsymbol{\eta}_0(\mathbf{Z}) + \sum_{m=0}^{\infty} \mathbf{C}_m^{(u')} \boldsymbol{\eta}_m(\mathbf{Z}), \tag{A.6}$$

$$\mathbf{v}^{\mathbf{D}}(h)/h = \boldsymbol{\eta}_0(\mathbf{Z}) + \sum_{m=2}^{\infty} \mathbf{C}_m^{(v)} \boldsymbol{\eta}_m(\mathbf{Z}), \tag{A.7}$$

$$\mathbf{v}^{\mathbf{D}\prime}(h) = \boldsymbol{\xi}(\mathbf{Z}) + \sum_{m=1}^{\infty} \mathbf{C}_m^{(v')} \boldsymbol{\eta}_m(\mathbf{Z}), \tag{A.8}$$

where $\xi(\mathbf{Z})$ and $\eta_m(\mathbf{Z})$ are the $n \times n$ diagonal matrices as defined in section 5.2.2.

The $\mathbf{C}_m$ coefficient matrices only depend upon the perturbation and the channel separation while the energy dependence is absorbed entirely in the $\mathbf{Z} = (\mathbf{V}_0^{\mathbf{D}} - E\mathbf{I})\, h^2$-dependent matrices of functions $\xi(\mathbf{Z})$ and $\eta_m(\mathbf{Z})$. Below we give a list of the $\mathbf{C}_m$ coefficients for the CPM{10,8} method as obtained by the MAPLE code listed in section B.3. For notational brevity the upper label $\mathbf{D}$ is suppressed and $\bar{\mathbf{V}}_i = \mathbf{V}_i h^{i+2}, i = 0, 1, \ldots, 8$. The coefficients are expressed in commutators and anticommutators of two matrices with $[\mathbf{A}, \mathbf{B}] = \mathbf{AB} - \mathbf{BA}$ the commutator and $\{\mathbf{A}, \mathbf{B}\} = \mathbf{AB} + \mathbf{BA}$ the anticommutator of the matrices **A** and **B**.

$$\begin{aligned}
\mathbf{C}_1^{(u)} = {} & -(\bar{\mathbf{V}}_1 + \bar{\mathbf{V}}_3 + \bar{\mathbf{V}}_5 + \bar{\mathbf{V}}_7)/2 + [\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]/24 \\
& +[-7\bar{\mathbf{V}}_1 + 3\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_2]/840 + \mathcal{O}(h^{11}), \\
\mathbf{C}_2^{(u)} = {} & (5\bar{\mathbf{V}}_3 + 14\bar{\mathbf{V}}_5 + 27\bar{\mathbf{V}}_7)/2 - \bar{\mathbf{V}}_1^2/24 - \bar{\mathbf{V}}_2^2/40 - \bar{\mathbf{V}}_3^2/56 \\
& +[4\bar{\mathbf{V}}_1 - 3\bar{\mathbf{V}}_2 + 3\bar{\mathbf{V}}_3 - 3\bar{\mathbf{V}}_4 + 3\bar{\mathbf{V}}_5 - 3\bar{\mathbf{V}}_6, \bar{\mathbf{V}}_0]/24
\end{aligned}$$

$$
\begin{aligned}
& +[-7\bar{\mathbf{V}}_1+3\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_2]/840+\big[[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_2],2\bar{\mathbf{V}}_1+7\bar{\mathbf{V}}_0\big]/3360\\
& -\big[[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0],7\bar{\mathbf{V}}_1+2\bar{\mathbf{V}}_3\big]/3360-\big[[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_2\big]/3360\\
& +\big[[\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_1\big]/1120-\big[[5\bar{\mathbf{V}}_1-\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big]/480+\mathcal{O}(h^{11}),\\
\mathbf{C}_3^{(u)} \;=\;& (-63\bar{\mathbf{V}}_5-297\bar{\mathbf{V}}_7)/2-3\bar{\mathbf{V}}_2^2/40+\bar{\mathbf{V}}_3^2/14+\{\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_3\}/4\\
& +\{\bar{\mathbf{V}}_1,2\bar{\mathbf{V}}_2+\bar{\mathbf{V}}_3+2\bar{\mathbf{V}}_4+\bar{\mathbf{V}}_5\}/8-[7\bar{\mathbf{V}}_1+32\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_2]/280\\
& +[-6\bar{\mathbf{V}}_3+7\bar{\mathbf{V}}_4-15\bar{\mathbf{V}}_5+18\bar{\mathbf{V}}_6,\bar{\mathbf{V}}_0]/8-\big[[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_1\big]/160\\
& -\big[[5\bar{\mathbf{V}}_1-6\bar{\mathbf{V}}_2+5\bar{\mathbf{V}}_3-5\bar{\mathbf{V}}_4,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big]/160\\
& +\Big((24\bar{\mathbf{V}}_2-21\bar{\mathbf{V}}_1)[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0]+[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0](18\bar{\mathbf{V}}_2-21\bar{\mathbf{V}}_1)\Big)/3360\\
& +\Big(13[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_2]\bar{\mathbf{V}}_1+\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_2]\Big)/3360-\Big([\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0](41\bar{\mathbf{V}}_3-28\bar{\mathbf{V}}_2)\\
& +(29\bar{\mathbf{V}}_3+42\bar{\mathbf{V}}_2)[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]+9\big[\bar{\mathbf{V}}_1,[\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_0]\big]\Big)/3360\\
& +\big[[[3\bar{\mathbf{V}}_1-\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big]/1920+\big[[[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big],\bar{\mathbf{V}}_1\big]/1920\\
& +[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]^2/1152+\mathcal{O}(h^{11}),\\
\mathbf{C}_4^{(u)} \;=\;& 1287\bar{\mathbf{V}}_7/2-15\bar{\mathbf{V}}_3^2/56+\bar{\mathbf{V}}_1^3/48-\{\bar{\mathbf{V}}_1,21\bar{\mathbf{V}}_4+9\bar{\mathbf{V}}_5\}/8\\
& +\{\bar{\mathbf{V}}_2,-15\bar{\mathbf{V}}_3+3\bar{\mathbf{V}}_4\}/8+\bar{\mathbf{V}}_1\{\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_2\}/80+3[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_3]/56\\
& +[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_4]/8-(5\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_2]+2[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_2]\bar{\mathbf{V}}_1)/560\\
& +[72\bar{\mathbf{V}}_5-99\bar{\mathbf{V}}_6,\bar{\mathbf{V}}_0]/8+\Big(10\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_0]+25[\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1\\
& +[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0](30\bar{\mathbf{V}}_3-28\bar{\mathbf{V}}_2)+5\bar{\mathbf{V}}_3[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]\Big)/560\\
& +\big[[\bar{\mathbf{V}}_3-2\bar{\mathbf{V}}_4,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big]/8+\big[[[2\bar{\mathbf{V}}_1-3\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big]/480\\
& -\big[[\bar{\mathbf{V}}_2\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big]/10+\big[[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big]\bar{\mathbf{V}}_1/480\\
& +\Big([\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0](36\bar{\mathbf{V}}_2+21\bar{\mathbf{V}}_1)+(48\bar{\mathbf{V}}_2-42\bar{\mathbf{V}}_1)[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0]\Big)/560\\
& +[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]^2/240+\mathcal{O}(h^{11})\\
\mathbf{C}_m^{(u)} \;=\;& \mathbf{0}+\mathcal{O}(h^{t(m)})\ \text{ with } t(m)\geq 11,\ \forall m\geq 5.
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{C}_0^{(u')} \;=\;& (\bar{\mathbf{V}}_2+\bar{\mathbf{V}}_4+\bar{\mathbf{V}}_6+\bar{\mathbf{V}}_8)/2+[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]/24+[-7\bar{\mathbf{V}}_1+3\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_2]/840\\
& +[\bar{\mathbf{V}}_4,\bar{\mathbf{V}}_3]/504+\mathcal{O}(h^{12}),\\
\mathbf{C}_1^{(u')} \;=\;& -(3\bar{\mathbf{V}}_2+10\bar{\mathbf{V}}_4+21\bar{\mathbf{V}}_6+36\bar{\mathbf{V}}_8)/2-\bar{\mathbf{V}}_1^2/24-\bar{\mathbf{V}}_2^2/40-\bar{\mathbf{V}}_3^2/56\\
& +[\bar{\mathbf{V}}_1-\bar{\mathbf{V}}_2+\bar{\mathbf{V}}_3-\bar{\mathbf{V}}_4+\bar{\mathbf{V}}_5-\bar{\mathbf{V}}_6+\bar{\mathbf{V}}_7,\bar{\mathbf{V}}_0]/8\\
& +\big[[-5\bar{\mathbf{V}}_1+\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\big]/480+\big[\bar{\mathbf{V}}_1-\bar{\mathbf{V}}_2,[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]\big]/480\\
& +\big[7\bar{\mathbf{V}}_1+\bar{\mathbf{V}}_2,[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0]\big]/3360+\big[[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_2],\bar{\mathbf{V}}_1\big]/1680\\
& -\big(3\big[\bar{\mathbf{V}}_1,[\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_0]\big]-2\big[\bar{\mathbf{V}}_3,[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]\big]\big)/3360+\big[[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_3],\bar{\mathbf{V}}_0\big]/1120\\
& +\mathcal{O}(h^{12}),\\
\mathbf{C}_2^{(u')} \;=\;& (35\bar{\mathbf{V}}_4+189\bar{\mathbf{V}}_6+594\bar{\mathbf{V}}_8)/2-7\bar{\mathbf{V}}_1^2/24-3\bar{\mathbf{V}}_2^2/20-15\bar{\mathbf{V}}_3^2/56\\
& -\{\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_3+\bar{\mathbf{V}}_5\}/4+([\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_4]-\{\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_4\})/8
\end{aligned}
$$

$$
\begin{aligned}
& +[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_2 + \bar{\mathbf{V}}_4 + \bar{\mathbf{V}}_6]/8 + 11[\bar{\mathbf{V}}_2^2, \bar{\mathbf{V}}_1]/3360 + [\bar{\mathbf{V}}_1^2, \bar{\mathbf{V}}_3]/672 \\
& +[3\bar{\mathbf{V}}_2 - 5\bar{\mathbf{V}}_3 + 10\bar{\mathbf{V}}_4 - 14\bar{\mathbf{V}}_5 + 21\bar{\mathbf{V}}_6 - 27\bar{\mathbf{V}}_7, \bar{\mathbf{V}}_0] \\
& +\big(77\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0] - 7[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1 + 20[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_2 \\
& +22\bar{\mathbf{V}}_2[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0]\big)/3360 + \big(-3\{\bar{\mathbf{V}}_1, [\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0]\} + 4\{\bar{\mathbf{V}}_2, [\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]\}\big)/480 \\
& +3\big[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_2], \bar{\mathbf{V}}_1\big]/1120 + \{\bar{\mathbf{V}}_4, [\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]\}/96 \\
& -\big[[15\bar{\mathbf{V}}_1 - 16\bar{\mathbf{V}}_2 + 15\bar{\mathbf{V}}_3 - 15\bar{\mathbf{V}}_4 + 15\bar{\mathbf{V}}_5, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big]/480 \\
& +\big[[[21\bar{\mathbf{V}}_1 - 7\bar{\mathbf{V}}_2 + \bar{\mathbf{V}}_3, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big]/13440 \\
& +\big(- 11\big[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], [\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0]\big] - 3\big[\bar{\mathbf{V}}_1, [[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0]\big] \\
& +5\big[\bar{\mathbf{V}}_2, [[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0]\big]\big)/13440 - \bar{\mathbf{V}}_1[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1/4480 \\
& -\big(\{\bar{\mathbf{V}}_3, [\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0]\} + 2\{\bar{\mathbf{V}}_2, [\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_0]\}\big)/560 \\
& +\big(-3\big[\bar{\mathbf{V}}_1, [[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0]\big] + 5[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]^2\big)/5760 + [\bar{\mathbf{V}}_1^3, \bar{\mathbf{V}}_0]/13440 \\
& +\big(-3\big[\bar{\mathbf{V}}_1, [\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_0]\big] + 37\big[\bar{\mathbf{V}}_3, [\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]\big]\big)/3360 + \mathcal{O}(h^{12}), \\
\mathbf{C}_3^{(u')} \quad = \quad & -693/2\bar{\mathbf{V}}_6 - 2574\bar{\mathbf{V}}_8 + 57/40\bar{\mathbf{V}}_2^2 + 249/56\bar{\mathbf{V}}_3^2 + \{\bar{\mathbf{V}}_1, 16\bar{\mathbf{V}}_3 + 43\bar{\mathbf{V}}_5\}/8 \\
& +13/4\{\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_4\} + 3/8[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_3 + \bar{\mathbf{V}}_5] + 7/8[\bar{\mathbf{V}}_4, \bar{\mathbf{V}}_1 + \bar{\mathbf{V}}_3] + 9/4[\bar{\mathbf{V}}_6, \bar{\mathbf{V}}_1] \\
& +[-35\bar{\mathbf{V}}_4 + 63\bar{\mathbf{V}}_5 - 189\bar{\mathbf{V}}_6 + 297\bar{\mathbf{V}}_7, \bar{\mathbf{V}}_0]/8 + 5[\bar{\mathbf{V}}_1^2, \bar{\mathbf{V}}_3]/336 \\
& +\big([\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0](154\bar{\mathbf{V}}_1 - 231\bar{\mathbf{V}}_2 + 309\bar{\mathbf{V}}_3 - 455\bar{\mathbf{V}}_4) \\
& +(336\bar{\mathbf{V}}_1 - 21\bar{\mathbf{V}}_2 + 111\bar{\mathbf{V}}_3 - 245\bar{\mathbf{V}}_4)[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0] + (-378\bar{\mathbf{V}}_1 + 45\bar{\mathbf{V}}_2)[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0] \\
& +[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0](-168\bar{\mathbf{V}}_1 + 207\bar{\mathbf{V}}_2) - 105[\bar{\mathbf{V}}_4, \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1 - 315\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_4, \bar{\mathbf{V}}_0] \\
& +114[\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1 + 306\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_0]\big)/3360 \\
& +\big[[-9\bar{\mathbf{V}}_2 + 25\bar{\mathbf{V}}_3 - 45\bar{\mathbf{V}}_4 + 70\bar{\mathbf{V}}_5, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big]/160 \\
& -\big(76\{\bar{\mathbf{V}}_3, [\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0]\} + 47\{\bar{\mathbf{V}}_2, [\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_0]\}\big)/1120 \\
& +\big([[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1 + [\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]^2 - 3\bar{\mathbf{V}}_1[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0]\big)/640 \\
& +\big(- 18\bar{\mathbf{V}}_2[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0] + 31[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0][\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0] + 46[[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1 \\
& -47[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0][\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0] + 33\bar{\mathbf{V}}_1[[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0] - 29[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_2\big)/13440 \\
& -\big[[[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big]/5760 + \big[[[63\bar{\mathbf{V}}_1 - 77\bar{\mathbf{V}}_2 + 73\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big] \\
& +7\bar{\mathbf{V}}_1\{\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_2\}/3360 - (20\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_2] - 13[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_2]\bar{\mathbf{V}}_1)/3360 \\
& +11/1120[\bar{\mathbf{V}}_2^2, \bar{\mathbf{V}}_1] - 11/8064\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1 - [\bar{\mathbf{V}}_1^3, \bar{\mathbf{V}}_0]/8064 + \mathcal{O}(h^{12}), \\
\mathbf{C}_4^{(u')} \quad = \quad & 19305\bar{\mathbf{V}}_8/2 - 1065/56\bar{\mathbf{V}}_3^2 - \big(261\{\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_5\} + 165\{\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_4\}\big)/8 \\
& +[693\bar{\mathbf{V}}_6 - 1287\bar{\mathbf{V}}_7, \bar{\mathbf{V}}_0]/8 + \big[[20\bar{\mathbf{V}}_4 - 63\bar{\mathbf{V}}_5, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0]\big]/32 \\
& +\big(99[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_6] + 27[\bar{\mathbf{V}}_5, \bar{\mathbf{V}}_2] + 15[\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_4]\big)/8 \\
& +\big[[[7\bar{\mathbf{V}}_2 - 25\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big]/1120 - 1/1920\big[[[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big] \\
& +(16\bar{\mathbf{V}}_2[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0] + 59[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0][\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0] + 25[[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1 \\
& +66\bar{\mathbf{V}}_1[[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0] + 74[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0][\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0] + 26[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_0)/2240 \\
& -(5\big[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big]\bar{\mathbf{V}}_1 + 18\bar{\mathbf{V}}_1\big[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big] + 16[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]^2)/960 \\
& +\big((-492\bar{\mathbf{V}}_2 + 705\bar{\mathbf{V}}_3)[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0] - 306[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0](\bar{\mathbf{V}}_2 + 495\bar{\mathbf{V}}_3)\big)/1120 \\
& +\big([\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0](-72\bar{\mathbf{V}}_3 + 154\bar{\mathbf{V}}_4) + (-152\bar{\mathbf{V}}_3 + 56\bar{\mathbf{V}}_4)[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0] \\
& +2171\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_4, \bar{\mathbf{V}}_0] + 119[\bar{\mathbf{V}}_4, \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1 + [\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_0](-88\bar{\mathbf{V}}_1 + 114\bar{\mathbf{V}}_2)
\end{aligned}
$$

