Investigating the Behavior of Electrons in Different Quantum Systems

Christopher Seymour

May 10, 2011

The focus of this project is to investigate the physical behavior of electrons in different quantum systems. There are very few systems that can be solved analytically. Many systems of interest encountered in research and teaching require numerical methods of solutions. We will study the use of stable, norm-preserving method to solve the time-dependent Schrödinger equation to simulate the behavior of quantum systems, including particles in the infinite potential well, finite square well or barrier, Dirac delta function, quantum harmonic oscillator, and hydrogen atoms. Interactions of these systems with laser fields will be simulated. Physical properties such as reaction rates will be investigated.

1 Problem

I am addressing a problem that involves numerically solving the time-dependant Schrödinger Wave Equation, which is a second order partial differential equation whose solutions tell us information about how the behavior of the particle unfolds with time.

$$-\frac{i\hbar\partial\Psi}{\partial t} = \hat{H}\Psi \tag{1a}$$

$$i\hbar\frac{\partial\Psi}{\partial t} = \frac{-\hbar^2}{2m}\nabla^2\Psi + V\Psi \tag{1b}$$

$$\hat{H} = \frac{-\hbar^2}{2m}\nabla^2 + V \tag{2}$$

Solutions to this Schrödinger equation, called "wave functions," are functions of position and time. the wave function Ψ is defined to describe the probability density and helps describe the behavior of the particle that the wave equation is describing. Equation (2) is the Hamiltonian energy operator.

The probability of finding the particle between a and d, at time t is

$$\int_a^b |\Psi|^2 \, \mathrm{d}x = \int_a^b \Psi^* \Psi \, \mathrm{d}x \tag{3}$$

The wave function has the properties:

- Ψ must be a solution to the Schrödinger equation
- Ψ(x, t) must be continuous and differentiable
- $\frac{d\Psi}{dt}$ must be continuous and differentiable

The probability density must be normalized.

$$\int_{-\infty}^{\infty} |\Psi|^2 \, \mathrm{d}x = 1 \tag{4}$$

$$\tag{5}$$

I am trying to solve this equation in different potentials, for example, the infinite potential well and the harmonic oscillator or Dirac-delta function. The potential "V(x)" in equation (1b) refers to the potential energy function of the particle that is in question. The potential function is essentially what governs the behavior of the particle in the system.

The focus of my project is to learn and implement numerical integration techniques in order to solve the differential equations at hand. In particular I am trying to solve the time-dependant Schrödinger equation (1b) also called "the" Schrödinger equation. Before I will be able to solve these more difficult problems, I need to build up the necessary machinery. In order to do this I will solve problems that are more simple and add the information I gain to what I have already learned.

2 Techniques

I am using numerical integration techniques to address the problem. To start I will use a separation of variable method to solve the Schrödinger equation.

In order to solve (1b), I will rewrite the wave function in the form

$$\Psi(x,t) = \psi(x)\phi(t) \tag{6}$$

Yielding the set of equations

$$\frac{d\phi}{dt} = -\frac{iE}{\hbar}\phi \tag{7a}$$

$$\frac{-\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2} + V\Psi = E\Psi \tag{7b}$$

Which allow us to rewrite solutions to (1b), equation (6), as

$$\Psi(x,t) = \psi(x)e^{-iEt/\hbar} \tag{8}$$

Where $\psi(x)$ is a solution to the time-*in*dendant Schrödinger equation.

$$\hat{H}\Psi = E\Psi \tag{9a}$$

$$\frac{-\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2} + V\Psi = E\Psi \tag{9b}$$

The time-indendant Schrödinger equation is an energy eigenvalue problem. Determinate states

$$\Psi(x,t) = \psi(x)e^{-iEt/\hbar}$$

called stationary states are eigenfunctions of (9b). Two or more *eigenfunctions* that share the same *eigenvalue* are degenerate states.

Unfortunately most of the more difficult situations can not be solved analytically. Even if an analytic solution is achievable it is often still useful to examine the results. So I will also solve some simple situations to get a feel for the algorithm and get an idea of reasonable values for certain parameters and results.

