Chapter 1

Simulation of 1D and 2D Wave-packets

1.1 Modeling the Free Particle State-function in 1-D:

1.1.1 Description:

A microscopic particle (object type) such as an electron which has no internal structure (object composition) of mass m_e, momentum p, energy E and finite size (object variables) is assumed to be free (interaction type). The particle is considered to be in 1-D, which fixes reference system as x-axis with particle considered to be at the origin at time $t = 0$. The stationary state function $\phi(x)$ describes state of the particle and together with energy E constitutes the state variables.

1.1.2 Formulation:

Even though, the electron is assumed to be free (that is, not being acted upon by any forces), yet it has a dispersion relation even in vacuum as well. It is dictated by the Einstein-DeBroglie relations and is given by

$$E = \hbar\omega = \frac{p^2}{2m_e} = \frac{\hbar^2 k^2}{2m_e} \tag{1.1}$$

Therefore, the interaction law is

$$\omega(k) = \frac{\hbar k^2}{2m_e} \tag{1.2}$$

The motion or dynamics of all particles in microscopic domain are governed by Time-Dependent Schrodinger Equation (TDSE) given by

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m_e}\frac{\partial^2\psi(x,t)}{\partial x^2} + V(x)\psi(x,t) \tag{1.3}$$

Assuming that the wave-function can be expressed as a product of independent functions of x and t as $\psi(x,t) = \phi(x)T(t)$ and using separation of variables technique, we obtain the Time-Independent Schrodinger equation (TISE) as

$$\frac{d^2\phi(x)}{dx^2} + \frac{2m_e}{\hbar^2}(E - V(x))\phi(x) = 0 \tag{1.4}$$

and the equation for T(t) as

$$i\hbar\frac{dT(t)}{dt} = ET(t) \tag{1.5}$$

solving which we obtain

$$T(t) = exp\left(-\frac{i}{\hbar}Et\right) \tag{1.6}$$

So, the characteristics of stationary state-function $\phi(x)$, that is a solution of TISE are

1. it should be single valued
2. both $\phi(x)$ and $\phi'(x)$ must be continuous
3. it must be square integrable, that is, it must satisfy

$$\int_{-\infty}^{\infty} \phi^*(x)\phi(x)dx = N \tag{1.7}$$

where N is a finite quantity. Dividing $\phi(x)$ by $\sqrt{N}$, the state function will satisfy the normalisation condition

$$\int_{-\infty}^{\infty} \phi^*(x)\phi(x)dx = 1 \tag{1.8}$$

Since, the particle has a finite spatial extent, this condition implies that $\phi(x) \to 0$ very quickly as $x \to \pm\infty$.

1.1.3 Ramification:

Since V(x) = 0 for a free particle, the solution of TISE is obtained by setting

$$k^2 = \frac{2m_eE}{\hbar^2} \tag{1.9}$$

Hence, the solutions are plane waves $exp(\pm ikx)$, suggesting the wave-vector associated with the particle could be either in the positive or negative x-direction. But such a function $\phi(x) = c_1exp(ikx) + c_2exp(-ikx)$ is not normalisable, since

$$\int_{-\infty}^{\infty} \phi^*(x)\phi(x)dx = \infty \tag{1.10}$$

footnote: One does come across plane wave solutions often in Quantum mechanics and they are made acceptable either using Box-normalisation or Delta-function normalisation. See ref[chap3, Amit Goswami].

Discussion: How to associate a wave with a particle?
It is not possible to associate a single wave with a fixed momentum $p = \hbar k$ and wavelength, 'say $\lambda = \frac{2\pi}{k}$ (or spatial frequency $f_s = 1/\lambda$)' to a particle. Since, it would extend its spatial extent to occupy all of space.
This is also not acceptable due to Heisenberg's uncertainty principle. Since the measurement of position of the particle is to be specified as its mean value $< x > \pm \Delta x$ (the uncertainty given via its standard deviation value, see Appendix). Even though the mean position might result in particle to be present at the origin, the uncertainty would certainly be atleast of an order such as finite spatial extent of particle. Hence, particle's momentum can not be a single value but necessarily has to be specified as a mean value $< p > (= \hbar < k >)$ with an uncertainty $\Delta p (= \hbar \Delta k)$ spread over an interval, such that $\Delta x \Delta p \geq \hbar$ (or $\Delta x \Delta k \geq 1$). This indicates that wave associated with that of a particle, called as wave packet, has to be due to a group of waves with wave-vectors 'k' (or spatial frequencies f_s) in an interval $[-\frac{\Delta k}{2}, \frac{\Delta k}{2}]$ (or $[-f_s/2, f_s/2]$) with amplitudes a(k).

