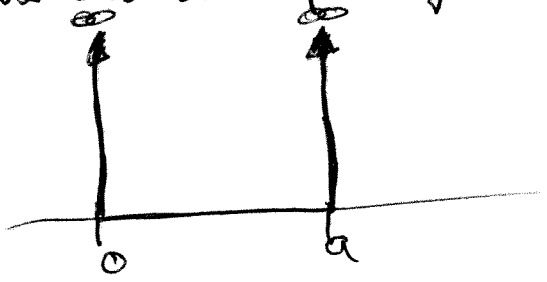
Quantum Mechanics

Problem 4.7, ~~4.18~~ 4.19

(4.7) A particle of mass m is moving in an infinite potential well

$$V(x) = \begin{cases} V_0 & 0 < x < a \\ \infty & \text{elsewhere} \end{cases}$$

a) Solve the Schrödinger equation and find the energy levels and corresponding normalized wavefunctions.



$$-\frac{\hbar^2}{2m}\frac{d^2}{dx^2}\psi(x) = E\psi(x)$$

$$\psi''(x) = -\frac{2mE}{\hbar^2}\psi(x) \qquad -k^2$$

$$\psi''(x) = -k^2\psi(x)$$

Then $\psi(x) = A\sin kx + B\cos kx$

Now apply the boundary conditions:

$$\psi(0) = 0 = \psi(a)$$

$\psi(0) = B = 0$ makes B go away

$\psi(a) = A\sin ka = 0$ makes $ka = n\pi$

So $\psi(x) = A\sin\left(\frac{n\pi x}{a}\right)$

To normalize we want $\langle\psi|\psi\rangle = 0$

So $\int_0^a |A|^2 \sin^2\left(\frac{n\pi x}{a}\right) dx = 1$

$$|A|^2 \int_0^a \sin^2\left(\frac{n\pi x}{a}\right) dx = |A|^2 \int_0^a \left[\frac{1-\cos\left(\frac{2n\pi x}{a}\right)}{2}\right] dx$$

$$= |A|^2 \cdot \left(\frac{a}{2} - 0\right) = 1$$

$$|A| = \sqrt{\frac{2}{a}}$$

$$\therefore \psi(x) = \sqrt{\frac{2}{a}} \sin\left(\frac{n\pi x}{a}\right)$$ for the asymmetric well

Energy levels satisfy

$$-\frac{2mE}{\hbar^2} = -k^2 = -\left(\frac{n\pi}{a}\right)^2$$

$$E = \frac{\hbar^2}{2m} \cdot \left(\frac{n\pi}{a}\right)^2$$

(b) Calculate $\langle \hat{x} \rangle_5$ $\langle \hat{p} \rangle_5$ $\langle \hat{x}^2 \rangle_5$ $\langle \hat{p}^2 \rangle_5$

$\langle \hat{x} \rangle_n \equiv \langle \psi_n | \hat{x} | \psi_n \rangle$ so

$$\langle \hat{x} \rangle_5 = \langle \psi_5 | \hat{x} | \psi_5 \rangle = \int_0^a \psi_5^*(x) \, x \, \psi_5(x) dx$$

$$= \frac{2}{a} \int_0^a x \sin^2\left(\frac{5\pi}{a}\right) dx$$

$\langle \hat{x} \rangle_n = \frac{a}{2}$ $\Rightarrow \langle \hat{x} \rangle_5 = \frac{a}{2}$

$\langle \hat{x}^2 \rangle_n = \frac{a^2}{3} - \frac{a^2}{2n^2\pi^2}$ $\Rightarrow \langle \hat{x}^2 \rangle_5 = \frac{a^2}{3} - \frac{a^2}{2 \cdot 25 \cdot \pi^2} = a^2\left(\frac{1}{3} - \frac{1}{50\pi^2}\right)$

$\langle \hat{p}^2 \rangle_n = \frac{n^2\pi^2\hbar^2}{a^2}$ $\Rightarrow \langle \hat{p}^2 \rangle_5 = \frac{25\pi^2\hbar^2}{a^2}$

