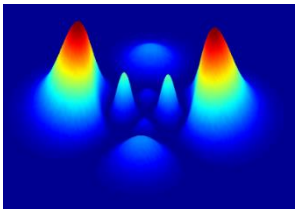

DOING PHYSICS WITH PYTHON

SOLVING THE [2D] SCHRODINGER EQUATION FOR PROPAGATING MATTER WAVES

Ian Cooper

matlabvisualphysics@gmail.com

DOWNLOAD DIRECTORY FOR PYTHON SCRIPTS

qm2DA.py Propagation of a free [2D] Gaussian pulse
qm2DB.py Potential barrier: wall U0 > 0 or cliff U0 < 0
qm2DC.py Potential barrier: single slit
qm2DD.py Potential barrier: double slit
qm2DE.py Potential barrier: Coulomb repulsion

GitHub

Google Drive

TIME-DEPENDENT QUANTUM-MECHANICAL SCATTERING IN TWO DIMENSIONS

THE FINITE DIFFERENCE TIME DEVELOPMENT METHOD

A finite difference time development method (FDTD) is used to solve the two-dimensional time dependent Schrodinger Equation. This method is applied to the free propagation of a Gaussian pulse and the scattering of the pulse from different potential energy functions: wall, cliff, single slit, double slit and Coulomb (Rutherford scattering). The results of the computations are presented as colour graphs portraying the probability density function. The amplitude of the probability for the scattered wave is often very small, so the probability density is scaled to better display the scattering. Arbitrary units are used for all quantities. Values of the wavefunction and potential energy are calculated on a [2D] mesh for a square of length 1. The centre of the square has Cartesian coordinates (0.5, 0.5). The initial wavefunction is a propagating [2D] Gaussian pulse. The wavefunction is zero at the boundaries of the square. So, when the wave packet strikes the boundaries, reflections occur. This results in interference patterns developing because of the superposition of the incident and reflected waves.

THE 2-DIMENSIONAL SCHRODINGER EQUATION

The Schrodinger equation in [2D] and Cartesian coordinates (x, y) can be expressed as

$$(1) \qquad i\hbar\frac{\partial\psi(x,y,t)}{\partial t}=\left[\frac{-\hbar^2}{2m}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)+U(x,y)\right]\psi(x,y,t)$$

The wavefunction $\psi(x,y,t)$ is a complex function

$$(2) \qquad \psi(x,y,t)=\psi_R(x,y,t)+\psi_I(x,y,t)$$

where the real part of the wavefunction is $\psi_R(x,y,t)$ and its imaginary part is $\psi_I(x,y,t)$.

Using equations 1 and 2 and equating the real and imaginary parts, the Schrodinger equation becomes

$$(3A) \qquad \frac{\partial\psi_I}{\partial t}=\left[\frac{\hbar}{2m}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)-\left(U/\hbar\right)\right]\psi_R$$

$$(3B) \qquad \frac{\partial\psi_R}{\partial t}=\left[-\frac{\hbar}{2m}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)+\left(U/\hbar\right)\right]\psi_I$$

We can use the FDTD method to solve the [2D] Schrodinger equation using equation 3.

Time and position are defined at a set of discrete points

(4A)

$$t(n_t) = n_t\,\Delta t \qquad n_t = 1,\ 2,\ 3,\ ...,\ N_t \qquad \Delta t = t(n_t + 1) - t(n_t)$$

(4B)

$$x(n_x) = n_x\,\Delta x \qquad n_x = 1,\ 2,\ 3,\ ...,\ N_x \qquad \Delta x = x(n_x + 1) - t(n_x)$$

(4C)

$$x(n_y) = n_y\,\Delta y \qquad n_y = 1,\ 2,\ 3,\ ...,\ N_y \qquad \Delta y = t(n_y + 1) - t(n_y)$$

The finite difference approximation assumes that the derivates of a function can be expanded in a Taylor series around every point of the mesh up to a desired order of accuracy.

The first derivative can be approximated as

(5A) $$\frac{\partial f(x_n)}{\partial t} = \frac{f(x_n) - f(x_{n-1})}{\Delta t}$$ forward difference

(5B) $$\frac{\partial f(x_n)}{\partial t} = \frac{f(x_{n+1}) - f(x_{n-1})}{2\,\Delta t}$$ central difference

The second derivative is approximated as

(5C) $$\frac{\partial^2 f(x_n)}{\partial x^2} = \frac{f(x_{n+1}) - 2\,f(x_n) - f(x_n - 1)}{\Delta x^2}$$

The finite difference approximation from equations 1 to 5 at time step n_t for the mesh point (n_x, n_y) becomes

