A numerical and analytical investigation into non-Hermitian Hamiltonians.

by

Gert Jermia Cornelus Wessels

Thesis presented at the University of Stellenbosch in partial fulfilment of the requirements for the degree of

Master of Physical and Mathematical Analysis

University of Stellenbosch
Private Bag X1, 7602 Matieland, South Africa

Study leaders:
Prof. H.B. Geyer  Prof. J.A.C. Weideman

November 2008
Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: ...............................  
G.J.C. Wessels

Date: .................................
Abstract

A numerical and analytical investigation into non-Hermitian Hamiltonians.

G.J.C. Wessels
University of Stellenbosch
Private Bag X1, 7602 Matieland, South Africa
Thesis: MSc (PMA)
November 2008

In this thesis we aim to show that the Schrödinger equation, which is a boundary eigenvalue problem, can have a discrete and real energy spectrum (eigenvalues) even when the Hamiltonian is non-Hermitian. After a brief introduction into non-Hermiticity, we will focus on solving the Schrödinger equation with a special class of non-Hermitian Hamiltonians, namely $\mathcal{PT}$-symmetric Hamiltonians. $\mathcal{PT}$-symmetric Hamiltonians have been discussed by various authors \cite{1,2,3,4,5} with some of them focusing specifically on obtaining the real and discrete energy spectrum.

Various methods for solving this problematic Schrödinger equation will be considered. After starting with perturbation theory, we will move on to numerical methods. Three different categories of methods will be discussed. First there is the shooting method based on a Runge-Kutta solver. Next, we investigate various implementations of the spectral method. Finally, we will look at the Riccati-Padé method, which is a numerical implemented analytical method. $\mathcal{PT}$-symmetric potentials need to be solved along a contour in the complex plane. We will propose modifications to the numerical methods to handle this.

After solving the widely documented $\mathcal{PT}$-symmetric Hamiltonian $H = p^2 - (ix)^N$ with these methods, we give a discussion and comparison of the obtained results.

Finally, we solve another $\mathcal{PT}$-symmetric potential, illustrating the use of paths in the complex plane to obtain a real and discrete spectrum and their influence on the results.
Uittreksel

A numerical and analytical investigation into non-Hermitian Hamiltonians.

G.J.C. Wessels

Universiteit van Stellenbosch

Privaatsak X1, 7602 Matieland, Suid Afrika

Tesis: MSc (FWA)

November 2008

In hierdie tesis is die hoofdoel om die Schrödinger vergelyking, ’n randvoorwaarde probleem, op te los waar die Hamilton-operatore nie Hermities is nie. Ons probeer wys dat hierdie operatore ’n diskrete en reële energie spektrum (eiewaardes) sal gee. Na ’n kort inleiding oor nie-Hermitisiteit, los ons die Schrödinger vergelyking op vir $PT$ simmetriese Hamilton-operatore, wat ’n nie Hermitiese subversameling is. $PT$ simmetriese Hamilton-operatore is al in baie besonderhede deur verskeie skrywers bespreek en opgelos met die doel om ’n reële spektrum te kry, [1, 2, 3, 4, 5].

Ons sal verskeie metodes vir die oplos van hierdie Schrödinger vergelykings oorweeg. Eers kyk ons na steuringsteorie voordat ons aanbeweeg na die numeriese metodes. Daar word drie verskillende numeriese metodes oorweeg. Eerste is die skietmetode wat Runge-Kutta gebruik om die differentiaal vergelykings op te los. Daarna kyk ons na ’n paar variasies van die spektraalmetode. Ons sluit dan hierdie gedeelte af met die Riccati-Padé metode. Die Riccati-Padé metode is nie ’n numeriese metode nie, maar ’n analitiese metode wat numeries toegepas word.

Aangesien $PT$ simmetriese potensiale soms langs ’n pad in die komplekse ruimte opgelos moet word, word ’n paar veranderinge aan die numeriese metodes voorgestel.

Nadat ons die $PT$ simmetriese Hamilton-operator $H = p^2 - (ix)^N$ opgelos het met hierdie metodes, word die resultate bespreek en vergelyk.
Ons sluit af met 'n ander \( PT \) simmetriese potensiaal en illustreer hoe 'n reële spektrum verkry kan word met behulp van 'n pad in die komplekse ruimte en hoe verkillende paaie die resultate kan beïnvloed.
Contents

Declaration i
Abstract iii
Uittreksel iv
Contents vi
List of Figures viii
List of Tables x

1 Introduction into non-Hermitian Hamiltonians 1
  1.1 Hermitian and Non-Hermitian Hamiltonians .......... 1
  1.1.1 Hermitian Hamiltonians .......................... 1
  1.1.2 Quasi- and Pseud-Hermitian Operators ............ 3
  1.1.3 \( \mathcal{PT} \)-Symmetry ......................... 5
  1.2 Contours in the Complex plane ....................... 7
  1.2.1 Stokes Wedges .................................. 9
  1.2.2 Quantum Toboggans .............................. 11

2 Perturbation Theory 13

3 Numerical Methods 17
  3.1 Shooting Methods .................................. 17
  3.1.1 Introduction .................................... 17
  3.1.2 Other Software Packages ........................ 24
  3.2 Spectral Collocation Methods ....................... 24
  3.2.1 Method 1a : DMSUITE ............................ 25
  3.2.2 Method 1b : DMSUITE, the Fourier approach .... 32
  3.2.3 Method 2 : HermiteEig .......................... 34
  3.3 Riccati-Padé ....................................... 38


<table>
<thead>
<tr>
<th>Contents</th>
<th>vii</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.1 $V(x) = x^2$</td>
<td>40</td>
</tr>
<tr>
<td>3.4 Summary</td>
<td>43</td>
</tr>
<tr>
<td>4 The ( \mathcal{PT} )-Symmetric Potential $V(x) = -(i \sinh x)^a \cosh^b x$</td>
<td>46</td>
</tr>
<tr>
<td>4.1 Paths in the Complex Plane</td>
<td>47</td>
</tr>
<tr>
<td>A MATLAB code</td>
<td>53</td>
</tr>
<tr>
<td>B Additional Tables and Figures</td>
<td>59</td>
</tr>
<tr>
<td>Bibliography</td>
<td>61</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Spectrum for the Hamiltonian $H = p^2 - (ix)^N$ as obtained by Bender and Boettcher [1]. ......................................................... 7
1.2 These are the Stokes wedges where $N = 4.5$ and the contour is $z = -2i\sqrt{1+iw}$. ......................................................... 10
1.3 These are the Stokes wedges where $N = 10$ and the contour is $z = i\mu(\sin(\alpha) + \sin(iw - \alpha))$. ................................. 10
1.4 This figure displays a tobogganic contour with $w = 3$. .......... 12
2.1 The first order approximation to the perturbed harmonic oscillator $(-i\mu)^N$ along with the spectrum of the potential. A higher order approximation will approach the spectrum more accurately at $N = 2$. ................. 16
3.1 Determining $\psi(x)$ by converting the boundary value problem into two initial value problems. ........................................... 18
3.2 The absolute error of the ground state, first and fifth energy level of the harmonic oscillator calculated with the shooting method for different values of $L$. ......................................................... 19
3.3 The matching function $\phi(E)$ for the harmonic oscillator. ....... 20
3.4 The effects of ode45’s relative tolerance on the accuracy and runtime (in seconds) of the shooting method. ............................ 21
3.5 The eigenvalues of $V(x) = -(ix)^N$ calculated with the shooting method. ................................................................. 22
3.6 The eigenvalues of $V(x) = -(ix)^N$ calculated with the shooting method on the contour $z = -2i\sqrt{1+iw}$. .......................... 23
3.7 Figures for the ground state and first level energy of $V(x) = -(ix)^3$, calculated for different matrix sizes ($n = 32$ and $n = 96$) and compared with the exact values of $E_0$ of $E_1$ respectively to illustrate the dependence on the scaling parameter $c$. ......................................................... 29
3.8 The logarithm of the absolute error of the ground state energy of the harmonic oscillator, $N = 2$, calculated at different matrix sizes and for different $c$-values. $c = 1$ is expected to give the exact eigenvalues for the harmonic oscillator. ......................................................... 30
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.9</td>
<td>The logarithm of the absolute error calculated for the first few energy levels, $k$, of the harmonic oscillator for three different matrix sizes ($n = 16, 32, 48$) and for $c = 2$. This figure illustrates the dependency of the accuracy of the eigenvalues on the matrix size.</td>
</tr>
<tr>
<td>3.10</td>
<td>The eigenvalues calculated using the Hermite spectral collocation method with a matrix size 128 and $c = 2$.</td>
</tr>
<tr>
<td>3.11</td>
<td>Spectrum obtained by solving the problem along the contour $z = -2i\sqrt{1 + i\omega}$ with the Hermite based approach.</td>
</tr>
<tr>
<td>3.12</td>
<td>Stokes wedges for $N = 2$ and the contour $z = -2i\sqrt{1 + i\omega}$.</td>
</tr>
<tr>
<td>3.13</td>
<td>First we observe that there is an instability where $N &lt; 2$ and $N \approx 6$. In the other two graphs we see that the contour does not stay within the wedges when $N &lt; 1$ and $N \geq 7$, which might explain the instability.</td>
</tr>
<tr>
<td>3.14</td>
<td>The eigenvalues calculated using the Fourier interpolation.</td>
</tr>
<tr>
<td>3.15</td>
<td>Spectrum obtained by solving the problem along the contour $z = -2i\sqrt{1 + i\omega}$ with the Fourier based approach.</td>
</tr>
<tr>
<td>3.16</td>
<td>The spectrum of the $\mathcal{PT}$-symmetric potential $V(x) = -(iz)^N$ obtained by the HermiteEig method.</td>
</tr>
<tr>
<td>3.17</td>
<td>This figure compares speed of the three techniques to calculate $H_2^D$. The dotted curve indicates the time it took to calculate a symbolic $H$ using (3.26), while the other two curves used the Hankel recursion formula (3.29) to calculate it for both a symbolic and numeric matrix. All of these results were obtained by MATLAB, using the symbolic toolbox for the symbolic computations. Table B.1 contains the data used to obtain this figure.</td>
</tr>
<tr>
<td>3.18</td>
<td>This figure displays the combination of the solution for the potential $V(x) = -(iz)^N$ obtained by integrating on the real and the complex axis with the spectral collocation method.</td>
</tr>
<tr>
<td>4.1</td>
<td>Spectrum of $V_{\alpha \beta}$ related to the reference value $\alpha_R = 2$, the solid curve, and $\alpha_R = 6$, the dashed curve as it was calculated in [3, p. 3113].</td>
</tr>
<tr>
<td>4.2</td>
<td>The ground state energies of $V_{\alpha \beta}$ where $\beta = 0$.</td>
</tr>
<tr>
<td>4.3</td>
<td>The value of the imaginary component of the contour $x + iy$ when $\beta = 0$. The two dashed lines, $y_+$ and $y_-$, indicates the boundaries of the integration area, while the solid line indicates the optimal $y$-value.</td>
</tr>
<tr>
<td>4.4</td>
<td>The eigenvalues of the potential $V(x) = -(i\sinh(x))^\alpha \cosh^\beta(x)$, calculated with the spectral collocation method for $\beta = 0$ and $\alpha_R = 2$. The spectrum up to $\alpha = 2$ was calculated on the real axis while the spectrum $\alpha &gt; 2$ was calculated along the contour $x + iy$ with the complex component (4.3).</td>
</tr>
<tr>
<td>4.5</td>
<td>The eigenvalues of the potential $V(x) = -(i\sinh(x))^\alpha \cosh^\beta(x)$, calculated for $\beta = 0$ and $\alpha_R = 6$.</td>
</tr>
</tbody>
</table>
4.6 The spectrum of the potential $V_{\alpha_0}$, calculated for the two reference values
$\alpha_R = 2$ (blue) and $\alpha_R = 6$ (red). The ground state eigenvalues determined
in [3] for $\alpha = 1, 2, 3$ are included in this figure. .......................... 52

B.1 A log graph for the comparison of the runtime of the HermiteEig and spec-
tral collocation method. ................................................................. 59

List of Tables

3.1 The progression of the algorithm as we approach the ground state eigenvalue
of the harmonic oscillator, as computed by the secant method (3.5) using
the initial guesses 0 and 0.75. ......................................................... 21
3.2 In this table we see the estimated errors given by MATSLISE for the eigen-
values of the potential $V(x) = -(ix)^N$ where $N = 2$ and $N = 6$. ....... 25
3.3 A MATLAB function for the computation of the eigenenergies of $V(x) =$
$-(ix)^N$. .............................................................................. 28
3.4 The MATLAB code for calculating the eigenvalues, $L$, of a potential, $fun$,
using the HermiteEig method. ...................................................... 37
3.5 The order of the determinant $H^s_{\beta}|_{x=0}$. .............................. 41
3.6 The ground state eigenvalue of the harmonic oscillator ($E_0 = 1$) calculated
with three different implementations of the RPM. .......................... 43

4.1 This table shows for which value of $\alpha$ the potential $V_{\alpha, \beta}$ real and imaginary
parts are positive or negative. After $\alpha = 4$ the pattern repeats itself. ... 47
4.2 The absolute error between the ground state eigenvalues of our spectrum,
calculated with the spectral collocation method, and the eigenvalues ob-
tained in [3] with the Riccati-Padé method. ................................. 51

A.1 MATLAB code for calculating eigenvalues by using the shooting method. .. 53
A.2 The shooting method uses this MATLAB program to solve the two parts
of the eigenfunction given an eigenvalue for the defined potential. ....... 54
A.3 For the contour calculations, the only difference in the shooting method is
the way the eigenfunctions are solved. ......................................... 55
A.4 Program that implements Wynn’s epsilon algorithm. ....................... 56
A.5  This is the MATLAB code used to calculate the Hankel determinant or the Riccati-Padé method. ........................... 57

A.6  $f_j$ is the MATLAB code used to determine the coefficients $f_j$, (3.24), which is the elements of the Hankel determinant calculated in Table A.5 ........................... 58

B.1  This table contains the data used to construct Figure 3.17. It shows the time it takes to calculate $H$ using three different techniques. The first 2 columns used symbolic calculations to determine $H$, while the last column used a numerical algorithm. While the first column calculated $H$ using brute force, the other 2 made use of the Hankel recursion formula (3.29). All the data was obtained by taking the average from 5 runs. .............................. 60
Chapter 1

Introduction into non-Hermitian Hamiltonians

Traditionally Hermiticity is required, in the context of quantum mechanics, if a Hamiltonian is to have a real energy spectrum. In the last few years it has become clear, especially in a quantum mechanical setting, that this is not a necessity for real energies. This implies that some complex, non-Hermitian Hamiltonians can have a discrete, positive and real spectrum.