$$
\begin{aligned}
& +(-136\bar{\mathbf{V}}_1 + 72\bar{\mathbf{V}}_2)[\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_0]\big)/224 + 9/112[\bar{\mathbf{V}}_2^2, \bar{\mathbf{V}}_1] \\
& +259\bar{\mathbf{V}}_1\{\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_2\}/1120 - (101\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_2] + 158[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_2]\bar{\mathbf{V}}_1)/1120 \\
& -9[\bar{\mathbf{V}}_1^2, \bar{\mathbf{V}}_3]/224 - 1/560\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1 - 31/6720[\bar{\mathbf{V}}_1^3, \bar{\mathbf{V}}_0] + \mathcal{O}(h^{12}), \\
\mathbf{C}_m^{(u')} \quad = \quad & \mathbf{0} + \mathcal{O}(h^{t(m)}) \ \text{ with } t(m) \geq 12, \ \forall m \geq 5.
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{C}_2^{(v)} \quad = \quad & -(\bar{\mathbf{V}}_2 + \bar{\mathbf{V}}_4 + \bar{\mathbf{V}}_6)/2) + [\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]/24 + [-7\bar{\mathbf{V}}_1 + 3\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_2]/840 + \mathcal{O}(h^{10}), \\
\mathbf{C}_3^{(v)} \quad = \quad & 7\bar{\mathbf{V}}_4/2 + 9\bar{\mathbf{V}}_6 - \bar{\mathbf{V}}_1^2/24 - \bar{\mathbf{V}}_2^2/40 + [\bar{\mathbf{V}}_2, 7\bar{\mathbf{V}}_1 - 3\bar{\mathbf{V}}_3]/280 \\
& -[\bar{\mathbf{V}}_3 - \bar{\mathbf{V}}_2 - \bar{\mathbf{V}}_4 + \bar{\mathbf{V}}_5, \bar{\mathbf{V}}_0]/8 + \big[-5\bar{\mathbf{V}}_1 + \bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big]/480 \\
& + \big[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_2], \bar{\mathbf{V}}_0\big]/480 + \big[\bar{\mathbf{V}}_1, [\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]\big]/480 + \mathcal{O}(h^{10}), \\
\mathbf{C}_4^{(v)} \quad = \quad & -99\bar{\mathbf{V}}_6/2 + 9\bar{\mathbf{V}}_2^2/40 + \{\bar{\mathbf{V}}_1, 2\bar{\mathbf{V}}_3 + \bar{\mathbf{V}}_4\}/8 + 3[\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_2]/56 \\
& +[-7\bar{\mathbf{V}}_4 + 9\bar{\mathbf{V}}_5, \bar{\mathbf{V}}_0]/8 + \big[[-3\bar{\mathbf{V}}_2 + 5\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big]/160 \\
& + \big[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big]/640 - \big([\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0](-4\bar{\mathbf{V}}_1 + 3\bar{\mathbf{V}}_2) \\
& +(-6\bar{\mathbf{V}}_1 + 9\bar{\mathbf{V}}_2)[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0] + 6[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1\big)/480 + \mathcal{O}(h^{10}), \\
\mathbf{C}_m^{(v)} \quad = \quad & \mathbf{0} + \mathcal{O}(h^{t(m)}) \ \text{ with } t(m) \geq 10, \ \forall m \geq 5.
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{C}_1^{(v')} \quad = \quad & (\bar{\mathbf{V}}_1 + \bar{\mathbf{V}}_3 + \bar{\mathbf{V}}_5 + \bar{\mathbf{V}}_7)/2 + [\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]/24 + [-7\bar{\mathbf{V}}_1 + 3\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_2]/840 + \mathcal{O}(h^{11}), \\
\mathbf{C}_2^{(v')} \quad = \quad & -(5\bar{\mathbf{V}}_3 + 14\bar{\mathbf{V}}_5 + 27\bar{\mathbf{V}}_7)/2 - \bar{\mathbf{V}}_1^2/24 - \bar{\mathbf{V}}_2^2/40 - \bar{\mathbf{V}}_3^2/56 \\
& +[\bar{\mathbf{V}}_2, 7\bar{\mathbf{V}}_1 - 3\bar{\mathbf{V}}_3]/840 + [-2\bar{\mathbf{V}}_1 + 3\bar{\mathbf{V}}_2 - 3\bar{\mathbf{V}}_3 + 3\bar{\mathbf{V}}_4 - 3\bar{\mathbf{V}}_5 + 3\bar{\mathbf{V}}_6, \bar{\mathbf{V}}_0]/24 \\
& + \big[[-5\bar{\mathbf{V}}_1 + \bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big]/480 + \big[\bar{\mathbf{V}}_1 - \bar{\mathbf{V}}_2, [\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]\big]/480 \\
& + \big[7\bar{\mathbf{V}}_1 + \bar{\mathbf{V}}_2, [\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0]\big]/3360 - \big[\bar{\mathbf{V}}_1, [\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_0]\big]/1120 + \big[\bar{\mathbf{V}}_3, [\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]\big]/1680 \\
& - \big[\bar{\mathbf{V}}_1, [\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_2]\big]/1680 + \mathcal{O}(h^{10}), \\
\mathbf{C}_3^{(v')} \quad = \quad & + \big(63\bar{\mathbf{V}}_5 + 297\bar{\mathbf{V}}_7\big)/2 + \{\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_3 - 2\bar{\mathbf{V}}_4 + \bar{\mathbf{V}}_5\}/8 - \{\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_1 + \bar{\mathbf{V}}_3\}/4 \\
& -3\bar{\mathbf{V}}_2^2/40 + \bar{\mathbf{V}}_3^2/14 + [4\bar{\mathbf{V}}_3 - 7\bar{\mathbf{V}}_4 + 13\bar{\mathbf{V}}_5 - 18\bar{\mathbf{V}}_6, \bar{\mathbf{V}}_0]/8 \\
& +[\bar{\mathbf{V}}_2, 7\bar{\mathbf{V}}_1 + 32\bar{\mathbf{V}}_3]/280 + \big[5\bar{\mathbf{V}}_1 - 12\bar{\mathbf{V}}_2 + 15\bar{\mathbf{V}}_3 - 15\bar{\mathbf{V}}_4, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big]/480 \\
& + \big(-3\bar{\mathbf{V}}_1[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0] + 5[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]^2 + 13[[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1\big)/5760 \\
& + \big([\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0](7\bar{\mathbf{V}}_1 - 4\bar{\mathbf{V}}_2) + (13\bar{\mathbf{V}}_1 + 2\bar{\mathbf{V}}_2)[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]\big)/480 \\
& + \big([\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0](-35\bar{\mathbf{V}}_1 + 18\bar{\mathbf{V}}_2) + (-7\bar{\mathbf{V}}_1 + 24\bar{\mathbf{V}}_2)[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0]\big)/3360 \\
& + \big(9\,\big[[\bar{\mathbf{V}}_3, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_1\big] + 41\bar{\mathbf{V}}_3[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0] + 29[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0]\bar{\mathbf{V}}_3\big)/3360 \\
& + \big[[[3\bar{\mathbf{V}}_1 - \bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0], \bar{\mathbf{V}}_0\big]/1920 \\
& +([\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_1]\bar{\mathbf{V}}_1 - 13\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_2])/3360 + \mathcal{O}(h^{10}), \\
\mathbf{C}_4^{(v')} \quad = \quad & -1287\bar{\mathbf{V}}_7/2 - 15\bar{\mathbf{V}}_3^2/56 - \bar{\mathbf{V}}_1^3/48 + \{\bar{\mathbf{V}}_2, 105\bar{\mathbf{V}}_3 + 21\bar{\mathbf{V}}_4\}/56 \\
& +63\{\bar{\mathbf{V}}_1, 147\bar{\mathbf{V}}_4 - 63\bar{\mathbf{V}}_5\}/56 + [\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_4]/8 - 3[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_3]/56 \\
& +\bar{\mathbf{V}}_1\{\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_2\}/80 + (-5\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_1\bar{\mathbf{V}}_2] + 2[\bar{\mathbf{V}}_1\bar{\mathbf{V}}_2]\bar{\mathbf{V}}_1)/560 - 27[\bar{\mathbf{V}}_5, \bar{\mathbf{V}}_0]/4 \\
& + \big((42\bar{\mathbf{V}}_1 - 15\bar{\mathbf{V}}_2)[\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0] + [\bar{\mathbf{V}}_2, \bar{\mathbf{V}}_0](21\bar{\mathbf{V}}_1 - 27\bar{\mathbf{V}}_2)\big)/560 \\
& + \big((35\bar{\mathbf{V}}_2 + 65\bar{\mathbf{V}}_3)[\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0] + [\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_0](21\bar{\mathbf{V}}_2 + 40\bar{\mathbf{V}}_3)\big)/560
\end{aligned}
$$

$$
\begin{aligned}
& - \left(12\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_0] + 9[\bar{\mathbf{V}}_3,\bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1\right)/112 + \left[[-\bar{\mathbf{V}}_3 + 3\bar{\mathbf{V}}_4,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\right]/16 \\
& + \left[[[3\bar{\mathbf{V}}_2 - \bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0\right]/960 - [[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0]\bar{\mathbf{V}}_1/480 \\
& -[\bar{\mathbf{V}}_1[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0],\bar{\mathbf{V}}_0]/160 + \mathcal{O}(h^{10}), \\
\mathbf{C}_m^{(v')} \quad = \quad & \mathbf{0} + \mathcal{O}(h^{t(m)}) \ \text{ with } t(m) \geq 11,\ \forall m \geq 5.
\end{aligned}
$$

Note that the number of matrix multiplications actually performed can be reduced substantially by computing the commutators and matrix products which occur several times (as e.g. $[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_0]$, $[\bar{\mathbf{V}}_2,\bar{\mathbf{V}}_0]$, $[\bar{\mathbf{V}}_1,\bar{\mathbf{V}}_2]$, $\bar{\mathbf{V}}_2^2$, $\ldots$) only once. In addition, we can remark that a commutator $[\bar{\mathbf{V}}_i,\bar{\mathbf{V}}_j]$ or an anticommutator $\{\bar{\mathbf{V}}_i,\bar{\mathbf{V}}_j\}$ needs only one matrix multiplication since $\bar{\mathbf{V}}_j\bar{\mathbf{V}}_i = (\bar{\mathbf{V}}_i\bar{\mathbf{V}}_j)^{\mathrm{T}}$ for $\bar{\mathbf{V}}_i$ and $\bar{\mathbf{V}}_j$ symmetric matrices.