(6A) time steps $t + n\Delta t$

$$\psi_R(n_x, n_y, n_t) = \psi_R(n_x, n_y, n_t - 1)$$
$$-\left(\frac{\Delta t\hbar}{2m\Delta y^2}\right)\left(\psi_I(n_x+1, n_y, n_t) - 2\psi_I(n_x, n_y, n_t) + \psi_I(n_x-1, n_y, n_t)\right)$$
$$-\left(\frac{\Delta t\hbar}{2m\Delta y^2}\right)\left(\psi_I(n_x, n_y+1, n_t) - 2\psi_I(n_x, n_y, n_t) + \psi_I(n_x, n_y-1, n_t)\right)$$
$$+\left(\frac{\Delta t U(n_x, n_y)}{\hbar}\right)\psi_I(n_x, n_y, n_t)$$

(6B) time steps $t + n\Delta t / 2$

$$\psi_I(n_x, n_y, n_t) = \psi_I(n_x, n_y, n_t - 1)$$
$$+\left(\frac{\Delta t\hbar}{2m\Delta x^2}\right)\left(\psi_R(n_x+1, n_y, n_t) - 2\psi_R(n_x, n_y, n_t) + \psi_R(n_x-1, n_y, n_t)\right)$$
$$+\left(\frac{\Delta t\hbar}{2m\Delta y^2}\right)\left(\psi_R(n_x, n_y+1, n_t) - 2\psi_R(n_x, n_y, n_t) + \psi_R(n_x, n_y-1, n_t)\right)$$
$$-\left(\frac{\Delta t V(n_x, n_y)}{\hbar}\right)\psi_R(n_x, n_y, n_t)$$

We can let

$$f = \frac{\Delta t\hbar}{2m\Delta x^2} = \frac{\Delta t\hbar}{2m\Delta y^2} < 0.2 \quad \hbar = 1 \;\; m = 1 \;\; \Delta x = \Delta y$$

$f < 0.2$ for stability of the numerical procedure. So, the time step must be $\Delta t = 2\Delta x^2 \, f$.

PYTHON CODE qm2DA.py

The Script **qm2DA.py** uses arbitrary units $(m=1\ \hbar=1)$ for all quantities. The initial pulse is a plane wave with a [2D] Gaussian envelope. The number 1 attached to a variable represents the current time and 2 represents the value of the variable at the next time step.

Set up for xy-grid

```
#%% SETUP XY GRID
# Number of time Steps [400]
nT = 600
# Mesh points X and y  [180] and XY grid
N = 208
G = linspace(0,1,N)
xG, yG = np.meshgrid(G,G)
dx = G[2] - G[1]
# Constant in S.E. and time step
f = 0.2          # f < 0.2
dt = 2*dx**2*f
```

Setting up the initial wavefunction

```
# [2D] GAUSSIAN PULSE (WAVE PACKET)
# Initial centre of pulse
x0 = 0.20;  y0 = 0.5
# Wavenumber
```

```
k0 = 100
# Initial amplitude of pulse
A = 10
# Pulse width: sigma squared
s = 10e-3

# Envelope
psiE = A*exp(-(xG-x0)**2/s)*exp(-(yG-y0)**2/s)
# Plane wave propagation in +X direction
psiP = exp(1j*k0*xG)
# Wavefunction
psi1 = psiE*psiP
# Probability Density
prob1 = np.conj(psi1)*psi1
# Extract Real and Imaginary parts
R1 = np.real(psi1);  I1 = np.imag(psi1)
```

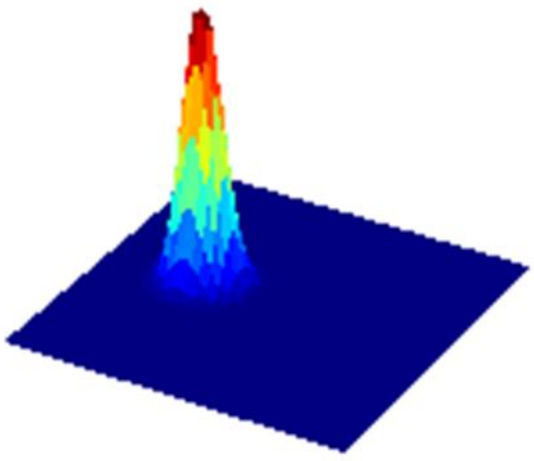

Fig 1. Initial pulse: grid 1x1, pulse centre (0.2, 0.5). **qm2DA.py**

The wavefunction is called at each successive time step and half-time step by calling the functions fI and fR

