5CCP211C — Introduction to Numerical Modelling
Physics Department — King's College London

KING'S College LONDON

Application of the FDTD algorithm to quantum tunelling in 1D

Written by M. Haider

January 21, 2019

Abstract

Using computational methods is an efficient way to solve the time-dependent 1D Schrödinger equation. Gaussian wave-packets can be used to visualise the propagation of the wavefunction. Application of the FDTD numerical algorithm utilising the Crank-Nicolson method is used to investigate the interactions of a single particle in a one dimension system with different interacting potentials. The results are found to deviate from classical physics and supports the results predicted by Quantum mechanics.

1 Introduction

The appearance and time evolution of the Schrödinger equation is often difficult to visualise especially when applied to a physical quantum mechanical system. This is because solving the Schrödinger equation analytically is very difficult. This investigation will use the Finite-difference time-domain method to discretise the problem such that the dynamical evolution of the wavefunction as constricted to a Gaussian can be seen. The effects of interacting potentials; step, double and well will also be investigated, with emphasis on the conditions which produce quantum tunnelling as predicted by theory.

2 Theoretical Background

The Schrödinger equation is a linear partial differential equation, *PDE*, which describes the time-evolution of a systems wave-function; the wave-function defines the state of a system at each spatial and temporal position. The one-dimensional time dependent Schrödinger equation is given by:

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi(x,t)}{\partial x^2} + V(x)\psi(x,t) \tag{1}$$

where $i = \sqrt{-1}$, $\hbar$ is the reduced Planck's constant, *ψ(x,t)* is the wavefunction in space (x) and time (t), $V(x)$ is the potential at position x and m is the mass of the particle in question.

This is a a complex second order *PDE* which is very difficult to solve analytically. By limiting the problem to one dimension, there are less variable coefficients which need to be obtained. The number of particles are also limited to one, (**N** particles in **3** spatial and **1** temporal dimension(s) require **3N+1** variables) to simplify the simulation.

This wave-function is a complex quantity and may be separated into its real and imaginary components:

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

$$\hbar\frac{\partial\psi_I(x,t)}{\partial t} = +\frac{\hbar^2}{2m}\frac{\partial^2\psi_R(x,t)}{\partial x^2} - V(x)\psi_R(x,t) \tag{3}$$

Solving the time-dependent Schrödinger equation analytically can be very difficult. This is because the complex relations of a systems interaction potentials form a *non-seperable PDE*; one which cannot be simplified into equations of lower dimensionality [1]. Following a numerical approach, the Finite Difference Method, a type of discretisation process, will be used to reduce the *PDE's* into a system of linear algebraic equations. More specifically, the Finite-difference time-domain, *FDTD*, will be

used to discretize the *PDE's* using central-difference approximations to the space and time partial derivatives.

Following the Explicit (forward difference) and Implicit (backward difference) Euler methods produce equations which result in unstability and large errors [2]. Whilst the central-difference method is unconditionally stable with smaller errors [2]. The central-difference approximation, more specifically, the Crank-Nicolson method is used with a grid-based representation of the wavefunction. This will be done by separating the spatial and temporal distances into a grid based system. Separation of spatial distance by Δx and temporal by Δt imposes the following conditions:

$$\begin{cases} x_l = x_0 + l\Delta x & , l \,\epsilon\, [0, L] \\ t_n = t_0 + n\Delta t & , n \,\epsilon\, [0, N] \end{cases} \tag{4}$$

where L and N are defined such that they form the total distance and time when multiplied by Δx and Δt, respectively. The initial conditions are usually taken such that $x_0, t_0 = 0$. To simplify calculations, these can be represented in the form of a compact *Stencil*, turning the wave-function into:

$$\psi(x_n, t_n) = \psi^n(l) \tag{5}$$

Using this notation, one can differentiate between spatial and temporal positions easily. This can be visualised to look like a grid where change in one dimension (spatial or temporal) is achieved by moving across the respective grid element.

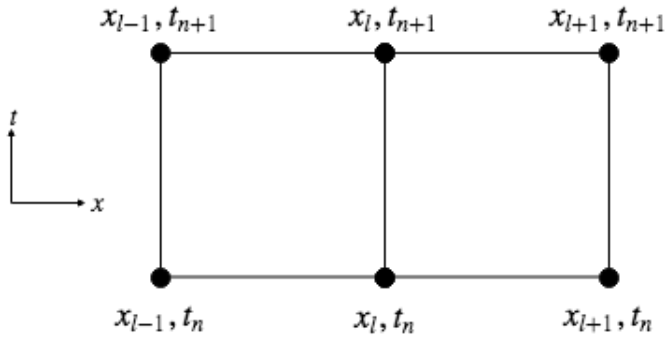


Figure 1: The Crank-Nicolson stencil for a 1D problem.

