Project for Numerical Methods and Programming

Title of the Project

Solving Time Dependent Schrodinger Wave Equation for Different Absorbing Boundary conditions using Crank Nikolson Method

Under Guidance Of

Dr. Nirmal Ganguli

Department of Physics, IISER Bhopal

A project authored by

Devashish Tiwari

Indian **I**nstitute **o**f **S**cience **E**ducation and **R**esearch, **B**hopal

Abstract

This project deals with the problem of reflection occurring at the boundaries. When a travelling wave strikes the boundaries of the Simulation box, a huge reflection occurs at the boundaries. This problem gives unambiguous results when the potential barrier is simulated. Here, we have tried to minimize the reflection coefficient at the boundaries to perfectly simulate the scattering of waves. We used three different methods to remove this problem, namely linear approximation, rational function approximation and new absorbing boundary conditions.

The Python code and the Presentation for the Project is attached here

Contents

Chapter 1

Introduction

The Schrödinger wave equation is a fundamental equation in quantum mechanics that describes the behaviour of quantum systems. It's not always possible to get accurate solutions of Schrodinger wave equations. So, the numerical methods becomes one of the most efficient way to obtain solutions for theoretically non-solvable systems. In many cases, the solution of the Schrödinger equation requires specifying boundary conditions that determine how the wave function behaves at the system's boundaries. In particular, for systems that are not isolated and are subject to external influences, such as scattering from a potential barrier or decay of a particle, it is necessary to consider absorbing boundary conditions. The boundaries on a spatial domain causes undesirable reflections that disturb the solution. This is especially problematic when simulating a scattering process, as the scattered wave function will reflect back at the potential barrier. While the domain can be enlarged to fix this problem, doing so either reduces accuracy or increases computation time. To solve this problem, many different methods have been developed, such as, so called Absorbing Boundary conditions.

In this Project work, we are using absorbing boundary conditions, with the Crank-Nicolson scheme Ref. [3] to solve the time-dependent Schrodinger equation numerically with Python, for a Gaussian wave packet using three different methods, namely, **Linear Approximation Method** Ref. [1], **Rational function Approximation** Ref. [2], **New Absorbing Boundary Conditions** Ref. [3]. Then there results, were plotted, and accuracy was determined. We also simulated the Scattering for single and double potential barriers with and without Absorbing boundary conditions.

Chapter 2

Absorbing Boundary Conditions

The Schrodinger wave equation is a fundamental equation in quantum mechanics that describes the behaviour of quantum systems. It is a partial differential equation that relates the wave function of a quantum system to its energy.

One of the reasons why solving the Schrodinger wave equation is very special is that it allows us to make predictions about the behaviour of quantum systems. By solving the Schrodinger equation, we can determine the energy levels and wave functions of quantum systems, which provide insights into the physical properties and behaviour of these systems.

2.1 Time Dependent Schrodinger wave equation

The time-evolution in a one-dimensional quantum mechanical system is determined by the time-dependent Schrödinger equation (TDSE)

$$i\hbar\frac{\partial}{\partial t}\psi(x,t) = \hat{H}\psi(x,t) \tag{2.1}$$

with the Hamilton operator

$$\hat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x,t) \tag{2.2}$$

Here, $\hbar$ is the reduced Planck constant, $i = \sqrt{-1}, m$ the mass of the particle, V the potential and ψ the wave function. If $\hat{H}$ is not time-dependent, the formal solution of the TDSE is given by

$$\psi(x,t) = \hat{U}(t)\psi(x,0) = e^{-\frac{i\hat{H}t}{\hbar}}\psi(x,0) \tag{2.3}$$

where $\hat{U}(t)$ is called the propagator. Since $\hat{U}$ is a unitary operator , the time-evolution operator $\hat{U}$

conserves the norm of the wave function

$$|\psi(x,t)|^2 = |\psi(x,0)|^2$$

Note that the norm squared of the wave function, $|\psi(x,t)|^2$, describes the probability density of the position of the particle. This guarantees that either the wave function is same, or it may maximum differ by a phase, such that there probability density is same.

2.2 Absorbing Boundary Conditions

When using computational methods to solve the TDSE, the spatial domain is necessarily restricted by the boundaries of the simulation domain. The boundaries generally cause unwanted reflections, which can disrupt the results of the computer simulation. However, simply enlarging the computation area to alleviate the problem is inefficient. Therefore, here we construct absorbing boundary conditions (ABCs) to minimize reflections at the borders.

