5CCP211C-Introduction to numerical modelling,
Physics Dept., King's College London

Quantum Tunnelling in One Dimension.

Taimur Ahmad Khan 1727874

January 29, 2019

1 Introduction and Theoretical Background.

Today, Quantum Theory is one of the leading fields of research due to not only its theoretical implications but also its real-world applications. An integral part of this research was the **Schrodinger Equation**. The time dependant equation in one dimension (x-direction) is;

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

A complex *eigen* problem that helps understand most of modern physics today. Not only is the equation useful to calculate and predict energy and potential values, but its solution, the $wavefunction$ contains all the information about a system; $\psi(x,t)$. A single valued function for any x,t (position and time in space) that can be used to model interactions in a set system or environment. Note that $\hbar$ is the reduced Planck Constant; $h/2\pi$ and j represents the imaginary term in the equation. After this, quantum theory research has grown with the discovery of **quantum tunneling**; a theory which defies classical physics, led to experiments and applications due to its unique basis where a particle could 'tunnel' or pass through potential barriers with a finite probability. This penetration of the barrier can theoretically occur even if its potential is greater than the Kinetic Energy of the particle. The wave function or solution to Equation 1 is used to simulate particles' interactions with chosen types of potential obstacles.

To study the Schrodinger equation's properties and implementations in experimentation and technological development, The Finite-Difference Time-Domain algorithm was developed. The $FDTD$ model is a popular method to numerically solve a partial differential equation. The basic principle behind it is to discretise the partial differential equation (PDE) in space and time and allow it to play itself out with discrete steps. A more detailed illustration of the process is given in the next section.

2 Simulation Procedure and FDTD Method.

The method involves discretising the partial derivative in time and space and approximate the derivatives with finite differences. Firstly, the Schrodinger Equation or Equation 1 is changed to fit the model. This is done by separating it into its *real* and *imaginary* components;

$$\psi(x,t) = \psi_R(x,t) + j\psi_I(x,t) \tag{2}$$

The subscripts R and I indicate real and imaginary terms with j representative of a complex number. By doing this, each component can be treated uniquely and computed. Substituting Equation 2 into Equation 1, produces two partial differential equations.

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

and

$$\hbar\frac{\partial\psi_I(x,t)}{\partial t} = +\frac{\hbar^2}{2m}\frac{\partial^2\psi_R(x,t)}{\partial x^2} - V(x)\psi_R(x,t) \tag{4}$$

Next step is lay down a basis onto which these PDEs can be discretised. For both position and time, quantised points are required to approximate the derivatives according to discrete changes in x and t;

$$x_l = l\triangle x \tag{5}$$

and

$$t_n = n\triangle t \tag{6}$$

Where $\triangle x$ and $\triangle t$ are the separation between the spatial and temporal points respectively and the conditions $0 \leq l \leq L$ and $0 \leq n \leq N$ are satisfied. To simplify the formulae and calculations below, we can use the notation $\psi(x_l, t_n) = \psi^n(l)$. The derivatives are approximated with the Central-Difference Method. Three equations are used;

$$\delta(f_n) = f_{n+1/2} - f_{n-1/2}, \tag{7}$$

$$\delta(f_{n+1/2}) = f_{n+1} - f_n, \tag{8}$$

$$\delta^2(f_n) = f_{n+1} - 2f_n + f_{n-1}, \tag{9}$$

Using these equations, one can approximate the time and space derivatives with respect to the real and imaginary parts. Note that if a derivative with respect to t is approximated, x will not change, only t will. This is because we are dealing with partial derivatives with respect to both x and t separately. During this the other variable will remain constant. After some manipulation, the derivatives are approximated as shown using Equations 7-9 and the simplification notation. The time derivative of the real component,

$$\frac{\partial\psi_R(x_l,t_{n+1/2})}{\partial t} \approx \frac{\psi_R^{n+1}(l) - \psi_R^n(l)}{\triangle t} \tag{10}$$

and the imaginary component,

$$\frac{\partial\psi_I(x_l,t_n)}{\partial t} \approx \frac{\psi_I^{n+1/2}(l) - \psi_I^{n-1/2}(l)}{\triangle t} \tag{11}$$

The second derivative approximations with respect to position are found using Equation 9. Again for the real component,

$$\frac{\partial^2 \psi_R(x_l, t_n)}{\partial x^2} \approx \frac{\psi_I^n(l+1) - 2\psi_R^n(l) + \psi_R^n(l-1)}{\triangle x^2} \tag{12}$$

and the imaginary part,

$$\frac{\partial^2 \psi_I(x_l, t_{n+1/2})}{\partial x^2} \approx \frac{\psi_I^{n+1/2}(l+1) - 2\psi_I^{n+1/2}(l) + \psi_R^{n+1/2}(l-1)}{\triangle x^2} \tag{13}$$