1.1 Hermitian and Non-Hermitian Hamiltonians

1.1.1 Hermitian Hamiltonians

For a consistent quantum mechanical theory, there need to be restrictions or assumptions. Among these is the requirement that an observable, an operator that describes a physical quantity, must be Hermitian\(^1\).

We start with the observable \(Q(x,p)\), where \(x\) denotes the displacement and \(p\) the momentum. The expectation value of \(Q\) can be written in terms of the inner-product,

\[
\langle Q \rangle = \int \Psi^* \hat{Q} \Psi dx = \langle \Psi | Q \Psi \rangle,
\]

where \(\Psi\) is the wave function. We foresee that the expectation value of the observable \(Q\) will be real, and so will the average of any number of

\(^1\text{Ref. [6, p. 96]}\)
measurements², \[ \langle Q \rangle = \langle Q \rangle^* . \]

The operator \( \hat{Q} \) is constructed from \( Q \) by replacing \( p \) with \( \hat{p} \equiv \left( \frac{\hbar}{i} \frac{\partial}{\partial x} \right) \). \( \hat{Q} \) will be Hermitian (self-adjoint) if
\[
\langle \Psi | \hat{Q} \Psi \rangle = \langle \hat{Q} \Psi | \Psi \rangle,
\]
which holds for any wave function³ \( \Psi \). Normally we would need to take the complex conjugate⁴ to change the order within an inner-product. This leads to an important property of Hermitian observables,
\[
\langle f | \hat{Q} g \rangle = \langle \hat{Q} f | g \rangle , \text{ for all } f(x) \text{ and } g(x).
\]

We can therefore apply the Hermitian operator \( \hat{Q} \) to the first or second part of the inner-product and still obtain the same results.

The Hermitian conjugate (or adjoint) of an operator \( \hat{Q} \) is \( \hat{Q}^\dagger \), such that
\[
\langle f | \hat{Q} g \rangle = \langle \hat{Q}^\dagger f | g \rangle , \text{ for all } f(x) \text{ and } g(x).
\]
This implies that the Hermitian operator \( \hat{Q} \) is equal to its Hermitian conjugate, \( \hat{Q} = \hat{Q}^\dagger \).

The eigenvalues of a Hermitian operator have two important properties: the eigenvalues are real and any two eigenfunctions corresponding to two different eigenvalues are orthogonal. We are interested in the first property.

**Theorem 1.** The eigenvalues of a Hermitian operator \( \hat{Q} \) are real.

**Proof.** Let \( \hat{Q} f = \lambda f \) where \( f \) is the eigenfunction \( (f(x) \neq 0) \) and \( \lambda \) the eigenvalue, and
\[
\langle f | \hat{Q} f \rangle = \langle \hat{Q} f | f \rangle ,
\]
\[
\langle f | \lambda f \rangle = \langle \lambda f | f \rangle ,
\]
because \( \hat{Q} \) is Hermitian. We now have
\[
\lambda \langle f | f \rangle = \lambda^* \langle f | f \rangle ,
\]
because \( \lambda \) is a scalar and the first function in the integral (1.1) is a complex conjugate, \( \lambda = \lambda^* \), which completes the proof. \( \square \)

---

²Notation: \(^* \) denotes the complex conjugate.
³In quantum mechanics the wave function lives in the Hilbert space \( \mathcal{H} \). In mathematics \( \mathcal{H} \) refers to the \( L^2 \) space.
⁴\( \langle f | g \rangle = \langle g | f \rangle^* \)
1.1.2 Quasi- and Pseudo-Hermitian Operators

Now that we have shown that Hermitian operators have real eigenvalues, we can look at non-Hermitian operators. In a correspondence between Bender and Bessis [1], a claim was made that non-Hermitian Hamiltonians (specifically $\mathcal{P}\mathcal{T}$-symmetric Hamiltonians) could have a real spectrum.

We start by defining a quasi-Hermitian operator as discussed by Scholtz, Geyer and Hahne, [7].

**Definition 1 (Quasi-Hermitian Operator).** An operator $A : \mathcal{H} \rightarrow \mathcal{H}$, where $\mathcal{H}$ denotes the Hilbert space, is classified as quasi-Hermitian if there exist a linear metric operator $\kappa : \mathcal{D}_\kappa \rightarrow \mathcal{H}$ which satisfies,

- $\mathcal{D}_\kappa \equiv \mathcal{H}$. (i.e. the domain of $\kappa$ is $\mathcal{H}$.)
- $\kappa^\dagger \equiv \kappa$. (i.e. $\kappa$ is Hermitian.)
- $\langle \phi | \kappa \phi \rangle > 0$, for all $\phi \in \mathcal{H}$ and $\phi \neq 0$. (i.e. $\kappa$ is positive definite.)
- $\| \kappa \phi \| \leq \| \kappa \| \| \phi \|$, for all $\phi \in \mathcal{H}$. (i.e. $\kappa$ is bounded.)
- $\kappa A \equiv A^\dagger \kappa$.

We can now define an inner-product using the new metric operator $\kappa$,

$$\langle \phi | \vartheta \rangle_\kappa \equiv \langle \phi | \kappa \vartheta \rangle$$

for all $\phi \in \mathcal{H}$, $\vartheta \in \mathcal{H}$. (1.2)

Suppose $A$ is quasi-Hermitian, which implies that it might not be self-adjoint, $A^\dagger \neq A$. It is in fact $\kappa$-Hermitian with regards to the newly defined inner-product in the Hilbert space $\mathcal{H}_\kappa$,

$$\langle \phi | A \vartheta \rangle_\kappa = \langle \phi | \kappa A \vartheta \rangle = \langle \phi | \kappa \vartheta \rangle = \langle A \phi | \kappa \vartheta \rangle = \langle A \phi | \vartheta \rangle_\kappa.$$

Following the same reasoning as Theorem 1, we can prove that the eigenvalues of a quasi-Hermitian operator are real.

**Theorem 2.** The eigenvalues of a quasi-Hermitian operator $A$ are real.

**Proof.** Let $A f = \lambda f$, where $A$ is quasi-Hermitian with the inner-product (1.2) and $\lambda$ is an eigenvalue of $A$. It can be shown that

$$\langle f | A f \rangle_\kappa = \lambda \langle f | f \rangle_\kappa.$$
Chapter 1. Introduction into non-Hermitian Hamiltonians

But

\[ \langle f|A|f \rangle_{\kappa} = \langle A|f|f \rangle_{\kappa} \]
\[ = \langle \lambda|f|f \rangle_{\kappa} \]
\[ = \lambda^* \langle f|f \rangle_{\kappa}. \]

We now have \( \lambda = \lambda^* \) which implies that the eigenvalues are real.

If the metric \( \kappa \) is the identity operator, \( \kappa = I \), we find that \( A^\dagger = A \). The quasi-Hermitian operator \( A \) now falls in the Hermitian set. This implies that,

Hermitian \( \subset \) Quasi-Hermitian.

A more general operator is the pseudo-Hermitian operators [8] since it does not necessarily need to be positive definite.

**Definition 2 (Pseudo-Hermitian Operator).** An operator \( A : \mathcal{H} \rightarrow \mathcal{H} \), where \( \mathcal{H} \) denotes a separable Hilbert space, is pseudo-Hermitian if there exist an operator \( \eta : \mathcal{H} \rightarrow \mathcal{H} \) such that,

\begin{itemize}
  \item \( \eta^{-1} \) exists. \( \eta \) is invertible.
  \item \( \eta^\dagger = \eta \). \( \eta \) is Hermitian (self-adjoint).
  \item \( \eta \) is linear.
  \item \( A^\dagger = \eta A \eta^{-1} \).
\end{itemize}

It is clear from this definition given by Mostafazadeh [8], that if there does not exist an operator \( \eta \) as defined by Def. 2, \( A \) is not pseudo-Hermitian. If there exist infinitely many \( \eta \) operators that satisfy \( A^\dagger = \eta A \eta^{-1} \), then \( A \) is pseudo-Hermitian.

Given a pseudo-Hermitian operator \( A \), we denote the set of all \( \eta \) operators satisfying \( A^\dagger = \eta A \eta^{-1} \) by \( \Omega (A) \). The operator \( A \) will be quasi-Hermitian, if \( \Omega (A) \) includes a positive operator \( \eta_+ \). A positive operator \( \eta_+ \) is self-adjoint with nonnegative eigenvalues.

According to Def. 2, all the elements of \( \Omega (A) \) are invertible, but \( \eta_+ \) is positive definite and will only have positive eigenvalues.

If \( B \) is a quasi-Hermitian operator, the operator \( \eta_+ \) belonging to \( \Omega (B) \) will not be unique. Therefore, any two operators, \( \eta_+ \) and \( \eta'_+ \), are related.

A quasi-Hermitian operator is always pseudo-Hermitian for some \( \eta \), where neither \( \eta \) nor \( -\eta \) are positive definite. We can now state that a pseudo-Hermitian operator may or may not be quasi-Hermitian. Therefore,

Hermitian \( \subset \) Quasi-Hermitian \( \subset \) Pseudo-Hermitian.
1.1.3 \( \mathcal{PT} \)-Symmetry

Hermiticity is a strong mathematical restriction and the physical requirements are quite far reaching. \( \mathcal{PT} \)-symmetry gives a more physical restriction to the Hamiltonian.

\( \mathcal{PT} \)-symmetry implies that a Hamiltonian is invariant under the combined reversal of parity (\( \mathcal{P} \)) and time (\( \mathcal{T} \)). A \( \mathcal{PT} \)-symmetric Hamiltonian will not be affected by the time reversal, \( p \to -p, \ x \to x, \ i \to -i \) and spatial reflection, \( p \to -p \) and \( x \to -x \).

Suppose we have the \( \mathcal{PT} \)-symmetric Hamiltonian \( H \), then it will be true that

\[
\mathcal{PT} H (\mathcal{PT})^{-1} = \mathcal{PT} H \mathcal{PT} = H. \tag{1.3}
\]

For \( H \) to have a real spectrum, \( \mathcal{PT} \)-symmetry needs to remain unbroken \([1, 2]\). Although Schrödinger’s equation and the boundary conditions are \( \mathcal{PT} \)-symmetric, the eigenfunctions may not be symmetric under the \( \mathcal{PT} \) operator. When the eigenfunctions, the solutions of the Schrödinger equation, retains the \( \mathcal{PT} \)-symmetry, we say that the \( \mathcal{PT} \)-symmetry is unbroken.

A complex, \( \mathcal{PT} \)-symmetric Hamiltonian will only have a real spectrum when the symmetry is not spontaneously broken. For example, if the potential remains \( \mathcal{PT} \)-symmetric after the transformation, but the corresponding eigenfunction does not, the symmetry is broken \([9]\). This implies that \( H \) and \( \mathcal{PT} \) must have the same eigenfunctions.

**Theorem 3.** Let the Hamiltonian \( H \) be \( \mathcal{PT} \)-symmetric and assume that the symmetry remains unbroken, then the energy spectrum of \( H \) is real.

**Proof.** Assume \( H \) is \( \mathcal{PT} \)-symmetric, \( \phi \) is an eigenstate of \( H \) with the eigenvalue \( E \) and \( \phi \) is also the eigenstate of \( \mathcal{PT} \) with the eigenvalue \( \lambda \).

We now have

\[
H \phi = E \phi \quad \text{(1.4)}
\]

\[
\mathcal{PT} \phi = \lambda \phi. \quad \text{(1.5)}
\]

Next we multiply (1.5) from the left with \( \mathcal{PT} \). Since \( \mathcal{P} \) and \( \mathcal{T} \) commute and \( \mathcal{P}^2 = 1 = \mathcal{T}^2 \) is true\(^5\), we find that

\[
\mathcal{PT}\mathcal{PT} \phi = \mathcal{PT} \lambda \phi, \\
\phi = \lambda^* \mathcal{PT} \phi, \\
\phi = \lambda^* \lambda \phi.
\]

\(^5\)In the article \([10]\) by Bender, preceding \([2]\), it was shown that \( \mathcal{PT} \)-symmetry is a generalization of Hermiticity and therefore \( \mathcal{P}^2 = \mathcal{T}^2 = 1 \) will hold.
We can now write $\lambda$ as $\lambda = e^{i\alpha}$ where $\alpha \in \mathbb{R}$. Now replace $\phi$ with $\phi e^{-i\alpha/2}$ which will give

$$\mathcal{PT} \phi = \phi.$$  

(1.6)

By multiplying (1.4) from the left with $\mathcal{PT}$, we find that

$$H\phi = E\phi,$$

$$\mathcal{PT} H \phi = \mathcal{PT} E \phi,$$

$$H \mathcal{PT} \phi = E^* \mathcal{PT} \phi,$$

$$H \phi = E^* \phi,$$

$$E \phi = E^* \phi,$$

$$E = E^*,$$

which implies that $E$ is real.

Consider the following Hamiltonian,

$$H = p^2 - (ix)^N, \text{ } N \text{ real.}$$

(1.7)

For most values of $N$ this Hamiltonian is entirely complex. When we apply the $\mathcal{PT}$ operator to $H$,

$$\mathcal{PT} H = (-p)^2 - (-ix)^N,$$

$$\mathcal{PT} \mathcal{PT} H = p^2 - (ix)^N,$$

we find that it is $\mathcal{PT}$-symmetric and the spectrum is real, see Figure 1.1.

Another Hamiltonian $p^2 + (ix)^3 + x$ is complex but when we apply the $\mathcal{PT}$ operator, we find that the symmetry is broken. The energy spectrum is therefore not real.