To do this, we have used three different methods, namely,

1. Linear Approximation Method (LAM)
2. Rational Function Approximation (RFA)
3. New Absorbing Boundary Conditions

The derivations of each of the methods follow from the Ref[1], Ref[2], and Ref[3], respectively. Though the LAM and RFA is done in the Finite difference method, **we independently generalized that using Crank-Nicolson Scheme** , and then compared the result with New Absorbing Boundary conditions.

Before deriving these results here, we should know the conditions which is getting approximated to get the result of Absorbing Boundary Conditions.

To construct absorbing boundary conditions, the boundary should always be almost transparent for a plane wave of the form

$$\psi(x,t) = \exp[-i(\omega t - kx)] \tag{2.4}$$

Due to this assumption, the expressions are only valid for scattering states. The idea is to construct an algebraic equation for the wave vector k and the frequency ω and then to use the correspondence

between the $x-t$ space and the $k-\omega$ space to construct a differential equation on the boundaries which is transparent for the plane waves. It is clear that it is impossible to design fully absorbing boundary conditions. From (2.1) and (2.4) one gets the dispersion relation for the wave vector k,

$$\hbar^2 k^2 = 2m[\hbar\omega - V(x)]. \tag{2.5}$$

This relation can be solved for k and yields

$$\hbar k = \pm\sqrt{2m[\hbar\omega - V(x)]}, \tag{2.6}$$

where plus sign describes waves moving to x = oo & the minus sign means waves moving x = —oo. The left boundary has to be transparent for left-going waves and the right boundary must be transparent for right-going waves. **To transform equation (2.6) back into the x-t space, one needs an approximation for the square root which can be easily transformed into a differential equation at the boundaries.**

We used three different approximations, whose details are as follows,

2.2.1 Linear Approximation Method

Unfortunately, function (2.6) is not rational and cannot be converted into a partial differential equation. Therefore we approximate this relation to

$$\hbar k = \pm\frac{(2m^*\alpha_2)^{1/2} - (2m^*\alpha_1)^{1/2}}{\alpha_2 - \alpha_1}(\hbar\omega - V) \pm \frac{\alpha_2\,(2m^*\alpha_1)^{1/2} - \alpha_1\,(2m^*\alpha_2)^{1/2}}{\alpha_2 - \alpha_1}. \tag{2.7}$$

Equation (2.7) is a straight line. The plus and minus signs in (5) correspond to the right-going and left-going waves, respectively. The correspondence of $\partial/\partial t = i\omega$ and $\partial/\partial x = ik$ leads us to rewrite Eq. (2.7) into a partial differential equation of

$$i\hbar\frac{\partial}{\partial t}\Psi(x,t) = \left(-i\hbar\frac{1}{g_1}\frac{\partial}{\partial x} + U - \frac{g_2}{g_1}\right)\Psi(x,t), \tag{2.8}$$

where

$$g_1 = \pm\frac{(2m^*\alpha_2)^{1/2} - (2m^*\alpha_1)^{1/2}}{\alpha_2 - \alpha_1}$$

and

$$g_2 = \pm\frac{\alpha_2\,(2m^*\alpha_1)^{1/2} - \alpha_1\,(2m^*\alpha_2)^{1/2}}{\alpha_2 - \alpha_1}.$$

Applying this in the numerical methods will might solve our problem. This method was then independently generalized using Crank-Nikolson Scheme. In Ref[1], it was generalized using the Finite difference method.

As the Ref[1], itself suggests, the parameter alpha1 and alpha2 are completely unphysical, and its very difficult to choose there accurate values. Thus, method of Rational Function Approximation was used, which introduces the meanings to the parameters, and hence its easy to choose there accurate values.

2.2.2 Rational Function Approximation

The rational function approximation

$$\sqrt{z - z_0} \approx \sqrt{z_0}\,(1 + 3z/z_0)\,/\,(3 + z/z_0) \tag{2.9}$$

is used. With the approximation (2.9) for the square root in the dispersion relation (2.6) one gets

$$\hbar k = \pm\hbar k_0 \left[\frac{2m(\hbar\omega - V)}{\hbar^2 k_0^2}\right]^{1/2} \approx \pm\hbar k_0 \frac{1 + 3z}{3 + z} \tag{2.10}$$

with $z = 2m(\hbar\omega - V)/\hbar^2 k_0^2$. k_0 plays the role of the expansion point in the rational function approximation. The mean wave vector of the localized wave function is still unknown when it arrives at the boundary. One has to choose a value for k_0 in such a way that the boundary is transparent for the incoming wave packet. By using the correspondence between $k \Longleftrightarrow i\partial/\partial x, \omega \Longleftrightarrow i\partial/\partial t$ one gets the partial differential equation.

$$-i\hbar\left(3\frac{\hbar^2 k_0^2}{2m} - V\right)\frac{\partial\psi}{\partial x}(x,t) + \hbar^2\frac{\partial^2\psi}{\partial t\partial x}(x,t) = \pm\hbar k_0\left(\frac{\hbar^2 k_0^2}{2m} - 3V\right)\psi(x,t) \pm 3i\hbar^2 k_0\frac{\partial\psi}{\partial t}(x,t) \tag{2.11}$$

The method of Rational Function Approximation introduces the meanings, related to energy to the parameters, and hence its easy to choose there accurate values. Here also, **we independently generalized this method to Crank Nikolson scheme**, and in Ref[2], it was done using Finite difference scheme.