FDTD method for solving equation 6

```
#%% FUNCTIONS
def fI(I1,R1):
  I2 = zeros([N,N])
  for x in range(2,N-1,1):
    for y in range(2,N-1,1):
      I2[x,y] = I1[x,y] + f*(R1[x+1,y] - 2*R1[x,y] + R1[x-1,y]
                    +    R1[x,y+1]  -2*R1[x,y] + R1[x,y-1]) - dt*U[x,y]*R1[x,y]
  return I2

def fR(I1,R1):
  R2 = zeros([N,N])
  for x in range(2,N-1,1):
    for y in range(2,N-1,1):
      R2[x,y] = R1[x,y] - f*(I1[x+1,y] - 2*I1[x,y] + I1[x-1,y]
                    +    I1[x,y+1] - 2*I1[x,y] + I1[x,y-1]) + dt*U[x,y]*I1[x,y]
  return R2
```

Computing the probability function after nT time steps. The Code is run for an increasing number of time steps to show the propagation of the matter wave in the x-direction.

```
for c in range(1,nT,1):
# Update real part of wavefunction
  R2 = fR(I1,R1); R1 = R2
# Update imaginary part of wavefunction
  I2 = fI(I1,R1)
# Probability Density Function
  probD = R2**2 + I1**2
  I1 = I2
```

Simulation 1: Free propagation of the pulse

We can model the free propagation of the pulse moving in the X direction. The potential energy function is set to zero at all mesh points.

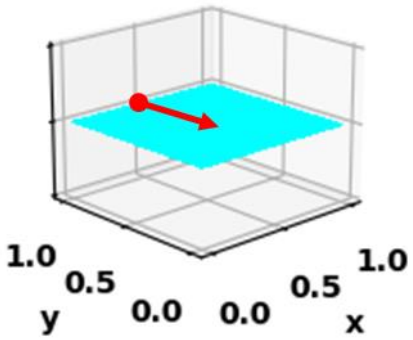

Fig. 2. Zero potential energy field. Red arrow shows direction of propagation of the [2D] pulse.

Figure 3 shows a time sequence view for the motion of the matter wave at time steps 200, 400, and 600 when k_0 = 100 and for k_0 =50 at time step 600. As the matter wave propagates, it spreads in all directions. The spreading is due to the different wavelength components of the wave packet travelling at different speeds. Components with larger wave numbers (smaller wavelengths) corresponds to waves of larger momentum, energy and velocity. Thus, the speed of propagation v is proportional to the propagation constant k.

(7) $$p = \hbar k = h/\lambda \quad E = p^2/2m = \hbar^2 k^2/2m \quad v = \hbar k/m$$

