Evolution of Guassian Wavepacket

Prof. O. S. K. S. Sastri

December 24, 2023

Contents

Abstract

1 Modeling the Physical System

1.1 Description of the System:

1.1.1 Object Description

- **Type of Object:** A microscopic particle
- **Object Composition:** such as electrons, neutrons whose internal structure is not considered
- **Object Variables:** mass of particle m,
  Size of particle described through its spatial extent σ and
  wavepacket associated with the particle that contains the information about momentum p and energy E of the particle

1.1.2 Interaction Description

- **Type of interaction:** The type of interaction of the particle depends on the potential in which the particle is present. For example, a free particle is one for which V(x) = 0. One can consider a particle in a 1-D box, where the size of the box is very large as compared to that of the particle as a good example.
- **Agent of interaction:** The agent of interaction could be either Coulomb or Nuclear depending upon the potential and particle under consideration
- **Interaction Variables:** Strength of interaction typically reflected by the depth or height of the potential 'V_0' and
  Range of interaction, mostly described by the width of the potential 'a'.

1.1.3 Process Description

- **Reference System:** We shall be considering only 1-D potentials for now and a typical cartesian co-ordinate system would be enough for the purpose. Generally, the origin could be chosen to reflect the potential symmetry, if present.
- **State Variables:** The wavefunction and energy of the particle could be considered as the variables that describe the state of the particle, at any given instant.

1.2 Formulation of Mathematical Model

1.2.1 Interaction law

The energy and momentum of the particle are given via the Einstein-Debroglie relations as

$$E = h\nu = \hbar\omega \tag{1}$$

where $\omega = 2\pi\nu$ and ν is the temporal frequency of the particle.

$$p = \frac{hc}{\lambda} = \hbar k \tag{2}$$

where k is the wave-vector and reflects the spatial frequency of the particle **Potential**

1. If its a free particle, V = 0 for all space
2. If its step potential, then V = 0 for x < 0 and V = V_0 for x > 0
3. For a Barrier potential, V = 0 for x < 0 and x > a and V = V_0 for 0 < x < a
4. For a Square well potential, V = V_0 for x < 0 and x > a and V = 0 for 0 < x < a

1.2.2 Dynamical law

Time Dependent Schrodinger Equation

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

1.2.3 Initial and Boundary Conditions

- **Gaussian Wavepacket initialisation:** Consider the wavepacket to be centered about x_0 and having initial momentum $p_0 = \hbar k_0$

$$\psi(x, x_0, t = 0) = \left(\frac{1}{2\pi\sigma^2}\right)^{1/4} e^{-\frac{(x-x_0)^2}{4\sigma^2}} e^{ik_0(x-x_0)} \tag{4}$$

  The uncertainity of this wavepacket Δx would be σ.

- **Normalisation Condition** The wavefunction and its derivative have to be continuous at the boundaries and it should normalised for all times. That is,

$$\int_{-\infty}^{\infty} \psi^*(x, x_0, t)\psi(x, x_0, t)dx = 1 \tag{5}$$

1.3 Ramification: Time evolution of a Gaussian wavepacket

First, we set out to determine the amplitude function $a(k, k_0)$ corresponding to the initial wavepacket, by using the fourier transform relation,

$$a(k, k_0) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dx\ \psi(x, x_0, 0)e^{-ik(x-x_0)} \tag{6}$$

Substituting from Eq.4

$$a(k, k_0) = \frac{1}{\sqrt{2\pi}} \left(\frac{1}{2\pi\sigma^2}\right)^{1/4} \int_{-\infty}^{\infty} dxe^{-(x-x_0)^2/(4\sigma^2)} e^{-i(k-k_0)(x-x_0)} \tag{7}$$

By putting $\alpha = 1/(4\sigma^2)$, $\beta = i(k - k_0)$

$$a(k, k_0) = \frac{1}{\sqrt{2\pi}} \left(\frac{1}{2\pi\sigma^2}\right)^{1/4} \int_{-\infty}^{\infty} dxe^{-\alpha(x-x_0)^2} e^{-\beta(x-x_0)}$$