The assumptions made is that the functions are continuous (and smooth) where the first and second derivatives can be found. Approximating the interaction to the nearest two neighbours, a 'three point stencil' can be used to find the elements contributing to the first and second derivatives. Utilising this approach, the imaginary component of the wave-function is defined to exist at half-step intervals from the real. The real and imaginary components thus become:

$$\frac{\partial \psi_R(x_l, t_{n+\frac{1}{2}})}{\partial t} \approx \frac{\psi_R^{n+1} - \psi_R^n}{\Delta t} \tag{6}$$

$$\frac{\partial \psi_I(x_l, t_n)}{\partial t} \approx \frac{\psi_I^{n+\frac{1}{2}} - \psi_I^{n-\frac{1}{2}}}{\Delta t} \tag{7}$$

Applying the same approach to the second spatial derivatives gives:

$$\frac{\partial^2 \psi_R(x_l, t_n)}{\partial x^2} \approx \frac{\psi_R^n(l+1) - 2\psi_R^n(l) + \psi_R^n(l-1)}{\Delta x^2} \tag{8}$$

$$\frac{\partial^2 \psi_I(x_l, t_n)}{\partial x^2} \approx \frac{\psi_I^{n+\frac{1}{2}}(l+1) - 2\psi_I^{n+\frac{1}{2}}(l) + \psi_I^{n+\frac{1}{2}}(l-1)}{\Delta x^2} \tag{9}$$

The functions $\psi_R^n(l)$ and $\psi_I^{n-\frac{1}{2}}(l)$ represent the present states whereas $\psi_R^{n+1}(l)$ and $\psi_I^{n+\frac{1}{2}}(l)$, are the future states. The aim of employing this numerical method is to observe the time evolution of the system, hence it is beneficial to view the progressive future states of the wave-function. Substituting these back into the real/imaginary separated Schrödingers equation and re-arranging for the future states give the following equations:

$$\psi_R^{n+1}(l) = -C_1[\psi_I^{n+\frac{1}{2}}(l+1) - 2\psi_I^{n+\frac{1}{2}}(l) + \psi_I^{n+\frac{1}{2}}(l-1)] + C_2 V(l)\psi_I^{n+\frac{1}{2}}(l) + \psi_R^n(l) \tag{10}$$

$$\psi_I^{n+\frac{1}{2}}(l) = +C_1[\psi_R^n(l+1) - 2\psi_R^n(l) + \psi_R^n(l-1)] - C_2 V(l)\psi_R^n(l) + \psi_I^{n+\frac{1}{2}}(l) \tag{11}$$

where $C_1 = \frac{\hbar \Delta t}{2m\Delta x^2}$ and $C_2 = \frac{\Delta t}{\hbar}$.

The equations above utilise the recursive relation that are inherent to numerical analysis. The initial conditions can be specified, the future states may be obtained from these. The subsequent evolution of the system is found by substituting these states as the current and obtaining the new set of future states, in the typical recursive manor. This procedure is repeated until the discretised time steps n=N.

It can be further shown that for a finite potential well $V(l)$=V_0, the condition for stability requires that (derivation from [3]):

$$\Delta t \leq \frac{\hbar}{\frac{\hbar^2}{m\Delta x^2} + \frac{V_0}{2}} \tag{12}$$

Here is the general structure of the code: The initial conditions are first specified. The wavefunction is also confined to a Gaussian envelope such that the wave propagates as a wave-packet. The initial width of the wave-packet can be defined through the value of its standard deviation, σ. The wavenumber can also be defined along with the 'time shift'. The necessity of the Guassian wave-packet arises due to the following;

Schorödinger's equation can be written as:

$$\hat{H}\psi = i\hbar\frac{\partial \psi}{\partial t}, \quad \hat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} \tag{13}$$

For a free particle, it has solutions:

$$\psi_k = Ae^{i(kx - \omega_k t)}, \quad \psi_{-k} = Be^{-i(kx + \omega_k)} \tag{14}$$

where $\hbar\omega_k = E_k = \frac{\hbar^2 k^2}{2m}$ and ψ_k is an eigenfunction for eigenvalue E. The general solution is found by the sum of the two independent solutions:

$$\psi(x,t) = \int_{-\infty}^{\infty} C(k)\psi_k(x,t)dk \tag{15}$$

It is found that the eigenfunctions ψ_k are not normalisbale, as

$$\int_{-\infty}^{\infty} |\psi_k|^2 \, dx = \int_{-\infty}^{\infty} |A|^2 \, dx = \infty \tag{16}$$

However, a wave-packet can be constructed which is a normalisable function of these eigenfunctions. This is where the Gaussian is paramount:

$$\psi(x,t) = A \int_{-\infty}^{\infty} e^{-a^2(k-k_0)^2} e^{i(kx - w_k t)} dk \tag{17}$$

It can be shown that $\int_{-\infty}^{\infty} |\psi|^2 \, dx$ does exist, upon normalising to unity (1), it can be shown that the energy of the Guassian:

$$E = \frac{\hbar^2}{2m} \frac{k_0^2 + 0.5}{\sigma^2} \tag{18}$$

is used to find the total energy of the wavefunction, after propagation.