What represents the velocity of the particle?
The overall wavefunction $\psi(x,t)$ is a product of $\phi(x)T(t)$ and hence shall be of the form $exp^{\pm i(kx-wt)}$ and the wavepacket would be in general a linear combination of many of these waves with wave-vectors in the interval $[-\frac{\Delta k}{2}, \frac{\Delta k}{2}]$ with different amplitudes a(k). The argument of the plane wave is called the phase, which physically represents the shape of the wave, that is supposed to remain constant as the wave progresses w.r.t. time 't'. This idea gives rise to the concept of phase velocity for individual plane waves and that of group velocity for a group of waves constituting a wave-packet.

The **phase velocity** is given by

$$\frac{d}{dt}(kx - \omega t) = 0 \tag{1.11}$$

$$v_{ph} = \frac{omega}{k} \tag{1.12}$$

Substituting from eq. 1.2, we get

$$v_{ph} = \frac{\hbar k}{2m_e} = \frac{v}{2} \tag{1.13}$$

the phase velocity to be half the particle velocity.

Group velocity is defined as

$$v_{gr} = \frac{d\omega}{dk} \tag{1.14}$$

and substituting for ω from eq 1.2, we obtain v_{gr} to be equal to velocity v of the particle.

Conclusion: So one has to have a superposition of waves of different wavelengths/spatial frequencies f_s and amplitudes a(k) in such a way that there is a finite region in which it exists and goes to zero over rest of the space. Such a superposition is called a wave-packet. It is the choice of different a(k) values that results in different shapes of wave-packets.

1.2 Simulation 1: Construction of a Wave-packet using Superposition

Formation of Beats: Adding two waves of slightly differing frequencies produces a beat (see Fig.), where in the amplitude of the beat (the envelop) dies down on either side to zero over a beat period.
What is needed is, to increase the periodicity ideally to infinity which would leave us with, a superposition-wave peaking at its center in a single region and dies down to zero on either side very quickly.

1.2.1 Fourier Superposition:

Any periodic waveform f(x) with periodicity λ can be expressed as a linear combination of simple harmonic waves. The wave-vector with $k_0 = \frac{2\pi}{\lambda} = 2\pi f_s$, where $f_s = \frac{1}{\lambda}$ is fundamental spatial frequency. All its harmonics mk_0 are added with appropriate amplitudes and is mathematically represented as

$$f(x) = \sum_{m=0}^{\infty} a(mk_0)cos(2\pi m f_s x) + b(mk_0)sin(2\pi m f_s x) \tag{1.15}$$

where the Fourier coefficients $a(mk_0)$ and $b(mk_0)$ are given by

$$a(mk_0) = \frac{2}{\lambda}\int_{-\frac{\lambda}{2}}^{+\frac{\lambda}{2}} f(x)cos(mk_0x)dx \tag{1.16}$$

$$b(mk_0) = \frac{2}{\lambda}\int_{-\frac{\lambda}{2}}^{+\frac{\lambda}{2}} f(x)sin(mk_0x)dx \tag{1.17}$$

1.2.2 Preparation of System for Numerical Solution:

Choice of units: Here, we use atomic units with $\hbar = 1$ and $m_e = 1$ (See Appendix for a detailed discussion). The energies are in Hartree and distances are in Bohr units.

Discretisaiton of continuous variables: The various continuous variables need to be discretised and infinitely large qunatities need to be limited to certain finite values so that it becomes suitable for numerical implementation in a computer.