Adding and subtracting $\frac{\beta^2}{4\alpha}$ in exponent and completing the squares

$$= \frac{1}{\sqrt{2\pi}} \left(\frac{1}{2\pi\sigma^2}\right)^{1/4} \int_{-\infty}^{\infty} dxe^{-\alpha\left((x-x_0)+\frac{\beta^2}{4\alpha}\right)^2} e^{\frac{\beta^2}{4\alpha}}$$

By setting $(x - x_0) + \frac{\beta^2}{4\alpha} = t, \int_{-\infty}^{\infty} dte^{-\alpha t^2} = \sqrt{\frac{\pi}{\alpha}}$,we get

$$a(k, k_0) = \left(\frac{4\sigma^2}{2\pi}\right)^{1/4} \exp\left[-(k - k_0)^2\sigma^2\right]$$

Now, the wavefunction at some time t can be obtained from

$$\psi(x, x_0, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dka(k, k_0)e^{i[k(x-x_0)-\omega t]} \tag{8}$$

$$\psi(x, t) = \frac{1}{\sqrt{2\pi}} \left(\frac{4\sigma^2}{2\pi}\right)^{1/4} \int_{-\infty}^{\infty} dk \exp\left[-(k - k_0)^2\sigma^2 + ik\left(x - x_0 - \frac{\hbar kt}{2m}\right)\right] \tag{9}$$

Adding and subtracting $ik_0\left(x - x_0 - \frac{\hbar k_0 t}{2m}\right)$ in exponent

$$\begin{aligned}\psi(x, t) = \frac{1}{\sqrt{2\pi}} \left(\frac{4\sigma^2}{2\pi}\right)^{1/4} \int_{-\infty}^{\infty} dk \exp\bigg[&- (k - k_0)^2\sigma^2 + i(k - k_0)\left(x - x_0 - \frac{\hbar(k - k_0)t}{2m} - \frac{\hbar k_0 t}{m}\right) \\ &+ ik_0\left(x - x_0 - \frac{\hbar(k - k_0)t}{2m} - \frac{\hbar k_0 t}{2m}\right)\bigg]\end{aligned}$$

Now clubbing $(k-k_0)^2, (k-k_0)$ terms, we get

$$\psi(x,t) = \frac{1}{\sqrt{2\pi}} \left(\frac{4\sigma^2}{2\pi}\right)^{1/4} \int_{-\infty}^{\infty} dk \exp\left[-(k-k_0)^2\left(\sigma^2 + \frac{i\hbar t}{2m}\right) + i(k-k_0)\left(x - x_0 - \frac{\hbar k_0 t}{m}\right)\right]$$
$$\exp\left[ik_0\left(x - x_0 - \frac{\hbar k_0 t}{2m}\right)\right]$$

Now putting $\alpha = \left(\sigma^2 + \frac{i\hbar t}{2m}\right), \beta = i\left(x - x_0 - \frac{\hbar k_0 t}{m}\right)$ and we get

$$\psi(x,t) = \frac{1}{\sqrt{2\pi}} \left(\frac{4\sigma^2}{2\pi}\right)^{1/4} \int_{-\infty}^{\infty} dk \exp\left[-\alpha(k-k_0)^2 + i\beta(k-k_0)\right] \exp\left[ik_0\left(x - x_0 - \frac{\hbar k_0 t}{2m}\right)\right]$$

completing the squares and using standard integral value as before

$$\psi(x,t) = \frac{1}{\sqrt{2\pi}} \left(\frac{4\sigma^2}{2\pi}\right)^{1/4} \exp\left[-\frac{\left(x - x_0 - \frac{\hbar k_0 t}{m}\right)^2}{4\left[\sigma^2 + \frac{i\hbar t}{2m}\right]}\right] \sqrt{\frac{2\pi}{2\left(\sigma^2 + \frac{i\hbar t}{2m}\right)}} \exp\left[ik_0\left(x - x_0 - \frac{\hbar k_0 t}{2m}\right)\right]$$

Introducing

$$\gamma^2 = \sigma^2 + \frac{i\hbar t}{2m}$$
$$\psi(x,t) = \left(\frac{\sigma^2}{2\pi}\right)^{1/4} \frac{1}{\sqrt{\gamma^2}} \exp\left[-\frac{\left(x - x_0 - \frac{\hbar k_0 t}{m}\right)^2}{\gamma^2}\right] \exp\left[ik_0\left(x - x_0 - \frac{\hbar k_0 t}{2m}\right)\right] \tag{10}$$