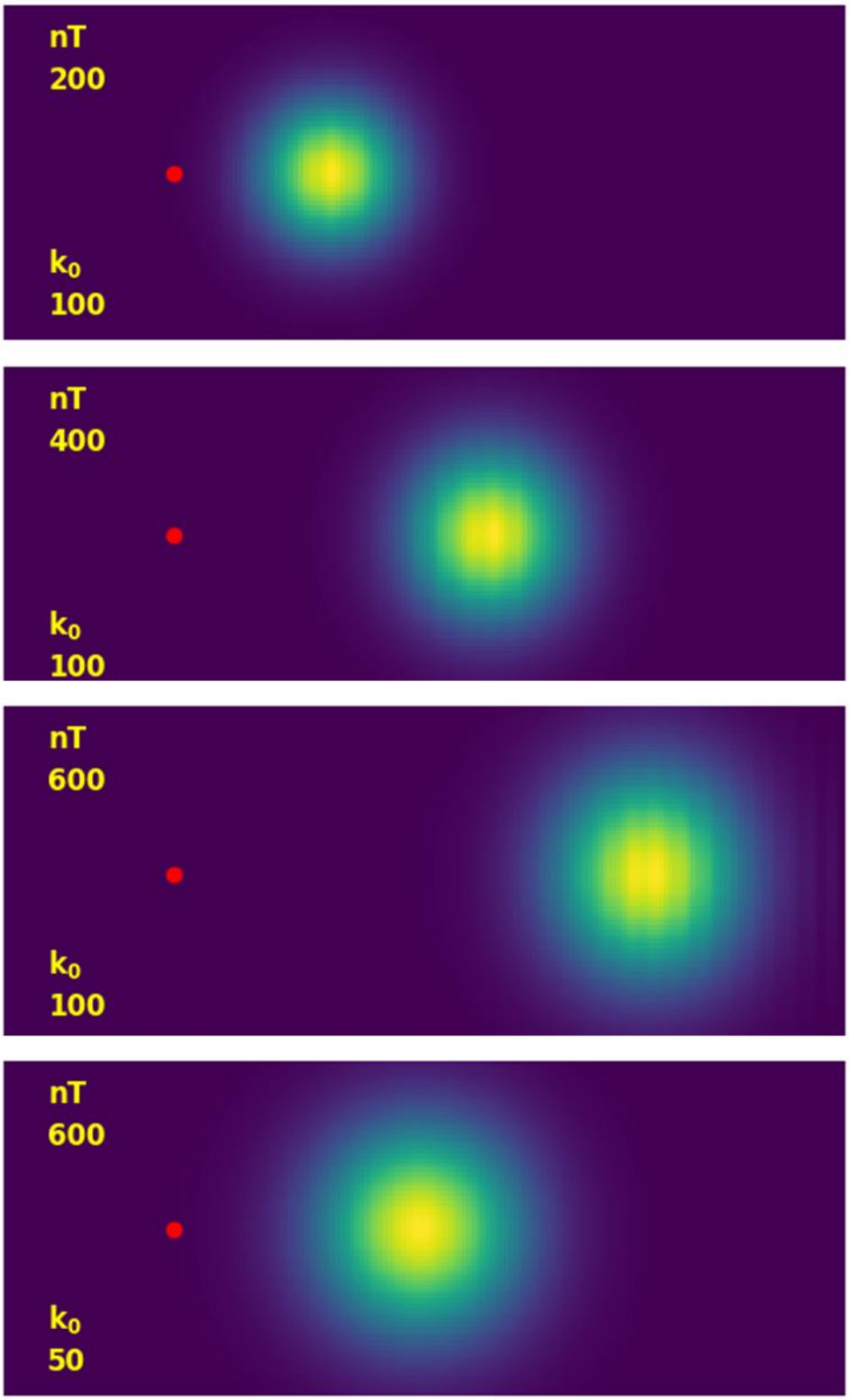

Fig. 3. Pulse propagation in the X-direction. Red dot shows the initial position of the pulse.

For k_0 = 100 and 600 time steps, the displacement of the pulse is 0.577 a.u. and the pulse velocity is 9.61 a.u. For k_0 = 50 and 600 time steps, the displacement of the pulse is 0.297 a.u. and the pulse velocity is 4.95 a.u. The ratio of the propagation constants is 100/50 = 2 and the ratio of the velocities is 9.61/4.95 = 1.94 ~ 2 as predicted by equation 7.

The spreading of a pulse is a subtle example of Heisenberg Uncertainty Principal and the superposition of states. Since the extent of the wavefunction is restricted, then the uncertainty in momentum must be greater than zero. Therefore, the wavefunction must contain components other than k_0 and these can combine in such a way that at time t = 0 for the wavefunction to travel only in the +X direction. However, at later times, the components will add so that the wavefunction propagates in the Y direction as well, and thus the wavefunction spreads in all directions as time evolves (figure 4).

Fourier analysis

Consider a function of x, $h(x)$. Then its Fourier transform $H(k)$ is

$$H(k) = \int_{-\infty}^{\infty} h(x) \exp\left(-i\,k\,x\right) dx \qquad (8)$$

It is a simple matter to evaluate the Fourier transform by direct integration using Simpson's rule rather than by the FFT method.