The energy of the initial wavefunction can be adjusted by changing any of the variables in the equation above. In this way, scenarios in which the wavefunction has energy above and below the finite potential interactions can be investigated.

Now that the energy of the wavefunction can be found at each given spatial and temporal position, the evolution of the wave can be considered. This is done by separation of the Gaussian into its real and imaginary components. These can then be used in the time evolution of the wave as they are fed into the future state equations derived earlier. The recursive relation is used to find all subsequent states until the simulation is concluded. The aim of this simulation is to study the propagation of the wavefunction by implementing the FDTD algorithm, investigating scenarios which may lead to divergence from classical physics by showing quantum tunneling and in effect analysing the effectiveness of the algorithm to such a problem.

3 Simulation procedure

3.1 Free Particle

The simplest application of the FDTD algorithm described previously is that to the propagation of the wavefunction of a free particle, without the presence of any potentials. Atomic units ($a.u.$) will be used throughout the code. This is where the relationship $m_e = \hbar = 1$ holds - mass of an electron m_e and the reduced Plank's constant $\hbar$ all become 1. As can be imagined, this simplifies the calculations greatly. Since the actual values require a great amount of precision to represent accurately, it is more beneficial to use atomic units and convert to SI when the exact units are required.

The code starts by defining the functions which will be used; the 'Gaussian', having arguments of the position variable x, time shift t and standard deviation σ. The 'free' function returns all positions where there are free particles. The 'fillax' function fills an array of y values for each x position such that the positions of where the wave has propagated is stored as an array. Next the initial conditions are specified, this includes defining the number of spatial points, time steps, spatial resolution, particle mass, the finite potential energy, thickness and separation, utilising the atomic units as explained before. These parameters may be tweaked to investigate the effects to the simulation. The introduction of the spatial points and time steps initialise the grid-based approach needed for the FDTD algorithm.

The parameters for the Gaussian are also specified; the standard deviation σ, time shift $x0$ (determining the time evolution of the wave-packet), the wavenumber K_0 and finally the energy of the wave-packet in accordance to equation 18. The code then implements an 'if' statement to choose which function to implement (the code is open to the introduction of more interactions which will be added later). The value of the critical time step (equation 12) is also stated explicitly by using the relevant variables listed in initial conditions. The values of the coefficients C_1 and C_2 are also calculated explicitly. These are done before the calculation of the future states as they remain constant for any parameter.

The 'Physical observables' module initialises the wavefunction by separating it into its real and imaginary components, along with the introduction of the temporal indexing- to represent the past, present and future states (represented as 0,1,2 respectively). The general solution to the wavefunction (equation 14) can be represented as a cosine (real component) and sine (imaginary component) via Euler's formula: $e^{\pm ikx} = cos(kx) \pm isin(kx)$. This Gaussian wavepacket (defined as fucntion previously) is multiplied by all wavefunctions to constrict them to the form of the Gaussian. The observable probability is also found through the application of equation 16 and is normalised to unity by following the same procedure. The total probability can now be found by the sum of the real and imaginary components at any given time step.

The 'Time evolution' module contains the recursive future state equations which will be quintessential to the algorithm. Before initiating this module, the indexing of the spatial positions is defined using the appropriate python zero-based indexing. Next the future states are explicitly applied, but with the values of ψ dependent on the potential V being calculated separately. This is because in certain interactions (like the free particle), the potential is not needed for calculations. After each future state is found, it is equated to the present state, in the typical recursive manor, such that a 'do-loop' is formed for all ranges of time step, t. Within the same loop, an 'if-statement' is used to compute the total probability of the wavefunction when the time step = 0 (the simulation is concluded). The 'Plotting' module plots the final time step of the wavefunction evolution into a graph which displays explicitly the amplitudes of the real, imaginary components of ψ along with the probabilities of the wavefunction at each spatial position.