1.3.1 Emergent Properties:

Since γ is complex in nature, it is better to determine the probability, so as to obtain the width of the packet. The probability is

$$\begin{aligned} P(x,t) =& |\psi(x,t)|^2 \\ =& \sqrt{\frac{\sigma^2}{2\pi|\gamma|^4}} \exp\left[-\sigma^2 \frac{\left(x - x_0 - \frac{\hbar k_0 t}{m}\right)^2}{|\gamma^4|}\right] \end{aligned}$$

Now, one can observe that the width would be given by

$$\begin{aligned} \Delta x(t) &= \frac{1}{\sigma}|\gamma|^2 \\ &= \frac{1}{\sigma}\left[\sigma^4 + \frac{\hbar^2 t^2}{4m^2}\right]^{1/2} \\ &= \sigma\left[1 + \frac{\hbar^2 t^2}{4m^2\sigma^4}\right]^{1/2} \end{aligned}$$

At t=0, $\Delta x = \sigma$ and with time t, choosing a time constant $T = \frac{m\sigma^2}{\hbar}$, we have

$$\Delta x(t) = \sigma\left[1 + \frac{1}{4}\left(\frac{t}{T}\right)^2\right]^{1/2} \tag{11}$$

One must also observe that the center of the wavepacket also changes w.r.t. t as $\frac{\hbar k_0 t}{m}$. That is, larger the initial momentum, the faster the center of the wavepacket moves and spreads equally faster.

1.4 Validation

Time evolution of wave packet in free space Typically, the results of analytical solution need to be verified with experimental results. Here, we are interested in solving the TDSE for a free particle using a numerical technique and validate the results of our simulation with the analytically expected outcomes. For this, we need to determine the probability density at regular time intervals for the wave packet obtained as solution of TDSE. While the amplitude of the wavepacket decreases at its center, given by $\frac{\hbar k_0 t}{m}$, the width of the wavepacket must change with time as

$$\sigma(t) = \sigma_0\left[1 + \frac{\hbar^2}{4m^2\sigma_0^4}t^2\right]^{1/2} \tag{12}$$

This is called dispersion of the wave packet. It happens in free space for quantum particles.
This is because, we have $\omega(k) = \hbar k^2/2m$

2 Preparing the system for Numerical Simulation

2.1 Choice of Numerical Technique: Crank-Nicholson Method

Why this method?
While solving Time dependent Schr*ö*dinger equation (TDSE),

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi(x,t)}{\partial x^2} + V(x) \tag{13}$$

it has the following advantages:

1. Both the time and space derivatives are determined to second order accuracy
2. It is an implicit method and is stable for small time steps
3. Expressed in Cayley form, it has advantage of keeping the wavefunction normalised at all time steps

2.1.1 Central Divided Difference (CDD) formulae

To determine $\frac{\partial\psi}{\partial t}$ and $\frac{\partial^2\psi}{\partial x^2}$ to second order accuracy using CDD, the steps involved are:

1. **Discretisation of variables**
   Sample x and t in steps of Δx and Δt respectively as:

   $$x = [x_1, x_2, ...x_i, x_{i+1}, ...x_n]; \tag{14}$$

   where $\Delta x = x_{i+1} - x_i$ and

   $$t = [t^1, t^2, ...t^\ell, t^{\ell+1}, ...t^n]; \tag{15}$$

   where $\Delta t = t^{\ell+1} - t^\ell$

2. **Discretising $\psi(x,t)$ and V(x)**
   $\psi(x_i, t^l)$ is written as ψ_i^ℓ, and
   V(x_i) is written as V$_i$

3. **CDD Formula for first derivative: Time derivative**
   Determine $\frac{\partial\psi}{\partial t}$ at the mid point of interval $[t^\ell$,$t^{\ell+1}]$ as:

   $$\frac{\partial\psi(x_i, t^\ell + \frac{\Delta t}{2})}{\partial t} = \frac{\partial\psi(x_i, t^{\ell+1/2})}{\partial t} = \frac{\psi_i^{\ell+1} - \psi_i^\ell}{\Delta t} = \frac{\psi(x_i, t^{\ell+1}) - \psi(x_i, t^\ell)}{t^{\ell+1} - t^\ell} \tag{16}$$

