Numerical Solution to Schrodinger's Equation

Hasti Hojabr

Student ID: 97216040

Professor: Nematollah Riazi

Date: January 9, 2022

Contents

1 Introduction

If the particle potential depends only on position x of the particle, the equation that governs the wave function of our quantum-mechanical system is **One-dimensional Schrödinger equation**. There are many problems for One-dimensional Schrödinger equation such as infinite square well, harmonic oscillator and etc. We also have one dimensional shape-invariant potentials such as Pöschl–Teller potential, Hydrogenic potential, Morse Potential and the Rosen-Morse potential.

In this project we will look at Hamiltonian with several different potentials shown in fig.1 . Simulation is done using python 3.8.5. Codes are provided in appendix.

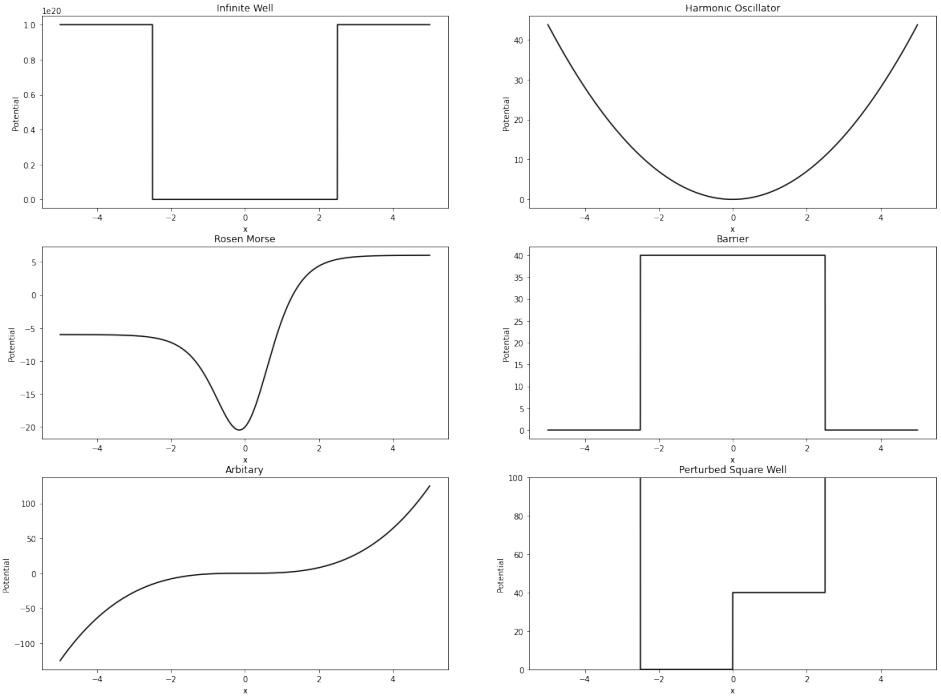

Figure 1: Several Potentials

2 Time-Independent Schrodinger Equation

A quantum particle in stationary state of definite energy E can be expressed using time-independent Schrödinger equation, which is an ordinary differential equation (ODE) .

$$-\frac{\hbar^2}{2m}\frac{d^2\psi}{dx^2} + V(x)\psi(x) = E\psi(x) \tag{1}$$

The eigenstate equation can be written as

$$\hat{H}|\psi\rangle = E|\psi\rangle \tag{2}$$

The boundary conditions makes Schrödinger equation different from ordinary differential problems. It in facts make it a eigenvalue problem. So we must find the solutions to ODE which satisfies the appropriate boundary conditions and are physically acceptable. In order to do so we need to use two algorithms,Numerov algorithm and Bisection algorithm, together.