Appendix B

Maple Code

The Maple codes listed in this appendix are available for download at [2] .

B.1 The generation of the coefficients for the one-dimensional CPM$\{P, N\}$

Maple code which generates the expressions of the corrections for the CPM$\{P, N\}$ methods of chapter 3.

```
restart;
# include package for the generation of orthogonal polynomials:
with(orthopoly):
# shifted Legendre polynomials:
Ps:=(n,x)->simplify(P(n,2*x-1)):

# a pruning procedure:
REDUCE:= proc(a,P)
 local operand, tmp, reduced, i:
 reduced:=0;
 tmp:=simplify(rem(rem(convert(a,’polynom’),deltaˆ(P),delta),
      hˆ(P),h)):
 for i from 1 to nops(tmp) do
   operand:=op(i,tmp);
   if degree(operand,{delta,h}) < (P) then
     reduced:=simplify(reduced+operand);
   end if;
 od:
 RETURN(reduced);
end proc:

# Construct elements of the propagation matrix for CPM{18,16}
# Here the CPM{18,16} method is constructed:
# (by changing Hmax and N other CPM{P,N} methods can be
# constructed)
Hmax:=18:
N:=16:
NumIt:=floor(2*N/3) + 1:

# DV(delta) = V^N(X+delta) - Vc[0] = perturbation
DV:=x->sum(Vc[n]*h^n*Ps(n,x/h),n=1..N):

# construct u1(delta) :
C[1,0]:=delta->integrate(DV(x),x=0..delta)/2:
# m from 1 to Hmax/2+1 suffices because C[i,m] and C’[i,m] are
# multiplied by delta^(2m+1) and terms with degree(delta) > Hmax
# are ignored later.
for m from 1 to Hmax/2+1 do
  C[1,m]:=unapply(simplify( -1/2/delta^m*int(delta1^(m-1)*
          diff(C[1,m-1](delta1),delta1$2),delta1=0..delta)),
          delta):
od:
u:=unapply(xi(delta)+sum(C[1,k](delta)*delta^(2*k+1)*eta[k],
   k=0..(Hmax/2+1)),delta):
up:=unapply(Z*eta[0]+C[1,0](delta)*xi+
    sum((simplify(diff(C[1,k](delta),delta$1)+delta*
    C[1,k+1](delta)))*delta^(2*k+1)*eta[k],k=0..(Hmax/2)),delta):

# calculate remaining corrections u_i(delta) :
for i from 2 to NumIt do
# --> REDUCE-method avoids calculating terms which will be
# ignored later.
  for m from 0 to Hmax/2 do
    R[i,m]:=unapply(REDUCE(expand(DV(delta)*C[i-1,m](delta)),
            Hmax-2*m+3),delta):
  od:
  C[i,0]:=delta->0:
  for m from 1 to Hmax/2+1 do
    C[i,m]:=unapply(simplify(1/2/delta^m*int(delta1^(m-1)*
            (R[i,m-1](delta1)-diff(expand(C[i,m-1](delta1)),
            delta1$2)),delta1=0..delta)),delta):
  od:
  u:=unapply(u(delta)+sum(C[i,k](delta)*delta^(2*k+1)*eta[k],
     k=0..(Hmax/2+1)),delta):
  up:=unapply(up(delta)+sum((diff(C[i,k](delta),delta)+
      delta*C[i,k+1](delta))*delta^(2*k+1)*eta[k],k=0..(Hmax/2)),
      delta):
end:
```

```
# construct v1(delta) :
R[1,0]:=unapply(DV(delta),delta):
for m from 1 to Hmax/2 do
  R[1,m]:=delta->0:
od:
C[1,0]:=delta->0:
for m from 1 to Hmax/2+1 do
  C[1,m]:=unapply(simplify(1/2/delta^m*int(delta1^(m-1)*
          (R[1,m-1](delta1)-diff(expand(C[1,m-1](delta1)),
          delta1$2)),delta1=0..delta)),delta):
od:
v:=unapply(delta*eta[0]+sum(C[1,k](delta)*delta^(2*k+1)*
   eta[k],k=0..(Hmax/2+1)),delta):
vp:=unapply(delta*xi+C[1,0](delta)*xi+
    sum((simplify(diff(C[1,k](delta),delta)+delta*C[1,k+1]
    (delta)))*delta^(2*k+1)*eta[k], k=0..(Hmax/2)),delta):

# calculate remaining corrections v_i(delta) :
for i from 2 to NumIt do
  for m from 0 to Hmax/2 do
    R[i,m]:=unapply(REDUCE(expand(DV(delta)*C[i-1,m](delta)),
            Hmax-2*m+3),delta):
  od:
  C[i,0]:=delta->0:
  for m from 1 to Hmax/2+1 do
    C[i,m]:=unapply(simplify(1/2/delta^m*int(delta1^(m-1)*
            (R[i,m-1](delta1)-diff(expand(C[i,m-1](delta1)),
            delta1$2)),delta1=0..delta)),delta):
  od:
  v:=unapply(v(delta)+sum(C[i,k](delta)*delta^(2*k+1)*eta[k],
     k=0..(Hmax/2+1)),delta):
  vp:=unapply(vp(delta)+sum((diff(C[i,k](delta),delta$1)+
      delta*C[i,k+1](delta))*delta^(2*k+1)*eta[k],
      k=0..(Hmax/2)),delta):
end:

# delta = h
# throw away terms with degree(h) > Hmax
u_ser:=convert(series(u(h),h,Hmax+1),polynom):
up_ser:=convert(series(up(h),h,Hmax+1),polynom):
v_ser:=convert(series(v(h),h,Hmax+1),polynom):
vp_ser:=convert(series(vp(h),h,Hmax+1),polynom):

# Vb[m] = {\bar V}_m
for m from 0 to N do
  Vc[m]:=Vb[m]/h^(m+2)
```

```
od:

ord:=[seq(Vb[i],i=1..N)]:
for n from 0 to NumIt do
  Cu[n]:=sort(simplify(coeff(u_ser,eta[n],1)),ord);
  Cup[n]:=sort(simplify(coeff(up_ser,eta[n],1)),ord);
  Cv[n]:=sort(simplify(coeff(v_ser,eta[n],1)),ord);
  Cvp[n]:=sort(simplify(coeff(vp_ser,eta[n],1)),ord);
od;
```

B.2 The generation of the corrections of the LPM[4,2] method

B.2.1 The analytic expressions for the first and second order correction

Maple code which generates the expressions of the first and second order correction for the LPM[4,1] and LPM[4,2] method (see 4.2.1).

```
restart;
with(orthopoly):                  # includes the orthopoly package
Ps:=(n,x)->simplify(P(n,2*x-1)): # shifted Legendre polynomials
N:=4:                             # N=4 (up to V_4)
DV:=unapply(sum(V[n]*h^n*Ps(n,x/h),n=2..N),x): # perturbation
```

Calculation of the first order correction:

```
# Using the procedure described in sections 3.3 and 3.5 from
# L. Gr. Ixaru, Numerical Methods for Differential Equations
# and Applications, Reidel (1984).

# We assume that u1(delta) is of the form :
u1:=unapply(a1(delta)*u0(delta)+b1(delta)*v0(delta)+c1(delta)*
    u0p(delta)+d1(delta)*v0p(delta),delta):
# (Note = Q=F[0]/F[1])
# where a1, b1, c1 and d1 satisfy the system :
eq1:=diff(a1(delta),delta$2)+2*diff(c1(delta),delta)*
     (Q*F[1]+F[1]*delta)+c1(delta)*F[1]=DV(delta):
eq2:=diff(b1(delta),delta$2)+2*diff(d1(delta),delta)*
     (Q*F[1]+F[1]*delta)+d1(delta)*F[1]=0:
eq3:=diff(c1(delta),delta$2)+2*diff(a1(delta),delta)=0:
eq4:=diff(d1(delta),delta$2)+2*diff(b1(delta),delta)=0:
# with initial conditions :
ini1:=a1(0)+d1(0)=0:
ini2:=D(a1)(0)+c1(0)*Q*F[1]+b1(0)+D(d1)(0)=0:
# solution of the system :
c1:=unapply(rhs(simplify(dsolve(subs(diff(a1(delta),‘$‘(delta,2))=
    -diff(c1(delta),‘$‘(delta,3))/2,eq1),c1(delta)),{_C1=0,_C2=0,
    _C3=0,_C4=0})),delta):
d1:=unapply(rhs(simplify(dsolve(subs(diff(b1(delta),‘$‘(delta,2))=
    -diff(d1(delta),‘$‘(delta,3))/2,eq2),d1(delta)),{_C1=0,_C2=0,
    _C3=0,_C4=0})),delta):
a1:=unapply(rhs(dsolve({eq3,ini1},a1(delta))),delta):
b1:=unapply(rhs(dsolve({eq4,ini2},b1(delta))),delta):
# the expression of the first derivative of u1(delta) w.r.t. delta:
u1p:=unapply(((diff(a1(delta),delta)+c1(delta)
     *(Q*F[1]+F[1]*delta))*u0(delta)+(diff(b1(delta),delta)
     +d1(delta)*(Q*F[1]+F[1]*delta))*v0(delta)+(a1(delta)
     +diff(c1(delta),delta))*u0p(delta)+(b1(delta)+diff(d1(delta),
     delta))*v0p(delta)),delta):

# We assume that v1(delta) is of the form :
v1:=unapply(e1(delta)*u0(delta)+f1(delta)*v0(delta)+g1(delta)*
    u0p(delta)+h1(delta)*v0p(delta),delta):
# where e1, f1, g1 and h1 satisfy the system :
eq1:=diff(e1(delta),delta$2)+2*diff(g1(delta),delta)*
     (Q*F[1]+F[1]*delta)+g1(delta)*F[1]=0:
eq2:=diff(f1(delta),delta$2)+2*diff(h1(delta),delta)*
     (Q*F[1]+F[1]*delta)+h1(delta)*F[1]=DV(delta):
eq3:=diff(g1(delta),delta$2)+2*diff(e1(delta),delta)=0:
eq4:=diff(h1(delta),delta$2)+2*diff(f1(delta),delta)=0:
# with initial conditions :
ini1:=e1(0)+h1(0)=0:
ini2:=D(e1)(0)+g1(0)*Q*F[1]+f1(0)+D(h1)(0)=0:
# solution of the system :
g1:=unapply(rhs(simplify(dsolve(subs(diff(e1(delta),‘$‘(delta,2))=
    -diff(g1(delta),‘$‘(delta,3))/2,eq1),g1(delta)),{_C1=0,_C2=0,
    _C3=0,_C4=0})),delta):
h1:=unapply(rhs(simplify(dsolve(subs(diff(f1(delta),‘$‘(delta,2))=
    -diff(h1(delta),‘$‘(delta,3))/2,eq2),h1(delta)),{_C1=0,_C2=0,
    _C3=0,_C4=0})),delta):
e1:=unapply(rhs(dsolve({eq3,ini1},e1(delta))),delta):
f1:=unapply(rhs(dsolve({eq4,ini2},f1(delta))),delta):
# expression of the first derivative of v1(delta)
v1p:=unapply(((diff(e1(delta),delta)+g1(delta)
     *(F[1]*Q+F[1]*delta))*u0(delta)+(diff(f1(delta),
     delta)+h1(delta)*(F[1]*Q+F[1]*delta))*v0(delta)
     +(e1(delta)+diff(g1(delta),delta))*u0p(delta)
     +(f1(delta)+diff(h1(delta),delta))*v0p(delta)),delta):

# These commands print out the expressions of the first
# order correction:
# first order correction for u (with Q=F[0]/F[1]) :
collect(simplify(u1(h)),[u0(h),v0(h),u0p(h),v0p(h),Q]);
```

```
# first order correction for v:
collect(simplify(v1(h)),[u0(h),v0(h),u0p(h),v0p(h),Q]);
# first order correction for u’:
collect(simplify(u1p(h)),[u0(h),v0(h),u0p(h),v0p(h),Q]);
# first order correction for v’:
collect(simplify(v1p(h)),[u0(h),v0(h),u0p(h),v0p(h),Q]);
```