2.2.3 New Absorbing Boundary Conditions

The absorbing boundary conditions derived here follow from the Research Paper by , Ref[3] by T. Fevens and H. Jiang. Here, we assumed that the potential V is either constant, or a slowly varying function, near the boundaries. We can then calculate group velocity from dispersion relation, (2.6):

$$v_g = \frac{\partial \omega(k)}{\partial k} = \frac{\hbar k}{m} \tag{2.12}$$

A positive group velocity represents a wave travelling to the right, while a negative one represents a wave travelling to the left. If we only want waves leaving the domain at the left boundary, we need to restrict the group velocity to be positive. In mathematical terms this means

$$\frac{\hbar k}{m} = \left| \frac{\hbar k}{m} \right|. \tag{2.13}$$

For the right boundary, k needs to be replaced with $-k$. But this boundary condition is not rational, due to the absolute value function, and therefore cannot be converted to a partial differential equation using our correspondence relation (to convert from k-w space to x-t). Thus we need to use an approximation. We choose,

$$\frac{\hbar k}{m} \equiv q \tag{2.14}$$

where q is positive and real. Using the correspondence relation, we obtained the following differential equation, which is a eigen value equation:

$$\left(i\frac{\partial}{\partial x} + \frac{mq}{\hbar} \right) \psi = 0 \tag{2.15}$$

If this differential equation is satisfied on the boundary, then no waves with group velocity q are reflected, therefore waves with this group velocity are essentially absorbed. Since waves, in general, consist of more than one component with different group velocities, the operator in (2.15) can be generalized by considering more group velocities, which are absorbed. This results in:

$$\prod_{l=1}^{p} \left(i\frac{\partial}{\partial x} + \frac{mq_l}{\hbar} \right) \psi = 0. \tag{2.16}$$

The effect of (2.16) at the boundary differs depending on the choice of q_l. If $q_k \neq q_l$ for $k \neq l$, then p different group velocities, of the computed wave solution, are absorbed to the first order. If $q_k = q_l$ for $k \neq l$, then the group velocity q_k is absorbed to p th order.

Let us now derive some more specific ABCs from Eq. (2.16), starting with $p = 2$. Here we use correspondence relation(C.R.) again,

$$\left(\pm k + \frac{mq_1}{\hbar}\right)\left(\pm k + \frac{mq_2}{\hbar}\right) = 0 \tag{2.17}$$

In this equation the top sign applies at the right and the bottom sign at the left boundary. If we multiply out (2.15) and use C.R. to substitute for k^2 we obtain

$$\mp\hbar k = \frac{2}{q_1 + q_2}(\hbar\omega - V) + \frac{mq_1q_1}{q_1 + q_2}$$

Continuing this process with $p = 3$

$$\left(\pm k + \frac{mq_1}{\hbar}\right)\left(\pm k + \frac{mq_2}{\hbar}\right)\left(\pm k + \frac{mq_3}{\hbar}\right) = 0,$$

leads to

$$\mp\hbar k = \frac{2mh_1(\hbar\omega - V) + h_3}{2m(\hbar\omega - V) + h_2}$$

$$\begin{aligned}
h_1 &= m\left(q_1 + q_2 + q_3\right), \\
h_2 &= m^2 q_1 q_2 q_3 \left(\frac{1}{q_1} + \frac{1}{q_2} + \frac{1}{q_3}\right), \\
h_3 &= m^3 q_1 q_2 q_3.
\end{aligned}$$

Lastly, setting $p = 4$, we obtain the following equation

$$4m^2(\hbar\omega - V)^2 + 2m\left(\pm g_1\hbar k + g_2\right)(\hbar\omega - V) \pm g_3\hbar k + g_4 = 0$$

where

$$\begin{aligned}
g_1 &= m\left(q_1 + q_2 + q_3 + q_4\right), \\
g_2 &= m^2\left(q_1q_2 + q_1q_3 + q_1q_4 + q_2q_3 + q_2q_4 + q_3q_4\right), \\
g_3 &= m^3 q_1 q_2 q_3 q_4 \left(\frac{1}{q_1} + \frac{1}{q_2} + \frac{1}{q_3} + \frac{1}{q_4}\right), \\
g_4 &= m^4 q_1 q_2 q_3 q_4.
\end{aligned}$$

This process can be repeated to derive ABCs of higher order in p.