2.1 Numerov Algorithm

A simple and efficient method for integrating second order differential equations is called the Numerov or Cowling's method. Because Schrödinger equation does not contain any first derivatives, the Numerov method is more efficient than fourth-order Runge-Kutta method and thus speeds up our calculations. We can write Eq. 2 in this generic form:

$$\frac{d^2\psi}{dx^2} + k^2(x)\psi(x) = 0 \tag{3}$$

In which k^2 is:

$$k^2 = \frac{2m}{\hbar^2}(E - V(x)) \tag{4}$$

We can write Taylor's series for $\psi(x+h)$:

$$f(x) = \sum_{n=0}^{\infty} f^n(h)\frac{(x-h)^n}{n!} \tag{5}$$

$$\psi(x+h) \sim \psi(x) + h\psi^{(1)}(x) + \frac{h^2}{2!}\psi^{(2)}(x) + \frac{h^3}{3!}\psi^{(4)}(x) + \frac{h^4}{4!}\psi^{(1)}(x) + O(5) \tag{6}$$

$$\psi(x-h) \sim \psi(x) - h\psi^{(1)}(x) + \frac{h^2}{2!}\psi^{(2)}(x) - \frac{h^3}{3!}\psi^{(3)}(x) + \frac{h^4}{4!}\psi^{(4)}(x) + O(5) \tag{7}$$

By adding these together we can cancel out all the odd powers of h.

$$\psi(x+h) + \psi(x-h) = 2\psi(x) + h^2\psi^{(2)}(x) + \frac{h^4}{12}\psi^{(4)}(x) + O(6) \tag{8}$$

So we can write the second derivative of Schrodinger equation as

$$\psi^{(2)}(x) = \frac{\psi(x+h) + \psi(x-h) - 2\psi(x)}{h^2} - \frac{h^2}{12}\psi^{(4)}(x) + O(6) \tag{9}$$

We can write the term involving the fourth derivative by second derivative by acting $1 + \frac{h^2}{12}\frac{d^2}{dx^2}$ on Eq 2.

$$\psi^{(2)}(x) + \frac{h^2}{12}\psi^{(4)}(x) + k^2(x)\psi(x) + \frac{h^2}{12}\frac{d^2}{dx^2}(k^2(x)\psi(x)) = 0 \tag{10}$$

This will give us

$$\psi(x+h) + \psi(x-h) - 2\psi(x) + h^2k^2\psi(x) + \frac{h^4}{12}(k^2(x)\psi(x)) + O(6) = 0 \tag{11}$$

$$\frac{d^2}{dx^2}(k^2(x)\psi(x)) \simeq \frac{k^2(x+h)\psi(x+h) + k^2(x-h)\psi(x-h) - 2k^2(x)\psi(x)}{h^2} \tag{12}$$

Submitting Eq.10 to Eq.9 will give us

$$\psi(x+h) = \frac{2(1 - \frac{5}{12}h^2k_m^2)\psi_n - (1 + \frac{1}{12}h^2k^2(x-h))\psi(x-h)}{1 + \frac{1}{12}h^2k^2(x+h)} + O(6) \tag{13}$$

$$\psi_{n+1} = \frac{2(1 - \frac{5}{12}h^2k_m^2)\psi_n - (1 + \frac{1}{12}h^2k_{n-1}^2)\psi_{n-1}}{1 + \frac{1}{12}h^2k_{n+1}^2} \tag{14}$$

So we can determine $\psi(n)$ for $n = 2, 3, ...$ given two initial conditions, $\psi(0)$ and $\psi(1)$. Number of steps needed to integrate is $(x_f - x_i)/h \sim 1/h$ so it will reduce the error from 6th to 5th order.

2.2 Bisection Algorithm

The most elementary trial-and-error technique is the bisection algorithm. Although it's slow but it's reliable. We want to find root for a function of f(x), the search procedure start with a guessed value for x.

$$f(x) \simeq 0 \tag{15}$$

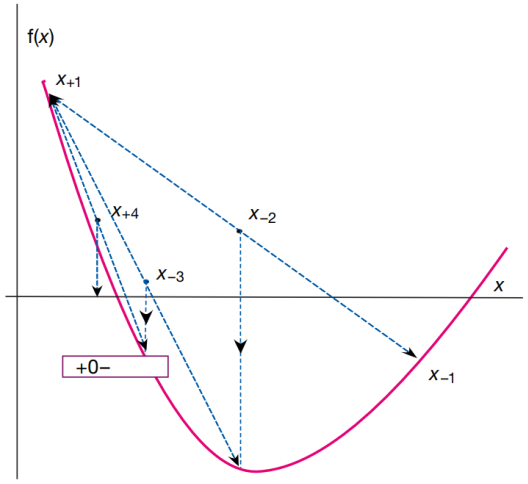

Figure 2: Bisection Method

If we know some interval in which f(x) changes sing, then bisection algorithm will converge to the root by finding smaller intervals within which the zero is.