Calculation of the second order correction:

```
# The same procedure as for the first order correction is used.
u2:=unapply(a2(delta)*u0(delta)+b2(delta)*v0(delta)+c2(delta)*
    u0p(delta)+d2(delta)*v0p(delta),delta):
eq1:=diff(a2(delta),delta$2)+2*diff(c2(delta),delta)*
     (Q*F[1]+F[1]*delta)+c2(delta)*F[1]=DV(delta)*a1(delta):
eq2:=diff(b2(delta),delta$2)+2*diff(d2(delta),delta)*
     (Q*F[1]+F[1]*delta)+d2(delta)*F[1]=DV(delta)*b1(delta):
eq3:=diff(c2(delta),delta$2)+2*diff(a2(delta),delta)=
     DV(delta)*c1(delta):
eq4:=diff(d2(delta),delta$2)+2*diff(b2(delta),delta)=
     DV(delta)*d1(delta):
ini1:=a2(0)+d2(0)=0:
ini2:=D(a2)(0)+c2(0)*Q*F[1]+b2(0)+D(d2)(0)=0:
eq3s:=(diff(eq3,delta)-(diff(c2(delta),‘$‘(delta,3))=
      diff(c2(delta),‘$‘(delta,3))))/2:
c2:=unapply(rhs(subs({_C1=0,_C2=0,_C3=0,_C4=0},dsolve(subs(eq3s,
    eq1),c2(delta)))),delta):
eq4s:=(diff(eq4,delta)-(diff(d2(delta),‘$‘(delta,3))=
      diff(d2(delta),‘$‘(delta,3))))/2:
d2:=unapply(rhs(subs({_C1=0,_C2=0,_C3=0,_C4=0},dsolve(subs(eq4s,
    eq2),d2(delta)):)),delta):
a2:=unapply(rhs(value(dsolve({eq3,ini1},a2(delta)))),delta):
b2:=unapply(rhs(value(dsolve({eq4,ini2},b2(delta)))),delta):
u2p:=unapply((diff(a2(delta),delta)+c2(delta)*(Q*F[1]+F[1]*delta))
    *u0(delta)+(diff(b2(delta),delta)+d2(delta)*(Q*F[1]
    +F[1]*delta))*v0(delta)+(a2(delta)+diff(c2(delta),delta))
    *u0p(delta)+(b2(delta)+diff(d2(delta),delta))*v0p(delta),
    delta):

v2:=unapply(e2(delta)*u0(delta)+f2(delta)*v0(delta)+g2(delta)*
    u0p(delta)+h2(delta)*v0p(delta),delta):
eq1:=diff(e2(delta),delta$2)+2*diff(g2(delta),delta)*
     (Q*F[1]+F[1]*delta)+g2(delta)*F[1]=DV(delta)*e1(delta):
eq2:=diff(f2(delta),delta$2)+2*diff(h2(delta),delta)*
     (Q*F[1]+F[1]*delta)+h2(delta)*F[1]=DV(delta)*f1(delta):
eq3:=diff(g2(delta),delta$2)+2*diff(e2(delta),delta)=
     DV(delta)*g1(delta):
eq4:=diff(h2(delta),delta$2)+2*diff(f2(delta),delta)=
     DV(delta)*h1(delta):
ini1:=e2(0)+h2(0)=0:
```

```
ini2:=D(e2)(0)+g2(0)*Q*F[1]+f2(0)+D(h2)(0)=0:
eq3s:=(diff(eq3,delta)-(diff(g2(delta),‘$‘(delta,3))=
      diff(g2(delta),‘$‘(delta,3))))/2:
g2:=unapply(rhs(subs({_C1=0,_C2=0,_C3=0,_C4=0},dsolve(subs(eq3s,
    eq1),g2(delta)))),delta):
eq4s:=(diff(eq4,delta)-(diff(h2(delta),‘$‘(delta,3))=
      diff(h2(delta),‘$‘(delta,3))))/2:
h2:=unapply(rhs(subs({_C1=0,_C2=0,_C3=0,_C4=0},dsolve(subs(eq4s,
    eq2),h2(delta)))),delta):
e2:=unapply(rhs(value(dsolve({eq3,ini1},e2(delta)))),delta):
f2:=unapply(rhs(value(dsolve({eq4,ini2},f2(delta)))),delta):
v2p:=unapply((diff(e2(delta),delta)+g2(delta)*(Q*F[1]+F[1]*delta))
     *u0(delta)+(diff(f2(delta),delta)+h2(delta)*(Q*F[1]+F[1]
     *delta))*v0(delta)+(e2(delta)+diff(g2(delta),delta))
     *u0p(delta)+(f2(delta)+diff(h2(delta),delta))*v0p(delta),
     delta):

# These commands print out the expressions of the second
# order correction:
# second order correction for u:
collect(expand(u2(h)),[u0(h),v0(h),u0p(h),v0p(h),Q]);
# second order correction for u’:
collect(expand(u2p(h)),[u0(h),v0(h),u0p(h),v0p(h),Q]);
# second order correction for v:
collect(expand(v2(h)),[u0(h),v0(h),u0p(h),v0p(h),Q]);
# second order correction for v’:
collect(expand(v2p(h)),[u0(h),v0(h),u0p(h),v0p(h),Q]);
```

B.2.2 The asymptotic forms for the zeroth, first and second order correction

Maple code which should be appended to the previous Maple code in order to generate the asymptotic forms of the zeroth, first and second order correction.

```
# asymptotic expansions of the Airy functions
# (see Eqs. 10.4.58-10.4.67 in M. Abramowitz and I. Stegun,
# Handbook of Mathematical Functions.):
c[0]:=1:
d[0]:=1:
M:=7:
for k from 1 to 2*M+1 do
  c[k]:=(6*k-5)*(6*k-1)*c[k-1]/(72*k):
  d[k]:=(-(6*k+1)/(6*k-1))*c[k]:
od:
zeta:=(2/3)*Zm^(3/2):
# asymptotic expansions of the Airy functions for Z large negative :
nAi:=unapply(Pi^(-1/2)*Zm^(-1/4)*(sin(zeta+Pi/4)*sum((-1)^i*c[2*i]
     *zeta^(-2*i),i=0..M)-cos(zeta+Pi/4)*sum((-1)^i*c[2*i+1]*zeta
     ^(-2*i-1),i=0..M)),Zm):
nAip:=unapply(-Pi^(-1/2)*Zm^(1/4)*(cos(zeta+Pi/4)*sum((-1)^i*d[2*i]
      *zeta^(-2*i),i=0..M)+sin(zeta+Pi/4)*sum((-1)^i*d[2*i+1]*zeta
      ^(-2*i-1),i=0..M)),Zm):
nBi:=unapply(Pi^(-1/2)*Zm^(-1/4)*(cos(zeta+Pi/4)*sum((-1)^i*c[2*i]
     *zeta^(-2*i),i=0..M)+sin(zeta+Pi/4)*sum((-1)^i*c[2*i+1]*zeta
     ^(-2*i-1),i=0..M)),Zm):
nBip:=unapply(Pi^(-1/2)*Zm^(1/4)*(sin(zeta+Pi/4)*sum((-1)^i*d[2*i]
      *zeta^(-2*i),i=0..M)-cos(zeta+Pi/4)*sum((-1)^i*d[2*i+1]*zeta
      ^(-2*i-1),i=0..M)),Zm):
# asymptotic expansions of the Airy functions for Z large positive:
pAi:=unapply((1/2)*Pi^(-1/2)*Zm^(-1/4)*exp(-zeta)*sum((-1)^i*c[i]
     *zeta^(-i),i=0..2*M),Zm):
pAip:=unapply(-(1/2)*Pi^(-1/2)*Zm^(1/4)*exp(-zeta)*sum((-1)^i*d[i]
      *zeta^(-i),i=0..2*M),Zm):
pBi:=unapply(Pi^(-1/2)*Zm^(-1/4)*exp(zeta)*sum(c[i]*zeta^(-i),
     i=0..2*M),Zm):
pBip:=unapply(Pi^(-1/2)*Zm^(1/4)*exp(zeta)*sum(d[i]*zeta^(-i),
      i=0..2*M),Zm):
```

Calculation of the asymptotic expansion of the zeroth order correction :

```
Z:=F[1]^(1/3)*h+F[0]/(F[1])^(2/3):      # Z(h) = alfa*(h+beta)
Z0:=F[0]/(F[1])^(2/3):                  # Z0 = alfa*beta
# zeroth order propagators must be calculated up to F1^14 to
# obtain the first correction up to F1^4 and the second order
# correction up to F1^3:
K:= 15:
# for Z and Z0 both large negative:
nu0:=convert(simplify(series(simplify(combine(Pi*(nAi(-Z)*nBip(-Z0)
     -nBi(-Z)*nAip(-Z0)),trig)),F[1],K)),polynom):
nv0:=convert(simplify(series(simplify(combine(Pi*(nBi(-Z)*nAi(-Z0)
     -nAi(-Z)*nBi(-Z0))/F[1]^(1/3),trig)),F[1],K)),polynom):
nu0p:=convert(simplify(series(simplify(combine(F[1]^(1/3)*Pi*
      (nAip(-Z)*nBip(-Z0)-nBip(-Z)*nAip(-Z0)),trig)),F[1],K)),
      polynom):
nv0p:=convert(simplify(series(simplify(combine(Pi*(nBip(-Z)*
      nAi(-Z0)-nAip(-Z)*nBi(-Z0)),trig)),F[1],K)),polynom):
# for Z and Z0 both large positive:
pu0:=convert(simplify(convert(series(combine(Pi*(pAi(Z)*pBip(Z0)-
     pBi(Z)*pAip(Z0))),F[1],K),trig)),polynom):
pv0:=convert(simplify(convert(series(combine(Pi*(pBi(Z)*pAi(Z0)-
     pAi(Z)*pBi(Z0))/F[1]^(1/3)),F[1],K),trig)),polynom):
pu0p:=convert(simplify(convert(series(combine(F[1]^(1/3)*Pi*
      (pAip(Z)*pBip(Z0)-pBip(Z)*pAip(Z0))),F[1],K),trig)),polynom):
pv0p:=convert(simplify(convert(series(combine(Pi*(pBip(Z)*pAi(Z0)
      -pAip(Z)*pBi(Z0))),F[1],K),trig)),polynom):
```

Calculation of the asymptotic expansion of the first order correction :

```
# Conversion of the first order correction (in the analytical
# form) into asymptotic form.
# First order correction in asymptotic form for Z and Z0 both
# large negative:
K:=5:   # up to F1^4
nu1:=collect(factor(convert(series(simplify(subs({u0(h)=nu0,
     v0(h)=nv0,u0p(h)=nu0p,v0p(h)=nv0p},subs({Q=F[0]/F[1]},
     u1(h)))),F[1],K),polynom)),F[1]);
nv1:=collect(factor(convert(series(simplify(subs({u0(h)=nu0,
     v0(h)=nv0,u0p(h)=nu0p,v0p(h)=nv0p},subs({Q=F[0]/F[1]},
     v1(h)))),F[1],K),polynom)),F[1]);
nu1p:=collect(factor(convert(series(simplify(subs({u0(h)=nu0,
      v0(h)=nv0,u0p(h)=nu0p,v0p(h)=nv0p},subs({Q=F[0]/F[1]},
      u1p(h)))),F[1],K),polynom)),F[1]);
nv1p:=collect(factor(convert(series(simplify(subs({u0(h)=nu0,
      v0(h)=nv0,u0p(h)=nu0p,v0p(h)=nv0p},subs({Q=F[0]/F[1]},
      v1p(h)))),F[1],K),polynom)),F[1]);
# First order correction in asymptotic form for Z and Z0 both
# large positive:
pu1:=collect(factor(convert(series(simplify(subs({u0(h)=pu0,
     v0(h)=pv0,u0p(h)=pu0p,v0p(h)=pv0p},subs({Q=F[0]/F[1]},
     u1(h)))),F[1],K),polynom)),F[1]);
pv1:=collect(factor(convert(series(simplify(subs({u0(h)=pu0,
     v0(h)=pv0,u0p(h)=pu0p,v0p(h)=pv0p},subs({Q=F[0]/F[1]},
     v1(h)))),F[1],K),polynom)),F[1]);
pu1p:=collect(factor(convert(series(simplify(subs({u0(h)=pu0,
      v0(h)=pv0,u0p(h)=pu0p,v0p(h)=pv0p},subs({Q=F[0]/F[1]},
      u1p(h)))),F[1],K),polynom)),F[1]);
pv1p:=collect(factor(convert(series(simplify(subs({u0(h)=pu0,
      v0(h)=pv0,u0p(h)=pu0p,v0p(h)=pv0p},subs({Q=F[0]/F[1]},
      v1p(h)))),F[1],K),polynom)),F[1]);
```

Calculation of the asymptotic expansion of the second order correction :

```
# Conversion of the second order correction (in the analytical
# form) into asymptotic form.
# Second order correction in asymptotic form for Z and Z0 both
# large negative:
K:=4:  #up to F1^3
nu2:=collect(factor(convert(series(simplify(subs({u0(h)=nu0,
     v0(h)=nv0,u0p(h)=nu0p,v0p(h)=nv0p},subs({Q=F[0]/F[1]},
     u2(h)))),F[1],K),polynom)),F[1]);
nv2:=collect(factor(convert(series(simplify(subs({u0(h)=nu0,
     v0(h)=nv0,u0p(h)=nu0p,v0p(h)=nv0p},subs({Q=F[0]/F[1]},
     v2(h)))),F[1],K),polynom)),F[1]);
nu2p:=collect(factor(convert(series(simplify(subs({u0(h)=nu0,
      v0(h)=nv0,u0p(h)=nu0p,v0p(h)=nv0p},subs({Q=F[0]/F[1]},
      u2p(h)))),F[1],K),polynom)),F[1]);
```

```
nv2p:=collect(factor(convert(series(simplify(subs({u0(h)=nu0,
      v0(h)=nv0,u0p(h)=nu0p,v0p(h)=nv0p},subs({Q=F[0]/F[1]},
      v2p(h)))),F[1],K),polynom)),F[1]);
# Second order correction in asymptotic form for Z and Z0 both
# large positive:
pu2:=collect(factor(convert(series(simplify(subs({u0(h)=pu0,
     v0(h)=pv0,u0p(h)=pu0p,v0p(h)=pv0p},subs({Q=F[0]/F[1]},
     u2(h)))),F[1],K),polynom)),[F[1],F[0]]);
pv2:=collect(factor(convert(series(simplify(subs({u0(h)=pu0,
     v0(h)=pv0,u0p(h)=pu0p,v0p(h)=pv0p},subs({Q=F[0]/F[1]},
     v2(h)))),F[1],K),polynom)),[F[1],F[0]]);
pu2p:=collect(factor(convert(series(simplify(subs({u0(h)=pu0,
      v0(h)=pv0,u0p(h)=pu0p,v0p(h)=pv0p},subs({Q=F[0]/F[1]},
      u2p(h)))),F[1],K),polynom)),[F[1],F[0]]);
pv2p:=collect(factor(convert(series(simplify(subs({u0(h)=pu0,
      v0(h)=pv0,u0p(h)=pu0p,v0p(h)=pv0p},subs({Q=F[0]/F[1]},
      v2p(h)))),F[1],K),polynom)),[F[1],F[0]]);
```