The initial conditions must be constricted to certain values in order to produce a successful plot. The number of spatial points is completely arbitrary and can be chosen to be any value. Choosing a greater number of points will constrict the waveform such that it might be difficult to observe its individual components. The time step chosen should be a multiple of the total number of steps in order to produce a plot at the last time step. The total time steps should also be chosen such that the simulation is concluded whilst the wavefucntion is within the boundaries of the spatial axis - as boundary conditions are not introduced in this algorithm. Changing the spatial resolution will change the distance propagated by the wavefucntion at each future state calculation. This can be chosen to be any arbitrary value, but choosing unity allows the entire wave propagation to be seen. Lastly the standard deviation of the Gaussian wave-packet can be changed to effect the initial width of the wave. Choosing a larger value will 'spread' the wave further. Since the wave-packet is normalised to unity, the wave will be seen to have a greater distribution of energies. This will effect how far the wave is able to propagate (due to dispersion) and will also come into effect when potential interactions are considered (less likely for quantum tunneling).

3.2 Potential Interactions

The *free code* is implemented into a general many interaction scenario through the introduction of other functions and variables. The provided code is very flexible as the terms involving potentials are separable such that other simulating conditions can be applied.

To apply the '**step potential**', a new function is added:

```
def step(npts,v0):
"Potential step"
v = free(npts)
v[npts/2:] = v0
return v
```

This function introduces a new argument 'v0' which is a variable defining the energy of a step potential. This is places the potential at the half way point of the spatial axis - defining v0 at *npts/2*. The variable v0 can be set to a given value when setting initial conditions. The 'if-statement' responsible for choosing the function is also altered as to introduce the new potential:

```
elif POTENTIAL=='step':
V = step(N,V0)
```

The rest of the code remains unchanged.

To apply the '**double barrier-potential**', a new function is added:

```
def barrier(npts,v0,thickness,ds):
"Barrier potential"
v = free(npts)
v[npts/2:npts/2+thickness] = v0
v[npts/2+ds:npts/2+thickness+ds] = v0
return v
```

A new argument is once again introduced, the 'thickness' represents distance along which each of the barrier operates and 'ds' is the separation of the two barriers. The 'if-statement' is altered to include:

```
elif POTENTIAL=='barrier':
V = barrier(N,V0,THCK,ds)
```

Where the variables *THCK* and *ds* can be set.
Lastly, the '**potential well**' is implemented:

```
def well(npts,vx,thickness):
"Well potential"
v = free(npts)
v[0:] = vx
v[1150:1150+thickness] = 0
return v
```

This is using the same logic as the step potential, but now placed at a specified position, the distance between the barriers is specified. The 'if-statement' now becomes:

```
elif POTENTIAL=='well':
V = well(N,V0,THCK_w)
```

with the variable *THCK w* being specified as the distance between the barriers.