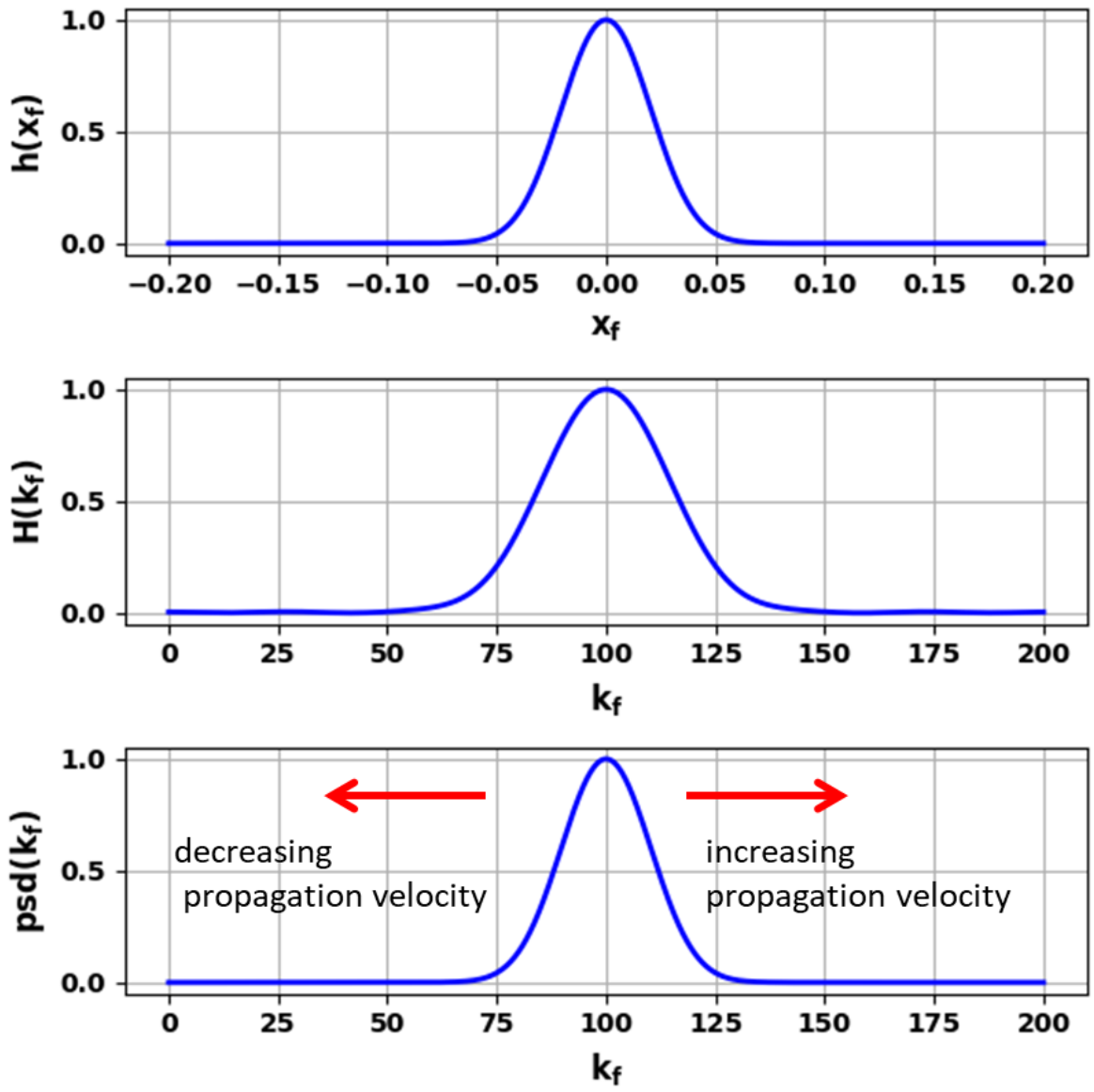

Fig. 4. Initial Gaussian pulse $h(x_f)$ with propagation constant k_0 = 100 and its Fourier transform $H(k_f)$ and the power spectral density $psd(k_f)$. **qm2DA.py**

Code **qm2DA.py** for the Fourier transform

```
xf = linspace(-0.2,0.2,2001)
h =  A*exp(-xf**2/s)*exp(1j*k0*xf)
kmax = 200
kmin = 0
nF = 2001
kf = np.linspace(kmin,kmax,nF)
HR = np.zeros(nF); HI = HR; H = HR+HR*1j
for c in range(nF):
    g = h*np.exp(-1j*kf[c]*xf)
    HR[c] = simps(np.real(g),xf)
    HI[c] = simps(np.imag(g),xf)
    H[c] = simps(g,xf)
psd = 2*(H**2)
psd = np.real(psd/max(psd))
```

Simulation 2 [2D] Pulse striking a potential wall

qm2DB.py

The scattering of the [2D] pulse depends upon the relative values of the pulse energy E and the height of the potential wall U_0.

The potential energy function $U(x,y)$ is given by

$U = 0$ $\quad x < 0.5$

$U = U_0$ $\quad x \geq 0.5$ $\quad U_0 > 0$

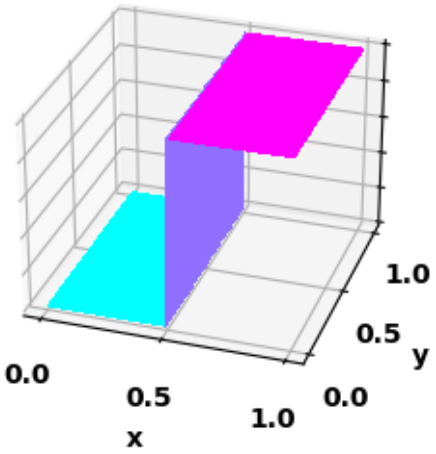

Fig. 5. The potential wall (repulsive barrier). **qm2DB.py**

The energy of the pulse can be taken as $E = k_0^{\,2}$ $\quad (\hbar = 1 \;\; m = 1)$.

When k_0 = 100 a.u., E = 10 000 a.u.

Figure 6 shows the scattering for a potential wall of different heights. At the repulsive barrier, both reflection and transmission occur. The incident pulse and reflected pulse interfere with each producing a sequence of interference fringes.