As shown in figure.1, we start with two nonzero values x_- and x_+ between where a zero occurs, where x_+ may be less than x_-. If this condition is satisfied then the algorithm then picks the new x as the bisection of the interval and selects as its new interval the half in which the sign change occurs. The process continues until the value of f(x) is less than the given precision choice. Bisection Algorithm can be written like:

```
def f(x):
def bisection:
    repeat N times
        x=(x+ + x-)/2
        if f(x+)f(x)>0: x+ = x
        else: x- = x
    return x
```

2.3 Algorithm

The Numerov and Bisection Algorithm together can give us the solution to one-dimensional Schrödinger equation. The algorithm can be breaked to five steps:

1. Take $\psi_0 = 0$ and choose ψ_1 any small number. We took 0.001 for ψ_1 in our simulation.

2. Choose a trial energy E_n.

3. Break ODE solver to 2 parts, right and left. Step $\psi_l(x)$, which satisfies its boundary condition, in toward the right. Do this also for ψ_R. These to must reach the matching radius x_{match} and satisfy this condition. In order to do so logarithmic derivative $\psi'(x)/\psi(x)$ to be continuous.

4. Normalize

5. Write a function that calculates the matching function $\Delta(E,x)$ as a function of energy and matching radius. This subroutine will be called by the bisection algorithm program to search for the energy at which $\Delta(E,x=2)$ vanishes.

$$\Delta(E,x) = \frac{\psi'_L(x)/\psi_L(x) - \psi'_R(x)/\psi_R(x)}{\psi'_L(x)/\psi_L(x) + \psi'_R(x)/\psi_R(x)}|_{x=x_{match}} \tag{16}$$

6. Choose value of tolerance and search until $\Delta(E,x)$ changes in only fourth decimal place.

7. Increase the value of the initial energy guess and search for excited states.

```
import libraries
def constants:
def potential_function:
def ODE:
def ODE_solution: #diff(h)
    Integrate left Psi -> from -Inf to Rmatch
    Integrate right Psi <- from Inf to Rmatch
def Normalize:
def Bisection:
    Quit after Nmax iterations
```

2.4 Results and Analysis

We can calculate eigenvalues and eigenstates using previous algorithm. We put $\hbar = m = 1$ in our simulation. In harmonic oscillator we put $\omega = 1$. In Rosen-Morse potential we put = 3 and $s = 4$. In infinite well because we cant use infinity in our coding we used a very large number(10^{20}) as its non-zero potential.

Infinite Well potential:

$$V(x) = \begin{cases} \infty(10^20) & \text{if x < 1/4 of spatial distance we are simulating} \\ \infty(10^20) & \text{if x > 3/4 of spatial distance we are simulating} \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

Rosen-Morse potential:

$$V_s(x) = 2tanh(x) - s(s+1)sech^2(x) \tag{18}$$

Harmonic Oscillator potential:

$$V(x) = m\omega^2 x^2/2 \tag{19}$$

Eigenvalues are in table down below. Eigenfunctions are plotted in Fig 3-5.

Eigenvalues			
State	Infinite Well	Rosen-Morse Potential	Harmonic Oscillator
Ground	0.034375	0.440443	0.5
First	0.136719	2.070249	1.5
Second	0.308594	3.698438	2.5
Third	0.548438	5.179150	3.5
Fourth	0.856250	6.108594	4.5
Fifth	1.234375	6.603125	5.5

We can check our results by looking to analytical results of harmonic oscillator.

$$E_n = \hbar\omega(n + \frac{1}{2}) \tag{20}$$

We will see for $\hbar = \omega = 1$ our results will correspond completely to analytical results.