Substituting Equations 10-13 in Equations 3 and 4 gives;

$$\hbar[\frac{\psi_R^{n+1}(l) - \psi_R^n(l)}{\triangle t}] = -\frac{\hbar^2}{2m}[\frac{\psi_I^{n+1/2}(l+1) - 2\psi_I^{n+1/2}(l) + \psi_I^{n+1/2}(l-1)}{\triangle x^2}] + V(l)\psi_I^{n+1/2}(l) \tag{14}$$

and

$$\hbar[\frac{\psi_I^{n+1/2}(l) - \psi_I^{n-1/2}(l)}{\triangle t}] = +\frac{\hbar^2}{2m}[\frac{\psi_R^n(l+1) - 2\psi_R^n(l) + \psi_R^n(l-1)}{\triangle x^2}] - V(l)\psi_R^n(l) \tag{15}$$

The FDTD uses present state values to predict future state ones. The future state for the real function is denoted by $\psi_R^{n+1}(l)$ and its present state used to theorise this value is $\psi_R^n(l)$. Similarly the imaginary component has also a predicted future state and present state; $\psi_I^{n+1/2}(l)$ and $\psi_I^{n-1/2}(l)$ respectively. The imaginary future state of the wave-function is calculated from its present state and the present real state. This future value in turn is involved in the computation of the future real value. This complex relationship is easily understood from the following resulting equations. Note that as the potential does not vary with x and is constant, V(l) = V_0. Therefore the equations for predicted future states is;

$$\psi_I^{n+1/2}(l) = +c_1[\psi_R^n(l+1) - 2\psi_R^n(l) + \psi_R^n(l-1)] - c_2 V_0 \psi_R^n(l) + \psi_I^{n-1/2}(l) \tag{16}$$

and

$$\psi_R^{n+1}(l) = -c_1[\psi_I^{n+1/2}(l+1) - 2\psi_I^{n+1/2}(l) + \psi_I^{n+1/2}(l-1)] + c_2 V_0 \psi_I^{n+1/2}(l) + \psi_R^n(l) \tag{17}$$

Representing the equations for the real and imaginary part respectively. The constants c_1 and c_2 are given by;

$$c_1 = \frac{\hbar \triangle t}{2m \triangle x^2}, \tag{18}$$

$$c_2 = \frac{\triangle t}{\hbar} \tag{19}$$

Hence the predicted equations have been derived. The FDTD model is an accurate approximation and helps solve Eigen problems and PDEs such as this to an extent where a simulation can be designed around it.

3 Description of the code.

Now we can implement this model into a code to run a simulation of Quantum Tunnelling. As this is only in one dimension, all motion is in the x-direction (to the right initially). Energy and/or potential level is given on the Y axis. A 'particle' or a wave-packet defined by the wave-function moves and interacts with various potential obstacles. Using the FDTD method to solve the PDE in space and time at discrete steps while gradually increasing the time variable. The code involves the initialisation of multiple variables and values used in calculations which result in a graphical illustration of the wave-packet.

3.1 Initialisation and Functions of the Potentials.

Important graphical and mathematical modules are imported. The simulation and plots are constructed using this built-in python module; pylab.ion(). A class for the Gaussian Distribution is defined, within it three objects; position, time shift and standard deviation. The function used is derived from;

$$e^{\frac{-(x-t)^2}{2\sigma^2}} \tag{20}$$

Here σ is the standard deviation, x is the variable and t is the time shift. This acts as an envelope for the E_k and the motion of free particle in 0 potential. The functions of the different potential environments (obstacles) are set. Firstly, the most simple function, for a free particle in free space with no potential barriers, wells or blocks. Packet propogates freely with 0 potential.

```
def free(npts):
    return np.zeros(npts)
```

The above line of code, returns a list of zeroes along the axis of wave packet motion to show zero potential and no potential obstacles. The other potentials are slightly more complex. The *step* function comprises of one potential barrier and sets the localised grid points where the potential V_0 will be placed. In the code below it is shown that the potential is spread over 600 spatial points.

```
def step(npts,v0):
    v = free(npts)
    v[600:] =v0
    return v
```

Both *doublebarrier* and *well* potentials have two potential localised widths. They are defined within a set range and apart from each other so that they interact individually with the wave function. The double barrier potential includes two barriers of 25 grid points width while the well potential coded in a similar fashion and in simplicity is just a region between two barriers. The last potential function to be defined is the *doublewell* potential.

```
def doublewell(npts,v0,thickness):
    v=free(npts)
    v[0:thickness] =v0
    v[600:600+thickness] =v0
    v[1225:1225+thickness] =v0
    return v
```

As you can see from the code above, three barriers are set to mimic two apparent wells. With Well 1 between Barrier 1 and 2, and Well 2 between Barrier 2 and 3. This arrangement

with set potentials and spread over grid points, simulates to an extent the interaction of a wave packet in these potential environments.