Then I will numerically integrate using the leap frog method (which I also saw called the position Verlet method).

$$x_{1/2} = x_0 + v_0 * \frac{h}{2} \tag{10a}$$

$$v_1 = v_0 + x_{1/2} * h \tag{10b}$$

$$x_1 = x_{1/2} + v_0 * \frac{h}{2} \tag{10c}$$

In order to solve the Schrödinger equation we need a method that will conserve energy as time passes, which makes the leap frog algorithm a good choice to solve the partial differential equations. In order to check that the method is area preserving, the determinant of the Jacobian transformation matrix must be equal to 1.

$$J = \begin{vmatrix} \frac{\partial x'}{\partial x} & \frac{\partial x'}{\partial v} \\ \frac{\partial v'}{\partial x} & \frac{\partial v'}{\partial v} \end{vmatrix} \tag{11}$$

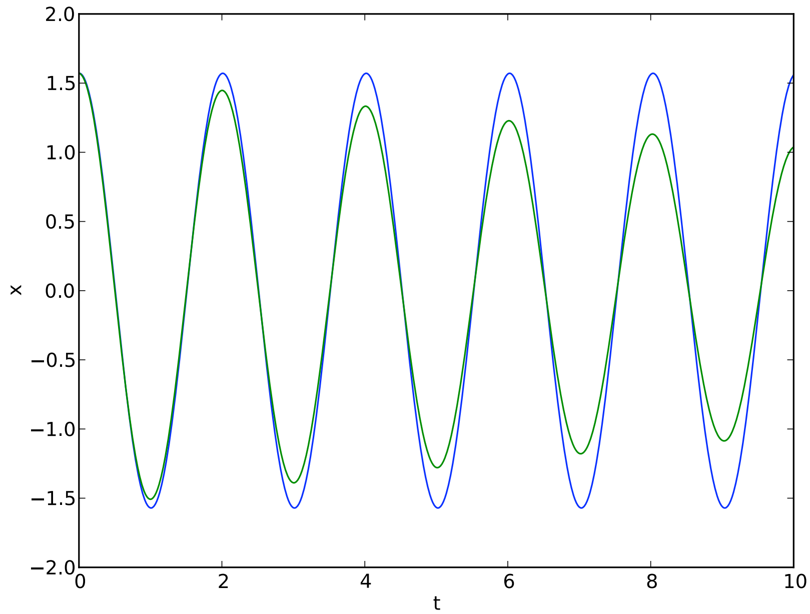


Figure 1: Simple Pendulum: Euler vs. Runga Kutta 4

3 Process

Over the course of the semester I have progressed quite a bit on my project. At first, Dr. Wang thought it would be good for me to learn about python. So I started learning python by writing simple codes and reading the necessary documentation. Once I learned a good deal about python I was able to start writing some real codes and tinkering around with them. I started off by writing a sample Eulers method algorithm and testing it for some simple cases like the classical harmonic oscillator. I continued to write more code including a Runge Kutta 2 and a Rungga Kutta 4 schemes and of course the Leap Frog method. I also worked toward learning about plotting my results neatly and in the manner that I want using the matplotlib software. I continued to mess around with python until I was able to create the graphs and visuals necessary for the presentations. After I have checked to make sure all of my algorithms are in working order I checked them all against each other to check how they compared. Below is a graph comparing the different methods numerical integration I mentioned.