Chapter 3

Computational Application

In this chapter, we derived the result for solving the time-dependent Schrodinger equation, as well as the numerical implementation of the ABC, to implement them on Python, using the numerical methods approach. We have used Crank-Nicolson Scheme to generalize all three of the methods.

3.1 Crank Nikolson Method for Schrodinger Wave Equation

In order to treat the TDSE numerically, we represent $\psi(x,t)$ by its values at a set of grid-points. For the spatial domain we choose $x = x_j = x_0 + j\Delta x$, with $j \in [0, J]$, where x_0 represents the left boundary and Δx is the grid spacing. Similarly, the time domain has the range $t = t_n = n\Delta t$, with $n \in [0, N]$. The values of the wave function at the grid points will be abbreviated by

$$\psi\left(x_j, t_n\right) \equiv \psi_j^n \tag{3.1}$$

After applying this discretization to the time-evolution given by Eq.(2.3), we obtain

$$\psi_j^{n+1} = e^{-i\hat{H}\Delta t}\psi_j^n = \hat{U}(\Delta t)\psi_j^n \tag{3.2}$$

It is important to note that, since $\hat{U}$ is unitary and preserves the norm of the wave function, any approximations of $\hat{U}(\Delta t)$ must be unitary as well. Therefore simply expanding $\hat{U}$

$$\psi_j^{n+1} = (1 - i\hat{H}\Delta t)\psi_j^n \tag{3.3}$$

does not provide the desired result.

To derive a unitary approximation, we start by splitting $\hat{U}(\Delta t)$ as follows

$$\psi_j^{n+1} = e^{-\frac{i\hat{H}\Delta t}{2}} e^{-\frac{i\hat{H}\Delta t}{2}} \psi_j^n \tag{3.4}$$

and multiply the equation from the left by $\hat{U}^\dagger\left(\frac{\Delta t}{2}\right)$ to obtain

$$e^{\frac{i\hat{H}\Delta t}{2}} \psi_j^{n+1} = e^{-\frac{i\hat{H}\Delta t}{2}} \psi_j^n \tag{3.5}$$

or by expanding the exponential functions into a Taylor series and cutting off after the second term

$$\left(1 + \frac{i\hat{H}\Delta t}{2}\right) \psi_j^{n+1} = \left(1 - \frac{i\hat{H}\Delta t}{2}\right) \psi_j^n \tag{3.6}$$

Thus we have found a unitary method to approximate $\hat{U}$ known as Cayley's form

$$\hat{U}(\Delta t) = e^{-i\hat{H}\Delta t} \simeq \frac{\left(1 - \frac{i\hat{H}\Delta t}{2}\right)}{\left(1 + \frac{i\hat{H}\Delta t}{2}\right)} \tag{3.7}$$

$$\psi_j^{n+1} - \frac{i\Delta t}{2}\left[\frac{\psi_{j+1}^{n+1} - 2\psi_j^{n+1} + \psi_{j-1}^{n+1}}{(\Delta x)^2} - V_j\psi_j^{n+1}\right] = \psi_j^n + \frac{i\Delta t}{2}\left[\frac{\psi_{j+1}^n - 2\psi_j^n + \psi_{j-1}^n}{(\Delta x)^2} - V_j\psi_j^n\right] \tag{3.8}$$

Finally, we introduce the vector $\boldsymbol{\psi}^n = \left(\psi_0^n, \cdots, \psi_j^n, \cdots, \psi_J^n\right)$, so we can write above equation as a matrix equation:

$$\mathcal{U}_1 \psi^{n+1} = \mathcal{U}_2 \psi^n \tag{3.9}$$

Here $\mathcal{U}_1$ and $\mathcal{U}_2$ represent two tridiagonal $(J+1)\times(J+1)$ matrices

$$\mathcal{U}_1 = \begin{pmatrix} \xi_0 & -\alpha & & & \\ -\alpha & \xi_1 & -\alpha & & \\ & \ddots & \ddots & \ddots & \\ & & -\alpha & \xi_{J-1} & -\alpha \\ & & & -\alpha & \xi_J \end{pmatrix}, \quad \mathcal{U}_2 = \begin{pmatrix} \gamma_0 & \alpha & & & \\ \alpha & \gamma_1 & \alpha & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha & \gamma_{J-1} & \alpha \\ & & \alpha & \gamma_J & \end{pmatrix} \tag{3.10}$$

with

$$\alpha = \frac{i\Delta t}{2\Delta x^2}, \xi_j = 1 + \frac{i\Delta t}{2}\left(\frac{2}{\Delta x^2} + V_j\right), \gamma_j = 1 - \frac{i\Delta t}{2}\left(\frac{2}{\Delta x^2} + V_j\right). \tag{3.11}$$

3.1.1 Implementing the method on Python

The code was implemented on python, and the graph obtained are attached in chapter 4. Link to the code is attached in the very first page of the Report.