B.3 The generation of the coefficients for the generalized CPM$\{P,N\}$

1. Construction of the elements of the $\mathrm{C}_m^{(u)}$ and $\mathrm{C}_m^{(u')}$ matrices

```
restart;
# built-in package for the generation of orthogonal polynomials:
with(orthopoly):
# shifted Legendre polynomials (n = degree):
Ps:=(n,x)->simplify(P(n,2*x-1)):

# By changing the parameters Mmax and Hmax, coefficients for other
# CPM{P,N} methods can be calculated :
Mmax:=8:          # V_0,V_1,...,V_8
Hmax:=10:         # CPM{10,8}=CPM{Hmax,Mmax}
NumberCorr:=5;    # 5 corrections is sufficient for CPM{10,8}

# The parameter N should be changed to obtain the coefficients
# for other NxN problems:
N:=2:             # 2x2 matrix


#a pruning procedure:
REDUCE:= proc(a,P)
local c, tmp,red:
red:=0;
tmp:=simplify(rem(rem(convert(a,'polynom'),delta^(P),delta),
     h^(P),h)):
for i from 1 to nops(tmp) do
 c:=op(i,tmp);
 if degree(c,{delta,h}) < (P) then
    red:=simplify(red+c);
 end if;
od:
RETURN(red);
end proc:

#The perturbation :
#Note: VD’s are symmetric
DV:=unapply(‘if‘(i<j,sum(VD[i,j,z]*h^z*Ps(z,x/h),z=1..Mmax),
    sum(VD[j,i,z]*h^z*Ps(z,x/h),z=1..Mmax)),i,j,x);

#First correction :
i:=1:
for II from 1 to N do
 for JJ from 1 to N do
  C[II,JJ,0,i]:=unapply(integrate(DV(II,JJ,x),x=0..delta)/2,
    delta):
  for m from 1 to (Hmax/2+1) do
   C[II,JJ,m,i]:=unapply(simplify(-1/2/delta^m*int(delta1^(m-1)
    *diff(C[II,JJ,m-1,i](delta1),delta1$2)+C[II,JJ,m-1,i](delta1)
    *VD[JJ,JJ,0]-VD[II,II,0]*C[II,JJ,m-1,i](delta1)),delta1=0..
    delta)),delta);
  od;
  u1[II,JJ]:=unapply(sum(C[II,JJ,w,i](delta)*delta^(2*w+1)*
   eta[w,JJ],w=0..(Hmax/2+1)),delta):
  up1[II,JJ]:=unapply(C[II,JJ,0,i](delta)*eta[-1,JJ]+
   sum((simplify(diff(C[II,JJ,k,i](delta),delta$1)+delta*
   C[II,JJ,k+1,i](delta)))*delta^(2*k+1)*eta[k,JJ],k=0..(Hmax/2)),
   delta):
 od;
od;

#Remaining corrections :
for i from 2 to NumberCorr do
 for II from 1 to N do
  for JJ from 1 to N do
   for m from 0 to Hmax/2 do
    R[II,JJ,m,i]:=unapply(REDUCE(expand(sum(DV(II,k,delta)*
      C[k,JJ,m,i-1](delta),k=1..N)),Hmax-2*m+1),delta):
   od:
   C[II,JJ,0,i]:=unapply(0,delta):
   for m from 1 to (Hmax/2+1) do
    C[II,JJ,m,i]:=unapply(simplify(1/2/delta^m*int(delta1^(m-1)*
     (R[II,JJ,m-1,i](delta1)-diff(C[II,JJ,m-1,i](delta1),delta1$2)
     -C[II,JJ,m-1,i](delta1)*VD[JJ,JJ,0]+VD[II,II,0]*C[II,JJ,m-1,i]
     (delta1)),delta1=0..delta)),delta);
   od:
   u1[II,JJ]:=unapply(u1[II,JJ](delta)+sum(C[II,JJ,w,i](delta)*
    delta^(2*w+1)*eta[w,JJ],w=0..(Hmax/2+1)),delta):
   up1[II,JJ]:=unapply(up1[II,JJ](delta)+C[II,JJ,0,i](delta)*
    eta[-1,JJ]+sum((simplify(diff(C[II,JJ,k,i](delta),delta$1)+
    delta*C[II,JJ,k+1,i](delta)))*delta^(2*k+1)*eta[k,JJ],k=0..
    (Hmax/2)),delta):
  od;
 od;
od;

#Construction of the C-matrices :
for II from 1 to N do
 for JJ from 1 to N do
  #delta = h
  #throw away terms with degree(h)>Hmax
  u[II,JJ]:=simplify(convert(series(u1[II,JJ](h),h,Hmax+1),
   polynom)):
  up[II,JJ]:=convert(series(up1[II,JJ](h),h,Hmax+1),polynom):
  for m from -1 to (Hmax/2+1) do
   Cu[II,JJ,m]:=simplify(coeff(u[II,JJ],eta[m,JJ],1));
  od;
  for m from -1 to (Hmax/2+1) do
   Cup[II,JJ,m]:=simplify(coeff(up[II,JJ],eta[m,JJ],1));
  od;
 od;
od;

t:=simplify(Cu[1,2,1]); #shows C^{(u)}_1 for i=1,j=2
```

2. Construction of the elements of the $\mathrm{C}_m^{(v)}$ and $\mathrm{C}_m^{(v')}$ matrices

```
restart;
with(orthopoly):
Ps:=(n,x)->simplify(P(n,2*x-1)):  # shifted Legendre polynomials
Mmax:=8:
Hmax:=10:  #CPM{10,8}
N:=2:   #2x2 matrix
NumberCorr:=5;

#pruning procedure :
REDUCE:= proc(a,P)
local c, tmp,red:
red:=0;
tmp:=simplify(rem(rem(convert(a,’polynom’),delta^(P),delta),
  h^(P),h)):
for i from 1 to nops(tmp) do
  c:=op(i,tmp);
  if degree(c,{delta,h}) < (P) then
    red:=simplify(red+c);
  end if;
od:
RETURN(red);
end proc:

#The perturbation :
DV:=unapply(‘if‘(i<j,sum(VD[i,j,z]*h^z*Ps(z,x/h),z=1..Mmax),
    sum(VD[j,i,z]*h^z*Ps(z,x/h),z=1..Mmax)),i,j,x);

#First correction :
i:=1;
for II from 1 to N do
 for JJ from 1 to N do
  R[II,JJ,0,i]:=unapply(DV(II,JJ,delta),delta):
  for m from 1 to Hmax/2 do
   R[II,JJ,m,i]:=unapply(0,delta):
  od:
  C[II,JJ,0,i]:=unapply(0,delta):
  for m from 1 to (Hmax/2+1) do
   C[II,JJ,m,i]:=unapply(simplify(1/2/delta^m*int(delta1^(m-1)*
    (R[II,JJ,m-1,i](delta1)-diff(C[II,JJ,m-1,i](delta1),delta1$2)-
    C[II,JJ,m-1,i](delta1)*VD[JJ,JJ,0]+VD[II,II,0]*C[II,JJ,m-1,i]
    (delta1)),delta1=0..delta)),delta);
  od:
  v1[II,JJ]:=unapply(sum(C[II,JJ,w,i](delta)*delta^(2*w+1)*
   eta[w,JJ],w=0..(Hmax/2+1)),delta):
  vp1[II,JJ]:=unapply(C[II,JJ,0,i](delta)*eta[-1,JJ]+
   sum((simplify(diff(C[II,JJ,k,i](delta),delta$1)+
   delta*C[II,JJ,k+1,i](delta)))*delta^(2*k+1)*eta[k,JJ],
   k=0..(Hmax/2)),delta):
 od;
od;

#Remaining corrections :
for i from 2 to NumberCorr do
 for II from 1 to N do
  for JJ from 1 to N do
   for m from 0 to Hmax/2 do
    R[II,JJ,m,i]:=unapply(REDUCE(expand(sum(DV(II,k,delta)*
     C[k,JJ,m,i-1](delta),k=1..N)),Hmax-2*m+1),delta):
   od:
   C[II,JJ,0,i]:=unapply(0,delta):
```

```
   for m from 1 to (Hmax/2+1) do
    C[II,JJ,m,i]:=unapply(simplify(1/2/delta^m*int(delta1^(m-1)*
     (R[II,JJ,m-1,i](delta1)-diff(C[II,JJ,m-1,i](delta1),delta1$2)
     -C[II,JJ,m-1,i](delta1)*VD[JJ,JJ,0]+VD[II,II,0]*C[II,JJ,m-1,i]
     (delta1)),delta1=0..delta)),delta);
   od:
   v1[II,JJ]:=unapply(v1[II,JJ](delta)+sum(C[II,JJ,w,i](delta)*
    delta^(2*w+1)*eta[w,JJ],w=0..(Hmax/2+1)),delta):
   vp1[II,JJ]:=unapply(vp1[II,JJ](delta)+C[II,JJ,0,i](delta)*
    eta[-1,JJ]+sum((simplify(diff(C[II,JJ,k,i](delta),delta$1)+
    delta*C[II,JJ,k+1,i](delta)))*delta^(2*k+1)*eta[k,JJ],
    k=0..(Hmax/2)),delta):
  od;
 od;
od;

for II from 1 to N do
 for JJ from 1 to N do
  v[II,JJ]:=simplify(convert(series(v1[II,JJ](h),h,Hmax+1),
   polynom)):
  vp[II,JJ]:=convert(series(vp1[II,JJ](h),h,Hmax+1),polynom):
  for n from -1 to (Hmax/2+1) do
   Cv[II,JJ,n]:=simplify(coeff(v[II,JJ],eta[n,JJ],1));
  od;
  for n from -1 to (Hmax/2+1) do
   Cvp[II,JJ,n]:=simplify(coeff(vp[II,JJ],eta[n,JJ],1));
  od;
 od;
od;

t:=simplify(Cv[1,2,3]);  #shows C^{(v)}_3 for i=1,j=2
t:=simplify(Cvp[1,2,3]); #shows C^{(v’)}_3 for i=1,j=2
```

Appendix C

List of test problems

In this appendix we list the problems predefined in the MATSLISE GUI. This test set of problems collects some problems from the set used by Pruess and Fulton to test SLEDGE and many problems from SLTSTPAK [107], a test package for Sturm-Liouville solvers. The nth problem in the SLTSTPAK test set is referred to as SLTSTPAK#n.

C.1 Schrödinger problems

Regular Schrödinger problems

1. **Coffey-Evans equation**. (`Coffey_Evans.mat`). (SLTSTPAK #7). Reference: [104].
   $V(x) = -2\beta \cos 2x + \beta^2 \sin^2 2x$
   $a = -\pi/2$ Regular $y(a) = 0$
   $b = \pi/2$ Regular $y(b) = 0$
   As β increases there are very close eigenvalue triplets $\{E_2, E_3, E_4\}, \{E_6, E_7, E_8\}, \ldots$ with the other eigenvalues well separated.
   $\beta = 20 : E_0 = 0.0000000000000$ $E_1 = 77.9161956771440$ $E_3 = 151.4632236576586$
   $\beta = 30 : E_0 = 0.0000000000000$ $E_1 = 117.946307662070$ $E_3 = 231.6649293129610$
   $\beta = 50 : E_0 = 0.0000000000000$ $E_1 = 197.968726516507$ $E_3 = 391.80819148905.$

2. **Mathieu equation**. (`Mathieu.mat`). (SLTSTPAK #2).
   $V(x) = 2r\cos(2x)$ r parameter
   $a = 0$ Regular $y(a) = 0$
   $b = \pi$ Regular $y(b) = 0$
   $r = 1 : E_0 = -0.1102488169921$ $E_5 = 36.0142899106282$ $E_9 = 100.0050506751595.$

3. **Paine problem 1**. (`Paine1.mat`). Reference: [97].
   $V(x) = e^x$
   $a = 0$ Regular $y(a) = 0$
   $b = \pi$ Regular $y(b) = 0$
   $E_0 = 4.896669379968$ $E_1 = 10.04518989325$ $E_9 = 107.11667613827.$

4. **Paine problem 2**. (`Paine2.mat`). (SLTSTPAK #1). Reference: [97].
   $V(x) = \dfrac{1}{(x+0.1)^2}$
   $a = 0$ Regular $y(a) = 0$
   $b = \pi$ Regular $y(b) = 0$
   $E_0 = 1.519865821099 \quad E_1 = 4.943309822145 \quad E_9 = 102.424988398249$.

5. **Pruess-Fulton problem 133**. (`Pruess_Fulton133.mat`). (SLTSTPAK #11).
   Reference: 133th problem in the Pruess-Fulton test set [102].
   $V(x) = \ln x$
   $a = 0$ Regular $y(a) = 0$
   $b = 4$ Regular $y(b) = 0$
   $E_0 = 1.1248168097 \quad E_{24} = 385.92821596$.

6. **Truncated Gelfand-Levitan**. (`Gelfand_Levitan_truncated.mat`). (SLTSTPAK #6).
   Reference: [39].
   $V(x) = 2(T \sin 2x + \cos^4 x)/T^2, \quad T = 1 + x/2 + \sin(2x)/4$
   $a = 0$ Regular $y(a) - y'(a) = 0$
   $b = 100$ Regular $y(b) = 0$
   Non-uniform oscillations of decreasing size in $V(x)$.