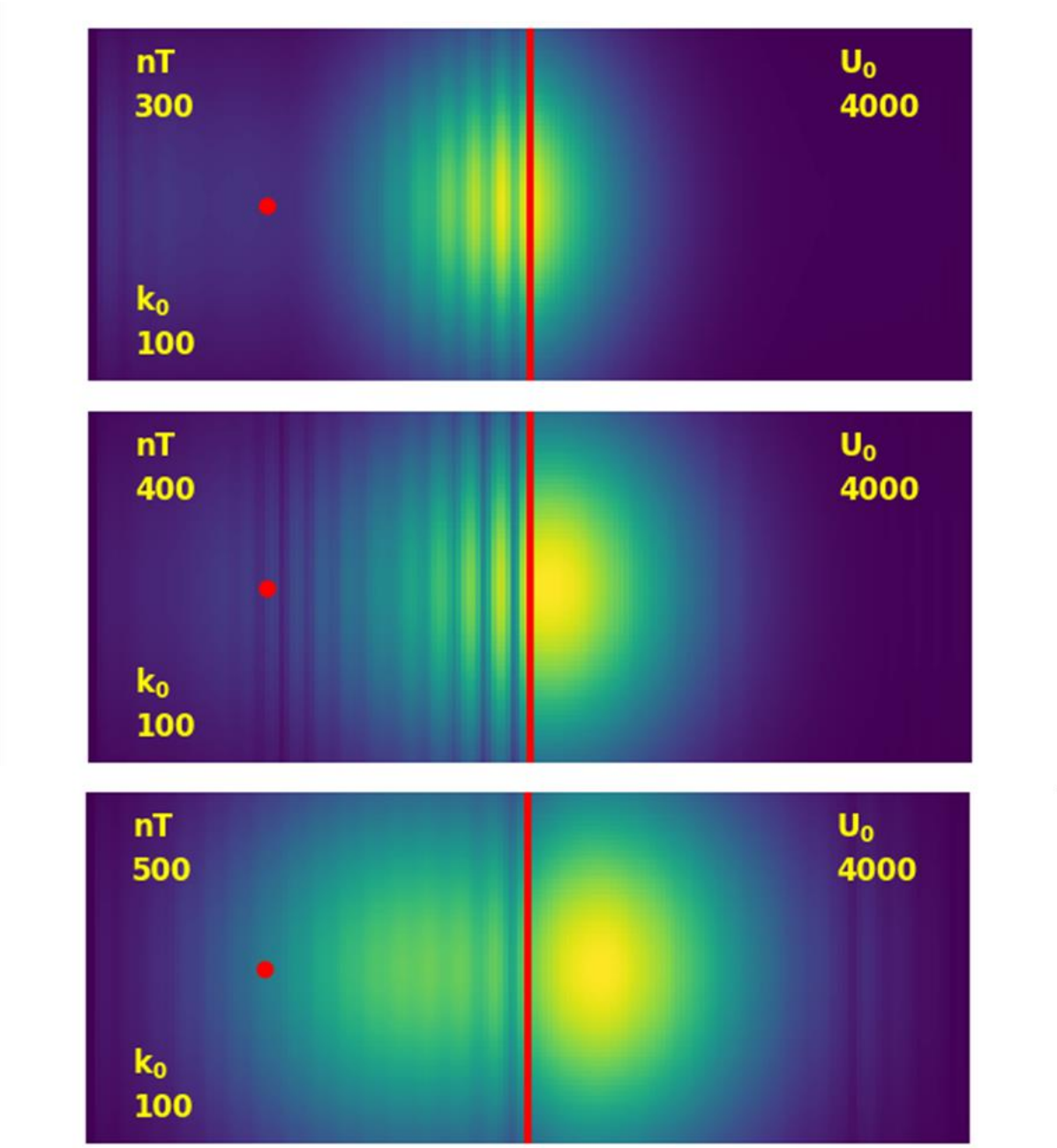

Fig. 6A. Pulse striking the repulsive barrier where $E > U_0$.The incident and reflected waves interfere and

