CSUMS First Semester

Logan Thomas
Department of Mathematics
Student– University of Massachusetts Dartmouth

Advisor: Dana Fine
Department of Mathematics
Professor– University of Massachusetts Dartmouth

May 10, 2011

Abstract

The propagator, which is a function of an initial position, a final position and a time, completely determines the quantum mechanics of a point particle in a given potential. The approximate propagators are explicitly calculated for the case of the harmonic oscillator potential. This project presents numerical investigation of the accuracy of certain approximate propagators for various values of N and t. This is feasible for this case, as the exact propagator is known.

1 Problem

the problem that I am addressing this semester is the analysis of the percent error from a propagator to the harmonic oscillator and a third order approximation of such a propagator. That was the initial problem to asses, which would involve writing some code, debugging the code, then trying for different values of variables. Then after that is squared away, The next problem to be faced is the problem of seeing why the approximation does well/badly, then try to remove elements to see how it affects the convergence of the approximation. The two equations given are equations 20 and 22 respectively:

$$\left\langle x_f \right| \exp(-iHt/\hbar) \left| x_0 \right\rangle = \left(\frac{m\omega}{2\pi i\hbar \sin \omega t} \right)^{1/2} \exp\left(\frac{im\omega}{2 \sin \omega \Delta t} [(x_f^2 + x_0^2) \cos \omega \Delta t - 2x_f x_0] \right)$$

$$\left\langle 0 \right| \exp(-iHt/\hbar) \left| 0 \right\rangle = \left(\frac{m}{2\pi i\hbar t} \right)^{1/2} \left(1 + \frac{\omega^2 t^2}{6N^2} \right)^{N/2} \prod_{k=1}^{N-1} \left[cot^2 \left(\frac{k\pi}{2N} \right) - \frac{\omega^2 t^2}{6N^2} - \frac{\omega^2 t^2}{12N^2} csc^2 \left(\frac{k\pi}{2N} \right) \right]^{-1/2}$$

The first equation (equation 20) is a propagator for the harmonic oscillator, whereas the second is an approximation in the third order, provided by Nancy Makri and William Miller.

2 Technique

To address the problem I am using my skills as a programmer to write a program to check the percent error difference between the approximation and the actual propagator. I used a few different techniques inside the code.

```
N=2;
error(14,1)=0;
while N<16
    clear M;
    clear c;
    clear Func22;
    clear Func20;
    x=pi;
    z=pi;
    k=1;
    M=((1-((x^2)/(6*(N^2))))^(N/2));
    m=((1-((z^2)/(6*(N^2))))^(N/2));
    Func20=(((x)/(sinh(x)))^(1/2));
    func20=(((z)/(sinh(z)))^(1/2));
    while k<N
        C(k,1)=((((cot((k*pi)/(2*N)))^2)+((x^2)/(6*(N^2)))+((x^2)/(12*(N^2)))*((csc((k*pi)/(2*N)))^2)
        E(k,1)=((((cot((k*pi)/(2*N)))^2)+((z^2)/(6*(N^2)))+((z^2)/(12*(N^2)))*((csc((k*pi)/(2*N)))^2)
        k=k+1;
    end
    d=prod(E);
    c=prod(C);%Multiplying all terms of teh matrix
    func22=m*d;
    Func22=M*c;
    zerror(N-1,1)=((func22-func20)/func20)*100;
    error(N-1,1)=((Func22-Func20)/Func20)*100;
    N=N+1;
end
```

To start off, the code defines a few variables, N, which is the starting value of the product series counting variable, then it calls an error(14,1)=0 which basically makes a matrix that is 14x1 and full of zeroes, we will use this later. Then it enters the while loop in which most of the code is executed. To start off with all the variables are cleared so that none effect each other upon repetitions of the loop (not sure if it would change just better safe than sorry). Then I defined $x = \pi$, x is a substitution I made in the actual equation for $x = \beta\omega\hbar$ and we are checking the special case where $x = \pi$ then it also sets $z = \pi$, basically so that I can basically run the code twice once using x and once using z, if I change the z value then I can see how the two compare. Then I have M and m which are two functions, basically the first part of equation 22 seen above, just one for x, one for z. Then I have Func 20, and func20 which are both representations of equation 20 seen above one with x and one with z. then I have another loop inside my loop that uses k as a counter for the product series, and it makes 2 matrices that are kx1 large and fills it with the different values for the terms in the product series in equation 22. Then after that loop finishes it makes 2 new variables d and c that are the products of the matrices that were made in the small loop, to give a value to the whole product series. Then It defines Func22 and func22 as M*c and m*d. then it checks the error for that particular value of N in the x and z cases then increases N by one and restarts the loop until N is 16. Though all this sounds complicated it is basically just running numbers through the equations given. This code makes it easy to do the later parts of my project such as checking how close the approximation was when the x variable was changed, in that case I would just change the z variable and compare the x and z values.