Infinite Well's Wave functions

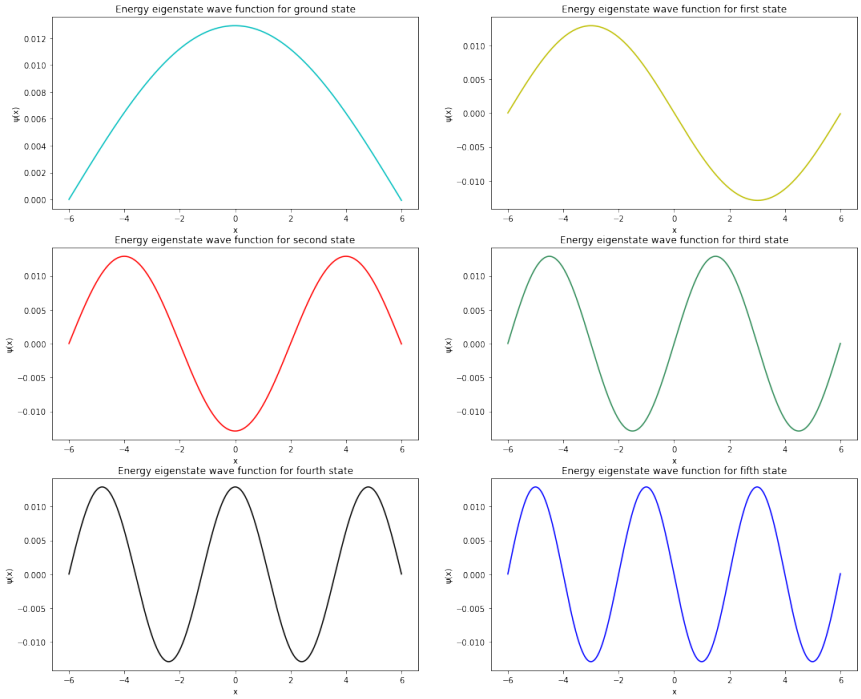

Figure 3: Infinite Square Well

Rosen-Morse Potential's Wave functions

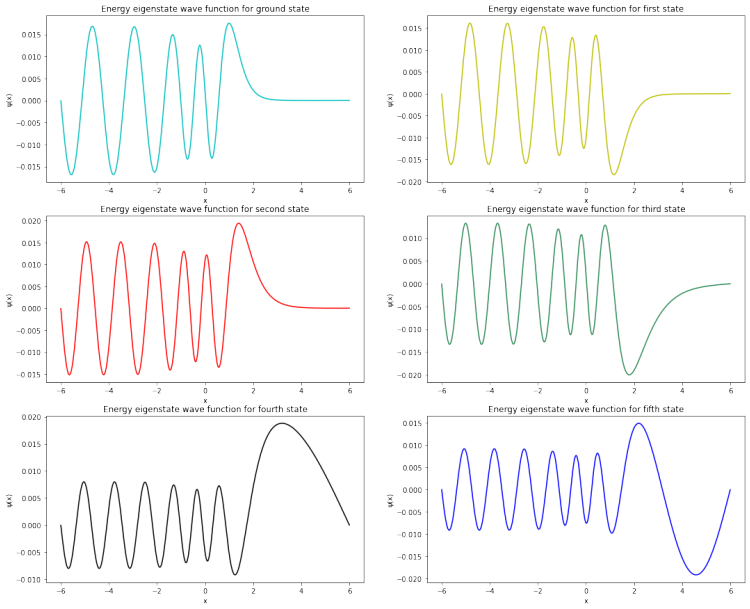

Figure 4: Rosen-Morse Potential

Harmonic Oscillator's Wave functions

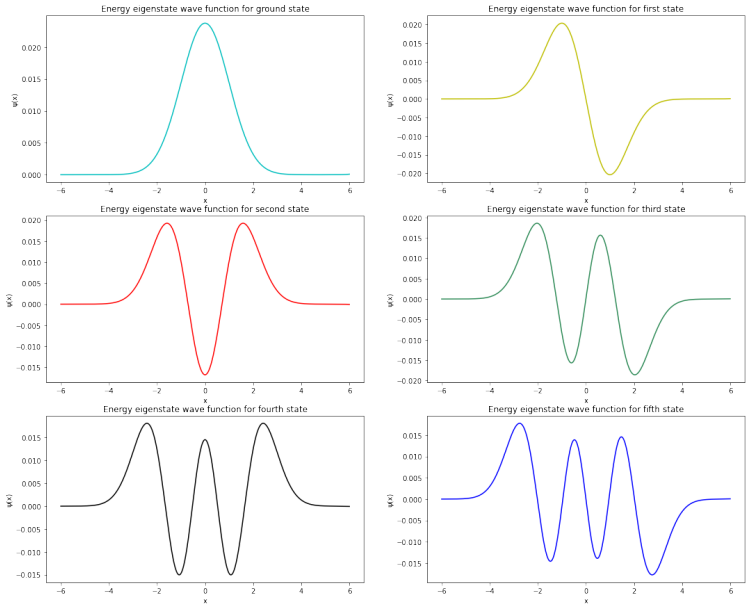

Figure 5: Harmonic Oscillator

3 Time-dependent Schrodinger Equation

The time evolution of the state vector $|\psi(t)\rangle$ is governed by time-dependent Schrödinger equation. H is the hamiltonian operator. Hamiltonian corresponds to the total energy of the system.