3.2 Simulation Parameters and Equations.

The variables used in the equations are set. Reduced physical constants are used instead of actual values. The Energy is in terms of atomic units or Hartrees. Position is measured in multiples of $10^{-10}m$ or unit Angstrom. Energy of particle is fixed at 0.0125 Hartrees while potential V_0 is varied from 0.0100 to 0.0150 to have different cases where Energy is less than and greater than the potential barrier or V_0. Note that the Energy here is the Kinetic Energy of the wave packet and is derived from the formula;

$$E_k = \frac{\hbar^2 k^2}{2m} \tag{21}$$

into;

$$E_k = \frac{\hbar^2}{2m}[k^2 + \frac{1}{2\sigma^2}] \tag{22}$$

This shows that the Kinetic Energy of the wave packet is in terms of the spreading of the Gaussian envelope or curve. Some of the reduced values used in the calculations are m=1, $\hbar$=1 and σ=40. Barrier thickness is kept at 25 grid points. The simulation domain has N=1250 grid points and runs for T time steps. T is calculated using the formula; T=5N where N is the simulation run-time domain.

```
X = dx*np.linspace(0,N,N)
x0 = round(N/2) - 5*sigma
k0 = 2.*np.pi/sigma
E = (hbar**2/2.0/m)*(k0**2+0.5/sigma**2)
```

A conditional statement so that a certain type of potential environment can be chosen. The following line of code depicts the use of if and $elif$ statements to produce a singular type simulation.

```
if POTENTIAL=='free':
        V = free(N)
elif POTENTIAL=='step':
     V = step(N,V0)
elif POTENTIAL=='doublebarrier':
     V= doublebarrier(N,V0,THCK)
elif POTENTIAL=='well':
     V = well(N,V0,THCK)
elif POTENTIAL == 'doublewell':
     V = doublewell(N,V0,THCK)
```

3.3 Algorithm Implementation.

Finally we arrive at the point where the FDTD algorithm is implemented. For a particle in free space with no potential, the solution to the Schrodinger Equation is;

$$\psi(x,t) = A_1 e^{j(kx-\omega t)} + A_2 e^{j(kx+\omega t)} \tag{23}$$

The real and imaginary parts after using the simplification notation are transformed to;

$$\psi_R^n(l) = cos(kl\triangle x - wn\triangle t) \tag{24}$$

and

$$\psi_I^n(l) = sin(kl\triangle x - wn\triangle t) \tag{25}$$

After substituting Equations 24 and 25 back into Equations 16 and 17, a value for critical $\triangle t$ is found. The critical value signifies the maximum allowed time increment to maintain stable simulation and no unbounded variables. Note that $\triangle t$ is dt in the code.

$$\triangle t \leq \frac{\hbar}{\frac{\hbar^2}{m\triangle x^2} + \frac{V_0}{2}} \tag{26}$$

In the code, the value of dt is slightly different from this equation but as a factor of two has been introduced as a multiple in the equations for c_1 and c_2 or Equations 18 and 19, the end result is conserved. Another important aspect of the wave function is looked on now. The square of the modulus of the wave function is the probability of finding a particle at that position and time. This is given by;

$$P = |\psi(x,t)|^2 \tag{27}$$

Real and Imaginary values of the wave functions are initialised with the proper variables including the observable probability.

```
psi_r = np.zeros((3,N))
psi_i = np.zeros((3,N))
psi_p = np.zeros(N,)
```

Indices for past, present and future are set as 0,1,and 2 and PA,PR, and FU respectively. The wave functions for both components, are enveloped in the Gaussian curve for normalised position coordinates. The past and present state value of the wave function for both imaginary and real parts are connected with the relationship;

```
psi_r[PR,xn] = cx*gg
psi_i[PR,xn] = sx*gg
psi_r[PA,xn] = cx*gg
psi_i[PA,xn] = sx*gg
```