Another way of looking at $\mathcal{PT}$-symmetry is that it is a condition for selecting a specific Riemann sheet which could cope with the branch point of the potential at $x = 0$, [3].

Although $\mathcal{PT}$-symmetry is not a sufficient condition for a Hamiltonian to have a real spectrum, it does open the doors to a new family of potentials which might be useful in quantum mechanics. In this thesis we will only consider the $\mathcal{PT}$-symmetric potentials $V(x) = -(ix)^N$ and $V(x) = -(i \sinh x)^\alpha \cosh \beta x$. Different numerical methods will then be used for solving these problems. The methods will be compared and their strengths and weaknesses will be discussed.
1. Introduction into non-Hermitian Hamiltonians

Figure 1.1: Spectrum for the Hamiltonian $H = p^2 - (ix)^N$ as obtained by Bender and Boettcher [1].

1.2 Contours in the Complex plane

In pseudo-Hermitian quantum mechanics the physical Hilbert space $\mathcal{H}_{\text{phys}}$ is determined by the eigenvalue problem, with a Hamiltonian that acts on a reference Hilbert space $\mathcal{H}$ [4]. When a system has a finite-dimensional domain, we define $\mathcal{H}$ to be the complex Euclidean space. With $\mathcal{P}\mathcal{T}$-symmetrical systems defined on the real axis, we choose $\mathcal{H}$ to be $L_2(\mathbb{R})$. Some $\mathcal{P}\mathcal{T}$-symmetrical Hamiltonians need to be defined on a complex contour $\Gamma$ and there is not a useful Hilbert space for these operators. We would like to redefine our $\mathcal{P}\mathcal{T}$-symmetric Hamiltonians in the Hilbert space $L_2(\mathbb{R})$.

Suppose $\mathcal{F}$ is a set of analytical functions $\psi : \mathbb{R} \rightarrow \mathbb{C}$ and $H$ is a linear operator $H : \mathcal{F} \rightarrow \mathcal{F}$ with the general form

$$H = (p - A(x))^2 + V(x),$$

where $A(x)$ and $V(x)$ are piecewise analytic functions in $\mathcal{F}$ and $p \psi(x) = -i \psi'(x)$, for all $\psi$ in $\mathcal{F}$. Consider the potential $V(x) = -(ix)^N$ where it has been shown that it will have a real and positive spectrum for $N \geq 0$ as long as it is solved along a contour $\Gamma$. The domain of this Hamiltonian is related to the choice of the asymptotic boundary conditions. It will have a real and discrete spectrum if we define the boundary conditions along an

---

$^6$Complex Euclidean inner-product: $\langle \psi|\phi \rangle = \psi^* \cdot \phi$
appropriate contour in the complex plane. We therefore need to identify
the eigenvalue problem of this Hamiltonian, expressed with a new complex
variable $z$:
\[
\left[ -\left( \frac{d}{dz} - iA(z) \right)^2 + V(z) \right] \Psi(z) = E_n \Psi_n(z), \quad \Psi_n(z) \to 0
\]
where $z$ moves along $\Gamma$.

The contour $\Gamma$ is generally a smooth curve parameterized with a real pa-
rameter $s$. Then there exists a continuous, piecewise differentiable function $\xi : \mathbb{R} \to \mathbb{C}$, such that $\Gamma = \{ \xi(s) | s \in \mathbb{R} \}$.

We can now redefine the boundary conditions of (1.9) as,
\[
\lim_{|s| \to \infty} |\Psi_n(\xi(s))| = 0 \quad \text{as} \quad s \to \pm \infty.
\]
The contour $\Gamma$ is not unique, but is required to stay within the Stokes
wedges as $|s|$ tends to infinity; we will discuss this in the next section.

We choose our contour $z = x + iy$, as well as our coordinate system
$\mathbb{R}^2 = \mathbb{C}$, in such a way that it is a smooth and increasing function of $x$. We
can now express the function $\xi$ in terms of another differentiable function $f : \mathbb{R} \to \mathbb{R}$,
\[
\xi(x) = x + if(x).
\]

Now that we have defined a general formula for our contour $\Gamma$, we would
like to write (1.9) in terms of a complex variable, $z = \xi(x)$, defined along
the contour $\Gamma$. With the change of variables $z = x + if(x)$, we get
\[
\left[ -g(x)^2 \left( \frac{d}{dx} - ia(x) \right)^2 + ig(x)^3 \left( \frac{d^2}{dx^2} f(x) \right) \left( \frac{d}{dx} - ia(x) \right) + v(x) \right] \psi_n(x) = E_n \psi_n(x),
\]
where the functions are defined by
\[
\begin{align*}
g(x) &= \xi'(x)^{-1} = \left( 1 + i \frac{d}{dx} f(x) \right)^{-1} \\
a(x) &= g(x)^{-1} A(x + if(x)) \\
v(x) &= V(x + if(x)) \\
\psi_n(x) &= \Psi_n(x + if(x)).
\end{align*}
\]

The boundary conditions of this new eigenvalue problem becomes $\psi(x) \to 0$, as $x \to \pm \infty$ and $\psi_n(x)$ lives in the domain $L^2(\mathbb{R})$. The eigenvalue prob-
lem (1.9) for the Hamiltonian (1.8) is equivalent to the eigenvalue problem
of the Hamiltonian,
\[
H' = g(x)^2 (p - a(x))^2 - g(x)^3 (p - a(x)) \frac{d^2}{dx^2} f(x) + v(x).
\]
Suppose \( \zeta : \mathbb{R} \to \mathbb{C} \) is \( \mathcal{PT} \)-symmetric, this implies that \( \mathcal{PT}\zeta\mathcal{PT} = \zeta \) according to (1.3). When we apply this to our Hamiltonian \( H' \), we get

\[
\mathcal{PT}H'\mathcal{PT} = (g(-x)^*)^2 (p - a(-x)^*)^2 - (g(-x)^*)^3 (p - a(-x)^*) \frac{d^2}{dx^2} f(-x)^* + v(-x)^*.
\]

For \( H' \) to be \( \mathcal{PT} \)-symmetric, the following must be true:

\[
(g(-x)^*)^2 = g(x)^2, \\
g(-x)^* f(-x)^* = g(-x)^* \frac{d^2}{dx^2} f(x), \\
a(-x)^* = a(x), \quad v(-x)^* = v(x).
\]

Because \( f \) is real and \( x \) can be 0, we have

\[
f(x) = -f(x), \\
A(u)^*|_{u = -(x + if(x))} = A(x + if(x)), \\
V(u)^*|_{u = -(x + if(x))} = V(x + if(x)).
\]

The conditions (1.11) imply that \( z(-x)^* = z(x) \). The fact that \( H' \) is \( \mathcal{PT} \)-symmetric therefore implies that the contour \( \Gamma \) has a spatial reflection about the imaginary axis. Because \( A \) and \( V \) can be analytically continued onto the contour \( \Gamma \), the equations (1.12) and (1.13) imply that they are separately \( PT \)-symmetric.

We can now conclude that the Hamiltonian \( H \) (1.8) and the contour \( \Gamma \) are \( \mathcal{PT} \)-symmetric if and only if the Hamiltonian \( H' \) (1.10) is \( \mathcal{PT} \)-symmetric.

### 1.2.1 Stokes Wedges

\( \mathcal{PT} \)-symmetric potentials, such as \( V(x) = -(ix)^N \), are usually complex except for some specific choices of the parameters, in this case \( N \). Therefore, we need to calculate the energies along a contour in the complex plane. More specifically, we need to solve it between the Stokes wedges. These wedges are areas in the complex plane where the eigenfunctions go to zero as \( x \to \pm \infty \). If we choose any contour within these wedges, we will get real, positive eigenvalues.

For example, consider the contour

\[
z = -2i \sqrt{1 + iw},
\]

where \( w \) is a real parameter.
Chapter 1. Introduction into non-Hermitian Hamiltonians

Figure 1.2: These are the Stokes wedges where \( N = 4.5 \) and the contour is \( z = -2i\sqrt{1 + iw} \).

These wedges are bounded by the \textit{Stokes lines} and has an opening angle of \( \Delta = \frac{2\pi}{N+2} \). In the middle of each wedge, lies an \textit{anti-Stokes line} at an angle of,

\[ \theta_{\text{Left}} = -\pi + \frac{\pi}{2} \left( \frac{N-2}{N+2} \right), \quad \theta_{\text{Right}} = -\frac{\pi}{2} \left( \frac{N-2}{N+2} \right). \]  

Figure 1.3: These are the Stokes wedges where \( N = 10 \) and the contour is \( z = i\mu(\sin(\alpha) + \sin(iw - \alpha)) \).
We can solve the problem as long as the contour goes to infinity within these wedges. This is not always the case as it is shown in Figure 1.3. We therefore need to choose our contours carefully and cannot assume that a certain contour will work for all values of $N$.

### 1.2.2 Quantum Toboggans

Quantum Toboggans [11, 12] are contours with more complex geometry than the contours we have discussed up to now.

Contours by definition can have any shape as long as they go to infinity as the real coordinate $x$ goes to $\pm \infty$. Furthermore, the boundary conditions need to hold for the new complex coordinate $z \in C(x)$, where $x \in \mathbb{R}$.

Toboggans are trajectories in the complex plane that spiral around the branch points of a potential and connect two different Riemann sheets.

We start with a basic contour

\[
z = z(x) = x - i\varepsilon, \quad x \in (-\infty, \infty),
\]

where $\varepsilon > 0$. If we solve the anharmonic oscillator, for example,

\[
\left[-\frac{d^2}{dx^2} + \frac{\ell(\ell + 1)}{z^2} + \omega^2 z^2\right] \phi_n(z) = E_n(\omega) \phi_n(z),
\]

we will get a real and discrete spectrum when we solve this eigenvalue problem. As we already mentioned, a contour can have any shape. We can now extend (1.17) to

\[
z(x) = x - i\varepsilon(x), \quad x \in (-\infty, \infty)
\]

where $\varepsilon(x) = \varepsilon(-x) > 0$. This general formula can further be transformed with the aim to form toboggans,

\[
\begin{align*}
z(x(\gamma)) &= -i\rho(\gamma)e^{i\gamma} \\
\rho(\gamma) &= \frac{\varepsilon(x(\gamma))}{\cos \gamma}
\end{align*}
\]

where $x(\gamma) = \epsilon \tan \gamma$ using $\gamma \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. The choice of the function $\varepsilon(x)$ is entirely arbitrary given that it has the required shape and $\epsilon$ is sufficiently small.

Finally, we allow the contour to run to other Riemann sheets. We modify the contour (1.19) for a last time by introducing a winding number $w$, which allows the contour to wind $w$-times around a branch point,

\[
z_w(x(\gamma)) = -i\rho(\gamma)^{2w+1}e^{i\gamma(2w+1)}.
\]
The new coordinate will be defined on the entire logarithmic Riemann surface $\mathcal{R}$ and the eigenfunctions will all have the general form,

$$\phi(z) \sim c_1 z^{\ell+1} + c_2 z^{-\ell}, \quad |z| \ll 1,$$

close to the origin and only the angle of (1.20) will show distinction between different Riemann sheets.

The eigenfunctions of (1.18) are still bound but now by the asymptotic boundary conditions

$$\lim_{\gamma \to \pm \infty} \phi_n(z(x(\gamma))) = 0,$$

which lie on two separate Riemann sheets.
Chapter 2

Perturbation Theory

In the book of Bender and Orszag [13, p. 330] a method is described that can be used to approximate eigenvalues for the Schrödinger equation of the form,

\[ \left( -\frac{d^2}{dx^2} + V(x) + \varepsilon W(x) - E \right) \psi(x) = 0, \tag{2.1} \]

subject to the boundary conditions

\[ \lim_{x \to \pm \infty} \psi(x) = 0. \]

The sum \( V(x) + \varepsilon W(x) \) represents the potential and is a continuous function that approaches \( \infty \) as \( x \to \pm \infty \). The assumption is made that the potential \( V(x) + \varepsilon W(x) \) makes it difficult to solve (2.1), except when \( \varepsilon = 0 \). When \( \varepsilon = 0 \), the potential reduces to \( V(x) \) and (2.1) is exactly solvable.

We are interested in the problem

\[ \left( -\frac{d^2}{dx^2} - (ix)^N - E \right) \psi(x) = 0 \tag{2.2} \]

where the potential \(-(ix)^N\) is \( \mathcal{PT}\)-symmetric. Before we start, we need to write equation (2.2) in the same form as (2.1).

First, we rewrite the potential by defining \( \varepsilon = N - 2 \). We continue by choosing \( V(x) = x^2 \) and \( W(x) = -(ix)^2 + \varepsilon - x^2 \). \( V(x) + W(x) \) becomes \( x^2 \) when \( \varepsilon \) tends to zero. The potential reduces to the harmonic oscillator when we look at the unperturbed problem, which have a known solution,

\begin{align*}
\psi_p &= C e^{-x^2/2} \text{He}_p(x), \quad p = 0, 1, 2, \ldots, \tag{2.3} \\
E_p &= 2p + 1, \tag{2.4}
\end{align*}
where $H_p(x)$ denotes the Hermite polynomial of degree $p$ and $C$ is an arbitrary normalization constant.

We have written the potential in the form $V(x) + W(x)$ but this will not give us a problem in the same form as (2.1). We do this by writing $W(x)$ as a power series in $\varepsilon$,

$$-x^2 - (ix)^{2+\varepsilon} = x^2 \log(i x) \varepsilon + \mathcal{O}(\varepsilon^2).$$

This yields, to first order in $\varepsilon$,

$$\left(-\frac{d^2}{dx^2} + x^2 + \varepsilon x^2 \log(i x) - E\right) \psi(x) = 0. \quad (2.5)$$

We now try to find a solution of the form,

$$E = \sum_{j=0}^{\infty} E_j \varepsilon^j, \quad (2.6)$$
$$\psi(x) = \sum_{j=0}^{\infty} \psi_j(x) \varepsilon^j. \quad (2.7)$$

Substituting (2.6) and (2.7) into (2.5) and by comparing the powers of $\varepsilon$ gives

$$\left[-\frac{d^2}{dx^2} + x^2 - E_0\right] \psi_j(x) = -\left[x^2 \log(i x)\right] \psi_{j-1}(x) + \sum_{k=1}^{j} E_k \psi_{j-k}(x).$$

We can obtain a recursion formula for $E_j$ and $\psi_j(x)$,

$$E_j = \int_{-\infty}^{\infty} dx \psi_0(x) \left(x^2 \log(i x) \psi_{j-1}(x) - \sum_{k=1}^{j-1} E_k \psi_{j-k}(x)\right), \quad (2.8)$$
$$\psi_j(x) = \psi_0(x) \int_{a}^{x} \frac{dt}{\psi_0^2(t)} \int_{-\infty}^{t} ds \psi_o(s) \left(s^2 \log(is) \psi_{j-1}(s) - \sum_{k=1}^{j} E_k \psi_{j-k}(s)\right).$$

Here $a$ is an arbitrary constant such that $\psi_j(a) = 0$ and $j = 1, 2, 3, \ldots,$

With these recursion formulas we can calculate $E_j$ and $\psi_j$ which will be used along with (2.6) and (2.7) to determine $E$ and $\psi$.