- The values of x which can extend from $(-\infty, \infty)$ have to be limited to a finite interval, say *[-L,L]*. This becomes the region of interest (RoI) in which the physics of the system under consideration shall be studied.
- Values of x are discretised uniformly in steps of *dx*. That is, *x = -L:dx:L*.
- The number of terms to be added has to be finite, say *n*.
- An ideal choice for all waves to be in phase, say at origin, whose superposition leads to a peak, would be cosine waves.
- So, let us set co-efficients of sine waves in the expansion to zero. That is, $b(mk_0) = 0$.
- The values of spatial frequencies to be added f_s need to be closely spaced so that strong constructive interference occurs at the origin and equally strong destructive interference results at regions beyond on either direction.
- Let the amplitudes of all the waves being added be equal, set $a(mk_0) = 1$. This is akin to chosing a rectangular pulse function in k-space.

Hence, the numerical algorithm parameters involved are *dx, n and L*. The system parameters are spatial frequency f_s to be written as fs, and the coefficents of expansion $a(mk)$ and $b(mk)$ for short *am* and *bm* respectively.

Algorithm: Typically, the algorithm involves specifying all the input variables, which can be identified from our modeling step as the object, interaction and state variables. This is followed by the process block, where in the central equation being worked out is generally specified as an iterative equation that can be looped over to obtain the required outputs. These outputs are then either tabulated or graphed, so that one could easily discuss the results and draw conclusions.

1. Choose system parameters: *am = 1, bm = 0, fs = 1*
2. Choose algorithm parameters:*n = 10, L = 1, dx = 0.001* or equivalently we can choose number of points to sample in RoI [-L,L], as *N = 30*n*fs*.
3. Initialisation: loop variable *m = 1*, state function: *phi = 0*.
4. Central equation to be iterated: *phi = phi + am*cos(2*π*m*fs*x)*.
5. Increment loop variable: *m = m + 1*.
6. Repeat the above two steps till *m = n*.
7. plot(x,phi)

1.2.3 Numerical Implementation in Python

Given below is the Python code, with in-line documentation to help understand the implementation.

```
def wavepacket(fs,n,L):
#written by Dr. O.S.K.S. Sastri
#for understanding how to construct a wave-packet
#[-L,L] is the region of interest in which the wave-packet
#is constructed
#n is the number of waves to be added
#fs is the fundamental spatial frequency whose harmonics are added
x = linspace(-L,L,1000);
#x is the discretised space variable on [-L,L]
phi=0;
#phi shall hold the sum of the series. Here it is initialised to 0
am = 1; #All the amplitudes are set to 1.
m = 1; #loop variable to keep track of number of terms added
while m <= n:
    phi = phi + am*cos(2*pi*m*fs*x); #general term of the series
    m = m + 1;                  #incrementing the loop variable
plot(x,phi)
```

1.2.4 Simulation and Discussion of Results:

There are three parameters to vary: number of terms to add n, fundamental frequency fs whose integral multiples are added and RoI of length $2L$, over which we wish to generate the superposition.

To start with, we run the code for adding $n = 10$ waves, of frequencies $fs = 1$ (fundamental) and its harmonics $2, 3, \ldots, 10$, on a region between $[-1, 1]$, that is $L = 1$. The output is shown in Fig. 1.1. The superposition gives a peak followed by regions of very low probability amplitude on either side but at the edges, we find the peaks appearing again. The wave-packet is periodic with periodicity of $\lambda = 1/fs$.

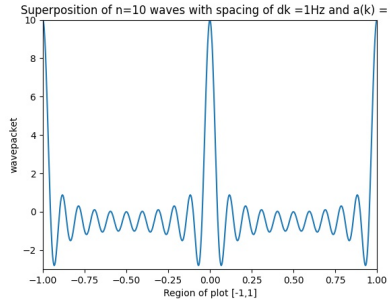
What happens when we add more number of waves, keeping fs and RoI constant?

When $n = 20$, doubling the number of waves being added, see Fig. 1.2(a), the destructive interference is more pronounced in the sense that the ratio of the central peak to that of the rest is further enhanced. But, the peaks at the edges do not disappear, that is the periodicity is not affected.