7. **Version of Mathieu equation**. (`Mathieu_version.mat`). (SLTSTPAK #5).
   $V(x) = c\cos(x)$ c parameter
   $a = 0$ Regular $y(a) = 0$
   $b = 40$ Regular $y(b) = 0$
   The lower eigenvalues form clusters of 6; more and tighter clusters as c increases.

Infinite integration interval

1. **Airy equation**. (`Airy.mat`). (SLTSTPAK #27). Reference: [119] p.91.
   $V(x) = x$
   $a = 0$ Regular $y(a) = 0$
   $b = +\infty$ LPN
   Number of eigenvalues: ∞ continuous spectrum: none
   Eigenvalues are the zeros of Airy function $Ai(E) = (J_{1/3} + J_{-1/3})(\frac{2}{3}E^{1/3})$.
   $E_0 = 2.338107410459 \quad E_9 = 12.82877675287$.

2. **Anharmonic oscillator potential**. (`anharm_oscillator.mat`). Reference: [37].
   $V(x) = x^2 + \lambda x^2/(1+gx^2)$ λ, g parameters
   $a = -\infty$ LPN
   $b = +\infty$ LPN
   Number of eigenvalues: ∞ continuous spectrum: none
   $\lambda = 0.1,\ g = 0.1:$ $E_0 = 1.04317371304$
   $\lambda = 10.0,\ g = 10.0:$ $E_0 = 1.58002232739$.

3. **Bender-Orszag potential**. (`Bender_Orszag.mat`). (SLTSTPAK #14).
   Reference: [23] p. 28.
   $V(x) = -m(m+1)/\cosh^2 x$ m parameter
   $a = -\infty$ LPN/O

$b = +\infty$ LPN/O
Number of eigenvalues: Number of integers in range $0 \le k < m$
Continuous spectrum: $(0, \infty)$
$E_k = -(m-k)^2, \quad 0 \le k < m.$

4. **Biswas potential**. (`Biswas.mat`). Reference: [24, 36].
   $V(x) = \mu x^2 + \nu x^4$ μ, ν parameters
   $a = -\infty$ LPN
   $b = +\infty$ LPN
   Number of eigenvalues: ∞ continuous spectrum: none
   $\mu = 0.0, \ \nu = 1.0: \quad E_0 = 1.0603620905$
   $\mu = 1.0, \ \nu = 1.0: \quad E_0 = 1.3923516415.$
5. **Close-eigenvalues problem**. (`Close_eigenvalues.mat`). (SLTSTPAK #38).
   $V(x) = x^4 - 25x^2$
   Double well version of quartic anharmonic oscillator
   $a = -\infty$ LPN Trunc. BC.: $y(a) = 0$
   $b = +\infty$ LPN Trunc. BC.: $y(b) = 0$
   Number of eigenvalues: ∞ continuous spectrum: none
   $E_0 = -149.219456142 \quad E_1 = -149.219456142$
   Half-range reduction makes the problem more tractable.
6. **Harmonic oscillator**. (`Harmonic_oscillator.mat`). (SLTSTPAK #28).
   Reference: [119] p.1536.
   $V(x) = x^2$
   $a = -\infty$ LPN
   $b = +\infty$ LPN
   Number of eigenvalues: ∞ continuous spectrum: $(0, \infty)$
   $E_k = 2k+1, \ k = 0, 1, ...$
7. **Half-range anharmonic oscillator**. (`HR_anharm_oscillator.mat`). (SLTSTPAK #17).
   Reference: [84].
   $V(x) = x^\alpha$, $\alpha > 0$
   $a = 0$ Regular $y(a) = 0$
   $b = +\infty$ LPN
   Number of eigenvalues: ∞ continuous spectrum: $(0, \infty)$
   $\alpha = 2$: $E_k = 4k+3, \ k = 0, 1, 2, ...$ (alternate eigenvalues of harmonic oscillator)
   $\alpha = 3$: $E_0 = 3.4505626899 \quad E_{24} = 228.520881389$
   $\alpha = 4$: $E_0 = 3.7996730298 \quad E_{24} = 397.141326781$
   $\alpha = 5$: $E_0 = 4.0891593149 \quad E_{24} = 588.178249691.$
8. **Morse potential**. (`Morse1.mat`). (SLTSTPAK #35). Reference: [91].
   $V(x) = 9e^{-2x} - 18e^{-x}$
   $a = -\infty$ LPN
   $b = +\infty$ LPN/O
   Number of eigenvalues: 3 continuous spectrum: $(0, \infty)$
   $E_k = -0.25 - (3-k)(2-k), \ k = 0, 1, 2.$
9. **Morse potential**. (`Morse2.mat`). (SLTSTPAK #39). Reference: [84]
   $V(x) = 8000e^{-3x} - 16000e^{-3x/2}$
   $a = -\infty$ LPN

$b = +\infty$ LPN/O
Number of eigenvalues: 60 continuous spectrum: $(0, \infty)$
With this deep well, a large truncated interval seems to be needed to give good approximations to higher eigenvalues.
$E_0 = -7866.39842135 \quad E_{57} = -10.19345525 \quad E_{58} = -2.86529795.$

10. **Problem with ‘pseudo-eigenvalue’**. `Pryce60.mat`. (SLTSTPAK #60). Reference: [84].
$V(x) = 3(x-31)/(4(1+x)(4+x)^2)$
$a = 0$ Regular $5y(a) + 8y'(a) = 0$
$b = \infty$ LPN/O
Number of eigenvalues: 1 continuous spectrum: none
$E_0 = -1.185214105.$

11. **Quartic anharmonic oscillator**. (`Quartic_anharm_oscillator.mat`). (SLTSTPAK #37). Reference: [115].
$V(x) = x^4 + x^2$
$a = -\infty$ LPN
$b = +\infty$ LPN
Number of eigenvalues: ∞ continuous spectrum: none
$E_0 = 1.3923516415 \quad E_9 = 46.965009506.$

12. **The Razavy potential**. (`Razavy.mat`). Reference: [37].
$V(x) = 1/8m^2(\cosh(4x) - 1) - m(n+1)\cosh(2x)$ n, m parameters
$a = -\infty$ LPN
$b = +\infty$ LPN
Number of eigenvalues: ∞ continuous spectrum: none
$n = 1, m = 1 : E_0 = -2,\ E_1 = 0$
$n = 2, m = 1 : E_0 = -2(1+\sqrt{2}),\ E_1 = -4,\ E_2 = 2(\sqrt{2}-1)$
$n = 1, m = 10 : E_0 = -11,\ E_1 = 9$
$n = 2, m = 10 : E_0 = -2(1+\sqrt{101}),\ E_1 = -4,\ E_2 = 2(\sqrt{101}-1).$

13. **Symmetric double-well potential**. (`symm_double_well.mat`). Reference: [37].
$V(x) = x^6 - Bx^2$ B parameter
$a = -\infty$ LPN
$b = +\infty$ LPN
Number of eigenvalues: ∞ continuous spectrum: none
$B = 11$: known exact eigenvalues $=\{-8, 0, 8\}$
$B = 13$: known exact eigenvalues $=\{-11.3137085, 0, 11.3137085\}$
$B = 15$: known exact eigenvalues $=\{-15.07750851, -3.55931694, 3.55931694, 15.07750851\}$.

14. **Wicke-Harris problem**. `Wicke_Harris.mat`. (SLTSTPAK #40). Reference: [130].
$V(x) = 1250e^{-83.363(x-2.47826)^2} + 3906.25(1 - e^{2.3237-x})^2$
$a = 0$ Regular $y(a) = 0$
$b = +\infty$ LPN/O
Number of eigenvalues: 61 continuous spectrum: $(3906.25, \infty)$
$E_0 = 163.223887 \quad E_9 = 1277.5368406$
This has a spike at the bottom of the well.

C.2 Sturm-Liouville problems

1. **Bessel equation, order** $1/2$. (`Bessel.mat`). (SLTSTPAK #19). Reference: [119].
   $p(x) = x \quad q(x) = \alpha/x \quad w(x) = x$
   $\alpha = \nu^2, \nu = \frac{1}{2}$
   $a = 0$ LCN
   $b = 1$ Regular $y(b) = 0$
   $E_k = ((k+1)\pi)^2$, this is $-v'' = Ev$ transformed by $v = x^{1/2}u$.

2. **Collatz problem**. (`Collatz.mat`). Reference: [31].
   $p(x) = 1 \quad q(x) = 0 \quad w(x) = 3 + \cos(x)$
   $a = -\pi$ Regular $y(a) = 0$
   $b = +\pi$ Regular $y(b) = 0$
   $E_0 = 0.071250472$.

3. **Infinite interval problem**. (`Pryce33.mat`). (SLTSTPAK #33). Reference: [20].
   $p(x) = 1 \quad q(x) = -7x^2 + 0.5x^3 + x^4 \quad w(x) = 0.5$
   $a = -\infty$ LPN
   $b = \infty$ LPN
   Number of eigenvalues: ∞ continuous spectrum: none
   $E_0 = -24.5175977072 \quad E_5 = 8.10470769427$.

4. **Klotter problem**. (`Klotter.mat`). (SLTSTPAK #3). Reference: [68] p.12.
   $p(x) = 1 \quad q(x) = 3/(4x^2) \quad w(x) = 64\pi^2/(9x^6)$
   $a = 8/7$ Regular $y(a) = 0$
   $b = 8$ Regular $y(b) = 0$
   $E_k = (k+1)^2$, $k = 0, 1, ...$
   Transformation of $-d^2v/dt^2 = Ev$, $v(\pi/48) = 0 = v(49\pi/48)$ by $t = \frac{4\pi}{3x^2}$, $u = x^{3/2}v$. (The original reference had $a = 1$, $b = 2$ corresponding to $v(\pi/3) = 0 = v(4\pi/3)$ which is much tamer.

5. **Paine problem**. `Paine_slp.mat`. Reference: [61].
   Using Liouville's transformation, this problem becomes a Schrodinger equation with $V(x) = 1/(x + 0.1)^2$, i.e. Paine problem 2.
   $p(x) = (u+x)^3 \quad q(x) = 4(u+x) \quad w(x) = (u+x)^5 \quad u = \sqrt{0.2}$
   $a = 0$ Regular $y(a) = 0$
   $b = -u + \sqrt{u^2 + 2\pi}$ Regular $y(b) = 0$.

6. **Pruess-Fulton problem 19**. (`Pruess_Fulton19.mat`). (SLTSTPAK #25).
   Reference: 19th problem in the Pruess-Fulton test set [102].
   $p(x) = x^4 \quad q(x) = -2x^2 \quad w(x) = x^4$
   $a = 0$ LCN
   $b = 1$ Regular $y(b) = 0$
   $E_k = ((k+1)\pi)^2$, $k = 0, 1, ...$

7. **Simple Sturm-Liouville problem 1**. (`simple_slp1.mat`).
   $p(x) = 1 \quad q(x) = 0 \quad w(x) = 1$
   $a = 0$ Regular $y(a) = 0$
   $b = 1$ Regular $y(b) = 0$
   Number of eigenvalues: ∞ continuous spectrum: none
   $E_k = ((k+1)\pi)^2$.

8. **Simple Sturm-Liouville problem 2**. (`simple_slp2.mat`).
   $p(x) = 1 \quad q(x) = 0 \quad w(x) = 1/x^2$
   $a = 1$ Regular $y(a) = 0$
   $b = e$ Regular $y(b) = 0$
   Number of eigenvalues: ∞ continuous spectrum: none
   $E_k = ((k+1)\pi)^2 + 1/4$.

C.3 Radial Schrödinger problems with a distorted Coulomb potential

Finite integration interval

1. **Bessel equation in normal form**. (`Bessel_normalform.mat`). (SLTSTPAK #13).
   $V(x) = (\alpha - 1/4)/x^2 \qquad (l = (-1 + 2\sqrt{\alpha})/2)$
   $\alpha = \nu^2$
   $a = 0$ LCN
   $b = 1$ Regular $y(b) = 0$
   Number of eigenvalues: ∞ continuous spectrum: none.
2. **Bessel equation in normal form, order 0**. (`Bessel_order0.mat`). (SLTSTPAK #18).
   Reference: [119].
   $V(x) = (\alpha - 1/4)/x^2 \qquad (l = (-1 + 2\sqrt{\alpha})/2)$
   $\alpha = 0$
   $a = 0$ LCN
   $b = 1$ Regular $y(b) = 0$
   $E_0 = 5.78318596295 \quad E_{19} = 3850.01252885$.
3. **Bessel equation in normal form, small α**. (`Pryce43.mat`). (SLTSTPAK #43).
   Bessel equation in normal form with $\alpha = 0.01$.
   LCN for small $\alpha \geq 0$
   Number of eigenvalues: ∞ continuous spectrum: none
   $E_0 = 6.540555712 \quad E_{24} = 6070.441468$.
4. **Truncated hydrogen equation**. (`hydrogen_truncated.mat`). (SLTSTPAK #4).
   $V(x) = -1/x + 2/x^2$
   $a = 0$ LPN
   $b = 1000$ Regular $y(b) = 0$
   $E_0 = -6.2500000000\ 10^{-2} \quad E_9 = -2.066115702478\ 10^{-3}$
   $E_{17} = -2.5757359232\ 10^{-4} \quad E_{18} = 2.873901310\ 10^{-5}$
   The lower eigenvalues approximate those of the infinite problem.