3.1.2 Implementing the Absorbing Boundary Conditions

Let us now derive the conditions for each of the three methods and then show their implementation in Python.

Linear Approximation Method

Using the differential equation obtained in section 2.2.1
The differential equation that we obtained was:-

$$i\hbar\frac{\partial}{\partial t}\Psi(x,t) = \left(-i\hbar\frac{1}{g_1}\frac{\partial}{\partial x} + U - \frac{g_2}{g_1}\right)\Psi(x,t), \tag{3.12}$$

Now, we aim to apply this equation to the Crank-Nicolson Method.
For that, we need the following formulae, obtained by the Taylor expansion, with respect to x and t.

$$\begin{aligned}
\psi &\approx \frac{1}{4}\left(\psi_{j\pm1}^{n+1} + \psi_j^{n+1} + \psi_{j\pm1}^{n} + \psi_j^{n}\right) \\
\frac{\partial\psi}{\partial x} &\approx \pm\frac{1}{2\Delta x}\left(\psi_{j\pm1}^{n+1} - \psi_j^{n+1} + \psi_{j\pm1}^{n} - \psi_j^{n}\right) \\
\frac{\partial\psi}{\partial t} &\approx \frac{1}{2\Delta t}\left(\psi_{j\pm1}^{n+1} + \psi_j^{n+1} - \psi_{j\pm1}^{n} - \psi_j^{n}\right)
\end{aligned} \tag{3.13}$$

$$\frac{\partial^2\psi}{\partial t\partial x}(x,t) \approx \frac{1}{\Delta t\Delta x}\left(\psi_{j+1}^{n+1} - \psi_j^{n+1} - \psi_{j+1}^{n} + \psi_j^{n}\right) \tag{3.14}$$

Now, keeping the necessarily required relations in equation (2.7), we obtained the following result,

$$i\hbar * (\frac{1}{2\Delta t}\left(\psi_{j\pm1}^{n+1} + \psi_j^{n+1} - \psi_{j\pm1}^{n} - \psi_j^{n}\right)) = \tag{3.15}$$

$$\left(-i\hbar\frac{1}{g_1} * (\pm\frac{1}{2\Delta x}\left(\psi_{j\pm1}^{n+1} - \psi_j^{n+1} + \psi_{j\pm1}^{n} - \psi_j^{n}\right))\right) + (U - \frac{g_2}{g_1}) * (\frac{1}{4}\left(\psi_{j\pm1}^{n+1} + \psi_j^{n+1} + \psi_{j\pm1}^{n} + \psi_j^{n}\right)) \tag{3.16}$$

Now, Simplifying this equation, we get,

$$(a-b+c)\psi_{j\pm1}^{n+1} + (-a-b+c)\psi_j^{n+1} = (-a-b-c)\psi_{j\pm1}^{n} + (a-b-c)\psi_j^{n} \tag{3.17}$$

where, a, b, and c are the parameters, whose values that we assumed are :-

$$a = \frac{i}{2\Delta x g1}, b = -\frac{i}{2\Delta t}, c = V - \frac{g_2}{g_1} \tag{3.18}$$

So, here our equation reduces to,

$$\zeta_1 \psi_j^{n+1} + \zeta_2 \psi_{j\pm1}^{n+1} = \zeta_3 \psi_j^n + \zeta_4 \psi_{j\pm1}^n \tag{3.19}$$

where,

$$\zeta_1 = a - b + c, \quad \zeta_2 = -a - b + c, \quad \zeta_3 = -a - b - c, \quad \zeta_4 = a - b - c \tag{3.20}$$

Now, our matrix for the Linear Approximation method turned out to be,

$$\mathcal{U}_1 = \begin{pmatrix} \zeta_1 & \zeta_2 & & & \\ -\alpha & \xi_1 & -\alpha & & \\ & \ddots & \ddots & \ddots & \\ & & -\alpha & \xi_{J-1} & -\alpha \\ & & & \zeta_2 & \zeta_1 \end{pmatrix}, \quad \mathcal{U}_2 = \begin{pmatrix} \zeta_3 & \zeta_4 & & & \\ \alpha & \gamma_1 & \alpha & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha & \gamma_{J-1} & \alpha \\ & & & \zeta_4 & \zeta_3 \end{pmatrix} \tag{3.21}$$

Now, this matrix is implemented on Python, whose code is attached in the first page of this report.