The eigenvalues and -functions of the harmonic oscillator are known and are given by (2.3) and (2.4). To illustrate this perturbation method we will only approximate the first two energy levels.
First, we approximate the ground level eigenvalue. By using the recursion formula for the eigenvalues (2.8), we have,

\[
E_1 = \int_{-\infty}^{\infty} dx \psi_0 \left( x^2 \log(ix) \psi_0 \right)
= \frac{1}{4} (2 - \gamma_E - \log 4),
\]

(2.9)

where \( \gamma_E \) is the euler-gamma constant\(^1\).

By substituting these energies into (2.6) we get the approximation,

\[
E_0 = \sum_{j=0}^{1} \mathcal{E}_j \varepsilon^j
= \mathcal{E}_0 + \mathcal{E}_1 \varepsilon
= 1 + \frac{1}{4} (N - 2)(2 - \gamma_E - \log 4)
= 0.981755 + 0.00912249 N.
\]

(2.10)

For the next energy level \((p = 1)\) we have \( \mathcal{E}_0 = 3 \) and \( \psi_0 = 2x e^{-x^2/2} \text{He}_1(x) \), and with the recursion formula we obtain

\[
E_1 = \int_{-\infty}^{\infty} dx \psi_0 \left( x^2 \log(ix) \psi_0 \right)
= \frac{1}{4} (8 - 3\gamma_E - \log 64).
\]

(2.11)

We again substitute these energies into (2.6) to get the approximation,

\[
E_1 = \sum_{j=0}^{1} \mathcal{E}_j \varepsilon^j
= 1 + \frac{1}{4} (N - 2)(8 - 3\gamma_E - \log 64)
= 1.94527 + 0.527367 N.
\]

(2.12)

When we continue in this fashion we can get first order approximations to all of the energy levels. The next few values are

\[
E_2 = 1.90878 + 1.54561 N,
E_3 = 1.53895 + 2.73052 N,
E_4 = 0.835795 + 4.0821 N.
\]

\(^1\gamma_E \approx 0.577216.\)
In Figure 2.1 we see the first order approximation plotted along with numerical results of the potential. The solution calculated with this perturbation method gives a good approximation for the eigenvalues near $N = 2$. It gets harder when we try to calculate the higher order approximations.

\[ (-ix^N) \]

Figure 2.1: The first order approximation to the perturbed harmonic oscillator $(-ix^N)$ along with the spectrum of the potential. A higher order approximation will approach the spectrum more accurately at $N = 2$. 
Chapter 3

Numerical Methods

In this chapter we will be looking at different numerical methods for solving a $\mathcal{PT}$-symmetric problem. For each method we will look at their strengths and weaknesses by comparing them with each other. These methods were implemented in MATLAB and Mathematica and most of the program code will be provided.

3.1 Shooting Methods

3.1.1 Introduction

Shooting methods are widely used for obtaining the eigenvalues of boundary value problems [14]. We are only interested in the eigenvalues of the Schrödinger equation,

\[ \psi''(x) = (V(x) - E) \psi(x), \quad \lim_{x \to \pm \infty} \psi(x) = 0. \] (3.1)

The shooting method solves the problem in two parts. First we need to choose a matching point $x_m$, which lies within the boundaries of the problem, as well as an estimated eigenvalue. Now we use the Runge-Kutta method to solve the boundary value problem as two initial value problems; each is solved by starting at a different boundary and working towards $x_m$. When the two parts of the eigenfunction match smoothly at $x_m$, we have chosen the correct eigenvalue.

Consider the following, more general, problem,

\[ \psi''(x) = w(x, E) \psi(x), \quad x \in (a, b) \] (3.2)
Figure 3.1: Determining $\psi(x)$ by converting the boundary value problem into two initial value problems.

where $w(x, E)$ represents $(V(x) - E)$. The boundary values are defined by

$$a_0 \psi(a) + b_0 \psi'(a) = 0,$$
$$a_1 \psi(b) + b_1 \psi'(b) = 0,$$

where $a_0$ and $b_0$ are not simultaneously zero. The same condition applies to $a_1$ and $b_1$.

Our first step is to transform this boundary value problem into two first order, initial value problems.

$$u'_L(x) = w(x, E)\psi_L(x) \quad u'_R(x) = w(x, E)\psi_R(x)$$
$$\psi'_L(x) = u_L(x) \quad \psi'_R(x) = u_R(x)$$
$$\psi_L(a) = -b_0 \quad \psi'_L(a) = a_0 \quad \psi_R(b) = -b_1 \quad \psi'_R(b) = a_1$$

Now we choose an initial estimate for the eigenvalue $E$ and solve $\psi_L(x)$ on the interval $[a, x_m]$ and $\psi_R(x)$ on $[x_m, b]$. We continue to adapt the value of $E$ until the two functions are equal at the matching point, in value and gradient; only then do we have the correct eigenvalue. This means that if we have chosen the exact eigenvalue, then $\psi_L(x_m) = \psi_R(x_m)$ and $\psi'_L(x_m) = \psi'_R(x_m)$. These results can be verified by calculating the mismatch function $\phi(E)$ at the match point,
Chapter 3. Numerical Methods

\[ \phi(E) = \begin{vmatrix} \psi_L'(x_m, E) & \psi_R'(x_m, E) \\ \psi_L(x_m, E) & \psi_R(x_m, E) \end{vmatrix}. \]  \hspace{1cm} (3.3)

By finding the roots of (3.3), we can find the eigenvalues.

We will now apply this method to the \( PT \)-symmetric potential \( V(x) = -(ix)^N \),

\[ -\psi''(x) = (E + (ix)^N)\psi(x), \quad \lim_{x \to \pm \infty} \psi(x) = 0. \] \hspace{1cm} (3.4)

First we create the two initial value problems. One for the function left of the matching point and one for the function on the right, each with their own initial values. For the Schrödinger problem the boundaries are \((-\infty, \infty)\) and need to be restricted for practical application. We therefore need to introduce a new parameter \( L \) such that the new boundaries are given by \([-L, L]\). This parameter can affect the accuracy of this method and will need to be adjusted for different potentials, Figure 3.2.

![Figure 3.2: The absolute error of the ground state, first and fifth energy level of the harmonic oscillator calculated with the shooting method for different values of \( L \).](image)

The boundary conditions of this problem are given by

\[ \psi_L(a) = 0, \quad \psi_L'(a) = 1, \]
\[ \psi_R(b) = 0, \quad \psi_R'(b) = 1, \]

where \( a = -L \) and \( b = L \).
In Figure 3.3 it is clearly seen that the roots of $\phi(E)$ approximate the eigenvalues\(^1\). Figures such as this can provide good initial guesses for solving

$$\phi(E) = 0$$

iteratively. One iterative solver is, for example, the secant method,

$$E_{n+1} = E_n - \frac{\phi(E_n)}{\phi(E_n) - \phi(E_{n-1})}.$$

(3.5)

A more efficient method is the function `fzero` built into MATLAB, based on Brent’s idea of inverse parabolic interpolation.

We know that if $N = 2$, we have the harmonic oscillator. In Table 3.1 we illustrate how this algorithm approaches the exact value of the ground state eigenvalue, $E_0 = 1$. We choose $x_m = 0$ and start with the initial eigenvalues 0 and 0.75.

Table 3.1 shows a rapid decrease in the matching function’s value as the eigenvalue approaches 1.

**Implementation and Analysis**

The MATLAB implementation of the shooting method can be found in the Appendix, Tables A.1 and A.2. The potential is entered in line 10 of

---

\(^1\)The eigenvalues are $E = 1, 3, 5, \ldots$ as given by (2.4).
Table 3.1: The progression of the algorithm as we approach the ground state eigenvalue of the harmonic oscillator, as computed by the secant method (3.5) using the initial guesses 0 and 0.75.

\[ E \begin{bmatrix} 0.00000 & 0.75000 & 1.15044 & 0.99441 & 0.99994 & 1.00000 \\ \phi(E) \end{bmatrix} \begin{bmatrix} 2.00000 & 0.69615 & -0.44442 & 0.01631 & 0.00018 & -1.59E-7 \end{bmatrix} \]

Before the method is applied, there are a few choices that need to be made. The function \texttt{ode45} has two options that need to be set, the \textit{Absolute Tolerance} and the \textit{Relative Tolerance}. The default values (\(1 \times 10^{-6}\) and \(1 \times 10^{-3}\) for the absolute and relative tolerance respectively) can be used but will have to be adjusted if higher accuracy is required. Another function with an option to set is \texttt{fzero}. Through experimentation we found that accuracy of the solution is not affected by adjusting the tolerances of \texttt{fzero}.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure34.png}
\caption{The effects of \texttt{ode45}'s relative tolerance on the accuracy and runtime (in seconds) of the shooting method.}
\end{figure}

It was found that adjusting the absolute tolerance does not show any improvement or decline in the accuracy or the runtime of the calculation. The default value for the absolute tolerance is therefore sufficient for this exercise. On the other hand, adjusting the relative tolerance does affect the solution. In Figure 3.4 we see two graphs. The first graph is a plot of the error against the relative tolerance. The accuracy of the eigenvalues increases dramatically as we choose a smaller relative tolerance. The price
of this accuracy is a steep incline in the cpu-time as seen in the second figure.

To compare the entire spectrum of $V(x) = -(ix)^N$, it is satisfactory to use the default values for the tolerances of ode45. When we need to investigate the eigenvalues corresponding to a specific choice of the parameter $N$, higher accuracy may be required. For now we are interested in reliably reproducing Figure 1.1.

The solution obtained with the shooting method shown in Figure 3.5 compares well with the one obtained by Bender and Boettcher [1] except for the discontinuity when $N = 4$. This is due to the complicated nature of the potential at $N = 4$.

When $N = 4$ the potential reduces to $-x^4$. This potential is not bounded from below and therefore does not have any real eigenvalues.

Up to now, we have only worked in the real space. To compute the spectrum in the complex plane we need to solve the Schrödinger equation along a contour within the Stokes wedges as discussed in Section 1.2.1. After we choose a contour, for example $z = -2i\sqrt{1+ix}$, we need to incorporate it into our Schrödinger equation (3.4). The problem can be converted to the new contour by the following transformations:

\footnote{See Figure 1.1 for the solution obtained by Bender.}
\[
\frac{d^2 \psi(z)}{dz^2} = \left( -E - (iz)^N \right) \psi(z)
\]
\[
\frac{d^2 \psi}{dx^2} \left( \frac{dx}{dz} \right)^2 + \frac{d\psi}{dx} \frac{d^2 x}{dz^2} = \left( -E - (iz)^N \right) \psi
\]
\[
- \frac{d^2 \psi}{dx^2} \left( \frac{z^2}{4} \right) + \frac{d\psi}{dx} \left( \frac{i}{2} \right) = \left( -E - (iz)^N \right) \psi
\]
\[
\frac{d^2 \psi}{dx^2} - \left( \frac{2i}{z^2} \right) \frac{d\psi}{dx} = \left( E + (iz)^N \right) \frac{4}{z^2} \psi
\]
\[
\psi''(x) = -\frac{E + 2^N (1 + ix)^{N/2}}{1 + ix} \psi(x) - \frac{i}{2(1 + ix)} \psi'(x).
\]

By solving the resulting differential equation (3.6), we get the spectrum shown in Figure 3.6.

Figure 3.6: The eigenvalues of \( V(x) = -(ix)^N \) calculated with the shooting method on the contour \( z = -2i \sqrt{1 + ix} \).

Note that the spectrum at the previous problem area (around \( N = 4 \)) is now computed accurately. The spectrum for smaller values of \( N \) cannot be used, because a contour only gives a good approximation for certain choices of \( N \).
3.1.2 Other Software Packages

There exists programs with already implemented numerical methods specifically designed for solving boundary value differential problems. These programs give reliable solutions. We will only briefly discuss a few of these programs in this section.

SLEIGN2 is a FORTRAN program that uses a Prüfer-based shooting method to compute the eigenvalues and eigenfunctions of Sturm-Liouville problems (which can be converted to the Schrödinger equation),

\[-(py')' + qy = \lambda wy, \text{ on } (a, b).\]

These second order linear differential equations, along with their boundary conditions, can be either regular or singular. A problem occurs when we look at the assumptions made by SLEIGN2. The assumption that hinders our calculations, is that \(p, q\) and \(w\) should be real-valued functions on the interval \((a, b)\). We can therefore not use SLEIGN2 (or its predecessor SLEIGN) to calculate the eigenvalues of \(\mathcal{PT}\)-symmetric potentials, at least not without modifications. SLEDGE is another Prüfer-based shooting method, but combines it with a piecewise constant midpoint approximation. The same assumptions prevent us from solving \(\mathcal{PT}\)-symmetric Schrödinger equations.

MATSLISE is a MATLAB package that calculates the eigenvalues of Sturm-Liouville, Schrödinger and distorted Coulomb problems using piecewise perturbation (PPM) and constant reference potential perturbation methods (CPM). This package is described in chapter 7 of [14]. It has been claimed that this MATLAB package is more accurate and efficient than the well-known SLEIGN or SLEDGE FORTRAN implementations.