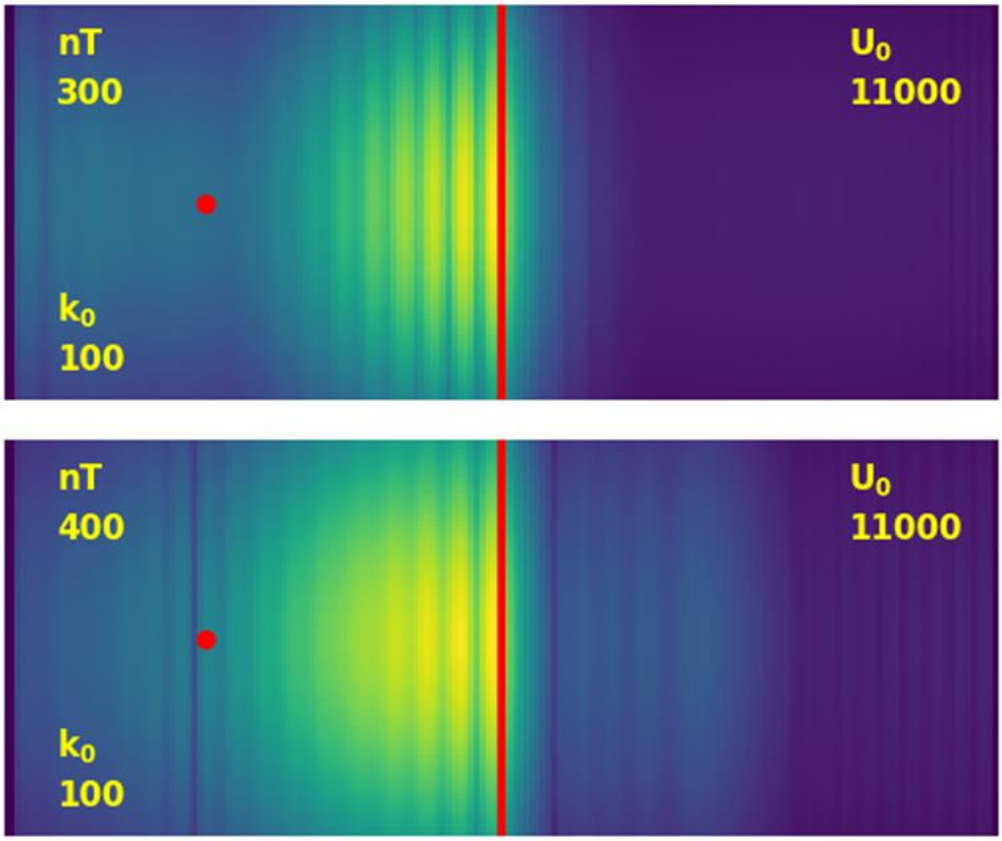

Fig. 6B. Pulse striking a repulsive barrier where $E < U_0$.

Simulation 3 [2D] Pulse striking a potential cliff

qm2DB.py The pulse is scattered by a potential cliff as shown in figure 7.

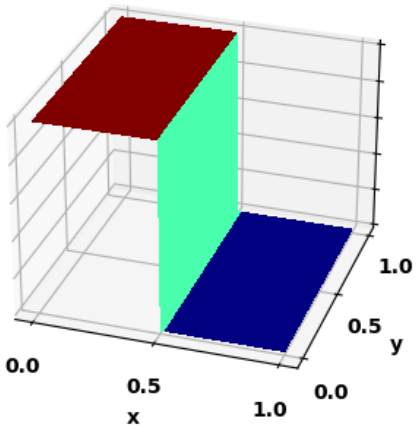

Fig. 7. A potential cliff: U_0 =-1000 a.u

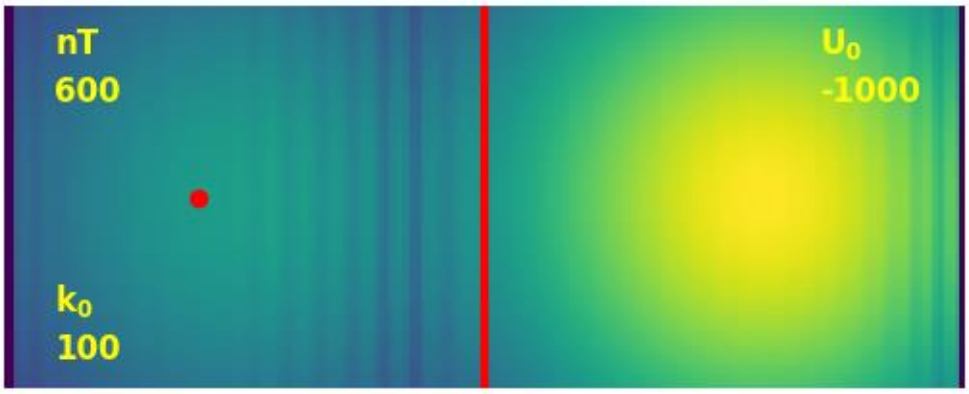

Fig. 8. Scattering by a potential cliff. The pulse is reflected and transmitted at the cliff edge. The incident and reflected waves interfere and produce the interference fringes. You also observe interference fringes at the extreme right of the image due to reflections at the boundary.