$\langle \hat{p} \rangle_n = 0$

HW5 computer problem: modelling a dye molecule

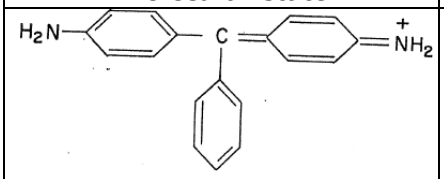
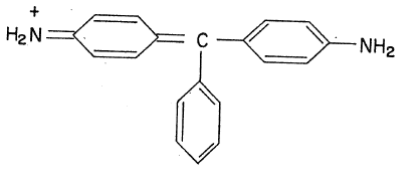
Objectives
By the end of this lab you should:

- Know how to manipulate complex numbers and matrices.

Programming environment: Spyder. To implement this, do the following:

- Go to the start menu icon and click it.
- Select **All Programs**
- Select **Anaconda 64-bit**
- Select **Spyder** (or whatever else you like, but this is what is on the computer in HA317)

Background: The a dye magenta has two molecular states as shown in the figure below, taken from Feynmann Lectures in Physics, Vol. 3, p. 10-12. In quantum physics, states can be represented as matrices, as shown.

Molecular state	Representation
	$\begin{bmatrix}1\\0\end{bmatrix}$
	$\begin{bmatrix}0\\1\end{bmatrix}$

Suppose the molecule is initially in state :

$$\vec{\psi}_{\text{initial}} = \begin{bmatrix}1\\0\end{bmatrix}$$

Suppose the Hamiltonian matrix (which determines how the molecular states change with time) is:

$$H = \begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}$$

It has eigenvectors we will designate:

$$v_1$$
$$v_2$$

and eigenvalues:

$$E_1$$
$$E_2.$$

The postulates of quantum physics state that the wavefunction at any time t for this system is:

$$\vec{\psi}(t) = \left(\vec{v}_1^{\,*} \cdot \vec{\psi}_{\text{inital}} \exp(-jE_1 t)\right)\vec{v}_1 + \left(\vec{v}_2^{\,*} \cdot \vec{\psi}_{\text{inital}} \exp(-jE_2 t)\right)\vec{v}_2$$

……..(1)

Note that $\left(\vec{v}_1^{\,*} \cdot \vec{\psi}_{\text{inital}} \exp(-jE_1 t)\right)\vec{v}_1$ means:

the dot product of the complex conjugate of $\vec{v}_1$ and $\vec{\psi}_{\text{inital}}$ times the exponential function of $-jE_1 t$. This will be a complex scalar, which then multiplies $\vec{v}_1$. Here $j = \sqrt{-1}$ and $\hbar$ has been set to one for simplicity.

The postulates further say that if you observe the system at time t, the probability of the wavefunction collapsing into the final state $\vec{\psi}_{\text{final}}$ is:

$$P = |\vec{\psi}_{\text{final}}^{\,*} \cdot \vec{\psi}(t)|^2 \text{ ……..(2)}$$

We will determine the probability that collapsing the wavefunction after time t gives the same state as the intial state:

$$\vec{\psi}_{\text{final}} = \vec{\psi}_{\text{initial}}$$

Part1: creating normalized wavefunctions
If it has been a while since you worked with Python, a quick refresher on complex numbers is here.

In quantum physics, states are vectors of complex numbers that are "normalized" to have unit length. You will write a function definition that:

- Normalizes a vector $\vec{x}$ using

$$\frac{\vec{x}}{\sqrt{\vec{x}^{\,*} \cdot \vec{x}}}$$

Some commands you may need:

from numpy import dot, conjugate	*dot* allows matrix multiplication and the dot product. *conjugate* takes the complex conjugate of a vector.
from numpy.linalg import eigh	The command *eigh* determines eigenvectors and eigen values for NxN arrays that are Hermitian.
from cmath import exp,sqrt	cmath is package that has functions that can take complex numbers as arguments.
def Normalize(x): *your code here* return	This is a function you will write to normalize the vector. It takes a vector x and passes back the normalized vector.