$$i\hbar\frac{\partial\langle\psi(t)|}{\partial t} = \hat{H} \tag{21}$$

Probability density is the physical meaning of a wave function. and it can be written as

$$P(\overrightarrow{r}, t) = |\psi(\overrightarrow{r}, t)|^2 \tag{22}$$

The total probability density must be

$$\int_{-\infty}^{\infty} |\psi(\overrightarrow{r}, t)|^2 d^3r = 1 \tag{23}$$

In this project we study one-dimensional Schrödinger equation, So we will have

$$-\frac{\hbar^2}{2m}\frac{d^2\psi(x,t)}{dx^2} + V(x)\psi(x,t) - i\hbar\frac{d\psi(x,t)}{dt} \tag{24}$$

If we put $\hbar = m = 1$ we will get

$$i\frac{d\psi(x,t)}{dt} = \frac{1}{2}\frac{d^2}{dx^2}\psi(x,t) + V(x)\psi(x,t) \tag{25}$$

3.1 Finite Difference Method

Finite difference is a suitable techniques to solve differential equations numerically. There are other methods as well, like finite element, finite volume methods, collocation methods, spectral methods and etc. Finite difference methods convert ordinary differential equations (ODE) or partial differential equations (PDE), they can be nonlinear, into a system of linear equations that can be solved by matrix algebra techniques.

In this method both the spatial domain and time interval are discretized, or broken into a finite number of steps. The value of the solution at these discrete points is approximated by solving algebraic equations containing finite differences and values from nearby points.

First we need to generate a grid, which is a finite set of points on which we seek the function values that represent an approximate solution to the differential equation. The parameter n can be chosen according to accuracy requirement. h is called the step size.

$$x_i = ih, \quad i = 0, 1, ..., n, \quad h = \frac{1}{n}$$

Then we represent derivative by some finite difference formula at every grid point where the solution is unknown, to get an algebraic system of equations.

$$\phi"(x) = \lim_{\Delta x \to 0} \frac{\phi(x-\Delta x) - 2\phi(x) + \phi(x+\Delta x)}{(\Delta x)^2} \tag{26}$$

So we can approximate

$$u"(x_i) = \frac{u(x_i - h) - 2u(x_i) + u(x_i + h)}{(h)^2} \tag{27}$$

We open the complex wave function to its real and imaginary part.

$$\psi(x,t) = Re\psi(x,t) + i\,Im\psi(x,t) \tag{28}$$

So in this problem we must generate a grid and use a time step. We've used a Crank-Nicolson time step.

$$i\frac{\psi_j^{n+1}-\psi_j}{dt} = -\frac{1}{2}\left(\frac{\psi_j^{n+1}-2\psi_j^{n+1}+\psi_{j-1}^{n+1}}{dx^2}+\frac{\psi_{j+1}^{n}-2\psi_j^{n}+\psi_{j-1}^{n}}{dx^2}\right)+\frac{V_j}{2}(\psi_j^{n+1}+\psi_j^n) \tag{29}$$

By rearranging we will get

$$\frac{1}{2}\frac{dt}{dx^2}\psi_{j+1}^{n+1}+\left(i-\frac{dt}{dx^2}-\frac{1}{2}V_j\right)+\frac{1}{2}\frac{dt}{dx^2}\psi_{j-1}^{n+1} = i\psi_j^n-\frac{1}{2}\frac{dt}{dx^2}\left(\psi_{j-1}^n-2\psi_j^n+\psi_{j-1}^n\right)+\frac{1}{2}V_j\psi_j^n \tag{30}$$