Although MATSLISE solves Schrödinger problems with real potentials (e.g. the harmonic oscillator), it cannot work for \(V(x) = -(ix)^N\). MATSLISE, like SLEIGN2, requires that the potential be real on the given interval. This implies that \(V(x)\) can only be solved where \(N\) gives a real potential that is bounded from below. Table 3.2 shows us the error given by MATSLISE for two special cases of \(V(x)\) where the potential is real and bounded, namely when \(N = 2\) and \(N = 6\).

3.2 Spectral Collocation Methods

Spectral methods are commonly used to solve ordinary and partial differential equations. The essence of the spectral method is that it approximates the solution of the differential equation with a linear combination of functions that are continuous over the domain of the solution.
Table 3.2: In this table we see the estimated errors given by MATSLISE for the eigenvalues of the potential $V(x) = -(ix)^N$ where $N = 2$ and $N = 6$.

<table>
<thead>
<tr>
<th>n</th>
<th>$N = 2$</th>
<th>$N = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$-1.2434 \times 10^{-14}$</td>
<td>$3.7748 \times 10^{-15}$</td>
</tr>
<tr>
<td>1</td>
<td>$-5.3291 \times 10^{-15}$</td>
<td>$-7.1054 \times 10^{-15}$</td>
</tr>
<tr>
<td>2</td>
<td>$3.5527 \times 10^{-14}$</td>
<td>$5.3291 \times 10^{-15}$</td>
</tr>
<tr>
<td>3</td>
<td>$2.6468 \times 10^{-13}$</td>
<td>$-4.2633 \times 10^{-14}$</td>
</tr>
<tr>
<td>4</td>
<td>$-1.7231 \times 10^{-13}$</td>
<td>$-2.1316 \times 10^{-13}$</td>
</tr>
<tr>
<td>5</td>
<td>$-1.4033 \times 10^{-13}$</td>
<td>$-1.6698 \times 10^{-13}$</td>
</tr>
<tr>
<td>6</td>
<td>$3.5527 \times 10^{-14}$</td>
<td>$8.5265 \times 10^{-14}$</td>
</tr>
<tr>
<td>7</td>
<td>$-1.4211 \times 10^{-14}$</td>
<td>$-5.6843 \times 10^{-14}$</td>
</tr>
<tr>
<td>8</td>
<td>$2.7001 \times 10^{-13}$</td>
<td>$5.8975 \times 10^{-13}$</td>
</tr>
<tr>
<td>9</td>
<td>$-1.0658 \times 10^{-14}$</td>
<td>$2.8422 \times 10^{-14}$</td>
</tr>
<tr>
<td>10</td>
<td>$9.5923 \times 10^{-14}$</td>
<td>$1.0942 \times 10^{-12}$</td>
</tr>
</tbody>
</table>

For the spectral collocation method, we choose a grid $\{x_0, x_1, \ldots, x_n\}$ as well as a set of basis functions for the approximation, $\{\rho_0, \rho_1, \ldots, \rho_n\}$.

We can now approximate a function $f(x)$ with

$$f(x) = a_0\rho_0(x) + a_1\rho_1(x) + a_2\rho_2(x) + \ldots + a_n\rho_n(x). \quad (3.7)$$

Furthermore, a collocation method requires that the left-hand and right-hand side of (3.7) be equal at the grid points, which gives

$$f(x_0) = a_0\rho_0(x_0) + a_1\rho_1(x_0) + a_2\rho_2(x_0) + \ldots + a_n\rho_n(x_0),$$

$$f(x_1) = a_0\rho_0(x_1) + a_1\rho_1(x_1) + a_2\rho_2(x_1) + \ldots + a_n\rho_n(x_1),$$

$$\vdots$$

$$f(x_n) = a_0\rho_0(x_n) + a_1\rho_1(x_n) + a_2\rho_2(x_n) + \ldots + a_n\rho_n(x_n).$$

This can be rewritten as,

$$
\begin{bmatrix}
  f(x_0) \\
  f(x_1) \\
  \vdots \\
  f(x_n)
\end{bmatrix} = 
\begin{bmatrix}
  \rho_0(x_0) & \rho_1(x_0) & \ldots & \rho_n(x_0) \\
  \rho_0(x_1) & \rho_1(x_1) & \ldots & \rho_n(x_1) \\
  \vdots & \vdots & \ddots & \vdots \\
  \rho_0(x_n) & \rho_1(x_n) & \ldots & \rho_n(x_n)
\end{bmatrix} 
\begin{bmatrix}
  a_0 \\
  a_1 \\
  \vdots \\
  a_n
\end{bmatrix}.
\quad (3.8)
$$

### 3.2.1 Method 1a: DMSUITE

The first method we are going to investigate is one proposed by Weideman [15]. We start by solving the Schrödinger equation

$$\psi''(x) + (E - V(x))\psi(x) = 0 \quad (3.9)$$
for the quantum harmonic potential, \( V(x) = x^2 \) [13, p. 28] where the eigenvalues and eigenfunctions are known, see (2.3) and (2.4).

Considering the form of (2.3), we can use this as a basis for an approximation method,

\[
\psi(x) = e^{-x^2/2} \sum_{k=0}^{\infty} c_k H_k(x),
\]

for \(-\infty < x < \infty\) where we need to find the coefficients.

There are a few methods available for determining \( c_k \), namely Galerkin, Rayleigh and Ritz among others. In the collocation method we would like to rewrite the polynomial in Lagrange form. Before applying this, we need to truncate the series (3.10) and define a real valued mesh of \( n + 1 \) nodes, \( \{x\}_{k=0}^{n} \).

\[
\psi(x) \approx e^{-x^2/2} \sum_{k=0}^{n} e^{x_k^2/2} \psi_k L_k(x).
\]

(3.11)

\( L_k(x) \) represents the Lagrange interpolating polynomial

\[
L_k(x) = \prod_{j=0, j \neq k}^{n} \frac{x - x_j}{x_k - x_j},
\]

and \( \psi_k \equiv \psi(x_k) \).

We now have a weighted polynomial interpolant (3.11). We can rewrite the interpolant in a more general form,

\[
\psi_n(x) = \sum_{k=0}^{n} \frac{w(x)}{w(x_k)} \psi_k L_k(x),
\]

(3.12)

where \( w(x) = e^{-x^2/2} \) in our case.

By approximating the eigenfunction of (3.9) with (3.12) we get,

\[
\psi_n'(x) + (E - V(x))\psi_n(x) \approx 0.
\]

Furthermore, a collocation method requires that the solution satisfies the given equation at all the nodes; this implies that we have equality for all \( x = x_j, \ j = 0, ..., n \):

\[
-\frac{d^2}{dx^2} \sum_{k=0}^{n} \frac{\psi_k}{w(x_k)} [w(x)L_k(x)]_{x=x_j} + (E - V(x_j))\psi(x_j) = 0.
\]
This equation can now be converted into matrix notation,

\[-D^{(2)}\psi + \text{diag}(V(x_j))\psi = E\psi,\]  \hspace{1cm} (3.13)

where \(\psi\) is a vector containing the function values \(\{\psi_j\}_{j=0}^n\) at the nodes \(\{x_j\}_{j=0}^n\) and \(D^{(2)}\) is a spectral derivative matrix given by,

\[D^{(2)}_{j,k} = \frac{1}{w(x_k)}(w''(x_j)L_k(x_j) + 2w'(x_j)L_k(x_j) + w(x_j)L_k(x_j)).\]

The diagonal matrix \(\text{diag}(V(x_j))\) contains the values of the potential calculated at the nodes. To obtain the eigenvalues and -functions, we numerically diagonalize the coefficient matrix in equation (3.13).

The algorithm to calculate \(D^{(2)}\) has been implemented as a MATLAB function called \texttt{herdif} and is part of the DMSUITE package [16]. The code uses the nodes, \(\{x_j\}_{j=0}^n\), and the weights \(\{w(x_j)\}_{j=0}^n\), \(\{w'(x_j)\}_{j=0}^n\) and \(\{w''(x_j)\}_{j=0}^n\) to produce the matrices \(D^{(\ell)}\) of any given order \(\ell\).

Theoretically, we can choose arbitrary nodes for a given weight \(w(x)\), but best accuracy is achieved by choosing the zeros of the Hermite polynomial, \(\text{He}_{n+1}\), which the function scales by a factor \(c\) at the mesh points. The Hermite differentiation matrix reproduces exact derivatives for the functions of the form,

\[\psi(x) = p(x)e^{-\frac{1}{2}(cx)^2},\]  \hspace{1cm} (3.14)

where \(p(x)\) is any polynomial with a degree of \(n\).

According to Boyd [17], the choice of the scaling parameter \(c\) is very important for convergence. This will be demonstrated in the next section.

**Implementation and Analysis**

Table 3.3 contains the MATLAB code that is used to calculate the eigenvalues of a given potential; in this case \(V(x) = -(ix)^N\). The MATLAB program \texttt{herdif} computes the differentiation matrices and \texttt{herroots} is used to calculate the roots of a Hermite polynomial of degree \(n\). Finally it makes use of MATLAB’s built-in function \texttt{eig} to calculate the eigenvalues of the matrix \(A\).

Before we apply the spectral collocation method on Schrödinger’s equation (3.9), we need to determine which value to use for the scaling parameter, \(c\). If we select a good value for \(c\), the eigenvalues determined by the spectral collocation method should be close to the exact eigenvalues. To speed up this process, we will only calculate the ground state energy at a specific value of \(N\) for different values of \(c\). By comparing these results with
function E = spectralHerm(n, c)
    [x, DM] = herdif(n, 2, c);
    D2 = DM(:, :, 2);
    for N = linspace(0, 6, 500)
        A = -D2 + diag(-(i*x).^N);
        E = sort(eig(A));
        E = E(1:10);
        ell = find(abs(imag(E)) < sqrt(eps));
        E = E(ell);
    end

Table 3.3: A MATLAB function for the computation of the eigenenergies of $V(x) = -(ix)^N$.

an exact value, we should be able to find the optimal $c$ value. Bender used a Runge-Kutta method in [1] to obtain $E_0 = 1.1562$ when $N = 3$. For our comparison we need a value with higher accuracy.

We obtain a more accurate reference value by using an extrapolation method. The epsilon algorithm, Table A.4, uses known approximate solutions to obtain a more accurate solution.

We get our first set of solutions by calculating the ground state energy for a specific $c$ and adjusting the matrix size. Because we do not know which $c$ will give us the best solution, we repeat this method for different values of $c$ and by doing this we increase our set of solutions for the epsilon algorithm to use.

It is clear from Figure 3.7 that $c \approx 1.8$ will give us a good approximation for a matrix of size $n = 32$. We run into trouble when the matrix size is increased. Because an increase in the matrix size increases the accuracy of the eigenvalues, we need a more accurate reference value. Although the number of significant digits has increased from 4 to 10 by using the epsilon algorithm, it still is not enough.

Another method for determining the scaling parameter is by comparing the derivative of an approximate eigenfunction with the derivative matrix calculated by `herdif`.

When we apply the spectral collocation method to a potential, where the spectrum is unknown, the choice of $c$ is arbitrary. Although $c$ is used to optimize our solutions, it can have a drastic effect on the entire spectrum. Ideally we would need to get approximate eigenvalues, for a certain choice of the parameters, obtained by another method. These eigenvalues can be used as reference values to confirm whether we have the correct spectrum.
Chapter 3. Numerical Methods

Figure 3.7: Figures for the ground state and first level energy of $V(x) = -ix^3$, calculated for different matrix sizes ($n = 32$ and $n = 96$) and compared with the exact values of $E_0$ of $E_1$ respectively to illustrate the dependence on the scaling parameter $c$.

For this potential.

Because this method is based on the harmonic oscillator, we would expect that it would give the exact eigenvalues when $N = 2$ and $c = 1$. We can now use this information to illustrate the dependence of the spectrum on the scaling parameter and the matrix size. Figure 3.8 shows the error using the known ground state energy of the harmonic oscillator and comparing it with the eigenvalues obtained at $N = 2$ for different matrix sizes and different $c$-values. In this figure we can see that the error decreases as we increase the matrix size $n$. At a certain point it seems as if the accuracy does not improve anymore; this is due to roundoff error.

Looking more closely at the error made at each energy level, see Figure 3.9, it is clearly visible that the accuracy diminishes as we look at higher level eigenvalues. Bigger matrices have higher accuracy but this only applies to the first few eigenvalues. The accuracy will start to decline at some point when we look at higher order energies due to the more oscillatory nature of the higher order modes.

Up until now a brief analysis was made of the method’s accuracy and it confirms that this is a highly accurate method, given a good choice of the scaling parameter, and was found that it calculates the spectrum in a short amount of time. The rest of this section will focus on the entire spectrum of $V(x) = -(ix)^3$, since this is what we are really interested in.
Figure 3.8: The logarithm of the absolute error of the ground state energy of the harmonic oscillator, $N = 2$, calculated at different matrix sizes and for different $c$-values. $c = 1$ is expected to give the exact eigenvalues for the harmonic oscillator.

Figure 3.9: The logarithm of the absolute error calculated for the first few energy levels, $k$, of the harmonic oscillator for three different matrix sizes ($n = 16, 32, 48$) and for $c = 2$. This figure illustrates the dependency of the accuracy of the eigenvalues on the matrix size.

Figure 3.10 displays the results for the $\mathcal{PT}$-symmetric potential $V(x) = -(ix)^N$. Considering that this was done purely on the real axis, it compares fairly well with Bender’s solution, Figure 1.1, except where $N \approx 4$ and $N < 0.5$, where a spurious spectrum appears.

When we look closely at the potential $-(ix)^4$, we see that it is unbounded from below and therefore have no real eigenvalues.
This is the result of our calculation on the real axis. Similar to the shooting method, we need to solve the problem along a contour that lies within the Stokes wedges. We will use the same contour used in Section 3.1, namely \( z(w) = -2i\sqrt{1 + iw} \) where \( w \in \mathbb{R} \).