- Write the normalization routine and check it with the vector $\vec{x} = \begin{bmatrix} 1 \\ j \end{bmatrix}$. The function definition should compute $\vec{x} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ j \end{bmatrix}$

Part2: determining the probability as a function of time

A quick refresher on eigenvectors and eigenvalues in Python is here.

You will now calculate probability as a function of time. Here are some commands:

def Psi(valH,vectH,PsiInitial,t): *your code* return *your code*	This will calculate $\vec{\psi}(t)$ using Eq.(1).
valH[0] vectH[:,0]	Eigenvalue with index 0 Eigenvector corresponding to eigenvalue valH[0]
1j	This is how you indicate $\sqrt{-1}$. If you forget the 1 python will complain.

- Create a function Psi(valH,vectH,PsiInitial,t) that takes the eigenvector array *vectH*, the eigenvalue array *valH*, the normalized initial wavefunction PsiInitial, and the time t. This should calculate $\vec{\psi}(t)$ based on Eq.(1), and return its value.
-

H=?	You figure out how to enter the H matrix. This does not change with time.
val,vect=eigh(H)	This returns the eigenvalues and eigenvectors of H.
PsiInitial=?	You figure out how to specify it.
abs(z)	This is how you calculate the modulus of a complex number $\|z\|$

- Create a loop that creates time values from times t=0. to 10. for N=100. Here are some other commands you may need. Calculate the probability from Eq.(2) for each time value.

for t in arange(t0,tf,dt):	This for loop changes the value of variable t from t0 to tf in steps of dt
PsiFinal=?	You figure this out. This calls the function you defined earlier and calculates state at each time t.
x.append(t)	Inside the for loop, this command builds an array called x by adding the new t value each iteration. Note: before the loop you will need to declare x=[ ] to create an empty array x.

- Plot the probability as a function of time. You should find a probability that oscillates in time, suggesting that the dye molecule swaps back and forth between states periodically. The frequency corresponds to the color of light that magenta absorbs.

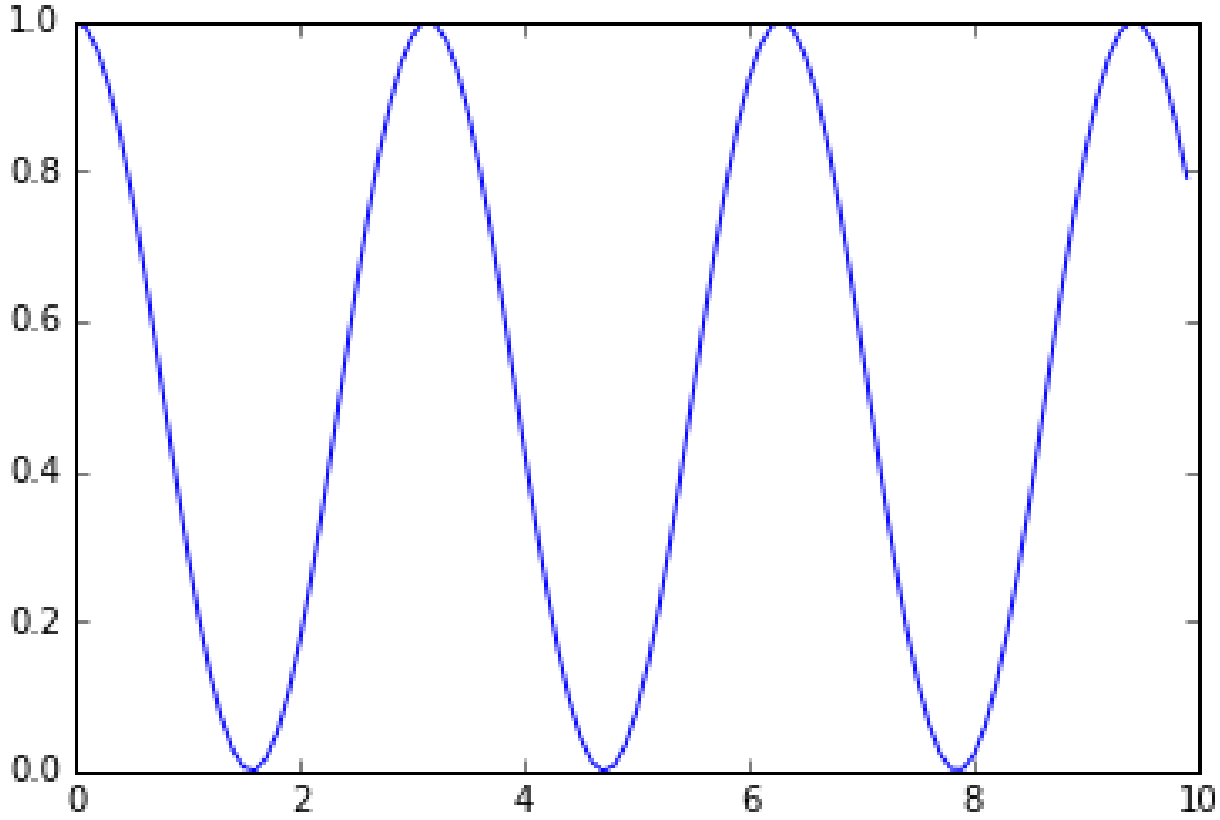
plot(x,y) show()	Here x and y represent one dimensional arrays. These commands will generate a plot of y versus x