Writing this in matrix form:

$$\begin{bmatrix} A_0 & A_- & 0 & \dots & 0 \\ A_+ & A_0 & A_- & \dots 0 & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & A_0 \end{bmatrix}\begin{bmatrix} \psi_0^{n+1} \\ \psi_1^{n+1} \\ \dots \\ \psi_{j-1}^{n+1} \end{bmatrix} = \begin{bmatrix} \psi_0^n \\ \psi_1^n \\ \dots \\ \psi_{j-1}^n \end{bmatrix}$$

The left matrix is known as tridiagonal matrix. In linear algebra, a tridiagonal matrix is a band matrix that has nonzero elements on the main diagonal, the first diagonal below this, and the first diagonal above the main diagonal only. We rewrite this as

$$Ax = b \tag{31}$$

3.2 Algorithm

In one dimension using Eq.28 we can write

$$\bar{H}\psi(x,t) = -\frac{\psi(x+\delta x)-2\psi(x,t)+\psi(x-\Delta x,t)}{\Delta x^2}+V(x)\psi(x,t) \tag{32}$$

We write the Hamiltonian in sparse matrix form for one dimension.

$$\bar{H} = \frac{-\hbar^2}{2m\Delta x^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix}+\begin{bmatrix} V(x_0) & 0 & 0 & 0 & 0 \\ 0 & V(x_1) & 0 & 0 & 0 \\ 0 & 0 & V(x_2) & 0 & 0 \\ 0 & 0 & 0 & V(x_3) & 0 \\ 0 & 0 & 0 & 0 & V(x_4) \end{bmatrix} \tag{33}$$

Equation 34 will be

$$\left(I+\frac{i\Delta t}{2}\bar{H}\right)\psi(x,t+\Delta t) = \left(I-\frac{i\Delta t}{2}\bar{H}\right)\psi(x,t) \tag{34}$$

$$\psi(m,n) = \psi(m\Delta x, n\Delta t) \tag{35}$$

Now everything is in finite-difference form. We must first solve the linear algebra problem with consideration of boundaries and then we will have eigenvalues and eigenstates. The algorithm consists of these steps.

1. Choose an intial wave. THe common choice is a Gaussian multiplied by a plane wave.

$$\psi(x, t_0) = exp\left[-\frac{1}{2}\left(\frac{x - x_0}{\sigma_0}\right)^2\right] e^{ik_0 x} \tag{36}$$

2. Normalize.
3. Write Hamiltonian as a sparse matrix like Eq.33.
4. Make linear algebra system given in Eq.34.
5. Solve it using Tridiagonl matrix algorithm. We seperated the real and imagenery part and solve each part using numpy library *np.linalg.solve*.
6. Be considerate of boundary conditions.
7. You have calculated the wave function.
8. Do a production run with 100 time steps and animate it.

See appendix A.3 from Computational Physics by Giordano and Nakanishi for more details. (6)

3.3 Results and Analysis

We put $\hbar = m = 1$ in our simulation. In harmonic oscillator we put $\omega = 70s^{-1}$. Simulation is done in Python 3.8.5. We used scipy Version 1.5.2 for calculation regarding tridiagonal matrix. Animation are made using matplotlib.animation but are converted to HTML representation of the animation. If you are using Jupiter Notebook, you can type %mathplotlib notebook. This way you have changed the back end and the animation can be shown in the terminal.