Rational Function Approximation

Now, going back to the equation that we had for Rational function approximation,

$$-i\hbar \left(3\frac{\hbar^2 k_0^2}{2m} - V\right) \frac{\partial \psi}{\partial x}(x,t) + \hbar^2 \frac{\partial^2 \psi}{\partial t \partial x}(x,t) = \pm \hbar k_0 \left(\frac{\hbar^2 k_0^2}{2m} - 3V\right) \psi(x,t) \pm 3i\hbar^2 k_0 \frac{\partial \psi}{\partial t}(x,t) \tag{3.22}$$

Now, in this, keeping the above relations written above,

$$-i\hbar \left(3\frac{\hbar^2 k_0^2}{2m} - V\right) * (\pm \frac{1}{2\Delta x} \left(\psi_{j\pm1}^{n+1} - \psi_j^{n+1} + \psi_{j\pm1}^n - \psi_j^n\right)) \tag{3.23}$$

$$+\hbar^2 * (\frac{1}{\Delta t \Delta x} \left(\psi_{j+1}^{n+1} - \psi_j^{n+1} - \psi_{j+1}^n + \psi_j^n\right)) \tag{3.24}$$

$$= \pm \hbar k_0 \left(\frac{\hbar^2 k_0^2}{2m} - 3V\right) * (\frac{1}{4} \left(\psi_{j\pm1}^{n+1} + \psi_j^{n+1} + \psi_{j\pm1}^n + \psi_j^n\right)) \tag{3.25}$$

$$\pm 3i\hbar^2 k_0 * (\frac{1}{2\Delta t}\left(\psi_{j\pm1}^{n+1} + \psi_j^{n+1} - \psi_{j\pm1}^{n} - \psi_j^{n}\right)) \tag{3.26}$$

On solving this expression, we get the following expression, with the required variables,

$$(a-b+c+d)\psi_{j\pm1}^{n+1} + (a+b+c-d)\psi_j^{n+1} = (a+b-c+d)\psi_{j\pm1}^{n} + (a-b-c-d)\psi_j^{n}$$

where,

$$a = \frac{3ik_0}{2\Delta t},\ \ b = \frac{3k_0^2 i}{2\Delta t},\ \ c = \frac{k_0^3}{4},\ \ d = \frac{1}{\Delta t \Delta x} \tag{3.27}$$

$$\zeta_1 \psi_j^{n+1} + \zeta_2 \psi_{j\pm1}^{n+1} = \zeta_3 \psi_j^{n} + \zeta_4 \psi_{j\pm1}^{n} \tag{3.28}$$

where,

$$\zeta_1 = a+b+c-d, \quad \zeta_2 = a-b+c-d, \quad \zeta_3 = a-b-c-d, \quad \zeta_4 = a+b-c+d \tag{3.29}$$

Now, our matrix for the Rational Function Approximation turned out to be,

$$\mathcal{U}_1 = \begin{pmatrix} \zeta_1 & \zeta_2 & & & \\ -\alpha & \xi_1 & -\alpha & & \\ & \ddots & \ddots & \ddots & \\ & & -\alpha & \xi_{J-1} & -\alpha \\ & & & \zeta_2 & \zeta_1 \end{pmatrix}, \quad \mathcal{U}_2 = \begin{pmatrix} \zeta_3 & \zeta_4 & & & \\ \alpha & \gamma_1 & \alpha & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha & \gamma_{J-1} & \alpha \\ & & & \zeta_4 & \zeta_3 \end{pmatrix} \tag{3.30}$$

Now, applying this in the python code.

New Absorbing Boundary Conditions

The Differential equation that we obtained for p = 2 (in section 2.2.3) is :-

$$\begin{gathered} \pm i\frac{\partial \psi}{\partial x} - ic_1 \frac{\partial \psi}{\partial t} + (c_1 V - c_2)\,\psi = 0, \\ c_1 = \frac{2}{q_1+q_2}, \quad c_2 = \frac{q_1 q_2}{2\,(q_1+q_2)}. \end{gathered} \tag{3.31}$$