- Checks:
  - If the initial state is an eigenvector of H, every H measurement will always give the same state. Thus the probability will always be 100%. Change the code and check this.
  - If you start in one eigenstate, make measurements of H, you never get the other eigenstate. Thus the probability of starting and ending in different eigenstates of H is zero. Change the code and check this.
- Discuss your checks in the comments of your code.

What to turn in:

- Hardcopy of program with your homework in class.
- Submit the python programs in dropbox on D2L by 11:50am Wednesday. Make sure:
  - The filename is HW04*YourName*.py
  - You save a copy on a flash drive or email the program to yourself.
  - Inside your program, the top line is: #HW04 Your name
  - You place comments in your code indicating what you are doing.

```
'''
This program plots the probability amplitude
of finding the system in state |psi1>, the first
energy eigenstate of the Hamiltonian, after a measurment
at time t.
This is for a 2x2 Hamiltonian
'''
from numpy import array,zeros,dot,conjugate,arange
from numpy.linalg import eigh
from cmath import exp,sqrt
from pylab import show, plot

def Normalize(x):
   CSquared=dot(conjugate(x),x)
   return x/sqrt(CSquared)

#This calculates |psi(t)> at any time
#by adding all the eigenstates of H
#and their corresponding exponentials of time
def Psi(valH,vectH,NormPsi,t):
   psi=exp(-1.j*valH[0]*t)*dot(conjugate(vectH[:,0]),NormPsi)*vectH[:,0]+\
         exp(-1.j*valH[1]*t)*dot(conjugate(vectH[:,1]),NormPsi)*vectH[:,1]
   return psi  #This is the |psi(t)>

t0=0.
tf=10.
N=100
dt=(tf-t0)/N

#This is the Hamiltonian
H=array([[1,1],[1,1]],complex)

#This calculates the eigenvalues and vectors
valH,vectH=eigh(H)

#This normalizes the initial state
PsiInitial=array([1,0],complex)  #This is the initial state
PsiFinal=array([1,0],complex)  #This is the final state

PsiInitial=Normalize(PsiInitial)
PsiFinal=Normalize(PsiFinal)

#This will be the time array.  Start it empty.
x=[]
#This will be the probability array.  Start it empty.
y=[]

for t in arange(t0,tf,dt):
   PsiOfT=Psi(valH,vectH,PsiInitial,t)
   probability=abs(dot(conjugate(PsiFinal),PsiOfT))**2
   x.append(t)
   y.append(probability)

plot(x,y)  #This plots probability versus time

show()
```


1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10