What happens if were to change the spacing between the waves, that is, vary fs keeping n and RoI constant?

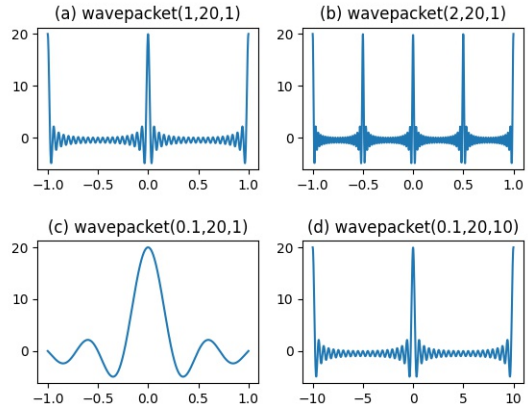
Doubling fs to 2, results in the peaks to be appearing at 0.5 units (See Fig. 1.2(b)). As expected doubling the fundamental frequency has halved the periodicity. Check that, if fs is halved, the periodicity doubles to 2 units. So, now if $fs = 0.1$, then the peaks would

Figure 1.1: Superposition of cosine waves for $fs = 1, n = 10$ and RoI is $[-1, 1]$



disappear, only to reappear when we increase the RoI to $[-10, 10]$ as seen in Figs. 1.2(c) and (d), respectively. So, the key is to add waves of closely spaced frequencies (fs tending to

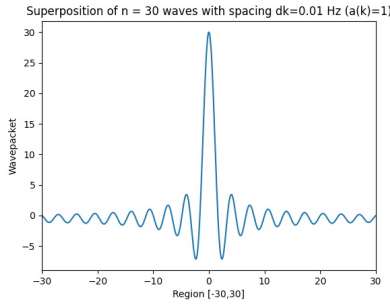
Figure 1.2: Wave-packet construction for different parameters



0) for increasing the periodicity of the peaks to move farther and more number of them (n tending to infinity) to have stronger destructive interference. A typical wavepacket has been shown in Fig.1.3, with a superposition adding up 30 waves with spacing of 0.01 and RoI as [-30,30]. Obviously, if we increase the region of plotting to [-100,100] or more, we shall see that the peaks return since they are $1/0.01 = 100$ units apart.

Conclusion: The key is to add waves of closely spaced frequncies ($fs \to 0$) for increasing the periodicity of the peaks to spread out farther and farther (to ∞) and add more number of them ($n \to \infty$) to have stronger destructive interference. A wave-packet requires an infinite number of waves superimposed with infinitesimally close frequencies. The choice of different

Figure 1.3: A typical wave-packet



amplitude functions a(k) would give rise to different wave-packets. Mathematically the wave-packet, which is a non-periodic function, is best represented using a Fourier transform.

1.3 Simulation 2: Wave-packet using Fourier Transform

Complex Fourier Series: Using the formulae for $\cos(\theta)$ and $\sin(\theta)$ given by

$$cos(\theta) = \frac{\left(e^{i\theta} + e^{-i\theta}\right)}{2} \tag{1.18}$$

$$sin(\theta) = \frac{\left(e^{i\theta} - e^{-i\theta}\right)}{2i} \tag{1.19}$$

in eq. 1.15 for Fourier series, we obtain the complex fourier series as

$$f(x) = \sum_{m=-\infty}^{\infty} c(2\pi m f_s) exp(i2\pi m f_s x) \tag{1.20}$$

where the co-efficients $c(2\pi m f_s)$, by remembering that $\lambda = 1/f_s$, are obtained as

$$c(2\pi m f_s) = f_s \int_{-\frac{1}{2f_s}}^{+\frac{1}{2f_s}} f(x) exp(-i2\pi m f_s x) dx \tag{1.21}$$

1.3.1 Fourier Transform:

Defining,

$$A(2\pi m f_s) = \frac{c(2\pi m f_s)}{f_s} \tag{1.22}$$

and rewriting, eq. 1.20 as

$$f(x) = \frac{1}{2\pi} \sum_{m=-\infty}^{\infty} \frac{c(2\pi m f_s)}{f_s} exp(i2\pi m f_s x)(2\pi f_s) \tag{1.23}$$