After checking and comparing the accuracy of the different methods, I cal-

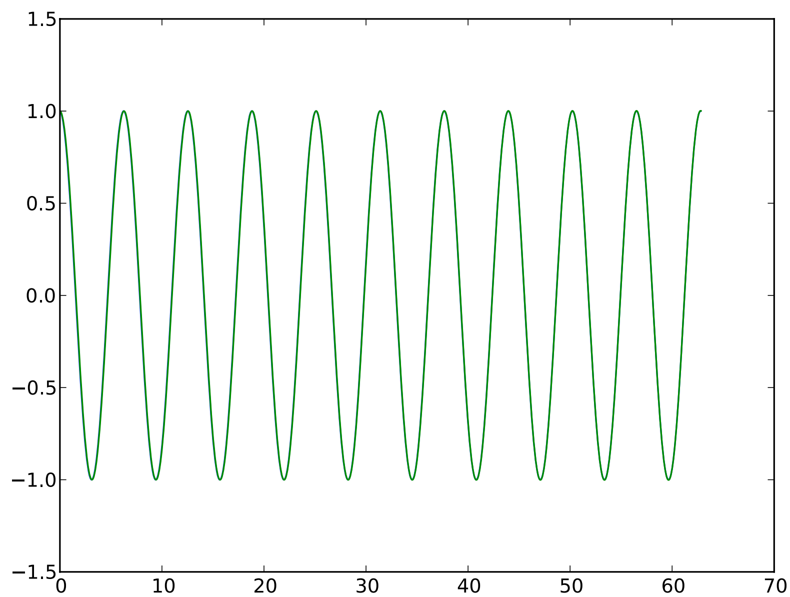
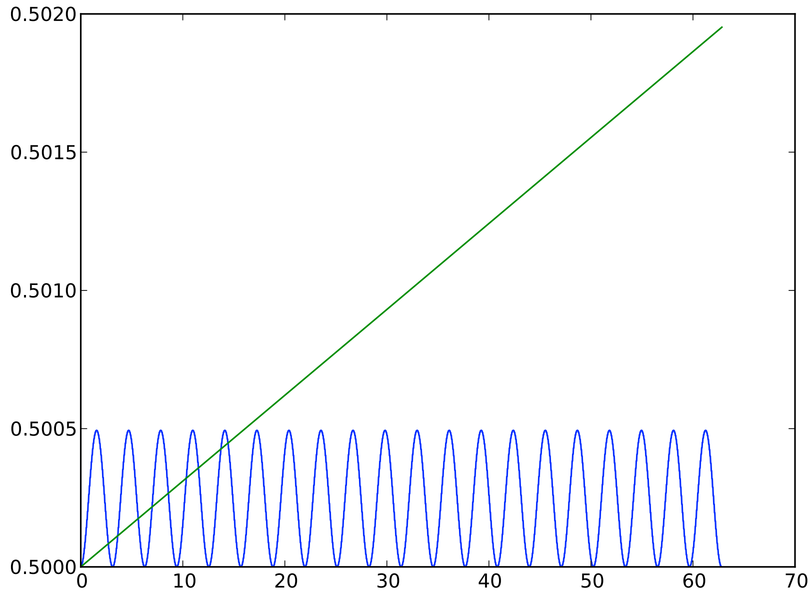


Figure 2: Harmonic Oscillator: Leap Frog vs. Runga Kutta 2

culated the Jacobian transformation matrix and found the determinant for each step of the leap frog method in order to be sure that the algorithm is symplectic. I have also compared the energy of the two systems since that is the quantity that we are interested in conserving. In a dynamical system the leap frog method is a great choice because it conserves angular momentum and is time reversible. Below is a graph comparing the energy of the harmonic oscillator for the leap frog method vs the Runge Kutta 2 method. Both the leap frog and Runga Kutta 2 method return the same solution. This is somewhat expected since they are both second order methods. However, comparing the energy of the two systems.



As you can see by the graph the energy of the leap frog method oscillates around it initial value while the energy of the Runge Kutta 2 method blows up rather quickly.

Along with learning python I have read a great deal about quantum mechanics, which I think has been very beneficial in my understanding of the subject and has been very helpful in dealing with the problem that I am trying to solve.

I have also spent a lot time learning LaTeX, and even more implementing the techniques I have learned to make presentations and posters for the various venues that have required them.

4 Accomplishments

My project seemed to be a lot more difficult for me than i had originally thought it would be. I thought it would be easier than it was since I have taken a course on differential equations and seen Eulers method once before. I also pride myself at being pretty decent at calculus so I figure that would help a lot. It turns out that the only calculus really needed was to set up the equations so that they could be numerically integrated. With that said it took me a lot of work just to get python up and running on my computer without a hitch since matplotlib was being quite finicky. After I had python installed and everything I spent the first three weeks to a month just messing around with python trying to learn the different commands and lots of the different useful techniques. Once I had python running smoothly and I had just started to implement my algorithms and get some results, however trivial, it was time to begin work on my first presentation. I spent a good amount of time trying to get my equations to work out in LaTeX to find out that I did not know enough about LaTeX to actually do all that I wanted. So I ended up doing my first presentation in powerpoint, which looking back was extremely unprofessional, which made it a great learning experience. That mistake will not happen again because after that presentation I spent the time to figure out the subtleties of TeX I had been missing.

After the first presentation I continued to work with python, writing the leap frog method code and another second order method Runge Kutta 2 in order to compare the two methods side by side. I continued to work with this until it came time to prepare a poster for the research conference at the University of Massachusetts Amherst. Preparing this poster took me probably ten to fifteen hours alone. It was a great experience, but writing the code for the entire poster, for that many equations, and custom formatting the whole thing in latex proved to be extremely taxing.