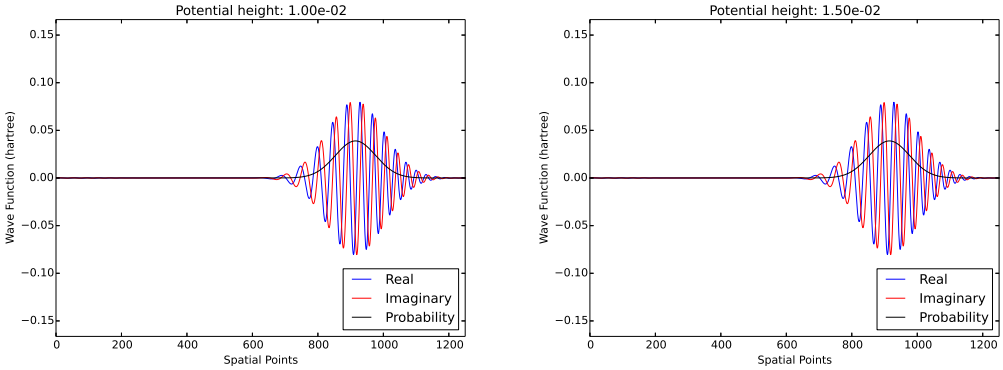
Here the variables are given by the following line of code;

```
gg = Gaussian(x,x0,sigma)
cx = np.cos(k0*x)
sx = np.sin(k0*x)
```

The update equations are applied to the functions of past and present states. Predicted future states are calculated using the equations 16 and 17 and are written into the code.

```
psi_i[FU,IDX1] = psi_i[PA,IDX1] + \c1*(psi_rPR[IDX2] - 2*psi_rPR[IDX1] +
                                       psi_rPR[IDX3])
psi_i[FU] -= c2V*psi_r[PR]
psi_r[FU,IDX1] = psi_r[PA,IDX1] - \c1*(psi_iPR[IDX2] - 2*psi_iPR[IDX1] +
                                       psi_iPR[IDX3])
psi_r[FU] += c2V*psi_i[PR]
```

Finally the present values are set to the past, the future values set to present and a new future value is found using the same method as before. This process is repeated until the time step reaches critical value and can no longer hold a stable simulation. The probably density is plotted with a factor so that it is more noticeable.



(a) $E > V_0$ (b) $E < V_0$

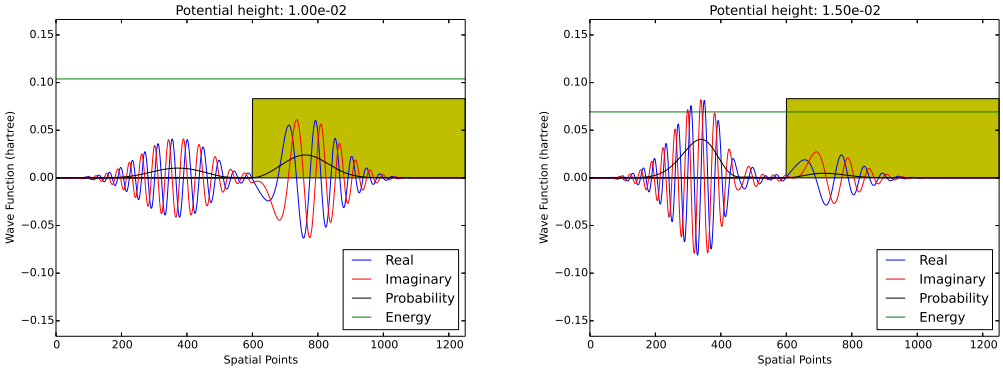
Figure 1: Free Particle In Space

4 Data Collection and Analysis.

When a simulation is chosen, certain values are outputted. Energy of the wave packets, type of potential, potential magnitude, barrier and well thickness. The only input after the variable parameters is the type of potential environment. You can see plots of these different types of potentials. Notice that the five types; Free, Step, Double Barrier, Well and Double Well all have two plots each. The value of the potential height or v_0 is kept at 0.0100 Hartree while the Kinetic energy of the wave packet is more as its equal to 0.0125 Hartree. For when the Energy is less than the Potential, E stays equal to 0.0125 Hartree but potential is increased to 0.0150 Hartree. Different effects are noticed in relation to the difference between E_k and V_0. For the particle in free space, it follows a gaussian curve or envelope in its motion. In the presence of a potential, the wave packet when collides, reflects back and penetrates through at finite probabilities. The energy of the particle remains the same, but the probabbility of finding it there decreases. Quantum Tunneling can be seen very clearly from these simulations as in both cases of Energy ¿ Potential and Energy ¡ Potential, there is a finite probability that a particle will pass through the potential obstacle and appear on the other side of it.