Observe that, the equation has been multiplied and divided by $2\pi f_s$. As discussed before, we need $f_s \to 0$, that is, we could think of it as becoming an infinitesimally small quantity df, whose multiples mdf would result in a continuous spatial frequency variable, say f. Then, $2\pi f$ would be the continuous wave-vector k and $2\pi df$, the infinitesimal change in k, is written as dk. The discrete sum in eq. 1.23 can be replaced by a continuous integral, to result in the Fourier transform as

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} A(k) e^{ikx} dk \tag{1.24}$$

and keeping in mind that $\frac{1}{df} \to \infty$, the co-efficients A(k) would be given by the Inverse Fourier transform

$$A(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ikx} dx \tag{1.25}$$

Note that the factor $\frac{1}{2\pi}$ in eq. 1.23 is distributed between the Fourier and Inverse Fourier transform expressions as $\frac{1}{\sqrt{2\pi}}$, so that the Parsavel's relation still holds [ref].
Now, we shall consider the formation of wave-packet for two typical amplitude functions A(k). One in which the co-efficients remain constant with in the interval of width Δk and the other in which they take the magnitudes as that of a Gaussian function, with its peak centered at a value k_0 and uncertainity Δk (width due to standard deviation).

1.3.2 Rectangular amplitude function

The rectangular function is defined as

$$A(k) = \begin{cases} \frac{1}{\sqrt{\Delta k}}, & k_0 - \frac{\Delta k}{2} \leq k_0 + \frac{\Delta k}{2} \\ 0, & Otherwise \end{cases} \tag{1.26}$$

The wave-function $\psi(x, 0)$ at time t = 0, is nothing but the stationary state function $\phi(x)$, which is obtained as Fourier transform of A(k) substituted in eq. 1.24 as

$$\psi(x, 0) = \phi(x) = \frac{1}{\sqrt{2\pi}} \int_{k_0 - \frac{\Delta k}{2}}^{k_0 + \frac{\Delta k}{2}} \frac{1}{\sqrt{\Delta k}} e^{ikx} dk \tag{1.27}$$

Integrating, we obtain

$$\psi(x, 0) = \frac{1}{\sqrt{2\pi\Delta k}} e^{ik_0 x} \frac{2}{x} sin\left(x \frac{\Delta k}{2}\right) \tag{1.28}$$

and the probability density as

$$P(x,0) = |\psi(x,0)|^2 = \psi^*(x,0) * \psi(x,0) = \frac{1}{2\pi\Delta k}\frac{4}{x^2} sin^2\left(x\frac{\Delta k}{2}\right) \tag{1.29}$$

The analytical functions A(k) for rectangular function and corresponding probability density using the Fourier transform to determine the wave function are plotted in Fig. [].

1.3.3 Gaussian amplitude function

The amplitudes could be chosen to be dictated by a Gaussian function, as

$$A(k) = \frac{1}{\sqrt{(2\pi(\Delta k)^2}} exp\left(-\frac{(k-k_0)^2}{2\Delta k^2}\right) \tag{1.30}$$

It is a bell shaped function in variable k defined over the interval (-∞, ∞). k_0 is the mean k value at which it peaks and symmetrically goes down to zero on either side with a standard deviation (rms value) of Δk, also referred to as width of the Gaussian function. Here, let us choose $k_0 = 0$, for sake of simplicity. The corresponding wavefunction $\psi(x,0)$ is obtained by taking the fourier transform as follows:

$$\psi(x,0) = \frac{1}{\sqrt{2\pi}}\frac{1}{\sqrt{(2\pi(\Delta k)^2}}\int_{-\infty}^{\infty} e^{-\frac{k^2}{2\Delta k^2}} e^{ikx} dk \tag{1.31}$$

$$= \frac{1}{\sqrt{2\pi}}\frac{1}{\sqrt{(2\pi(\Delta k)^2}}\int_{-\infty}^{\infty} e^{-\frac{1}{2\Delta k^2}\left(k^2 - 2ikx\Delta k^2\right)} dk \tag{1.32}$$