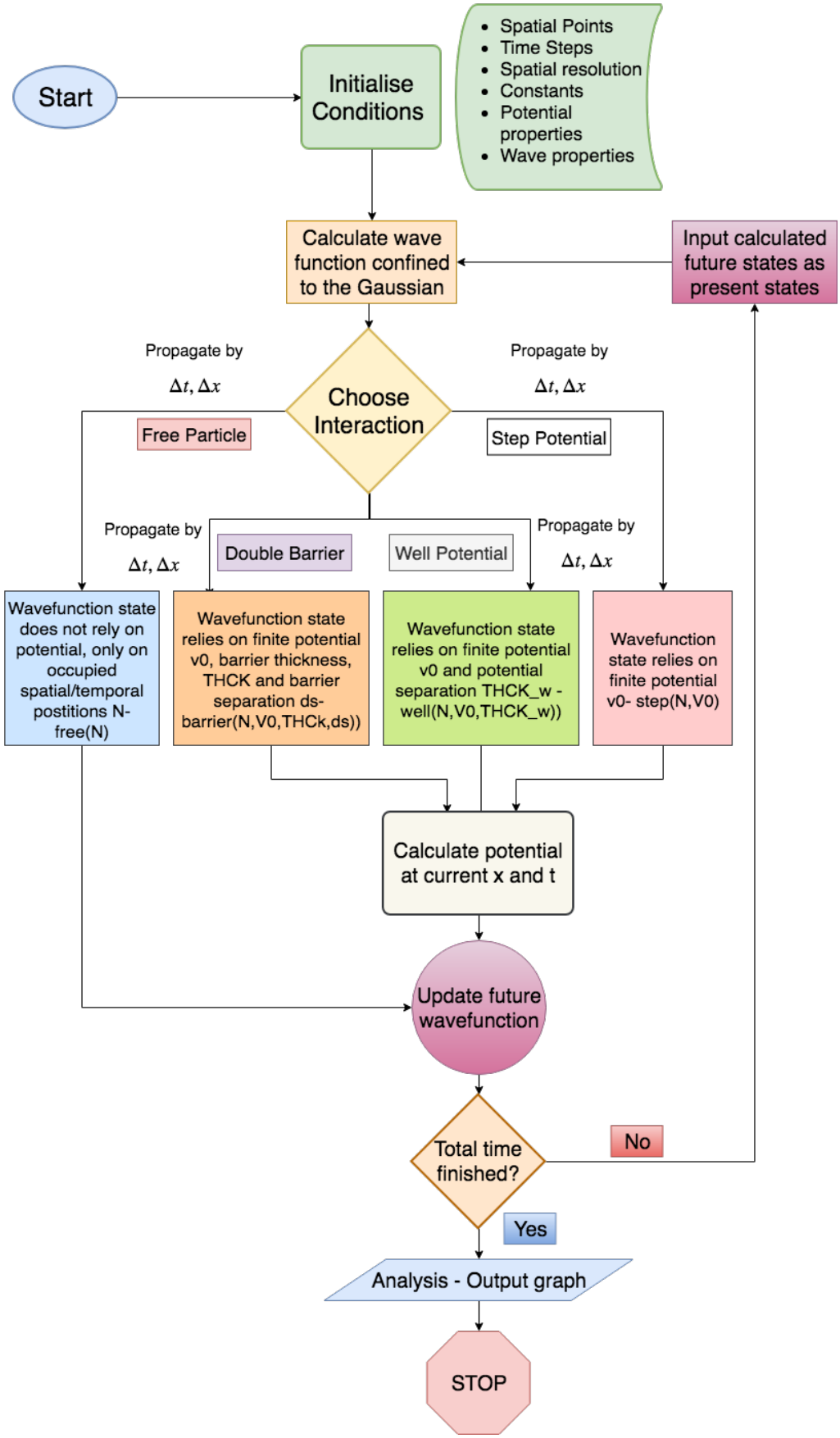
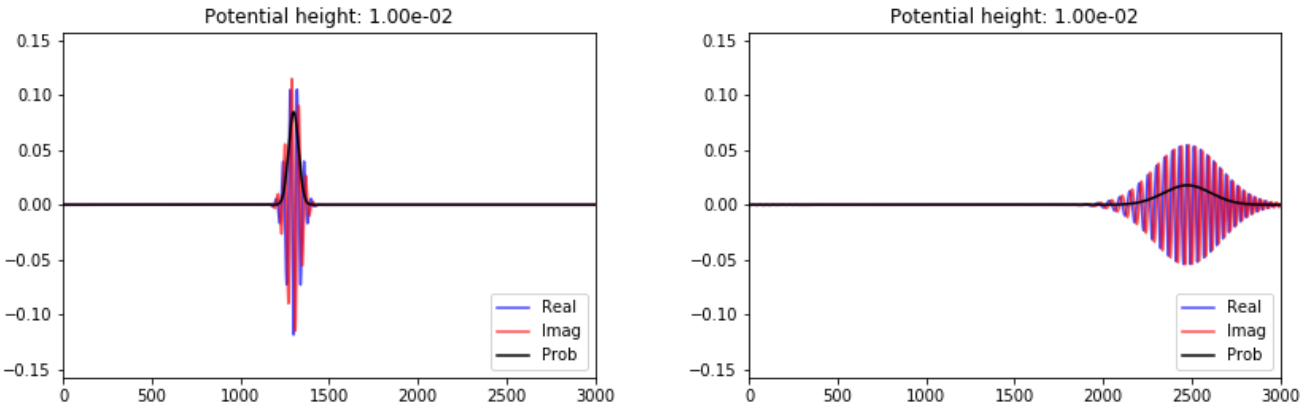


Figure 2: Basic structure of the FDTD algorithm used in the code.

4 Data collection and Analysis

4.1 Free Particle



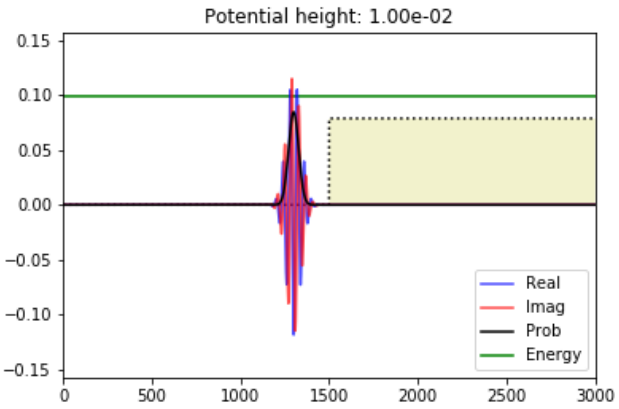
(a) Initial form of the wavefunction at t=0