After keeping the standard results that are used in LAM, we get the following equation,

$$
\begin{aligned}
&\zeta_1\psi^{n+1}_{(0,J)} + \zeta_2\psi^{n+1}_{(1,J-1)} = \zeta_3\psi^{n}_{(0,J)} + \zeta_4\psi^{n}_{(1,J-1)},\\
&\zeta_1 = \left(-\frac{i}{2\Delta x} - \frac{ic_1}{2\Delta t} + \frac{\left(c_1V_{(0,J)} - c_2\right)}{4}\right),\\
&\zeta_2 = \left(\frac{i}{2\Delta x} - \frac{ic_1}{2\Delta t} + \frac{\left(c_1V_{(0,J)} - c_2\right)}{4}\right),\\
&\zeta_3 = \left(\frac{i}{2\Delta x} - \frac{ic_1}{2\Delta t} - \frac{\left(c_1V_{(0,J)} - c_2\right)}{4}\right),\\
&\zeta_4 = \left(-\frac{i}{2\Delta x} - \frac{ic_1}{2\Delta t} - \frac{\left(c_1V_{(0,J)} - c_2\right)}{4}\right).
\end{aligned}
\tag{3.32}
$$

Thus, the modified matrix are,

$$
\mathcal{U}_1 = \begin{pmatrix}
\zeta_1 & \zeta_2 & & & \\
-\alpha & \xi_1 & -\alpha & & \\
 & \ddots & \ddots & \ddots & \\
 & & -\alpha & \xi_{J-1} & -\alpha \\
 & & & \zeta_2 & \zeta_1
\end{pmatrix}, \quad
\mathcal{U}_2 = \begin{pmatrix}
\zeta_3 & \zeta_4 & & & \\
\alpha & \gamma_1 & \alpha & & \\
 & \ddots & \ddots & \ddots & \\
 & & \alpha & \gamma_{J-1} & \alpha \\
 & & & \zeta_4 & \zeta_3
\end{pmatrix}.
\tag{3.33}
$$

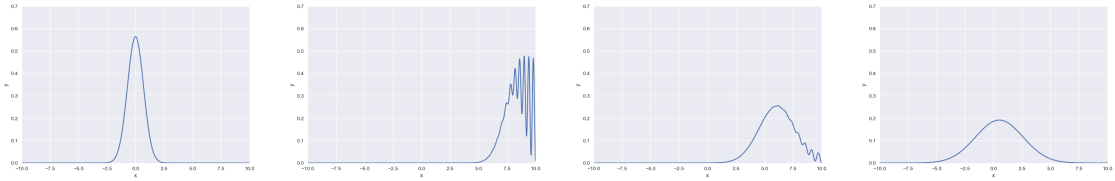
This was then applied in Python.

Chapter 4

Results and Comparison of the Numerical Methods

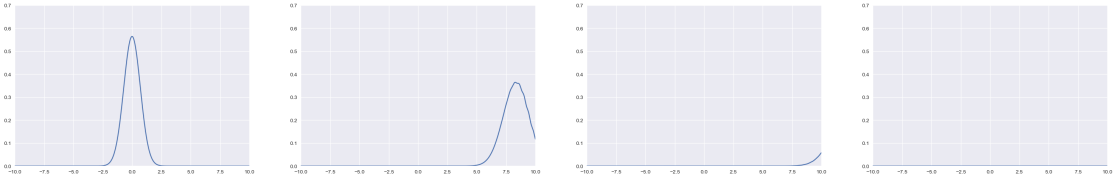
The results that we obtained for different methods are mentioned in this chapter.

Plots were obtained from a simple Crank-Nicholson Method when no absorbing boundary conditions were applied.



(a) t = 0 (b) It = 500 (c) t = 1000 (d) t= 1500

Figure 4.1: Plots without ABC



(a) t = 0 (b) It = 500 (c) t = 1000 (d) t= 1500

Figure 4.2: Plots with ABC, p =3

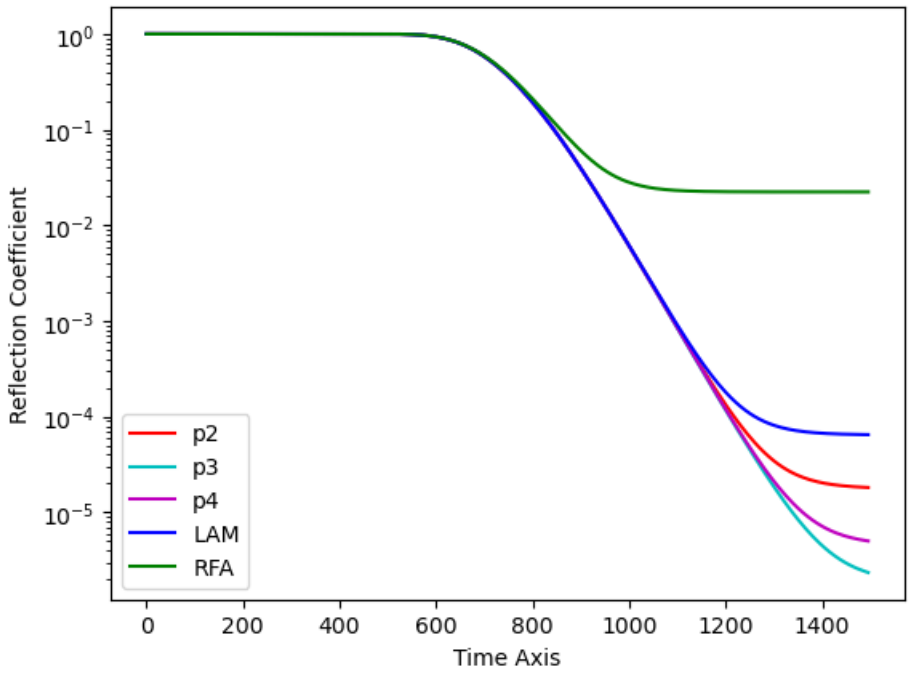
Also, we compared the reflection coefficients of various methods that we used. The Graph for their comparison is shown in the next section.