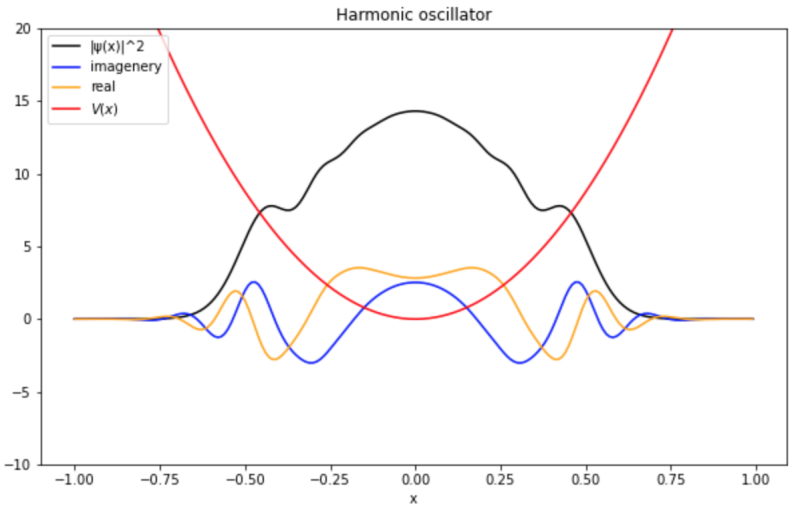

Figure 6: Harmonic oscillator

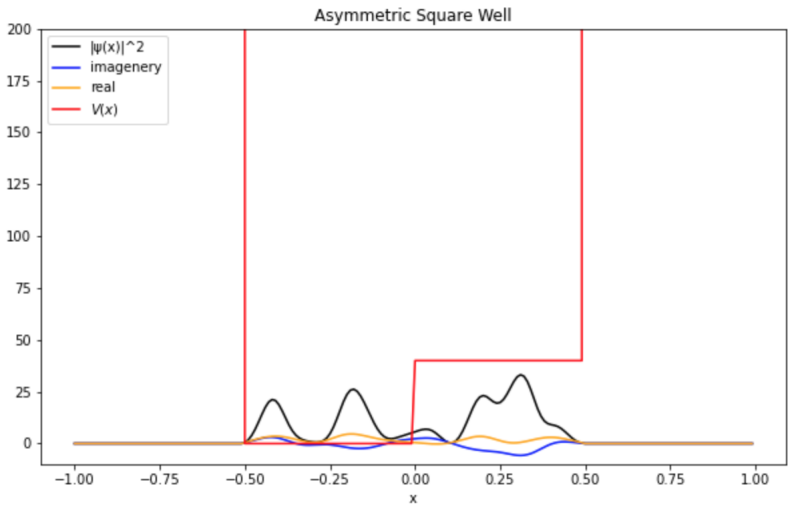

Figure 7: Asymmetric Square Well

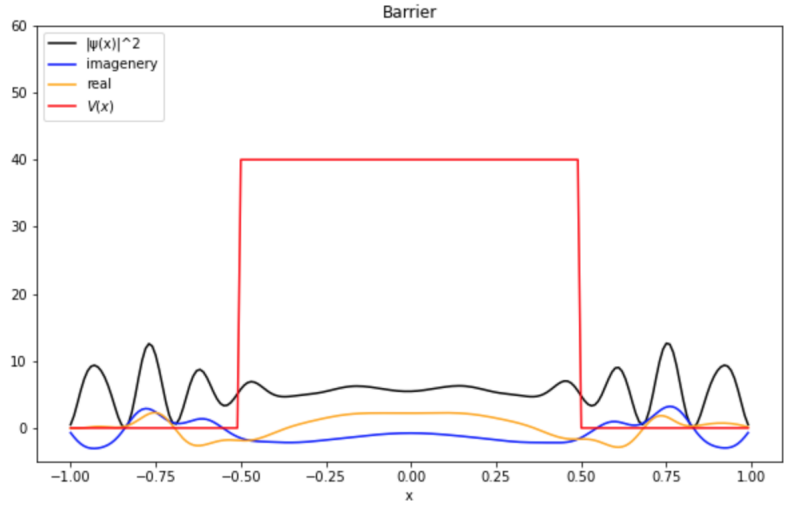

Figure 8: Barrier

References

[1] Landau R.H, Páez M.J, Bordeianu, C.C. *Computational Physics*. 3rd rev. Germany:Wiley-VCH;2015.

[2] Landau R.H., Páez M.J.*Computational Problems for Physics*. 2nd rev. USA:CPD Press;2018.

[3] Hecht, K.T. *Quantum Mechanics*. 1st rev. USA:Springer;2000.

[4] Zettili N. *Quantum Mechanics Concepts and Application*. 2nd rev. USA:Wiley;2009.

[5] Mcintyre D.H. *Quantum Mechanics A Paradigms Approach*. 1st rev. USA:Pearson Addison-Wesley:2012.

[6] Giordano N.J, Nakanishi H. *Computational Physics*. 2nd rev. USA:Person;2006.

[7] Mazumder S. *Numerical Methods for Partial Differential Equations Finite Difference and Finite Volume Methods.* 1st rev. UK:Elsevier;2016.

[8] Li Z, Qiao Z, Tang T. *Numerical Solution of Differential Equations: Introduction to Finite Difference and Finite Elements Methods.* 1st rev. UK:University Printing House;2018.