(b) Final form of the wavefunction at t=T

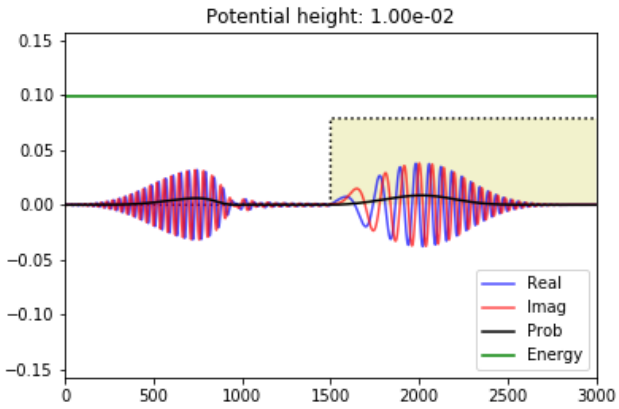
Figure 3: Initial and final wavefunction of the free particle. Mean K.E= 0.0125 Ha ($\sigma = 40.d0$)

The graph above shows the clear time evolution of the free particle wavefunction. The waveform starts as a sharp peak at the set initial position, as it propagates across the spatial and temporal dimensions it is seen to become broader with a decreased amplitude- the imaginary and real components of the wave can be seen more clearly as they become more spread out. This agrees with the dispersive nature of the Schrödinger equation. Since the Gaussian is normalised to unity, the waveform will continue to disperse until it occupies a single frequency and spreads out over space. It is also characteristic of the Heisenberg uncertainty principle which states that $\Delta E \Delta t > \frac{\hbar}{2}$. This is most evidently seen when a potential with energy greater than the particles is introduced.

4.2 Step Potential

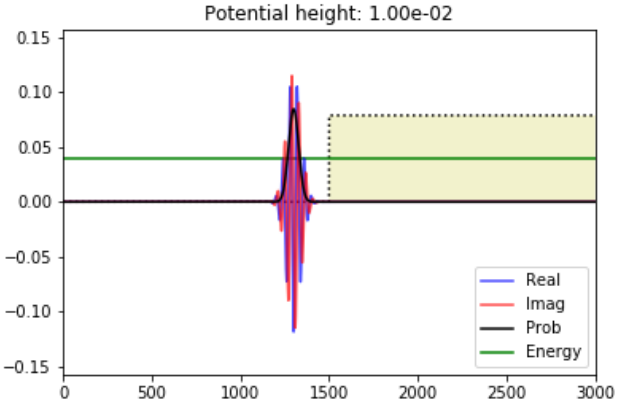


(a) Initial form of the wavefunction at t=0

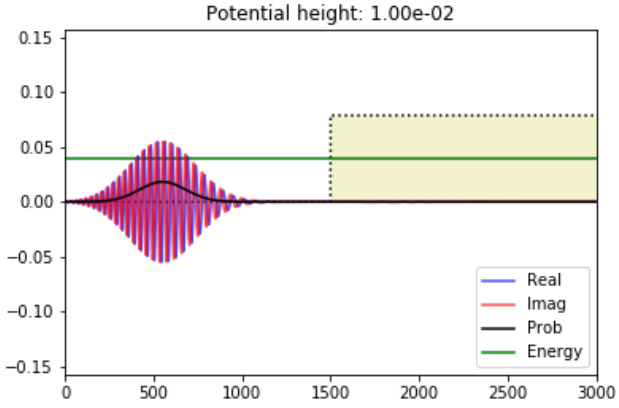


(b) Final form of the wavefunction at t=T

Figure 4: Initial and final wavefunction of the particle when a step potential is introduced. $E > V0$, Mean K.E = 0.0125 ha ($\sigma = 40.d0$)