The code in Table 3.3 needs to be modified for this contour calculations. We basically need to apply this change of variables to the differentiation matrix\(^3\),

\[
\tilde{D}^{(2)} = \text{diag} \left( \frac{dw}{dz} \right) D^{(2)} + \text{diag} \left( \frac{d^2w}{dz^2} \right) D^{(1)}.
\]

\(^3\)The derivatives of \( w \) can be obtained by: \( \frac{dw}{dz} = 1/\frac{dz}{dw} \) and \( \frac{d^2w}{dz^2} = -\frac{d^2z}{dw^2}/\left( \frac{dz}{dw} \right)^3 \).
In Figure 3.11 it is clearly visible that there is an improvement in the results in the region where \( N = 4 \), but now there is a problem for \( N < 2 \). If we recall the section about Stokes wedges, a contour does not work for all values of \( N \).

![Figure 3.12: Stokes wedges for \( N = 2 \) and the contour \( z = -2i\sqrt{\Gamma + iw} \).](image)

In Figure 3.12 we can clearly see that the contour does not go to infinity between the Stokes lines and therefore we do not get good results for values of \( N \) around, and smaller than 2. What will happen when we choose another contour?

When we try the contour \( z = i(\sin \alpha + \sin(iw - \alpha)) \) there is once again a drastic change in the spectrum. There is an improvement around \( N = 4 \), although the rest of the spectrum cannot be used.

It is clearly visible in Figure 3.13, that our results is again entirely dependent on the choice of the contour.

### 3.2.2 Method 1b: DMSUITE, the Fourier approach

Another approach is to solve the differential equation on a periodic domain. Previously we used the zeros of the Hermite polynomial when we calculated the differentiation matrix as well as the Lagrange interpolant; this Fourier method uses a trigonometric polynomial where the nodes are equispaced on a domain \([0, 2\pi]\). This can be rescaled to a more general domain \([-L, L]\), by the transformation \( x \rightarrow (x - \pi)L/\pi \). The interpolant on \([0, 2\pi]\) is given by

\[
t_N(x) = \sum_{j=1}^{N} \varphi_j(x)\psi_j
\]  

(3.15)
Figure 3.13: First we observe that there is an instability where $N < 2$ and $N \approx 6$. In the other two graphs we see that the contour does not stay within the wedges when $N < 1$ and $N \geq 7$, which might explain the instability.

where

$$\varphi_j(x) = \begin{cases} \frac{1}{N} \sin\left(\frac{N}{2}(x - x_j)\right) \cot\left(\frac{1}{2}(x - x_j)\right), & N \text{ even,} \\ \frac{1}{N} \sin\left(\frac{N}{2}(x - x_j)\right) \csc\left(\frac{1}{2}(x - x_j)\right), & N \text{ odd.} \end{cases}$$

Although the Schrödinger equation is not periodic, the solutions tend to zero as $x \to \pm\infty$ and can therefore be approximated as a periodic problem.

We can now solve the potential $V(x) = -(ix)^N$ using the Fourier approach. In Figure 3.14 we see that there is only a slight improvement in the results compared to the spectrum obtained with the Hermite approach in the previous section; the spurious eigenvalues for $N < 0.5$ dissapears while the rest of the energies are as we expected from Figure 3.10.

As before we can solve our Schrödinger equation along a contour. By comparing Figure 3.15 with Figure 3.11 it is visible that once again we get more or less the same spectrum as with the Hermite based spectral method.
Chapter 3. Numerical Methods

3.2.3 Method 2: HermiteEig

Another Hermite-based spectral method was recently implemented in MATLAB by Damien Trif [18]. It starts of by introducing the linear transformation \( \xi = cx \), where \( c > 0 \). The constant \( c \), as in the spectral collocation method, discussed in Section 3.2.1, represents a scaling parameter. When this is applied to Schrödinger’s equation (3.9) we get

\[
-\frac{d^2 \Psi}{d\xi^2} + c^{-2}V(c^{-1}\xi)\Psi = \mathcal{E}(c)\Psi,
\]

where \( \xi \in (-\infty, \infty) \). The eigenvalues \( \mathcal{E} \) of this problem is dependent on \( c \) and the eigenvalues of our original problem (3.9) is connected to \( \mathcal{E} \), \( E = c^2\mathcal{E}(c) \).

To solve (3.16), we do a series expansion on \( \Psi \),
\[ \Psi(\xi) = \sum_{n=0}^{\infty} c_n \phi(\xi) \]

where \( \phi_n(\xi) = e^{-\xi^2/2} He_n(\xi)/\sqrt{2^n n! \pi} \). By rewriting this expansion we get,

\[
\Psi(\xi) = \sum_{n=0}^{\infty} c_n \frac{1}{\sqrt{2^n n! \pi}} e^{-\xi^2/2} He_n(\xi)
\]

\[
= e^{-\xi^2/2} \sum_{n=0}^{\infty} c_n \frac{1}{\sqrt{2^n n! \pi}} He_n(\xi)
\]

\[
\Psi(\xi) = e^{-\xi^2/2} y(\xi), \quad (3.17)
\]

where \( y(\xi) = \sum_{n=0}^{\infty} \frac{1}{\sqrt{2^n n! \pi}} He_n(\xi) \).

Furthermore

\[
\frac{d^2 \Psi}{d \xi^2} = e^{-\xi^2/2} \left[ y''(\xi) - 2\xi y'(\xi) + (\xi^2 - 1)y(\xi) \right].
\]

Through substitution and elimination (3.16) becomes,

\[
-y''(\xi) + 2\xi y'(\xi) - (\xi^2 - 1)y(\xi) + e^{-2} V(c^{-1}) y(\xi) = E y(\xi). \quad (3.18)
\]

We now want to approximate the function \( y(\xi) \). From (3.17) we can write \( y(\xi) \) as a series,

\[
y_N(\xi) = \sum_{n=0}^{N-1} c_n \frac{1}{\sqrt{2^n n! \pi}} He_n(\xi)
\]

\[
= \sum_{n=0}^{N-1} c_n \vartheta_n(\xi), \quad (3.19)
\]

where

\[
\vartheta_n(\xi) = \frac{He_n(\xi)}{\sqrt{2^n n! \pi}} = e^{-\xi^2/2} \phi_n(\xi). \quad (3.20)
\]

To simplify the implementation of this method, we define the vectors

\[
c^T = \{c_0, c_1, c_2, \ldots\},
\]

\[
t^T = \{\vartheta_0, \vartheta_1, \vartheta_2, \ldots\},
\]
in such a way such that \( y_N(\xi) = c^T t(\xi) = t^T(\xi)c \). We now want to find a matrix \( X \) that will satisfy,

\[
\xi y(\xi) = \xi c^T t(\xi) = (Xc)^T t(\xi).
\]

Note that \( Xc \) is the coefficients of \( \xi y(\xi) \). By using the recursion formula of Hermite polynomials, \( \text{He}_{n+1}(\xi) = 2\xi \text{He}_n(\xi) - 2n \text{He}_{n-1}(\xi) \), along with (3.19), we get

\[
\sqrt{2^{n+1}(n+1)!\pi} \vartheta_{n+1} = 2\xi \sqrt{2^n n!\pi} \vartheta_n - 2n \sqrt{2^{n-1}(n-1)!\pi} \vartheta_{n-1}
\]

\[
\xi \vartheta_n = \sqrt{\frac{n+1}{2}} \vartheta_{n+1} + \sqrt{\frac{n}{2}} \vartheta_{n-1},
\]

where \( \vartheta_{-1}(\xi) = 0 \) and \( \vartheta_0(\xi) = \pi^{-1/4} \) by definition.

From (3.20) and the fact that \( Xc \) is the coefficients of \( \xi y(\xi) \), we get the tri-diagonal matrix \( X \),

\[
X_{i,i+1} = X_{i+1,i} = \sqrt{\frac{i}{2}}, \quad i = 1, 2, \ldots
\]

Finally we need to determine the differentiation matrix \( D \),

\[
\frac{dy}{d\xi} = (Dc)^T t(\xi).
\]

The derivatives of the Hermite polynomials satisfy \( \text{He}_n' = 2n \text{He}_{n-1} \). From (3.20) we can show that,

\[
\sqrt{2^n n!\pi} \vartheta_n' = 2n \sqrt{2^{n-1}(n-1)!\pi} \vartheta_{n-1}
\]

\[
\vartheta_n' = \sqrt{2n} \vartheta_{n-1}.
\]

The non-zero elements of \( D \) will therefore yield,

\[
D_{i,i+1} = \sqrt{2i}, \quad i = 1, 2, \ldots
\]

Using equation (3.18) and the matrices \( X \) and \( D \), we get

\[
[-D^2 + 2XD - X^2 + V(X) + I_N] c = Ec.
\]

We can now solve this eigenvalue problem.

The MATLAB code for this method is freely available\(^4\) and is easy to implement. This package consist of five essential programs.

Chapter 3. Numerical Methods

The first two programs are $X = \text{mult}(N)$ and $D = \text{deriv}(N)$ that calculates the $N \times N$ coefficient matrices $X$ and $D$. The program $[x,w] = \text{pd}(n)$ calculates the nodes and the grid points of the Gauss-Hermite quadrature, while $t = x2t(n,x)$ calculates the vector $t$. Finally there is the main program $\text{sph}$, Table 3.4.

```matlab
function [L,psi,x,w,t] = sph(fun,n,a)
    [x,w] = pd(n); t = x2t(n,x);
    D = deriv(n); X = mult(n);
    V = feval(fun,X/a);
    A = -D^2 + 2*X*D + V/a^2 - X^2 + eye(n);
    [v,l] = eig(A);
    L = diag(l)*a^2;
    [L,ind] = sort(L);
    psi = v(:,ind);
```

Table 3.4: The MATLAB code for calculating the eigenvalues, $L$, of a potential, $fun$, using the HermiteEig method.

After applying this method to the $PT$-symmetric potential $V(x) = -(ix)^N$, we find that it gives satisfactory results, Figure 3.16, except, of course, near $N = 4$.

The main difference between Trif’s and Weideman’s methods is that Trif uses a Hermite function base for the spectral collocation method, while Weideman uses a Lagrange base. Furthermore, the differentiation matrix obtained by HermiteEig is a banded matrix compared to the dense matrices that DMSUITE returns.

The HermiteEig method compares extremely well with the spectral collocation method discussed earlier. It has been found that there is not a big difference between the results of these two methods, given that the same parameters are chosen. It can be shown, and is illustrated in Damien Trif’s article [18], that HermiteEig gives better results. The main difference is that the differentiation matrix obtained by HermiteEig is a banded matrix, unlike the matrix created by DMSUITE. The result of these banded matrices is a solution that is less affected by roundoff errors.
3.3 Riccati-Padé

In a 1989 article by F.M. Fernández, Q. Ma and R.H. Tipping [19], a Riccati-Padé method (RPM) was proposed for finding the eigenvalues of the Schrödinger equation,

\[ \psi''(x) = (V(x) - E)\psi(x). \]  

(3.22)

This method starts by transforming Schrödinger’s equation into a Riccati equation. The solution will then be approximated by Padé approximants. By finding the roots of the resulting determinant, we can find the eigenvalues.

First we convert the Schrödinger equation into a Riccati equation\(^5\). This is done by defining a function \( \Phi(x) = x^{-s}\psi(x) \), where \( s = 0 \) will yield even energy levels and \( s = 1 \) the odd energy levels. The result of this transformation is that the logarithmic derivative

\[ f(x) \equiv -\Phi'(x)/\Phi(x) \]

will be regular at the origin. Through substition we can now rewrite (3.22) as a Riccati equation

\[ f'(x) - f(x)^2 + 2sf(x)/x = E - V(x). \]

\(^5\text{Riccati Equation: } y'(x) = a(x)y(x)^2 + b(x)y(x) + c(x).\)
Chapter 3. Numerical Methods

Note that we have now reduced (3.22) from a second order problem to a first order problem. The cost of this reduction of order is that we now have a non-linear problem compared to the linear problem we started with.

For simplicity we make the assumption that these potentials are symmetric. We now need to expand the potential \( V(x) \) and the logarithmic derivative \( f(x) \) in a Taylor series around the origin,

\[
V(x) = \sum_{j=1}^{K} v_j x^{2j}, \quad v_K > 0, \quad (3.23)
\]

\[
f(x) = \sum_{j=0}^{\infty} f_j x^{2j+1},
\]

where the coefficients \( f_j \) satisfy the formula

\[
f_j = \sum_{i=0}^{j-1} f_i f_{j-1-i} + E \delta_{j0} - \sum_{i=0}^{K} v_i \delta_{ji} \left( 2j + 2s + 1 \right)^{-1}. \quad (3.24)
\]

Now that the Schrödinger equation has been transformed into a Riccati equation, we can approximate \( f(x) \) by using a sequence of rational functions,

\[
g(x) = \frac{A(x)}{B(x)},
\]

\[
A(x) = \sum_{j=0}^{M} a_j x^{2j+1},
\]

\[
B(x) = \sum_{j=0}^{N} b_j x^{2j}, \quad b_0 = 1.
\]

Given that \( g(x) \) is a Padé approximant, \( f(x) - g(x) = O(x^{2(M+N)+3}) \). There are \( M + N + 2 \) adjustable parameters in the approximating function; namely the coefficients \( a_j, b_j \) and the eigenvalue \( E \). From (3.24) we see that \( f_j \) will be a function in \( E \); more specifically a polynomial of degree \( j + 1 \). If we choose \( E \) such that \( f(x) - g(x) = O(x^{2(M+N)+5}) \), we will get a more accurate solution for the Riccati equation around \( x = 0 \) and the coefficients \( a_j \) and \( b_j \) will satisfy
\[ \sum_{i=0}^{j} b_j f_{j-i} = a_j, \quad j = 0, \ldots, M, \]
\[ \sum_{i=0}^{j} b_j f_{j-i} = 0, \quad j = M + 1, \ldots, M + N + 1, \quad (3.25) \]

where \( b_i = 0 \) for \( i > N \).

The \( N \) coefficients \( b_j \) with \( j = 1, \ldots, N \) cannot satisfy the \( N + 1 \) homogeneous, linear equations (3.25) unless

\[ H_D^d = \begin{vmatrix} f_{d+1} & f_{d+2} & \cdots & f_{d+D} \\ f_{d+2} & f_{d+3} & \cdots & f_{d+D+1} \\ \vdots & \vdots & \ddots & \vdots \\ f_{d+D} & f_{d+D+1} & \cdots & f_{d+2D-1} \end{vmatrix} = 0, \quad (3.26) \]

where \( d = M - N \geq 0 \) and \( D = N + 1 \).