   Note: *Check that* $\frac{\partial\psi}{\partial t}$ *determined using above formula is accurate to second order.*

4. **CDD for second derivative: Space derivative** We know $\frac{\partial^2\psi}{\partial x^2}$ to second order accuracy using CDD as:

   $$\frac{\partial^2\psi(x_i, t^\ell)}{\partial x^2} = \frac{\psi(x_{i+1}, t^\ell) - 2\psi(x_i, t^\ell) + \psi(x_{i-1}, t^\ell)}{\Delta x^2} = \frac{\psi_{i+1}^l - 2\psi_i^\ell + \psi_{i-1}^\ell}{\Delta x^2} \tag{17}$$

   We need $\frac{\partial^2\psi}{\partial x^2}$ at $t^{\ell+1/2}$, i.e at the mid-point of the interval. So, we also determine $\frac{\partial^2\psi(x_i, t^{\ell+1})}{\partial x^2}$ as:

   $$\frac{\partial^2\psi(x_i, t^{\ell+1})}{\partial x^2} = \frac{\psi_{i+1}^{\ell+1} - 2\psi_i^{\ell+1} + \psi_{i-1}^{\ell+1}}{\Delta x^2} \tag{18}$$

5. **Space derivative at mid-point** Now $\frac{\partial^2\psi}{\partial x^2}$ at $t = t^{\ell+1/2}$ would be the average of the two values of $\frac{\partial^2\psi}{\partial x^2}$ at t = t^ℓ and $t = t^{\ell+1}$ and is obtained as:

   $$\frac{\partial^2\psi(x_i, t^{\ell+1/2})}{\partial x^2} = \frac{1}{2\Delta x^2}\left[(\psi_{i+1}^\ell - 2\psi_i^\ell + \psi_{i-1}^\ell) + (\psi_{i+1}^{\ell+1} - 2\psi_i^{\ell+1} + \psi_{i-1}^{\ell+1})\right] \tag{19}$$

6. **Therefore TDSE using CDD formulae is obtained as:**

   $$i\hbar\frac{\psi_i^{\ell+1} - \psi_i^\ell}{\Delta t} = -\frac{\hbar^2}{2m}\left[\frac{(\psi_{i+1}^\ell - 2\psi_i^\ell + \psi_{i-1}^l) + (\psi_{i+1}^{l+1} - 2\psi_i^{\ell+1} + \psi_{i-1}^{\ell+1})}{2\Delta x^2}\right] + \frac{V_i}{2}[\psi_i^{\ell+1} + \psi_i^\ell]$$

   Hence,

   $$\psi_i^{\ell+1} - \psi_i^\ell = \frac{i\hbar\Delta t}{4m\Delta x^2}\left[(\psi_{i+1}^\ell - 2\psi_i^\ell + \psi_{i-1}^\ell) + (\psi_{i+1}^{\ell+1} - 2\psi_i^{\ell+1} + \psi_{i-1}^{\ell+1})\right] + \frac{-iV_i\Delta t}{2\hbar}[\psi_i^{\ell+1} + \psi_i^\ell]$$

2.1.2 Tri-diagonal system of equations in Matrix form:

1. **Simplification with notational factors**
   Introducing:
$$f_i = \frac{V_i \Delta t}{2\hbar} \tag{20}$$
   and
$$g = \frac{\hbar \Delta t}{4m\Delta x^2} \tag{21}$$
   Rearranging the following equation
$$\psi_i^{\ell+1} - \psi_i^{\ell} = \frac{i\hbar\Delta t}{4m\Delta x^2}\left[(\psi_{i+1}^{\ell} - 2\psi_i^{\ell} + \psi_{i-1}^{\ell}) + (\psi_{i+1}^{\ell+1} - 2\psi_i^{\ell+1} + \psi_{i-1}^{\ell+1})\right] + \frac{-iV_i\Delta t}{2\hbar}[\psi_i^{\ell+1} + \psi_i^{\ell}]$$
   by keeping all $\psi^{\ell+1}$'s to L.H.S and ψ^{ℓ}'s to RHS:
$$(1 + if_i + 2ig)\psi_i^{\ell+1} - ig\psi_{i+1}^{\ell+1} - ig\psi_{i-1}^{\ell+1} = (1 - if_i - 2ig)\psi_i^{\ell} + ig\psi_{i+1}^{\ell} + ig\psi_{i-1}^{\ell} \tag{22}$$