Completing the square by adding and subtracting $(ix\Delta k^2)^2$ term inside the brackets, and taking out the subtracted term outside the integral, we obtain,

$$\psi(x,0) = \frac{1}{\sqrt{2\pi}}\frac{e^{-\frac{x^2\Delta k^2}{2}}}{\sqrt{(2\pi(\Delta k)^2}}\int_{-\infty}^{\infty} e^{-\frac{1}{2\Delta k^2}\left(k - ix\Delta k^2\right)^2} dk \tag{1.33}$$

Using the standard integral,

$$\int_{-\infty}^{\infty} e^{-\alpha(x-x_0)^2} dx = \sqrt{\frac{\pi}{\alpha}} \tag{1.34}$$

the wave-function is given by

$$\psi(x,0) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2(1\Delta k)^2}} \tag{1.35}$$

A gaussian function in x-space with width $\Delta x = \frac{1}{\Delta k}$ thus satisfies the Heisenberg uncertainity principle $\Delta x \Delta k = 1$.

1.3.4 Preparation of System for Numerical Solution:

The Fourier transform (See Appendix) is implemented very efficiently and is available in Python as a single command, *fft (fast fourier transform)*. Fourier transform has a complex kernel of transformation and hence the output of the transform is in general complex. The transform has both negative and positive frequencies present in it. One should understand the *fft command* before using it.

1. Firstly it produces output array of complex numbers.
2. Secondly, since the array index starts from 0 in Python and has no negative indices, the fft algorithm stores the negative frequencies after it completes the storage of the positive frequencies.
3. So, the negative frequencies which are supposed to come before the first index of the array, actually appear from N down to $N/2$ if there are N points in the fft.
4. To obtain output in correct order, one needs to use *fftshift* command in Python
5. The value of N should be preferably a power of 2, that is $N = 2^n$, where n is an integer for very efficient implementation of *fft* command.

Algorithm: Now, we are ready to write the algorithm for implementation in Scilab/Python.

1. Let $k = 2\pi f_s$ representing the wave-vectors be chosen in the interval (-1,1) with N points equally spaced at every dk units, say 0.01. That is, with sampling rate as SR = 1/dk = 100.
2. Define the Rectangular/Gaussian function $A(k)$ at these k points.
3. The fft command on $A(k)$ would produce an output array, say '*phi*' of N values, whose samples are $2 * \pi * SR/N$ apart and are complex numbers.
4. So, the x-vector with N values centred about zero would be obtained as $(-\frac{N}{2} : \frac{N}{2} - 1) * 2 * \pi * \frac{SR}{N}$.
5. The absolute value of 'phi' is obtained using 'abs' command to see the magnitude spectrum and fftshift command is used to swap the values of the array about the mid-point.
6. Plot the Rectangular/Gaussian function A(k) w.r.t. k and the fft-shifted magnitude spectrum phi w.r.t. x.

1.3.5 Numerical Implementation in Python

The program for a wave-packet with rectangular amplitude function of width w = deltak is given below:

```
from numpy.fft import fft,fftshift
import matplotlib.pyplot as plt
def rectangular(w):
    #written by Dr O.S.K.S. Sastri
    #w is the width of the rectangular amplitude function
    #RoI is chosen as interval [-1,1]
    k = linspace(-1,1,100) #Discretisation of variable k in spatial frequency domain
    N = len(k); #Number of points corresponding to width of RoI, here 2 units
    M = round(w*N/2);#Number of points corresponding to width w
    A = zeros(N);  #A contains an array of N zeros
    A[int(N/2-M/2):int(N/2+M/2)] = 1/sqrt(w); #
    x =linspace(-int((N/2)*2*pi*100/N),int(((N-1)/2)*2*pi*100/N),100); #corresponding x-val
    phi = fftshift(abs(fft(A)));
    subplot(2,1,1)
    plt.xlabel("k")
    plt.ylabel("a(k)")
    plt.title("Amplitude function")
    plt.plot(k,A)
    plt.tight_layout(pad=3.0)
    subplot(2,1,2)
    plt.xlabel("x")
    plt.ylabel("y")
    plt.title("Sinc wave-packet")
    plt.plot(x,phi)
```