After the research conference at Amherst I had a little time to start making progress on my project. It seemed like as soon as the conference was over it was time for me to start preparing my second presentation. The second slide show was much better than the first, which I spent a bit more time working on. Since I had spent so much time working on my latex between the two presentations, I did not really have many new results to present. I think that the presentation portion was weak, but the slideshow was good so I still got something out of it.

I have taken a class on programming with C before but python is very different and in some cases I think it is easier to use since the notation is more casual. However i have noticed that the processing times are a bit slower with python since it is interpreted. Have said that, python is a great programming language and I have learned a great deal about it over the course of the semester.

5 Reflection

I have learned a great deal while participating in the Computational Science Training for Undergraduates in the Mathematical Sciences. It has been one of my best experiences at the University of Massachusetts Dartmouth. I have seen what it is like to participate in a real research environment. The knowledge I have gained is irreplaceable and I do not think that I could have learned some of the lessons I did any other way.

Lessons I have learned while working my project have been abundant since the beginning. I am not a great public speaker; at first I was quite nervous when I found out that there were going to be a lot of presentations. I soon found out that presenting your work well is one of the most important aspects of the research process. My take on it is that properly sharing the information with others is beneficial to both parties. The listener is hopefully gaining some new knowledge on a subject they already know about or being introduced to a new subject or idea they have yet to encounter. The presenter is forced to explain their project to whoever the audience might be. I found out the hard way how difficult it can be explaining my project to the different types of audiences. Information that you are very familiar with and have been working on for months is extremely difficult to explain to a layman or even someone who may know the math, but not the jargon of the particular field in question.

I have learned a tremendous amount of python code and syntax since the beginning of the course. While I have a tiny amount of programming background the differences between the interpreted nature of python and the compilation of C code were substantial.

I knew nothing about LaTeX at the beginning of the semester except that it was a really nice program used to write pdfs, I also knew that I wanted to learn about it, and learn how to use it. I was glad to start using LaTeX since I frequently have to create presentations or write papers with equations written in them and I am was tired of struggling with sloppy formatting inside word and powerpoint. My LaTeX experience will be very beneficial, writing lab reports and including diagrams will be much easier while writing lab reports.

6 Feedback

I think the CSUMS class is a great experience. It requires a lot of work and time, but in the end it is worth it for the knowledge that can be gained when the effort is put forth. The concept of the class is great, students get to research topics they are interested in so they can get a feel for what a real research environment is like, including the aspects that my not be initially thought of, like giving talks and creating posters to present.

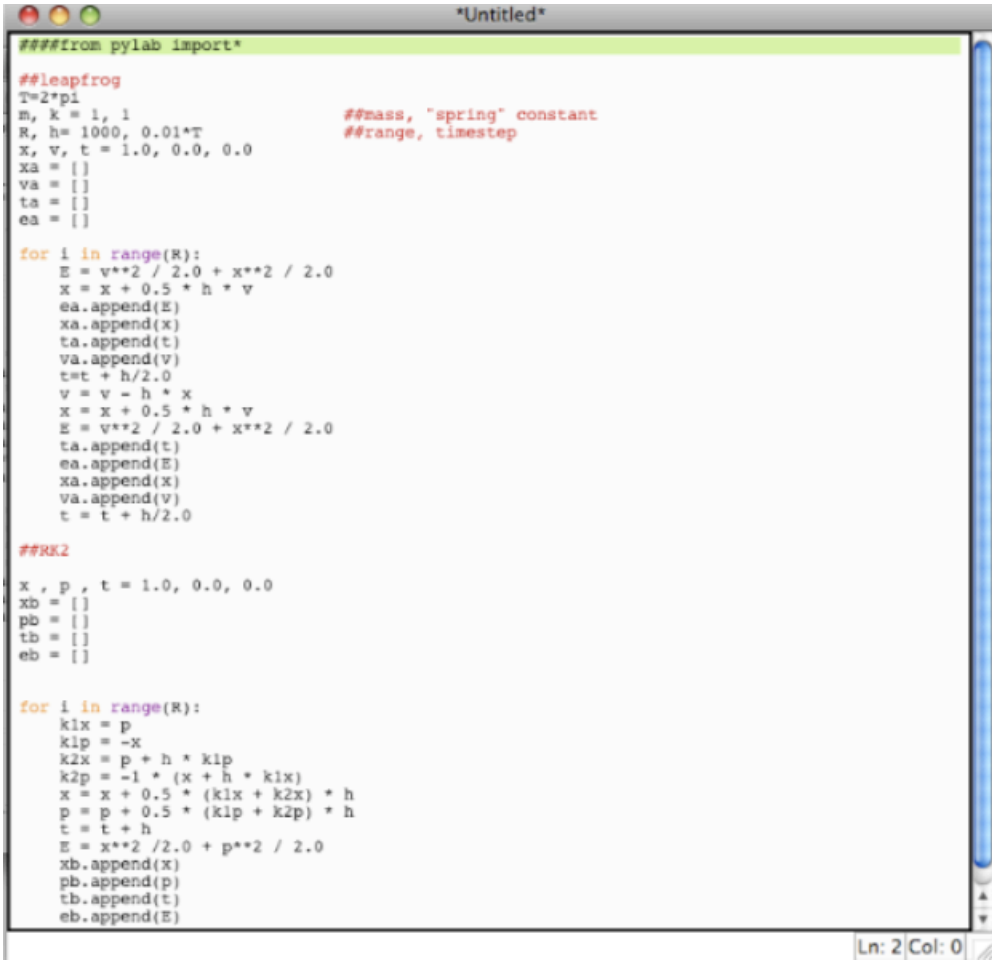
I think that the best feature of the CSUMS experience was definitely the SIAM conference in Reno. It was great to have the opportunity to attend the conference and listen to a lot of talks. Although I did not understand any of the talks completely, I was still able to pick up small bits of general information from them that simply expanded my understanding of the math and techniques being used. The worst feature of the CSUMS experience would be that it was so early in the morning. The only thing I would change would be to maybe have everyone write a couple comments during or after presentations, and give them to the presenter after to give them some feedback. To give them an idea of what they may need to bolster, students should write down one thing they explained well and one thing that they explained poorly. I think that you should continue the individual aspect because it forces students to become motivated and learn on their own in order to complete their research projects. I also think the elevator talks are a great idea. As a suggestion maybe gear one elevator talk to the layman and the next version to a slightly more advanced listener. Since explaining to different audiences is quite difficult.