Infinite integration interval

1. **Pure attractive Coulomb potential**. (`pure_coulomb.mat`).
   $V(x) = l(l+1)/x^2 - 2Z/x$
   $a = 0$ LCN
   $b = +\infty$ LPN/O
   Number of eigenvalues: ∞ continuous spectrum: $(0, \infty)$
   $E_k = -Z^2/(n+l+1)^2,\ k = 0, 1, ...$

2. **Chemical model potential**. (`Pryce42.mat`). (SLTSTPAK #42). Reference: [123].
   $V(x) = l(l+1)/x^2 + (-1+5e^{-2x})/x$
   $a = 0$ LCN ($l = 0$), LPN ($l = 1$)
   $b = +\infty$ LPN/O
   Number of eigenvalues: ∞ continuous spectrum: $(0, \infty)$
   $l = 0 : E_0 = -0.156358880971 \quad E_2 = -0.023484895664$
   $l = 1 : E_0 = -0.061681846633 \quad E_2 = -0.015501561691.$
3. **Coulomb potential**. (`Coulomb.mat`). (SLTSTPAK #30). Reference: [18, 119].
   With $b = 1$, $u(b) = 0$ also called Boyd equation.
   $V(x) = -1/x$
   $a = 0$ LCN
   $b = +\infty$ LPN/O
   Number of eigenvalues: ∞ continuous spectrum: $(0, \infty)$
   $E_k = -1/[4(k+1)^2],\ k = 0, 1, ...$
4. **Partially screening exponential-cosine potential**. (`Expon_cosine_part_screening.mat`). Reference: [62].
   $V(x) = l(l+1)/x^2 - 2Z_0V_{ec}(x,\lambda,\mu) - 2Z_{as}(1/x - V_{ec}(x,\lambda,\mu))$
   $V_{ec}(x,\lambda,\mu) = e^{-\lambda x}\cos(\mu x)/x.$
   $a = 0$ LCN ($l = 0$) LPN ($l = 5, 10$)
   $b = +\infty$ LPN/O
   Number of eigenvalues: ∞ continuous spectrum: $(0, \infty)$
   $l = 0$, $Z_0 = 50$, $Z_{as} = 1$, $\lambda = \mu = 0.025$: $E_0 = -2497.550000612$
   $l = 5$, $Z_0 = 50$, $Z_{as} = 1$, $\lambda = \mu = 0.025$: $E_0 = -66.9947751270$
   $l = 10$, $Z_0 = 50$, $Z_{as} = 1$, $\lambda = \mu = 0.025$: $E_0 = -18.2144512404.$
5. **Screening exponential-cosine potential**. (`Expon_cosine_screening.mat`). Reference: [62].
   $V(x) = l(l+1)/x^2 - 2ZV_{ec}(x,\lambda,\mu)$
   $V_{ec}(x,\lambda,\mu) = e^{-\lambda x}\cos(\mu x)/x.$
   $a = 0$ LCN ($l = 0$) LPN ($l = 5, 10$)
   $b = +\infty$ LPN/O
   Number of eigenvalues: ∞ continuous spectrum: $(0, \infty)$.

6. **Hulthén partially screening potential**. (`Hulthen_part_screening.mat`). Reference: [62].
   $V(x) = l(l+1)/x^2 - 2Z_0V_H(x,\lambda) - 2Z_{as}(1/x - V_H(x,\lambda))$
   $V_H(x,\lambda) = \dfrac{\lambda e^{-\lambda x}}{1-e^{-\lambda x}} = \dfrac{e^{-\lambda x/2}}{x\,\eta_0((\lambda x/2)^2)}.$
   $a = 0$ LCN ($l = 0$) LPN ($l = 5, l = 10$)
   $b = +\infty$ LPN/O
   Number of eigenvalues: ∞ continuous spectrum: $(0, \infty)$
   $l = 0$, $Z_0 = 50$, $Z_{as} = 1$, $\lambda = 0.025$: $E_0 = -2498.775153125$
   $l = 5$, $Z_0 = 50$, $Z_{as} = 1$, $\lambda = 0.025$: $E_0 = -68.2234257245$
   $l = 10$, $Z_0 = 50$, $Z_{as} = 1$, $\lambda = 0.025$: $E_0 = -19.4490716959.$
7. **Hulthén screening potential**. (`Hulthen_screening.mat`). Reference: [62].
   $V(x) = l(l+1)/x^2 - 2Z\,V_H(x,\lambda)$
   $V_H(x,\lambda) = \dfrac{\lambda e^{-\lambda x}}{1-e^{-\lambda x}}.$

$a = 0$
$b = +\infty$ LPN/O
Number of eigenvalues: ∞ continuous spectrum: $(0, \infty)$
Exact eigenvalues only known for $l = 0$: $E_k = -[2Z - (k+1)^2\lambda]^2/4(k+1)^2$, $k = 0, 1, ..., k_{max} = \left\lfloor \sqrt{2Z/\lambda} \right\rfloor - 1$.

8. **Hydrogen atom**. (`hydrogen.mat`). (SLTSTPAK #29). Reference: [119].
$V(x) = -1/x + 2/x^2$
$a = 0$ LPN
$b = +\infty$ LPN/O
Number of eigenvalues: ∞ continuous spectrum: $(0, \infty)$
$E_k = -1/(2k+4)^2,\ k = 0, 1, \ldots.$

9. **Laguerre's equation**. (`Laguerre.mat`). (SLTSTPAK #32).
$V(x) = x^2 + 3/(4x^2)$
$a = 0$ LPN Trunc. BC: $y(a) = 0$
$b = +\infty$ LPN Trunc. BC: $y(b) = 0$
Number of eigenvalues: ∞ continuous spectrum: none
$E_k = 4(k+1)$, $k = 0, 1, \ldots.$

10. **Morse potential**. (`Pryce36.mat`.) (SLTSTPAK #36). Reference: [116].
$V(x) = 2/x^2 - 2000(2e^{-1.7(x-1.3)} - e^{-3.4(x-1.3)})$
$a = 0$ LPN
$b = +\infty$ LPN/O
Number of eigenvalues: 26 continuous spectrum: none
$E_0 = -1923.529655$ $E_1 = -1777.290819$ $E_{13} = -473.29712549.$

11. **Woods-Saxon potential**. (`Woods_Saxon.mat`). (SLTSTPAK#41). Reference: [123].
$V(x) = l(l+1)/x^2 - 50(1 - 5t/(3(1+t)))/(1+t)$
$t = e^{(x-7)/0.6}$
$a = 0$ Regular ($l = 0$) LPN ($l = 2$) $y(a) = 0$
$b = +\infty$ LPN/O
$l = 0$: Number of eigenvalues: 14 continuous spectrum: $(0, \infty)$
$E_0 = -49.457788728$ $E_{10} = -18.094688282$
$l = 2$: Number of eigenvalues: 13 continuous spectrum: $(0, \infty)$
$E_0 = -48.349481052$ $E_{10} = -13.522303353.$

References

1. http://www.cpc.cs.qub.ac.uk/.
2. http://www.nummath.ugent.be/SLsoftware.
3. ABRAMOWITZ, M., AND SEGUN, I. A. *Handbook of Mathematical Functions*. Dover, 1965.
4. ALEXANDER, M. H., AND MANOLOPOULOS, D. E. A stable linear reference potential algorithm for solution of the quantum close-coupled equations in molecular scattering theory. *J. Chem. Phys. 86* (1987), 2044–2050.
5. ALLISON, A. C. The numerical solution of coupled differential equations arising from the Schrödinger equation. *J. Comput. Phys. 6* (1970), 378–391.
6. ALLISON, A. C. The numerical solution of the equations of molecular-scattering. *Adv. At. Mol. Phys. 25* (1988), 323–341.
7. AMOS, D. E. A subroutine package for Bessel functions of a complex argument and nonnegative order. Sandia national laboratory report, sand85-1018, Sandia National Laboratories, Albuquerque, 1985.
8. AMOS, D. E. Algorithm 644. a portable package for Bessel functions of a complex argument and nonnegative order. *ACM Trans. Math. Software 12* (1986), 265–273.
9. ANDERSSEN, R. S., AND DE HOOG, F. R. On the correction of finite difference eigenvalue approximations for Sturm-Liouville problems with general boundary conditions. *BIT 24* (1984), 401–412.
10. ANDREW, A. L. The accuracy of Numerov's method for eigenvalues. *BIT 26* (1986), 251–253.
11. ANDREW, A. L. Asymptotic correction of finite difference eigenvalues. *Computational Techniques and Applications: CTAC85*. Amsterdam, North-Holland (1986), 333–341.
12. ANDREW, A. L., AND PAINE, J. W. Correction of Numerov's eigenvalue estimates. *Numer. Math. 47* (1985), 289–300.
13. ANDREW, A. L., AND PAINE, J. W. Correction of finite element estimates for Sturm-Liouville eigenvalues. *Numer. Math. 50* (1986), 205–215.
14. ATKINSON, F. V. *Discrete and Continuous Boundary Problems*. Academic Press, 1964.

15. BAILEY, P. B. Sturm-Liouville eigenvalues via a phase function. *SIAM Journal of Applied Math. 14* (1966), 242–249.

16. BAILEY, P. B. Modified Prüfer transformations. *J. Comp. Phys. 29* (1978), 306–310.

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

18. BAILEY, P. B., EVERITT, W. N., AND ZETTL, A. Computing eigenvalues of singular Sturm-Liouville problems. *Results Math. 20* (1991), 391–423.

19. BAILEY, P. B., GARBOW, B. S., KAPER, H. G., AND ZETTL, A. Eigenvalue and eigenfunction computations for Sturm-Liouville problems. *ACM Trans. Math. Software 17* (1991), 491–499.

20. BAILEY, P. B., GORDON, M. K., AND SHAMPINE, L. F. Solving Sturm-Liouville eigenproblems. Technical report sand76-0560, Sandia National Laboratories, Albuquerque, 1976.

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

22. BANDYRSKII, B., GAVRILYUK, I., LAZURCHAK, I., AND MAKAROV, V. Functional-discrete method (FD-method) for matrix Sturm-Liouville problems. *Comput. Meth. in Appl. Math. 5* (2005), 362–386.

23. BENDER, C. M., AND ORSZAG, S. A. *Advanced Mathematical Methods for Scientists and Engineers*. McGraw-Hill, 1978.

24. BISWAS, S. N., DATTA, K., SAXENA, R. P., SRIVASTAVA, P. K., AND VARMA, V. S. (eigenvalues of Ex^{2m} anharmonic oscillators. *J. Math. Phys. 14* (1973), 1190.

25. BLATT, J. M. Practical points concerning the solution of the Schrödinger equation. *J. Comput. Phys. 1* (1967), 382–396.

26. BREZINSKI, C., AND REDIVO ZAGLIA, M. *Extrapolation Methods. Theory and Practice*. North-Holland, 1991.

27. BRILLOUIN, L. Remarques sur la méchanique ondulatoire. *J. Phys. Radium 7* (1926), 353–368.

28. CANOSA, J., AND DE OLIVEIRA, R. G. A new method for the solution of the Schrödinger equation. *J. Comp. Phys. 5* (1970), 188–207.

29. CHRISTLEY, J. A., AND THOMPSON, I. J. CRCWFN: coupled real Coulomb wave functions. *Comput. Phys. Commun. 79* (1994), 143–155.

30. CODDINGTON, E. A., AND LEVINSON, N. *Theory of Ordinary Differential Equations*. McGraw-Hill, 1955.

31. COLLATZ, L. *Differential equations, An Introduction with Applications*. Wiley, 1986.

32. DEGANI, I., AND SCHIFF, J. RCMS: Right Correction Magnus Series approach for oscillatory ODEs. *J. Comput. Appl. Math. 193* (2006), 413–436.

33. DUNFORD, N., AND SCHWARTZ, J. T. *Linear Operators: Part II: Spectral Theory,*

self adjoint operators in Hilbert space. Wiley-Interscience, 1963.

34. DWYER, H. I., AND ZETTL, A. Eigenvalue computations for regular matrix Sturm-Liouville problems. *Electr. J. Diff. Eqns. 5* (1995), 1–13.
35. FABIJONAS, B. R. Algorithm 819: routine for the computation of complex Airy functions. *ACM Trans. Math. Software 30* (2004), 491–501.
36. FACK, V., AND VANDEN BERGHE, G. (a finite difference approach for the calculation of perturbed oscillator energies. *J. Phys. A: Math. Gen. 18* (1985), 3355–3363.
37. FACK, V., AND VANDEN BERGHE, G. (extended) Numerov method for computing eigenvalues of specific Schrödinger equations. *J. Phys. A: Math. Gen. 20* (1987), 4153–4160.
38. FULTON, C. T., AND PRUESS, S. The computation of spectral density functions for singular Sturm-Liouville problems involving simple continuous spectra. *ACM Trans. Math. Software 24* (1998), 107–129.
39. GELFAND, I., AND LEVITAN, B. On the determination of a differential equation from its spectral function. *AMS Translations 1* (1955), 253–304.
40. GIL, A., SEGURA, J., AND TEMME, N. Algorithm 819: routine for the computation of complex Airy functions. *ACM Trans. Math. Software 28* (2002), 325–336.
41. GORDON, R. G. New method for construcing wavefunctions for bound states and scattering. *J. Chem. Phys. 51* (1969), 14–25.
42. GORDON, R. G. Quantum scattering using piecewise analytic solutions. *Methods in Computational Physics - Academic Press 10* (1971), 81–109.
43. GREENBERG, L. A Prüfer method for calculating eigenvalues of self-adjoint systems of ordinary differential equations, parts 1 and 2. Technical report, Department of Mathematics, University of Maryland, 1991.
44. HAIRER, E., NØRSETT, S. P., AND WANNER, G. *Solving Ordinary Differential Equations I*. Springer-Verlag, 1987.
45. HARTMAN, P. *Ordinary Differential Equations*. Wiley, 1963.
46. HOCHBRUCK, M., AND LUBICH, C. On Magnus methods for time-dependent Schrödinger equations. *SIAM J. Numer. Anal. 41* (2003), 945–963.
47. HUTSON, J. M. Coupled channel methods for solving the bound-state Schrödinger equation. *Comput. Phys. Commun. 84* (1994), 1–18.
48. HUTSON, J. M., AND GREEN, S. *MOLSCAT computer code, version 14*. distributed by Collaborative Computational Project No. 6 of the Engineering and Physical Sciences Research Council (UK), 1994.
49. ISERLES, A. Think globally, act locally: solving highly-oscillatory ordinary differential equations. *BIT 43* (2002), 145–160.
50. ISERLES, A. On the method of Neumann series for highly oscillatory equations. *BIT 44* (2004), 473–488.
51. ISERLES, A., MUNTHE-KAAS, H. Z., NØRSETT, S. P., AND ZANNA, A. Lie group methods. *Acta Numer. 9* (2000), 215–365.