(a) Initial form of the wavefunction at t=0

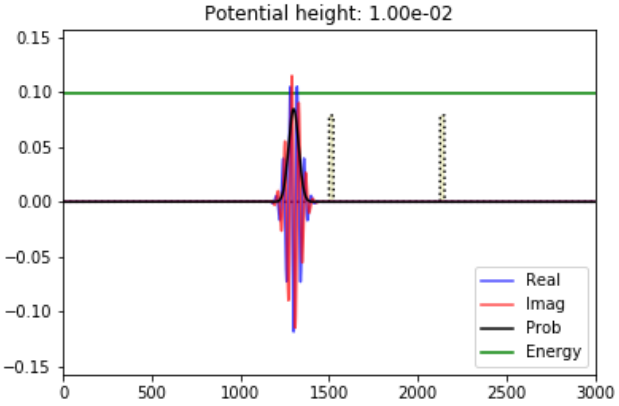


(b) Final form of the wavefunction at t=T

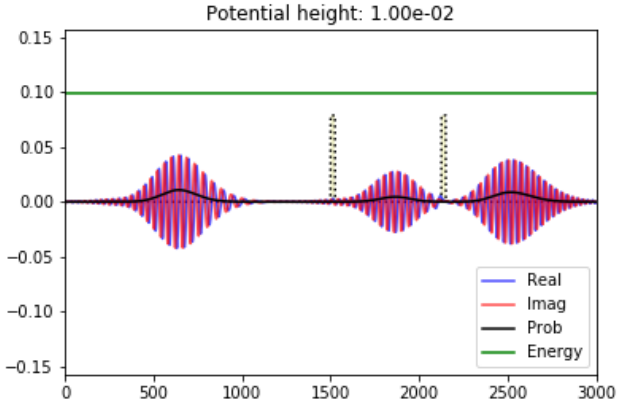
Figure 5: Initial and final wavefunction of the particle when a step potential is introduced. $E < V0$, Mean K.E = 0.005 ha ($\sigma = 40.d0, m = 2.5a.u.$)

When $E > V0$, some form of the wavefunction is able to penetrate into the step potential whilst most is reflected back. For $E < V0$, none of the wavefunction is able to penetrate potential and a complete reflection is seen, the waveform is identical to that seen with the free particle but at a different final position.

4.3 Double barrier potential

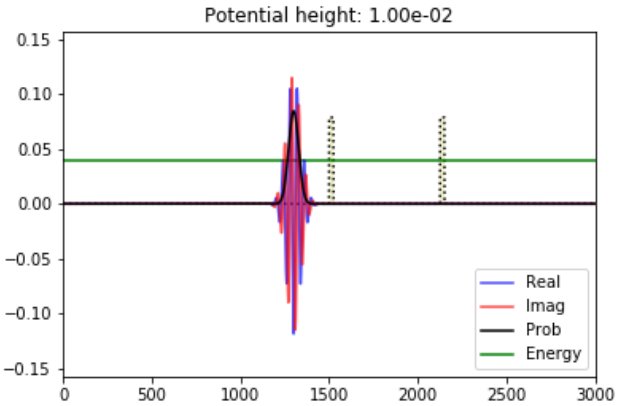


(a) Initial form of the wavefunction at t=0

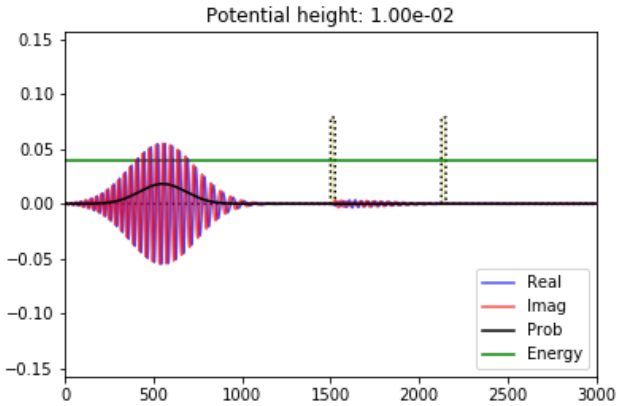


(b) Final form of the wavefunction at t=T

Figure 6: Initial and final wavefunction of the particle when a double barrier potential is introduced. $E > V0$, Mean K.E = 0.0125 ha ($\sigma = 40.d0$)



(a) Initial form of the wavefunction at t=0

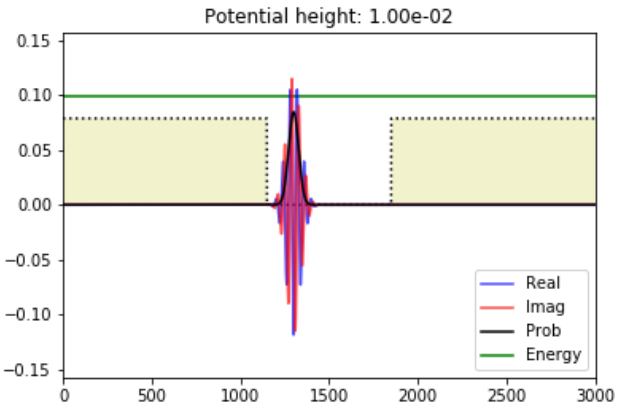


(b) Final form of the wavefunction at t=T

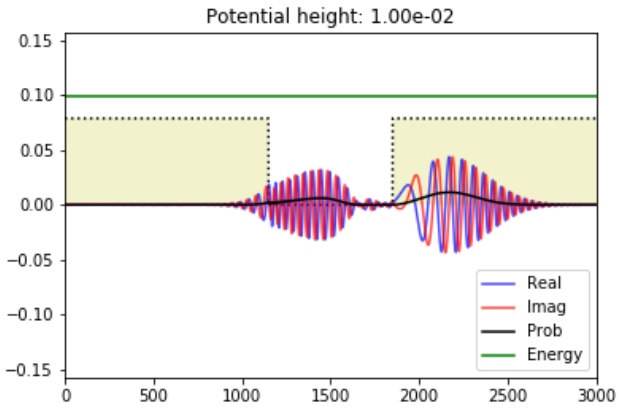
Figure 7: Initial and final wavefunction of the particle when a double barrier potential is introduced. $E < V0$, Mean K.E = 0.005 ha ($\sigma = 40.d0, m = 2.5a.u.$)