2. **Tridiagonal matrix:** This is a system of equations and can be put in tri-digonal matrix form as:
$$(I + iH)\begin{bmatrix}\psi_1\\ \psi_2\\ \vdots\\ \psi_n\end{bmatrix}^{t^{\ell+1}} = (I - iH)\begin{bmatrix}\psi_1\\ \psi_2\\ \vdots\\ \psi_n\end{bmatrix}^{t^{\ell}} \tag{23}$$
   where I is unit/Identity matrix and H is given by:
$$H = \begin{bmatrix} f_1+2g & -g & 0 & 0 & \cdots & 0 & 0 & 0\\ -g & f_2+2g & -g & 0 & \cdots & 0 & 0 & 0\\ 0 & -g & f_3+2g & -g & \cdots & 0 & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\ 0 & 0 & 0 & 0 & \cdots & -g & f_{n-1}+2g & -g\\ 0 & 0 & 0 & 0 & \cdots & 0 & -g & f_n+2g \end{bmatrix}$$

3. **TDSE in Cayley Form**
   The numerical difference equation in Cayley form is
$$\psi^{\ell+1} = \left(\frac{I - iH}{I + iH}\right)\psi^{\ell} \tag{24}$$
   Normalisation for all times The temporal operator that relates ψ^{ℓ} to $\psi^{\ell+1}$ is numerically stable and is also unitary, since:
$$\left(\frac{I - iH}{I + iH}\right)^{*}\left(\frac{I - iH}{I + iH}\right) = \left(\frac{I + iH}{I - iH}\right)\left(\frac{I - iH}{I + iH}\right) = I \tag{25}$$

4. **Decoupling the Complex Equations**
   Now since $\Psi(x,t)$ is, in general a complex wave function, we express eq.24 as:
$$Re(\psi^{\ell+1}) + i * Im(\psi^{\ell+1}) = \frac{I^2 - H^2 - i2IH}{I^2 + H^2}[Re(\psi^{\ell}) + i * Im(\psi^{\ell})] \tag{26}$$
   That is, the real and imaginary parts are
$$Re(\psi^{\ell+1}) = \left(\frac{I^2 - H^2}{I^2 + H^2}\right)Re(\psi^{\ell}) + \left(\frac{2IH}{I^2 + H^2}\right)Im(\psi^{\ell}) \tag{27}$$
$$Im(\psi^{\ell+1}) = \left(\frac{I^2 - H^2}{I^2 + H^2}\right)Im(\psi^{\ell}) + \left(\frac{-2IH}{I^2 + H^2}\right)Re(\psi^{\ell}) \tag{28}$$
   These are the final numerical equations that can be solved using Scilab

2.1.3 Rephrasing the problem in Appropriate Units

Atomic units We choose $\hbar = 1$ and $m_e = 1$. All energies are in Hartree and distances are in Bohr.

2.1.4 Discretising Continuous Variables and Limiting the Region of Interest

- x is discretised in steps of dx in the interval [-L,L] as x = -L:dx:L;
  Typical values are L = 1 and dx = 0.005;
- t is discretised in steps of dt. Typical values are dt = 10^{-5}
- t0 and tf are initial and final times respectively

3 Implementation of Crank Nicholson Algorithm in Python

3.1 Initialising Gaussian Wavepacket

First step is to initialise the wavepacket for the particle. Gaussian wavefunction at t = 0 as
function Psi0 = Gaussian(x,x0,sigma0,k0)
Choosing that the wavepacket is
centered about x0 (= 0)
with intial width as sigma0 (= 0.1) and
having an initial momentum p0 = k0 ($\hbar$=1) given as input from user or from calling function.

function for initialising Gaussian in Python

```
def Gaussian(x,x0,sigma0,k0):
    RePsi0 = ((2*np.pi*sigma0**2)**(-0.25))*(np.exp(-(x-x0)**2/(4*sigma0**2)))*np.cos(k0*x)
    ImPsi0 = ((2*np.pi*sigma0**2)**(-0.25))*(np.exp(-(x-x0)**2/(4*sigma0**2)))*np.sin(k0*x)
    j=complex(0,1)
    Psi0 = RePsi0+j*ImPsi0;
    axs[0][0].plot (x,RePsi0,label='j=0')
    axs[0][0].grid()
    axs[0][0].set_title('For j={}'.format(0))
    axs[0][0].plot(x,V)
    return RePsi0
```