It can be shown that the roots of (3.26) converge towards the eigenvalues as the size \( (D) \) of the matrix increases\(^6\).

### 3.3.1 \( V(x) = x^2 \)

To illustrate the Ricatti-Padé Method we will apply it to the well-known quantum harmonic oscillator, \( V(x) = x^2 \). For simplicity, we will only do this for a small matrix:

\[ H_2^0 = \begin{vmatrix} f_1 & f_2 \\ f_2 & f_3 \end{vmatrix}. \quad (3.27) \]

The coefficients \( f_j \) can be found by using the recursion formula (3.24).

\(^6\)For asymmetric potentials we will get the Riccati equation \( f'(x) = f(x)^2 + E - V(x) \). Because \( f(x) \) is regular at the origin, we can do a Taylor expansion where the coefficients satisfy, \( f_{j+1} = \left[ \sum_{i=0}^{j} f_i f_{j-i} - \sum_{i=2}^{2K} v_i \delta_{ji} + E \delta_{j0} \right] (j+1)^{-1} \). Another main difference between the solving of symmetric and asymmetric potentials is that now \( H_B^d = H_B^{d+1} = 0 \). For a more in depth discussion, consult [19].
For this exercise we will only compute the even energies \((s = 0)\).

\[
\begin{align*}
  f_0 &= \frac{E}{2s + 1} = E \\
  f_1 &= \frac{f_0^2 - v_1}{2s + 3} = \frac{1}{3}(E^2 - 1) \\
  f_2 &= \frac{2f_0f_1}{2s + 5} = \frac{2}{15}(E^3 - E) \\
  f_3 &= \frac{2f_0f_2 + f_1^2}{2s + 7} = \frac{1}{315}(17E^4 - 22E^2 + 5)
\end{align*}
\]

Therefore \(H_2^0 = \frac{(E^2 - 25)(E^2 - 1)^2}{4725} = 0\) \((3.28)\)

When we solve \(H_2^0 = 0\) for \(E\), we find that \(E = \pm 1\) or \(E = \pm 5\).

This is exactly the ground state and second energy\(^7\) according to the known formula for the eigenvalues of the harmonic oscillator, \(E_n = 2n + 1\).

\[
\begin{array}{c|c}
  H_D^S & O(H_D^S) \\
  \hline
  H_2^0 = \frac{(E^2 - 25)(E^2 - 1)^2}{297675} & 6 \\
  H_2^1 = \frac{(E^2 - 25)(E^2 - 1)^2(E^2 + 3)}{42047375} & 8 \\
  H_2^2 = \frac{(E^2 - 25)(E^2 - 1)^2(135 - 28E^2 + 13E^4)}{442047375} & 10 \\
  H_2^3 = \frac{(E^2 - 81)(E^2 - 25)^2(E^2 - 1)^3}{46414974375} & 12 \\
  H_2^4 = \frac{4E(E^2 + 29)(E^2 - 81)(E^2 - 25)^2(E^2 - 1)^3}{896041080309375} & 15 \\
\end{array}
\]

\textbf{Table 3.5}: The order of the determinant \(H_D^S|_{s=0}\).

In Table 3.5 we have shown a few determinants where \(s = 0\). As we can see, the order of the polynomials increase as we increase the parameters \(d\) and \(D\). This leads to more eigenenergies. Through inspection it can be shown that an increase in \(d\) increases the possibility of repetition of eigenenergies, while an increase in \(D\) (the matrix size) provide more distinct eigenvalues.

There is a disadvantage to using a larger matrix. Because we work with a symbolic matrix, the time it takes to calculate the roots of \(H_D^S\), increases drastically, see Figure 3.17.

\(\text{\(^7\)We ignore the negative energies because physically it does not mean anything.}\)
Chapter 3. Numerical Methods

This figure compares speed of the three techniques to calculate $H_D^d$. The dotted curve indicates the time it took to calculate a symbolic $H$ using (3.26), while the other two curves used the Hankel recursion formula (3.29) to calculate it for both a symbolic and numeric matrix. All of these results were obtained by MATLAB, using the symbolic toolbox for the symbolic computations. Table B.1 contains the data used to obtain this figure.

First we notice that $H_D^d$ is the Hankel determinant [20, p. 595] and a recursion formula can be derived from Jacobi’s Identity\(^8\) to calculate $H_D^d$:

$$H_n^k = \frac{H_{k-1}^n H_k^{n+3} - (H_{k-1}^{n+1})^2}{H_{k-2}^{n+2}}.$$  \hspace{1cm} (3.29)

This did not have the effect we hoped for, Figure 3.17. When we use the recursion formula (3.29) to determine $H_D^d$, more symbolic calculations are needed compared to (3.26), which leads to an increase in runtime.

To improve the speed of the calculations, we steered away from symbolic functions and used numerical vectors to represent the polynomials, $f_k$ (The MATLAB implementation can be viewed in Tables A.5 and A.6). For example, the polynomial $2x^2 - 3x^5$ becomes $[-3 \ 0 \ 0 \ 2 \ 0 \ 0]$. The multiplication and division of these vectorized polynomials can be done by using MATLAB’s \texttt{conv} and \texttt{deconv} respectively. This made a huge improvement in the speed.

Looking at Table 3.6 it is clearly visible that there is a disadvantage to using numerical methods. The runtime might have increased, but the algorithm is very unstable. The accuracy of the eigenvalues decrease as the matrix size increases.

\(^8\) Jacobi’s Identity: $(H_n^k)^2 - H_{k-1}^n H_k^{n+1} + H_{k+1}^{n-1} H_k^{n+1} = 0$. 

Table 3.6: The ground state eigenvalue of the harmonic oscillator ($E_0 = 1$) calculated with three different implementations of the RPM.

<table>
<thead>
<tr>
<th>D</th>
<th>Symbolic $H_D^2$</th>
<th>Symbolic Hankel</th>
<th>Numerical Hankel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>$1.00 \pm 7.467i \times 10^{-9}$</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1.000008313</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1.001549110</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1.237479374</td>
</tr>
</tbody>
</table>

To conclude, the Riccati-Padé method is not an efficient method. The analytical approach is time consuming while the numerical method is unstable. Finally, we will not be able to apply it to the $\mathcal{PT}$-symmetric potential $V(x) = -(ix)^N$ without modification because this function cannot be expanded into a power series, of the form (3.23), for non integer $N$ values.

3.4 Summary

We started this chapter by looking at the shooting method. This method is easy to understand and implement. There are a few variations of this method. Different numerical solvers can be used on the two initial value problems. There are also different root finding methods available to solve the matching function.

The implementation used in this thesis gives accurate eigenvalues if the appropriate tolerances are chosen for the MATLAB function ode45, see Figure 3.4. The cost for these solutions, is a relatively long runtime.

Second, we considered three different implementations of the spectral collocation method. The first two were variants of the method implemented by DMSUITE.

We started with the Lagrangian implementation of the spectral collocation. By using the supplied MATLAB code, it is relatively easy to apply this method to other $\mathcal{PT}$-symmetric potentials. The only factor that might delay obtaining the solutions, is the choosing of an appropriate scaling parameter $c$. Since the choice of $c$ is entirely arbitrary, another approximation or numerical method can be used to obtain a set of reference eigenvalues. These values can be used to verify that your choice of $c$ gives an accurate representation of the spectrum.

Another important point to remember when solving a $\mathcal{PT}$-symmetric potential is that it is sometimes required to solve the Schrödinger equation
Figure 3.18: This figure displays the combination of the solution for the potential \( V(x) = -(ix)^N \) obtained by integrating on the real and the complex axis with the spectral collocation method.

on a complex contour. The spectrum obtained by this contour calculation will probably differ from our solution obtained on the real axis, but this should be expected, since the contour is dependent on the parameter \( N \).

The next variation of the spectral collocation method we looked at had a Fourier basis. Considering the decaying shape of the eigenfunctions\(^9\), we can consider them as periodic. Instead of the zeros of a Hermite polynomial, trigonometric polynomials, which requires rescaling to \([-L, L]\), was used. Similar to the scaling parameter \( c \), we have an adjustable parameter \( L \) that can influence the outcome of the algorithm.

With this Fourier implementation we obtain similar results to that of the Lagrange based method. The only visible difference between these solutions is the disappearance of the spurious eigenvalues which occurred when \( N < 1 \) (compare Figure 3.10 and 3.14).

Finally we look at a Hermite based spectral collocation method implemented in the package HermiteEig. The main difference between this method and the other two is that it has a banded differentiation matrix. The structure of this algorithm is similar to the other two methods, except that the math is more elaborate. The result is a solution which is less affected by roundoff errors.

The last numerical method we will consider is the Riccati-Padé method.

\(^9\)\( \lim_{x \to \pm \infty} \psi(x) = 0 \).
This method obtains the eigenvalues of the Schrödinger equation using an analytical approach. The advantage of this is that it is not affected by roundoff and other errors resulting from numerical methods. The drawback of this method is that it can become time consuming. To speed up the process, we implemented this method numerically.

The result was a solution obtained in a short amount of time. The problem with this method is that the error increases as we try to get more distinct eigenvalues (by increasing $D$). We can therefore conclude that the analytical approach is time consuming, while the numerical solution can be inaccurate.

This method is inefficient but still useful. Given that the $\mathcal{PT}$-symmetric potential can be expanded in a Taylor series, for a specific choice of parameters, it can be used to determine the accuracy of your solutions obtained by other numerical methods.
Chapter 4

The $\mathcal{PT}$-Symmetric Potential

$V(x) = -(i \sinh x)^{\alpha} \cosh^{\beta} x$

In this chapter we will be look at another $\mathcal{PT}$-symmetric potential as discussed in [3],

$V_{\alpha \beta}(x) = -(i \sinh(x))^\alpha \cosh^\beta x,$

(4.1)

with real parameters $\alpha$ and $\beta$. The potential tends to $\infty$ or $-\infty$ when $x \to \pm \infty$. The $\alpha$-parameter is responsible for the “shape” of the potential, since $\alpha$ determines whether the real and imaginary parts of $V_{\alpha \beta}$ are negative or positive. The $\beta$-parameter does not have a major effect on the potential although some negative $\beta$ choices may result in an unbounded potential. For this reason we restrict our parameters to values which satisfies $\alpha + \beta > 0$.

When $\alpha = 2$ we have a positive and bounded potential. Using $\alpha = 2$ as a reference value, we can now find the eigenvalues for $\alpha > 2$ and $\alpha < 2$. We can choose any other $\alpha$ as a reference value given that the potential is real, positive and bounded; for example, $\alpha_R = 6$ or $\alpha_R = 10$. In fact, when we look closely at the potential’s real and imaginary parts, we can see that a pattern emerges, Table 4.1. In Figure 4.1 we can see two separate spectra with different reference values as calculated in [3].

In Figure 4.2 we have the ground state eigenvalues of $V_{\alpha \beta}$ and we see that there is a discontinuity at $\alpha = 4$. This is because the potential is negative and unbounded from below and therefore does not have any real eigenvalues. When $\alpha \in [4,8]$ the eigenvalues are real again. These are the eigenvalues connected to $\alpha_R = 6$. This is consistent with the pattern $\alpha_R = 2 + 4N$, where $N$ is a positive integer, visible in Table 4.1. Everyone of these reference values has a different energy spectrum related to it. We
Chapter 4. The $\mathcal{PT}$-Symmetric Potential $V(x) = -(i \sinh x)\alpha \cosh \beta x$

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
$\alpha$ & Real $V_{\alpha \beta}$ & Imaginary $V_{\alpha \beta}$ \\
\hline
0 & $V_{R} < 0$ & $V_{I} = 0$ \\
0 < $\alpha$ < 1 & $V_{R} < 0$ & $V_{I} < 0$ \\
$\alpha = 1$ & $V_{R} = 0$ & $V_{I} < 0$ \\
1 < $\alpha$ < 2 & $V_{R} > 0$ & $V_{I} < 0$ \\
$\alpha = 2$ & $V_{R} > 0$ & $V_{I} = 0$ \\
2 < $\alpha$ < 3 & $V_{R} > 0$ & $V_{I} > 0$ \\
$\alpha = 3$ & $V_{R} = 0$ & $V_{I} > 0$ \\
3 < $\alpha$ < 4 & $V_{R} < 0$ & $V_{I} > 0$ \\
$\alpha = 4$ & $V_{R} < 0$ & $V_{I} = 0$ \\
\hline
\end{tabular}
\end{center}

Table 4.1: This table shows for which value of $\alpha$ the potential $V_{\alpha \beta}$ real and imaginary parts are positive or negative. After $\alpha = 4$ the pattern repeats itself.

\begin{figure}
\centering
\includegraphics[width=0.5\textwidth]{spectrum.png}
\caption{Spectrum of $V_{\alpha \beta}$ related to the reference value $\alpha_R = 2$, the solid curve, and $\alpha_R = 6$, the dashed curve as it was calculated in [3, p. 3113].}
\end{figure}

Therefore need to find a way to solve the differential equation related to a specific $\alpha_R$.

## 4.1 Paths in the Complex Plane

We need to find a path in the complex plane along which to integrate.

This potential, like many others, has a branch point at $x = 0$. The result of this branch point is that the real part is symmetric and the imaginary part is antisymmetric. We can now rewrite $V_{\alpha \beta}$ as
Chapter 4. The $\mathcal{PT}$-Symmetric Potential $V(x) = -(i \sinh x)^\alpha \cosh^\beta x$

![Image: The ground state energies of $V_{\alpha \beta}$ where $\beta = 0$.](image)

Figure 4.2: The ground state energies of $V_{\alpha \beta}$ where $\beta = 0$.

\[
V(x) = e^{i\pi(2+\alpha)/2} \sinh^\alpha x \cosh^\beta x, \quad x > 0
\]

\[
V(x) = e^{-i\pi(2+\alpha)/2} \sinh^\alpha x \cosh^\beta x, \quad x < 0.
\]

This potential is $\mathcal{PT}$-symmetric and needs to cut the complex plane from 0 to $-\infty$. Before we continue to find the paths, we need to verify whether we have bounded solutions. When $x \to +\infty$, the potential reduces to

\[
\lim_{x \to \infty} V_{\alpha \beta}(x) = e^{i\pi(2+\alpha)/2} e^{(\alpha+\beta)x/2} 2^{-(\alpha+\beta)}.
\]

If we assume that the solution has the form $\Psi(x) = e^{G(x)}$, as in the WKB approximation, the leading order of the expansion will yield

\[
G(x) \to \pm e^{i\pi(2+\alpha)/4} \frac{2 e^{(\alpha+\beta)x/2}}{(\alpha + \beta) e^{(\alpha+\beta)/2}}. \tag{4.2}
\]

Excluding the case when $\alpha = 4N$, $G(x)$ is negative and its magnitude increases exponentially, which satisfies the requirement $\lim_{x \to \pm \infty} \Psi(x) = 0$. There appears to be discrete eigenvalues with localized solutions.