Simulation 4 Pulse striking a single slit

qm2DC.py

This is a simulation of the scattering of an electron by a single slit. The pulse is reflected by the barrier and diffracted at the slit as it passes through the it.

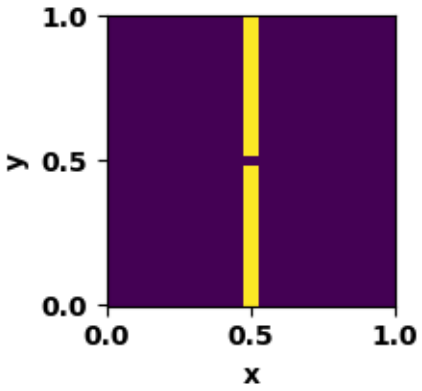

Fig. 9. Potential energy function for single slit.

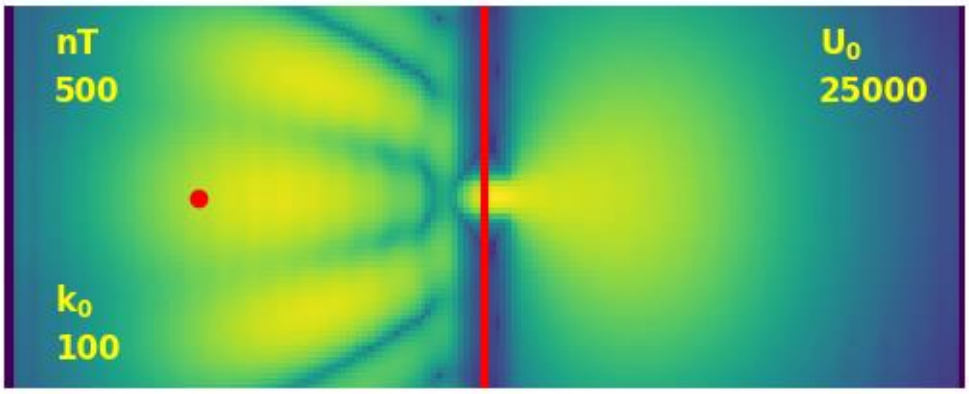

Fig. 10. Single slit diffraction. The pulse spreads (diffracts) in passing through opening. Most of the pulse is reflected at the potential barrier and interferes with the incident wave creating an interface pattern.

Simulation 5 Pulse striking a double slit

qm2DD.py

This is a simulation of the scattering of an electron by a double slit. The matter wave is reflected by the barrier and diffracted at the slit as it passes through the it.

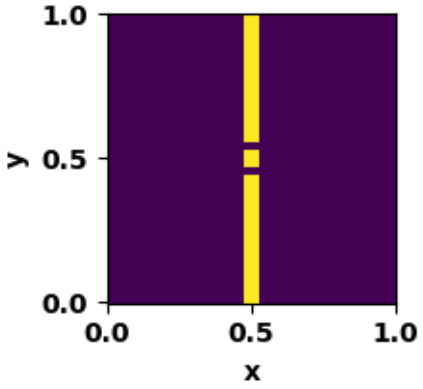

Fig. 11. Potential energy function for double slit.

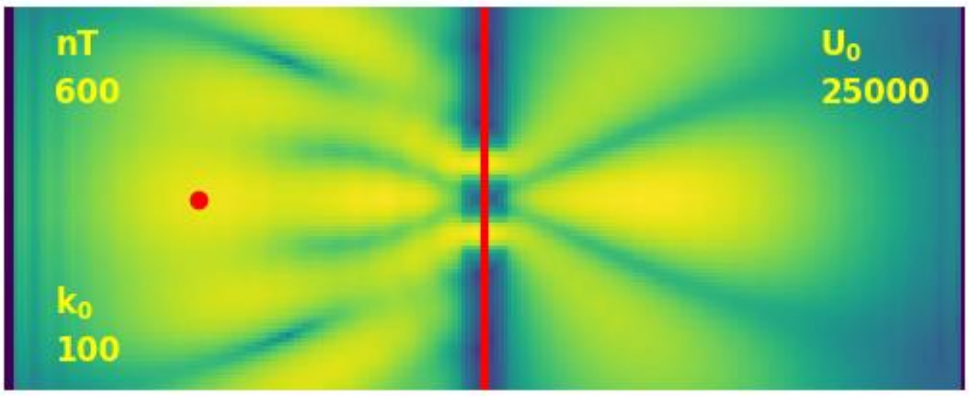

Fig. 12. Double slit diffraction. Pulse spreads (diffracts) in passing through opening. A very predominant interference pattern is displayed with very distinct regions of constructive and destructive interference. The pulse is reflected at the potential barrier and interferes with the incident wave creating an interface pattern.