3.2 Defining the potential function

Considering a free particle but restricting the region of interest to [-L,L] implies solving for a particle in 1-D box. The potential height at the ends is ideally infinity and anyways does not appear in the numerical solution as the wavefunction $u(x)$ is assumed to be zero at both the ends. That is, $u(1) = 0$ as $u(x = -L) = 0$ and also $u(N) = 0$ as $u(x = +L) = 0$.
Just to ensure that V0 is not zero at the ends, we set it equal to a large value, such as 10^6.

Potential function in Scilab

```
def Potential(V0,x,N):
# V0 and x are inputs obtained into the function
# So also N: Number of points representing x-vector
N=len(x)
    V=np.zeros(N);#Initialising V to be zero at all points
    V[1] = V0; V[N-1] = V0;\\Setting V at two ends as large value
    plt.subplot(1,1,1)
    plt.plot(x,V)
    return V
```

3.3 Preparing the H-matrix

```
def Hmatrix(V,dx,dt,N):
#step sizes dx, dt are obtained as inputs
#So also potential: V, and total number of samples of x: N
    f = V*dt/2;
    g = dt/(4*dx**2);
    H = np.diag(f+2*g)+np.diag(-g*np.ones(N-1),-1)+np.diag(-g*np.ones(N-1),1);
    return H
```

3.4 Code for Crank-Nicholson step

```
def CrankNicholsonTDSE(k0,n):
    //Written by Prof.O.S.K.S. Sastri
    //n is the number of iterations
    count_r,count_c=0,1
    Psi=[] ; j=complex(0,1)
    x0=0 ; k0=0 ; sigma0=0.1
    dx=0.005 ; dt=0.001 ; L=2
    x=np.arange(-L,L,dx)
    N=len(x)
    psi0=Gaussian(x,x0,sigma0,k0)
    V=Potential(x)
    i = 1; #Psi at ith time
    t = dt;#initialising time step
I=np.eye(N)
    H= Hmatrix(V,dx,dt,N)
    #wavefunction for i=1 is initial wavepacket
Psi.append(psi0);
    i = 2;
    j = 1; # iteration number
    t = t+dt
    M = np.linalg.inv(I + j*H)@(I - j*H);
    ow=3 ; col=3
    fig,axs=plt.subplots(row,col)
    plt.ylim(0,2)
    plt.figure(figsize=(20,20))
    fig.tight_layout()
    while j<=n:
        Psi.append(M@Psi[j-1])
        i+=1 ; j+=1 ; t+=1
        if j%20==1:
            axs[count_r][count_c].plot(x,Psi[j-1].real)
            axs[count_r][count_c].plot(x,V)
            axs[count_r][count_c].grid()
            axs[count_r][count_c].set_ylim(0,2)
            axs[count_r][count_c].set_title(’For j={}’.format(j-1))
            count_c+=1
        if count_c==col:
            count_r+=1 ; count_c=0
    return Psi
```

4 Simulation of Results and Discussion:

Initially, we will study the case where in the particle is assumed to be stationary at the origin. That is, energy of the particle is taken to be zero. This is given through the wave-vector k in the gaussian wavepacket.