7 Code

```
from pylab import*

h=0.01                                ##step
n=1000                          ##iterations
g,L=9.8,1                          ##gravity [m/s^2]/pendulum Length [m]
w0=g/L                             ##angular frequency

##Euler
x,w,t,e=pi/2,0,0,0                ##initial condition
xa=[]                              ##define vectors
wa=[]
ta=[]
ea=[]
for i in range(n):
    x=x+w*h
    w=w-w0*x*h
    t=t+h
    xa.append(x)
    wa.append(w)
    ta.append(t)

##Runge Kutta
x,w,t=pi/2,0,0
xb=[]                              ##define vectors
wb=[]
tb=[]
for i in range(n):
    a=w
    b=w-w0*x*h/2.0
    c=b-w0*x*h/2.0
    d=c-w0*x*h
    x=x+1/6.0*h*(a+b*2+c*2+d)
    w=w-w0*x*h
    t=t+h
    xb.append(x)
    wb.append(w)
    tb.append(t)

figure()
plot(ta,xa)
plot(tb,xb)
xlabel("t")
ylabel("x")
show()
```

Ln: 41 Col: 11

Euler + RK4



```
*Untitled*
####from pylab import*

##leapfrog
T=2*pi
m, k = 1, 1                              ##mass, "spring" constant
R, h= 1000, 0.01*T                       ##range, timestep
x, v, t = 1.0, 0.0, 0.0
xa = []
va = []
ta = []
ea = []

for i in range(R):
    E = v**2 / 2.0 + x**2 / 2.0
    x = x + 0.5 * h * v
    ea.append(E)
    xa.append(x)
    ta.append(t)
    va.append(v)
    t=t + h/2.0
    v = v - h * x
    x = x + 0.5 * h * v
    E = v**2 / 2.0 + x**2 / 2.0
    ta.append(t)
    ea.append(E)
    xa.append(x)
    va.append(v)
    t = t + h/2.0

##RK2

x , p , t = 1.0, 0.0, 0.0
xb = []
pb = []
tb = []
eb = []


for i in range(R):
    k1x = p
    k1p = -x
    k2x = p + h * k1p
    k2p = -1 * (x + h * k1x)
    x = x + 0.5 * (k1x + k2x) * h
    p = p + 0.5 * (k1p + k2p) * h
    t = t + h
    E = x**2 /2.0 + p**2 / 2.0
    xb.append(x)
    pb.append(p)
    tb.append(t)
    eb.append(E)
Ln: 2 Col: 0
```

Leap Frog + RK2