52. ISERLES, A., AND NØRSETT, N. P. On the solution of linear differential equations in Lie groups. *Philos. Trans. R. Soc. Lond. A 357* (1999), 983–1019.

53. IXARU, L. GR. The algebraic approach to the scattering problem. Internal Report IC/69/7, International Centre for Theoretical Phyics, 1969.

54. IXARU, L. GR. An algebraic solution of the Schrödinger equation. Internal Report IC/69/6, International Centre for Theoretical Phyics, 1969.

55. IXARU, L. GR. The error analysis of the algebraic method for solving the Schrödinger equation. *J. Comput. Phys. 9* (1972), 159–163.

56. IXARU, L. GR. Perturbative numerical methods for the Schrödinger equation. *Comput. Phys. Commun. 20* (1980), 97–112.

57. IXARU, L. GR. Simple procedure to compute accurate energy levels of an anharmonic oscillator. *Physical Review D 25* (1982), 1557–1564.

58. IXARU, L. GR. *Numerical Methods for Differential Equations and Applications*. Reidel, 1984.

59. IXARU, L. GR. LILIX - a package for the solution of the coupled channel Schrödinger equation. *Comput. Phys. Commun. 174* (2002), 834–852.

60. IXARU, L. GR., DE MEYER, H., AND VANDEN BERGHE, G. CP methods for the Schrödinger equation, revisited. *J. Comput. Appl. Math. 88* (1997), 289–314.

61. IXARU, L. GR., DE MEYER, H., AND VANDEN BERGHE, G. SLCPM12 - a program for solving regular Sturm-Liouville problems. *Comput. Phys. Commun. 118* (1999), 259–277.

62. IXARU, L. GR., DE MEYER, H., AND VANDEN BERGHE, G. Highly accurate eigenvalues for the distorted Coulomb potential. *Phys. Rev. E 61* (2000), 3151–3159.

63. IXARU, L. GR., RIZEA, M., AND VERTSE, T. Piecewise perturbation methods for calculating eigensolutions of a complex optical potential. *Comput. Phys. Commun. 85* (1995), 217–230.

64. JANHKE, T., AND LUBICH, C. Numerical integrators for quantum dynamics close to the adiabatic limit. *Numer. Math. 94* (2003), 289–314.

65. JOHNSON, B. R. Renormalized Numerov method applied to calculating bound-states of coupled-channel Schrödinger equation. *J. Chem. Phys. 69* (1978), 4678–4688.

66. KELLER, H. B. On the accuracy of finite difference approximations to the eigenvalues of differential and integral operators. *Numer. Math. 7* (1965), 412–419.

67. KELLER, H. B. *Numerical solution of two point boundary value problems*. Lecture Notes in Mathematics 1258. SIAM, 1976.

68. KLOTTER, K. *Technische Schwingungslehre, I*. Springer, 1978.

69. KODAIRA, K. The eigenproblem for ordinary differential equations of the second order and Heisenberg's theory of s matrices. *Amer. J. Math. 71* (1949), 921–945.

70. KOSOWSKI, P. The relative error in the Pruess method for Sturm-Liouville prob-

lems. *Linear Algebra Appl. 309* (2000), 325–337.

71. KRAMERS, H. A. Wellenmechanik und halbzahlige quantizierung. *Z. Physik 39* (1926), 828–840.

72. LAMBERT, J. D. *Computational methods in ordinary differential equations*. Wiley, 1973.

73. LEDOUX, V., IXARU, L. GR., RIZEA, M., VAN DAELE, M., AND VANDEN BERGHE, G. Solution of the Schrödinger equation over an infinite integration interval by perturbation methods, revisited. *Comput. Phys. Commun. 175* (2006), 612–619.

74. LEDOUX, V., RIZEA, M., IXARU, L. GR., VANDEN BERGHE, G., AND VAN DAELE, M. Solution of the Schrödinger equation by a high order perturbation method based on a linear reference potential. *Comput. Phys. Commun. 175* (2006), 424–439.

75. LEDOUX, V., VAN DAELE, M., AND VANDEN BERGHE, G. CP methods of higher order for Sturm-Liouville and Schrödinger equations. *Comput. Phys. Commun. 162* (2004), 151–165.

76. LEDOUX, V., VAN DAELE, M., AND VANDEN BERGHE, G. Matslise: A matlab package for the numerical solution of Sturm-Liouville and Schrödinger equations. *ACM Trans. Math. Software 31* (2005), 532–554.

77. LEDOUX, V., VAN DAELE, M., AND VANDEN BERGHE, G. CPM{P, N} methods extended for the solution of coupled channel Schrödinger equations. *Comput. Phys. Commun. 174* (2006), 357–370.

78. LEDOUX, V., VAN DAELE, M., AND VANDEN BERGHE, G. Piecewise constant perturbation methods for the multichannel Schrödinger equation. *Lect. Notes in Comput. Sc. 3994* (2006), 716–723.

79. LEDOUX, V., VAN DAELE, M., AND VANDEN BERGHE, G. A numerical procedure to solve the multichannel Schrödinger eigenvalue problem. *Comput. Phys. Commun.* (in press).

80. LEVIN, D. Development of nonlinear transformations for improving convergence of sequences. *Int. J. Comput. Math. 3* (1973), 371–388.

81. LEVINE, R. Adiabatic approximation for nonreactive subexcitation molecular collisions. *J. Chem. Phys. 49* (1968), 51.

82. LIMBER, M. N., AND CURRY, J. H. On the dynamics of shooting methods for solving Sturm-Liouville problems. *Comm. Appl. Nonlinear Anal. 1* (1994), 1–24.

83. MAGNUS, W. On the exponential solution of differential equations for a linear operator. *Comm. Pure and Appl. Math. 55* (1954), 213–229.

84. MARLETTA, M. *Theory and Implementation of Algorithms for Sturm-Liouville Computations*. PhD thesis, Royal Military College of Science, Shrivenham, UK, 1991.

85. MARLETTA, M. Automatic solution of regular and singular vector Sturm-Liouville

problems. *Numer. Algorithms 4* (1993), 65–99.

86. Marletta, M. Numerical solution of eigenvalue problems for Hamiltonian systems. *Adv. Comput. Math. 2* (1994), 155–184.

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

88. Marletta, M., and Pryce, J. D. Automatic solution of Sturm-Liouville problems using the Pruess method. *J. Comput. Appl. Math. 39* (1992), 57–78.

89. Marti, J. T. Small potential corrections for the discrete eigenvalues of the Sturm-Liouville problem. *Numer. Math. 57* (1990), 51–62.

90. Moan, P. C. Efficient approximation of Sturm-Liouville problems using Lie group methods. Technical report 1998/NA11, DAMTP University of Cambridge, 1998.

91. Morse, P. M. Diatomic molecules according to the wave mechanics. ii. vibrational levels. *Phys. Rev. 34* (1929), 59–61.

92. Murphy, G. M. *Ordinary Differential Equations and Their Solutions*. D. Van Nostrand Company, 1960.

93. NAG Fortran Library Manual. *Mark 15*. The Numerical Algorithms Group Limited, 1991.

94. Niessen, H. D., and Zettl, A. Singular Sturm-Liouville problems; the Friedrichs extension and comparison of eigenvalues. *Proc. London Math. Soc. 64* (1992), 545–578.

95. Paine, J. W. Correction of Sturm-Liouville eigenvalue estimates. *Math. Comp. 39* (1982), 415–420.

96. Paine, J. W., and de Hoog, F. R. Uniform estimation of the eigenvalues of Sturm-Liouville problems. *J. Austral. Math. Soc. Ser. B 21* (1980), 365–383.

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

98. Press, W. H., Teuksky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes in Fortran - The Art of Scientific Computing*. University Press, 1996.

99. Pruess, S. Estimating the eigenvalues of Sturm-Liouville problems by approximating the differential equation. *SIAM J. Numer. Anal. 10* (1973), 55–68.

100. Pruess, S. High-order approximations to Sturm-Liouville eigenvalues. *Numer. Math. 24* (1975), 241–247.

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

102. Pruess, S., Fulton, C. T., and Xie, Y. Performance of the sturm-liouville software package SLEDGE. Technical report MCS-91-19, Colorado School of Mines, 1991.

103. PRÜFER, H. Neue herleitung der Sturm-Liouvilleshen reihenentwicklung stetiger funktionen. *Math. Ann. 95* (1923), 499–518.

104. PRYCE, J. D. Error control of phase function shooting methods for Sturm-Liouville problems. *IMA J. Numerical Analysis 6* (1986), 103–123.

105. PRYCE, J. D. *Numerical Solution of Sturm-Liouville Problems*. Oxford University Press, 1993.

106. PRYCE, J. D. Efficient, reliable computation of resonances of the one-dimensional Schrödinger equation. *J. Comp. Phys. 112* (1994), 234–246.

107. PRYCE, J. D. Algorithm 789: SLTSTPAK: a test package for Sturm-Liouville solvers. *ACM Trans.Math. Software 25* (1999), 58–69.

108. PRYCE, J. D. A test package for Sturm-Liouville solvers. *ACM Trans.Math. Software 25* (1999), 21–57.

109. RALSTON, A., AND RABINOWITZ, P. *A first course in numerical analysis*. McGraw Hill, 1978.

110. REID, W. T. *Ordinary Differential Equations*. Wiley, 1971.

111. REID, W. T. *Sturmian theory for ordinary differential equations*. Springer Verlag, 1980.

112. RIZEA, M. Persys - a program for the solution near the origin of coupled channel Schrödinger equation with singular potential. *Comput. Phys. Commun. 143* (2002), 83–99.

113. RYKACZEWSKI, K., BATCHELDER, J. C., AND BINGHAM, C. R. ET AL. Proton emitters ^{140}Ho and ^{141}Ho: Probing the structure of unbound Nilsson orbitals. *Phys. Rev. C. 60* (1999), 011301.

114. SCHRÖDINGER, E. An undulatory theory of the mechanics of atoms and molecules. *Phys. Rev. 28* (1926), 1049–1070.

115. SCOTT, M. R., SHAMPINE, L. F., AND WING, G. M. Invariant imbedding and the calculation of eigenvalues of Sturm-Liouville systems. *Computing 4* (1969), 10–23.

116. SECREST, D., CASHION, K., AND HIRSCHFELDER, J. O. Power-series solutions for energy eigenvalues. *J. Chem. Phys. 37* (1962), 830–835.

117. SMOOKE, M. D. Piecewise analytical perturbation series solutions of the radial Schrödinger equation: one-dimensional case. *SIAM J. Sci. Stat. Comput. 3* (1982), 195–222.

118. TIKHONOV, A. N., VASIL'EVA, A. B., AND SVESHNIKOV, A. G. *Differential Equations*. Springer-Verlag, 1985.

119. TITCHMARSH, E. C. *Eigenfunction expansions associated with second order differential equations*. Clarendon Press, 1962.

120. VALLÉE, O., AND SOARES, M. *Airy functions and applications to physics*. Imperial College Press, 2004.

121. VANDEN BERGHE, G., AND DE MEYER, H. Accurate computation of higher Sturm-Liouville eigenvalues. *Numer. Math. 59* (1991), 243–254.

122. VANDEN BERGHE, G., AND DE MEYER, H. On a correction of Numerov-like eigenvalue approximations for Sturm-Liouville problems. *J. Comp. Appl. Math. 37* (1991), 179–186.

123. VANDEN BERGHE, G., FACK, V., AND DE MEYER, H. Numerical methods for solving radial Schrödinger equations. *J. Comp. Appl. Math. 28* (1989), 391–401.

124. VANDEN BERGHE, G., AND VAN DAELE, M. Exponentially-fitted Numerov methods. *J. Comput. Appl. Math.* (in press).

125. VANDEN BERGHE, G., VAN DAELE, M., AND DE MEYER, H. Accurate high-lying eigenvalues of Schrödinger and Sturm-Liouville problems. *Physics Letters A 193* (1994), 341–347.

126. WEIDMANN, J. *Spectral theory of ordinary differential operators*. Lecture Notes in Mathematics 1258. Springer-Verlag, 1987.

127. WENIGER, E. J. Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series. *Comput. Phys. Rep. 10* (1989), 191–371.

128. WENTZEL, G. Eine verallgemeinerung der quantenbedingungen für die zwecke der wellenmechanik. *Z. Physik 5* (1926), 518–529.

129. WEYL, H. Über gewöhnliche differentialgleichungen mit singularitäten und die zugehörigen entwicklungen willkürlicher funktionen. *Math. Ann. 68* (1910), 220–269.

130. WICKE, B. G., AND HARRIS, D. O. Comparison of three numerical techniques for calculating eigenvalues of an unsymmetrical double minimum oscillator. *J. Chem. Phys. 64* (1976), 5236–5242.

131. ZETTL, A. Sturm-Liouville problems, spectral theory and computational methods of Sturm-Liouville problems. Proceedings of the 1996 Knoxville Barrett conference, *Marcel Dekker 191* (1997), 1–104.

132. ZETTL, A. *Sturm-Liouville Theory*. American Mathematical Society, 2005.

133. ZHANG, S., AND JIN, J. M. *Computation of Special Functions*. Wiley, 1996.

Index