4.1 Dispersion of stationary particle: A wavepacket with k0 = 0:

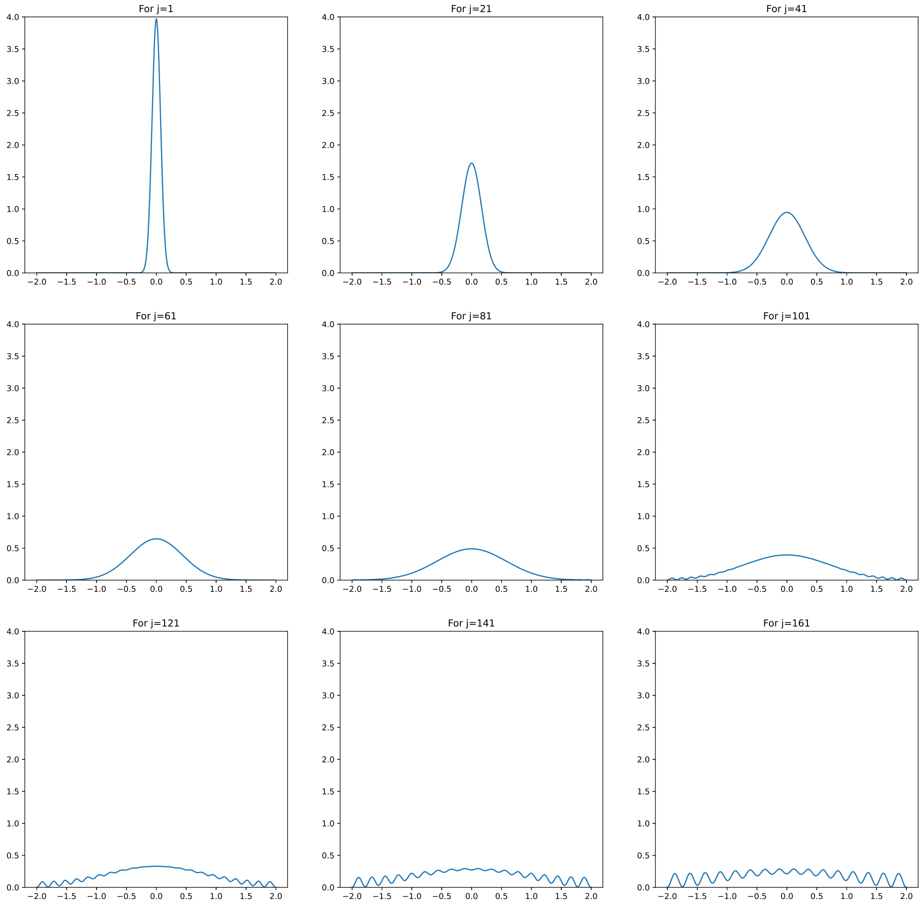
The results are generated by creating 3×3 matrix of plots, which show the wavefunction at the end of 20 iterations. So, a total of 9 plots to be observed. This implies, the number of iterations should be 9*20 = 180.

Since, the time step is chosen as dt = 0.001, we are observing the wavefunction at the end of every 0.02 sec. On running the simulation using the command:

```
--> Psi=CrankNicholsonTDSE(0,180);
```

the wavepackets at the end of every 0.02 secs will appear in various subplots, as shown in Fig. 1. One can observe that the center of wavepacket remains at the origin as the energy of the wavepacket, that is, the value of k0 has been chosen to be 0. The expected values for width of the wavepacket from analytical formula Eq.12, with $\hbar = m_e = 1$, at these times are obtained in Scilab using the following commands:

Figure 1: Dispersion of Wavepacket with time



```
--> sigma0 = 0.1
--> t = np.arange(0.001,0.001*180,0.001*20)
--> print(t)
[0.001 0.021 0.041 0.061 0.081 0.101 0.121 0.141 0.161]

--> width = sigma0*(1 + 0.25*t**2/sigma0**4)**(1/2)
-->  print(width)
[0.10012492 0.145      0.22808989 0.32097508 0.41716304 0.51480579
 0.61320877 0.71205688 0.8111874 ]
```

Now, to obtain the width from simulation, we need to determine the expectation values of x and x^2. From which the width is obtained as

$$\Delta x = sqrt(< x^2 > - < x >^2) \tag{29}$$

This can be acheived by adding the following function to the code:

```
function [meanx,w] = deltax(x,psi)
    f = abs(conj(psi).*psi);
    x=x';
    meanx = intsplin(x,f.*x);
    //disp(meanx)
    meanxsqr = intsplin(x,f.*x.^2);
    //disp(meanxsqr)
    w = sqrt(meanxsqr - meanx^2);
endfunction
```

Then, call this function in the loop for plotting as follows:

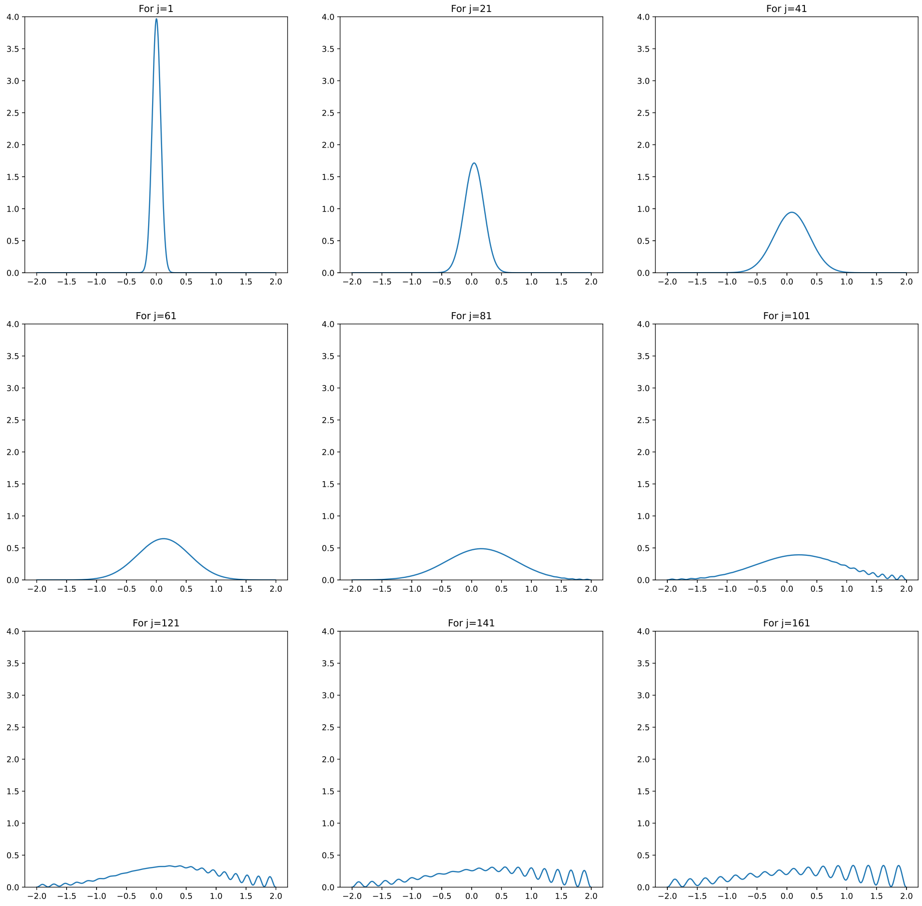
```
//Plot after every 20 iterations
        if modulo(j,20)==1
            scf(2);
            subplot(3,3,p)
            plot2d(x',conj(Psi(1:N,i)).*Psi(1:N,i),rect=[-L 0 L 4])
            plot(x,V,'r')
            [meanx,delx] = deltax(x,Psi(1:N,i));
            disp("<x> =",meanx,"width =", delx)
            p=p+1;
        end
```

The outputs are 0.1001247, 0.1449319, 0.2279250, 0.3207156, 0.4168103, 0.5142845, 0.6115932, 0.7057447 and 0.7928531. The first five values match with the theortically expected ones to three decimal places, the next three to two decimal places and the last one to one decimal place. So, the error due to numerical method and choice of step size propagates with more and more iterations. The mean position of the wavepacket remains at the origin.

4.2 Dispersion of travelling particle: A wavepacket with k0 $\neq$ 0:

A particle travelling would have certain momentum of energy, which will be reflected in the wavepacket definition through the value of k0. Now, the value of sin(k0*x) does not become 0. Running the code for k0 =2 gives the evolution of wavepacket as shown in Fig.2. One can clearly observe the shift of the center of the wavepacket with time, that is, number of iterations. It is interesting to note that the wavepacket reflects from the boundary after 0.1 secs and interferes with the part of wavepacket that has not yet reached. This can be verified by increasing the RoI to larger value so that the dispersed wavepacket does not reach the boundary.
Once again, one can easily check that the expected mean position values are simply, given by $k0 * t$, which are 0.002, 0.042, 0.082, 0.122, 0.162, 0.202, 0.242, 0.282, 0.322. The obtained values match closely.

Figure 2: Dispersion of Wavepacket with time


For j=1
For j=21
For j=41
For j=61
For j=81
For j=101
For j=121
For j=141
For j=161