For $E > V0$, most of the wavefunction is able to penetrate through the first barrier, whilst less is able to go through the second, this is seen as some waveform is reflected off the second barrier. For $E < V0$: One can see upon observation that despite the particle having substantially lower energy than the first barrier potential, some of the wavefucntion still penetrates through. This is a consequence of Heisenberg's uncertainty principle since the energy of the particle cannot be found exactly for a given time, there exists a possibility that some form of the wavefunction has enough energy to overcome the potential barrier.

4.4 Well potential

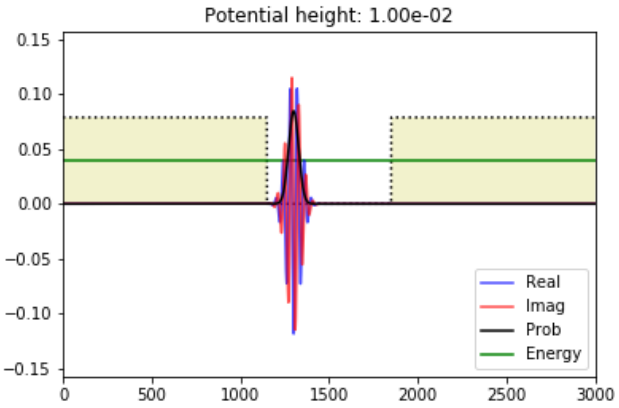


(a) Initial form of the wavefunction at t=0

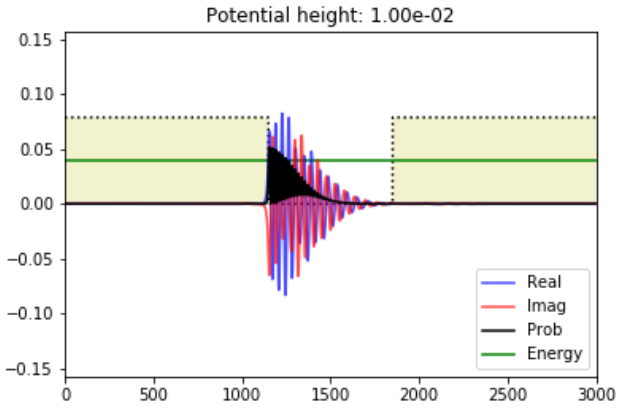


(b) Final form of the wavefunction at t=T

Figure 8: Initial and final wavefunction of the particle when a well potential is introduced. $E > V0$, Mean K.E = 0.0125 ha ($\sigma = 40.d0$)



(a) Initial form of the wavefunction at t=0



(b) Final form of the wavefunction at t=T

Figure 9: Initial and final wavefunction of the particle when a well potential is introduced. $E < V0$, Mean K.E = 0.005 ha ($\sigma = 40.d0, m = 2.5a.u.$)

For $E > 0$, some elements of the wavefunction have enough energy to overcome the well function on the right, however due to the large thickness of the barrier the wavefunction is depleted - seen by the broadening of the probability curve. Some of the reflected wavefunction is also able to penetrate the well potential on the left. When $E < V0$, none of the wave is able to go through the well potential on the right, and is reflected back. Upon reflection, no noticeable transmission is seen through the well potential on the left.

5 Conclusion

We conclude that the FDTD algorithm is a suitable numerical approach to solve the 1D Schrödinger's equation. The stability and error of the Crank-Nicolson method was not investigated and comparisons to results in classical theory were not made. The code produced provided an easy way to investigate different interacting potentials which were simulated correctly and outputted on graphs. The conditions to produce quantum tunnelling were also briefly discussed along with the application of Heisenburg's uncertainty principle. Improvements can be made to discuss comparisons with other Finite difference methods and a more rigid approach to the processes invlolved in quantum tunelling.

References

[1] Griffiths, David J. *Introduction to Quantum Mechanics*, Upper Saddle River, NJ: Pearson Prentice Hall, 2005.

[2] *Finite difference approaches to the Schrödinger equation* , available at www.phys.au.dk

[3] *One dimensional FDTD algorithm applied to Schrödinger equation*, available at www.scipy.cookbook.readthedocs.io