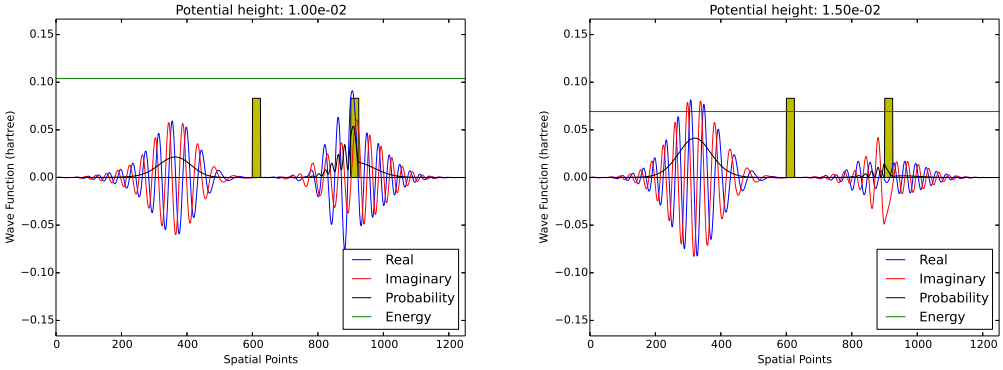
5 Bibliography

- Whittaker, E. T. and Robinson, G. "Central-Difference Formulae.
- James R. Nagel "The One Dimensional Finite-Difference Time-Domain (FDTD) Algorithm Applied to the Schrodinger Equation



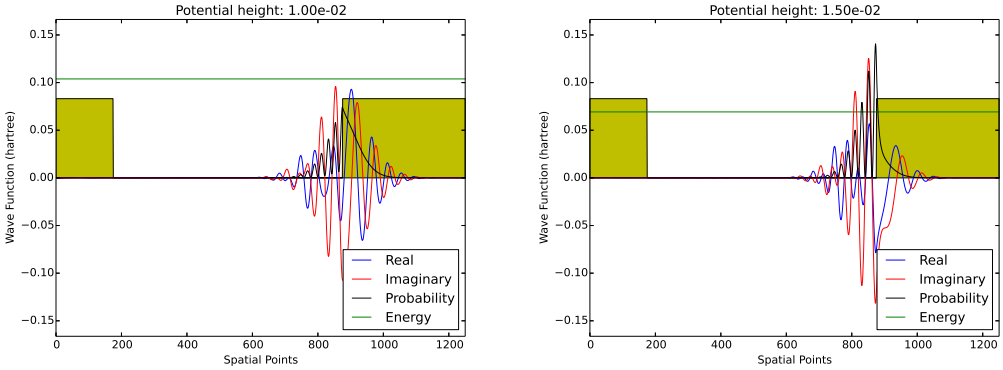
(a) $E > V_0$ (b) $E < V_0$

Figure 2: Step Potential (singular potential wall)



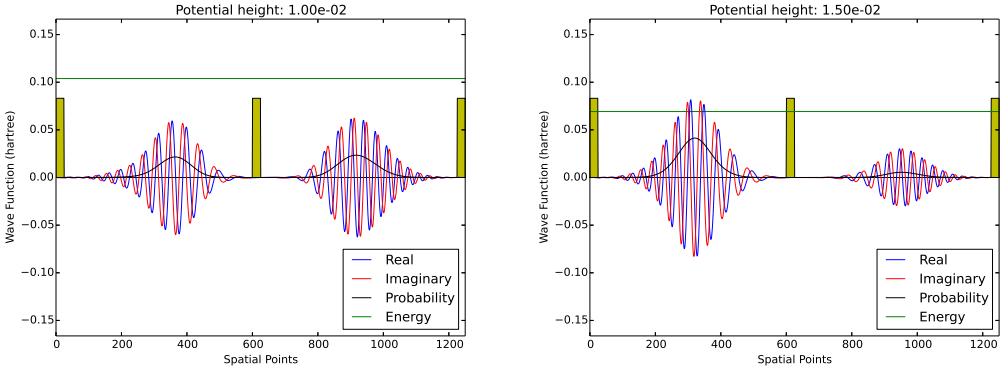
(a) $E > V_0$ (b) $E < V_0$

Figure 3: Double Potential Barrier



(a) $E > V_0$ (b) $E < V_0$

Figure 4: Singular Potential Well



(a) $E > V_0$ (b) $E < V_0$

Figure 5: Double Potential Well