4.0.1 Reflection Coefficient

The reflection ratio/reflection coefficient is a parameter that describes how much of a wave is reflected by an impedance discontinuity in the transmission medium. It is equal to the ratio of the amplitude of the reflected wave to the incident wave.

$$R = \frac{\sum_{j=0}^{J} \left|\psi_j^n\right|^2 \Delta x}{\sum_{j=0}^{J} \left|\psi_j^0\right|^2 \Delta x} \tag{4.1}$$

- If R = 1, The wave packet is completely reflected.
- If R = 0, The wave packet is completely absorbed.‘



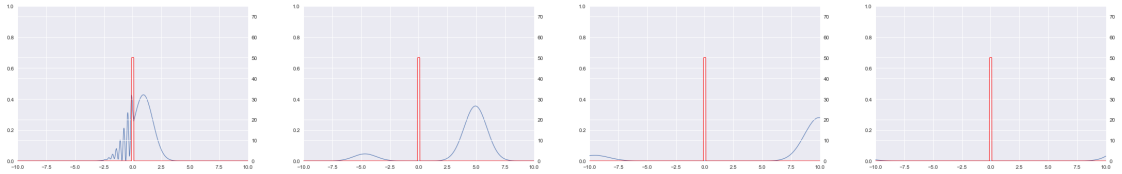
- As we can observe, Rational Function approximation is the poor method of all the methods.
- New Absorbing Boundary condition for p = 3 is the best method that gives the answer with high accuracy.

We also observed that, the reflection ratio vs time curve dies from 1 to 0 as the wave progress towards the boundaries. It’s a one-step decay curve.

4.0.2 Potential Step

The main aim of applying ABC was to get an accurate simulation of the scattering of waves. So now, we introduce the rectangular potential barrier in between the path of the traveling wave packet. The result that we got for the single barrier, as well as the double barrier, are attached for different times. We can see that after ABC is applied, no further reflection is observed on the simulational boundary.

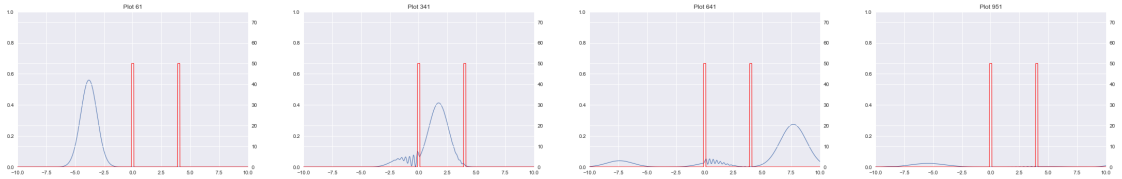
Single Barrier



(a) t = 0 (b) It = 500 (c) t = 1000 (d) t= 1500

Figure 4.3: Plots of SIngle Barrier Potential after ABC was applied, p = 3

Double Barrier



(a) t = 0 (b) It = 500 (c) t = 1000 (d) t= 1500

Figure 4.4: Plots for double barrier when ABC was applied, p = 3

Chapter 5

References

The majority of the help for this project is taken from the following research papers :-

1. T. Shibata. Absorbing boundary conditions for the finite-difference time-domain calculation of the one-dimensional Schrodinger equation. Phys. Rev. B, 43(8):6760– 6763, 1991

2. J.-P. Kuska. Absorbing boundary conditions for the Schr¨odinger equation on finite intervals. Phys. Rev. B, 46(8):5000–5003, 1992

3. T. Fevens and H. Jiang. Absorbing Boundary Conditions for the Schr¨odinger Equation. SIAM J. Sci. Comput., 21(1):255–282, 1999.