To find out how these contours are constructed, we will focus on the case where $\alpha = 2$. When $\alpha = 2$, the phase factor of $G(x)$ reduces to $e^{i\pi}$. We would like to extend our spectrum away from $\alpha = 2$, ideally to $\alpha > 4$. Let us introduce a new parameter $\delta$ such that $\alpha = 2 + \delta$. The phase angle now becomes $\theta = \pi + \delta \frac{\pi}{4}$ and the solution tends to
Chapter 4. The \( \mathcal{PT} \)-Symmetric Potential \( V(x) = -(i \sinh x)^{\alpha} \cosh \beta x \)

\[
\Psi(x) \rightarrow e^{-i(\cos(\delta \pi/4)+i \sin(\delta \pi/4))} e^{(\alpha+\beta)x/2}
\]
as \( x \rightarrow \infty \). To extend the integration past \( \alpha = 4 \), we need to integrate along a contour \( x + iy \). The imaginary component needs to be negative for \( \mathcal{PT} \)-symmetry to hold. When we substitute \( x \) with \( x + iy \) in \( G(x) \), the phase angle changes to

\[
\theta = \frac{\pi(\alpha + 2)}{4} + \frac{y(\alpha + \beta)}{2}.
\]

When \( \theta = \pi \) the exponential part will decrease the fastest. This is the *anti-stokes line* as discussed in Section 1.2.1. For the solution to converge, the contour is bounded by the region \( \theta = \pi \pm \pi/2 \). The optimal contour suggests that \( y \) is equal to

\[
y = \frac{(2 - \alpha)\pi}{2(\alpha + \beta)}.
\tag{4.3}
\]

This value of \( y \) is negative when \( \alpha > 2 \), which implies that the \( \mathcal{PT} \)-symmetry remains unbroken. However, we can still solve the differential equation for any \( y \)-value within the regions indicated in Figure 4.3 and by equations (4.4) and (4.5).

![Figure 4.3](image.png)

*Figure 4.3:* The value of the imaginary component of the contour \( x + iy \) when \( \beta = 0 \). The two dashed lines, \( y_+ \) and \( y_- \), indicates the boundaries of the integration area, while the solid line indicates the optimal \( y \)-value.
Chapter 4. The $\mathcal{PT}$-Symmetric Potential $V(x) = -(i \sinh x)^{\alpha} \cosh^{\beta} x$

\[ y_+ = \frac{(4 - \alpha)\pi}{2(\alpha + \beta)} \quad (4.4) \]
\[ y_- = \frac{-\alpha\pi}{2(\alpha + \beta)} \quad (4.5) \]

In Figure 4.2 we recall that there are segments of real eigenvalues which are connected to a specific reference value, for example $\alpha_R = 2$. Figure 4.3 verifies this by showing us that when we solve the differential equation on the real axis, $y = 0$, we can get real eigenvalues up to $\alpha = 4$, without breaking the $\mathcal{PT}$-symmetry.

Using the contour (4.3) to calculate the eigenvalues when $\alpha > 2$, we find that the spectrum now extends past $\alpha = 4$, Figure 4.4.

**Figure 4.4:** The eigenvalues of the potential $V(x) = -(i \sinh(x))^{\alpha} \cosh^{\beta}(x)$, calculated with the spectral collocation method for $\beta = 0$ and $\alpha_R = 2$. The spectrum up to $\alpha = 2$ was calculated on the real axis while the spectrum $\alpha > 2$ was calculated along the contour $x + iy$ with the complex component (4.3).

We can now do the same to obtain more spectra connected to other reference values. For example, the imaginary component of the contour connected to $\alpha_R = 6$ is given by,

\[ y = \frac{\pi(6 - \alpha)}{2(\alpha + \beta)} \quad (4.6) \]
with the boundaries given by,

\[ y_- = \frac{\pi (4 - \alpha)}{2(\alpha + \beta)} \]
\[ y_+ = \frac{\pi (8 - \alpha)}{2(\alpha + \beta)} \]

By using (4.6) we get the spectrum in Figure 4.5.

![Figure 4.5: The eigenvalues of the potential \( V(x) = -(i \sinh x)^\alpha \cosh^\beta x \), calculated for \( \beta = 0 \) and \( \alpha R = 6 \).](image)

We have now obtained two spectra related to different \( \alpha R \) values. To determine how accurate our results are, we need to return to [3]. The only values available to us for a comparison was obtained by the Riccati-Padé method.

\[
\begin{array}{cccc}
\alpha = 1 & \alpha = 2 & \alpha = 3 & \alpha = 3 \\
(\text{real axis}) & (\text{real axis}) & (\alpha_R = 2) & (\alpha_R = 6) \\
4.248 \times 10^{-13} & 9.253 \times 10^{-5} & 4.547 \times 10^{-10} & 2.604 \times 10^{-11} \\
\end{array}
\]

Table 4.2: The absolute error between the ground state eigenvalues of our spectrum, calculated with the spectral collocation method, and the eigenvalues obtained in [3] with the Riccati-Padé method.

In Table 4.2 we see that the ground state eigenvalues calculated with the spectral collocation method, compares well with the eigenvalues of [3].
Figure 4.6: The spectrum of the potential $V_{\alpha_0}$, calculated for the two reference values $\alpha_R = 2$ (blue) and $\alpha_R = 6$ (red). The ground state eigenvalues determined in [3] for $\alpha = 1, 2, 3$ are included in this figure.

Figure 4.6 is a combination of the Figures 4.4 and 4.5, which is similar to the figure generated in [3], see Figure 4.1.
Appendix A

MATLAB code

```matlab
global flComp;
flComp = false;

% Method used to solve the 2 initial value problems
% and returns \phi(E).
fname = 'rk4_ode';

abt = 1e-6; % Absolute Tolerance of ode45
relt = 1e-12; % Relative Tolerance of ode45
E = [];

for N = linspace(0.8,5,500)
    for ee = linspace(0,20,20)
        f = fzero(fname,ee,1e-10,N,abt,relt);
        if flComp == false
            E = [E; N, f];
        end
    end
end
```

Table A.1: MATLAB code for calculating eigenvalues by using the shooting method.
function phi = rk4_ode(E,N,abtl,reltl)

global flComp;

xm = 0;  \%The matching point
a = -10; \%The left boundary
b = 10; \%The right boundary
y0 = 0;  \%The initial values
v0 = 1;
options = odeset('Abstol',abtl,'RelTol',reltl);

F = @(x,w) \[w(2); (-(i*x)^N-E)*w(1)\];

\% Solves the left part of the eigenfunction
[XL,WL] = ode45(F,[a xm], [y0 v0],options);
YL = WL(:,1); VL = WL(:,2);
YL = YL/max(abs(YL)); VL = VL/max(abs(VL));

\% Solves the right part of the eigenfunction
[XR,WR] = ode45(F,[b xm], [y0 v0],options);
YR = WR(:,1); VR = WR(:,2);
YR = YR/max(abs(YR)); VR = VR/max(abs(VR));

\% phi = The matching function
phi = (VR(end)*YL(end)-VL(end)*YR(end));
if imag(phi) == 0
    flComp = false;
else
    flComp = true;
end
phi = real(phi); \% Returns the matching function phi(E)

Table A.2: The shooting method uses this MATLAB program to solve the two parts of the eigenfunction given an eigenvalue for the defined potential.
function phi = rk4_ode_modified(E,N,abtl,reltl)

    global f1Comp;
    global tE;
    global tN;
    tE = E; tN = N;

    xm = 0;  %The matching point
    a = -10; %The left boundary
    b = 10;  %The right boundary
    y0 = 0;  %The initial values
    v0 = 1;
    options = odeset('Abstol',abtl,'RelTol',reltl);

    % Solves the left part of the eigenfunction
    [XL,WL] = ode45(@contour,[a xm], [y0 v0],options);
    YL = WL(:,1); VL = WL(:,2);
    YL = YL/max(abs(YL)); VL = VL/max(abs(VL));

    % Solves the right part of the eigenfunction
    [XR,WR] = ode45(@contour,[b xm], [y0 v0],options);
    YR = WR(:,1); VR = WR(:,2);
    YR = YR/max(abs(YR)); VR = VR/max(abs(VR));

    phi = (VR(end)*YL(end)-VL(end)*YR(end));
    phi = real(phi); % Returns the matching function phi(E)
end

% Defines the new ODE with the Contour
function dy = contour(t,y)
    global tE;
    global tN;

    dy = zeros(2,1);

    dy(1) = y(2);
    dy(2) = (2^tN *(1+i*t)^(tN/2) - tE)*y(1)/(1+i*t) - (i/2)*y(2)/(1+i*t);
end

Table A.3: For the contour calculations, the only difference in the shooting method is the way the eigenfunctions are solved.
function lim = weps(a);
% Program that implements Wynn’s epsilon algorithm.
a = a(:);
N = length(a);
n = [1:1:N];
S = a; E = [];
E(:,1:2) = [zeros(size(S)) S];

for j = 3:N
  D = [E(2:N,j-1); 0]-[E(1:N-1,j-1); 0];
  L = (D==0);
  D(L) = eps;
  R = 1./D;
  R(L) = NaN;
  E(:,j) = [E(2:N,j-2); 0] + R;
end

b = E(1,2:2:N);
b(isnan(b)) = [];
lim = b(length(b));

Table A.4: Program that implements Wynn’s epsilon algorithm.
function HH = Han(n,k,s)
% Han(n,k,s) calculates the Hankel determinant
% This is a recursion formula as described in "Applied and
% computational Complex analysis Vol 1, P Henrici" p595.

if k == 1
    % fj is a program that calculates f_j using the recursion formula
    HH = fj(n+1,s);
elseif k == 2
    m1 = conv(fj(n+1,s),fj(n+3,s)); m1 = polyreduce(m1);
    m2 = conv(fj(n+2,s),fj(n+2,s)); m2 = polyreduce(m2);
    dm = length(m2)-length(m1);
    m1 = [zeros(1,dm),m1];
    m2 = [zeros(1,-dm),m2];
    HH = m1 - m2;
else
    m1 = conv(Han(n,k-1,s),Han(n+2,k-1,s)); m1 = polyreduce(m1);
    m2 = conv(Han(n+1,k-1,s),Han(n+1,k-1,s)); m2 = polyreduce(m2);
    dm = length(m2)-length(m1);
    m1 = [zeros(1,dm),m1];
    m2 = [zeros(1,-dm),m2];
    m3 = m1-m2; m3 = polyreduce(m3);
    tmpH = Han(n+2,k-2,s); tmpH = polyreduce(tmpH);
    [HH, r] = deconv(m3, tmpH);
    HH = polyreduce(HH);
end

function p = polyreduce(p)
fi = find(p = 0);
if fi > 1
    p(1:(fi-1)) = [];
end

Table A.5: This is the MATLAB code used to calculate the Hankel determinant or
the Riccati-Padé method.
function y = fj(j,s)

% x^2
v = [1];

global f
y = [];

if isempty(f);
    f = {};
end
jmax = length(f)-1; % Highest value of j already calculated

if j > jmax || (j <= jmax && isempty(cell2mat(f(j+1))))
    if j == 0
        y = [1/(2*s+1) 0];
    else
        som = 0;
        for k = 0:(j-1)
            mult = conv(fj(k,s),fj(j-k-1,s));
            dm = length(mult) - length(som);
            som = [zeros(1,dm), som];
            mult = [zeros(1,-dm), mult];
            som = som + mult;
        end
        if j <= length(v)
            som(end) = som(end) - v(j);
        end
        som = som./(2*j+2*s+1);
        som = polyreduce(som);
        y = som;
    else
        y = cell2mat(f(j+1));
    end
else
    y = cell2mat(f(j+1));
end

if j>jmax || isempty(f(j+1))
    f(j+1) = y;
end

Table A.6: fj is the MATLAB code used to determine the coefficients $f_j$, (3.24), which is the elements of the Hankel determinant calculated in Table A.5
Appendix B

Additional Tables and Figures

Figure B.1: A log graph for the comparison of the runtime of the HermiteEig and spectral collocation method.
Table B.1: This table contains the data used to construct Figure 3.17. It shows the time it takes to calculate $H$ using three different techniques. The first 2 columns used symbolic calculations to determine $H$, while the last column used a numerical algorithm. While the first column calculated $H$ using brute force, the other 2 made use of the Hankel recursion formula (3.29). All the data was obtained by taking the average from 5 runs.

<table>
<thead>
<tr>
<th>$D$</th>
<th>Symbolic $H^2_D$</th>
<th>Symbolic Hankel</th>
<th>Numerical Hankel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3716</td>
<td>0.3500</td>
<td>0.0562</td>
</tr>
<tr>
<td>2</td>
<td>0.4622</td>
<td>0.4404</td>
<td>0.0688</td>
</tr>
<tr>
<td>3</td>
<td>0.6344</td>
<td>0.7654</td>
<td>0.0748</td>
</tr>
<tr>
<td>4</td>
<td>1.3596</td>
<td>8.5124</td>
<td>0.0844</td>
</tr>
<tr>
<td>5</td>
<td>5.2746</td>
<td>519.7243</td>
<td>0.1186</td>
</tr>
<tr>
<td>6</td>
<td>139.5598</td>
<td>-</td>
<td>0.2434</td>
</tr>
</tbody>
</table>
Bibliography