The program for a wave-packet with an amplitude function as Gaussian of width w is given below:

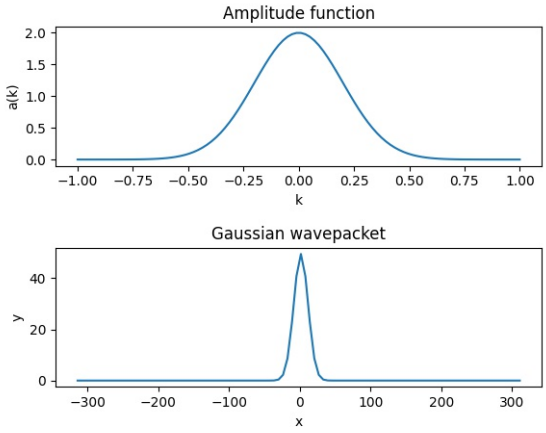
```
from numpy.fft import fft,fft2, fftshift
import matplotlib.pyplot as plt
def gaussian(w):
    #written by Dr O.S.K.S. Sastri
    #w is the width of the gaussian amplitude function
k = linspace(-1,1,100); #Discretisation of variable k in frequency domain
N = len(k);
A = exp(-k**2/(2*w**2))/sqrt(2*pi*w**2);#Definition of Gaussian amplitude function
x = linspace(-int(N/2)*2*pi*100/N,int(((N-1)/2)*2*pi*100/N),100); #0corresponding x-values
phi = fftshift(abs(fft(A)))
subplot(2,1,1)
```

```
plt.xlabel("k")
plt.ylabel("a(k)")
plt.title("Amplitude function")
plt.plot(k,A)
plt.tight_layout(pad=3.0)
subplot(2,1,2);
plt.xlabel("x")
plt.ylabel("y")
plt.title("Gaussian wavepacket")
plt.plot(x,phi)
```

1.3.6 Simulation and Discussion of Results:

The output of running this program is shown in Fig.2.5. We observe that the Fourier transform of a Gaussian function A(k) is also a Gaussian function phi(x). Another interesting feature to observe is that the widths of the Gaussian functions in the spatial and frequency domains are inversely related. DeBroglie relation relates wave-vector k to the momentum p as $\hbar$k and so dk is directly reflective of dp. dx is the width of the Gaussian y(x) and is the dispersion or uncertainty in position x of the particle. dk is the width of the Guassian a(k) and dp=$\hbar dk$ is the uncertainty in p. The product of the two widths is dx.dk=1 which leads to dx.dp=$\hbar$. This proves that the uncertainty product is minimum for a Gaussian wave-packet.

Figure 1.4: Wavepacket for a Gaussian amplitude function



1.3.7 2-D Gaussian Wave-packet

Just to showcase some interesting 3-D plotting features of Python, let us build a 2D Gaussian wave-packet by choosing 2-D amplitude function as a Gaussian. Here, is the program:

```
from numpy.fft import fft,fft2, fftshift
import matplotlib.pyplot as plt
def gaussian2D(w):
    #written by Dr O.S.K.S. Sastri
    #w is the width of the gaussian amplitude function
n1 = linspace(-1,1,100);
n2 = linspace(-1,1,100)
[k1,k2] = meshgrid(n1,n2);
a = exp(-k1**2/(2*w**2))* exp(-k2**2/(2*w**2));
y = fftshift(abs(fft2(a)));
fig = figure(1)
ax = fig.gca(projection='3d')
surf = ax.plot_surface(k1,k2,a)
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_zlabel("z")
ax.set_title("Gaussian wavepacket")
fig1  =figure(2)
ax = fig1.gca(projection='3d')
surf = ax.plot_surface(k1,k2,y)
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_zlabel("z")
ax.set_title("Gaussian amplitude function")
```

The output is given below:

Figure 1.5: 2D Gaussians using 3D plots