3 Process

Through this semester I have accomplished a lot for me, but on paper it doesnt look like much, But seeing as this is my first research project I am very proud of myself for my work. My first hurdle to jump was the problem about simplifying the equations to be able to be put into software. This means I needed to get rid of imaginaries or anything too small. So I did just that, through a long process that took multiple weeks to make sure I was doing everything 100 percent correct. here is the progression of the equations:

$$(1)\left\langle x_f\right| \exp(-iHt/\hbar)\left|x_0\right\rangle = \left(\frac{m\omega}{2\pi i\hbar\sin\omega t}\right)^{1/2} \exp\left(\frac{im\omega}{2\sin\omega\Delta t}[(x_f^2+x_0^2)\cos\omega\Delta t-2x_f x_0]\right)$$

$$(2)\left\langle 0\right| \exp(-iHt/\hbar)\left|0\right\rangle = \left(\frac{m}{2\pi i\hbar t}\right)^{1/2}\left(1+\frac{\omega^2t^2}{6N^2}\right)^{N/2}\prod_{k=1}^{N-1}\left[cot^2\left(\frac{k\pi}{2N}\right)-\frac{\omega^2t^2}{6N^2}-\frac{\omega^2t^2}{12N^2}csc^2\left(\frac{k\pi}{2N}\right)\right]^{-1/2}$$

$$t = -i\hbar\beta$$

$$x = \omega\hbar\beta$$

Simplification of Equation (1) in the special case where $x_f = x_0 = 0$:

$$(1)\left\langle 0\right| \exp(-iHt/\hbar)\left|0\right\rangle = \left(\frac{mx}{2\pi\beta\hbar^2\sinh x}\right)^{1/2} * 1$$

$$(2)\left\langle 0\right| \exp(-iHt/\hbar)\left|0\right\rangle = \left(\frac{m}{2\pi\hbar^2\beta}\right)^{1/2}\left(1-\frac{x^2}{6N^2}\right)^{N/2}\prod_{k=1}^{N-1}\left[cot^2\left(\frac{k\pi}{2N}\right)+\frac{x^2}{6N^2}+\frac{x^2}{12N^2}csc^2\left(\frac{k\pi}{2N}\right)\right]^{-1/2}$$

so then we can implement an error equation of:

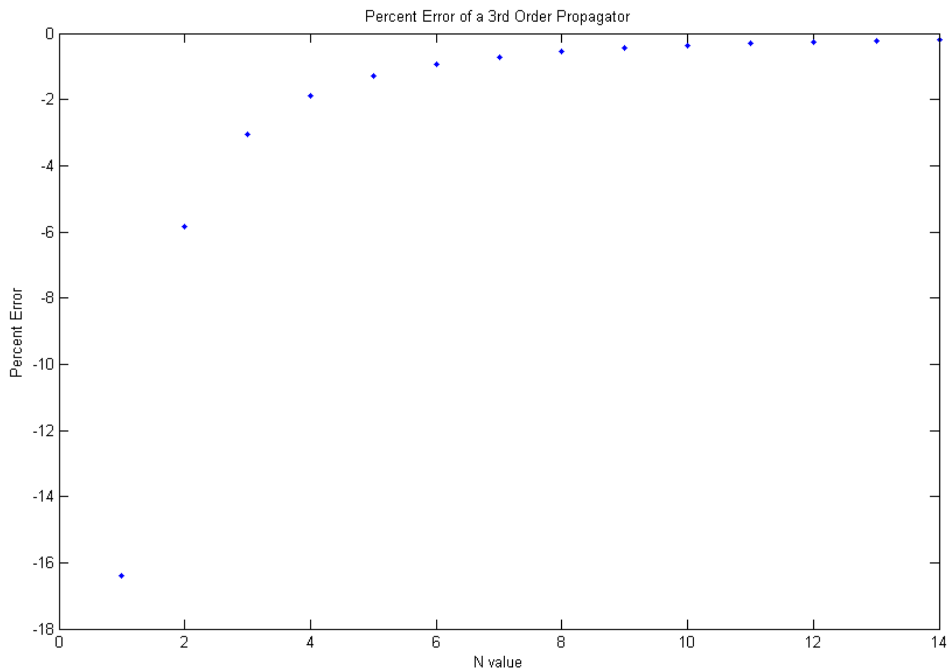
$$\frac{expirimental - actual}{actual}$$

$$\frac{\left(\left(1-\frac{x^2}{6N^2}\right)^{N/2} \prod_{k=1}^{N-1} \left[cot^2\left(\frac{k\pi}{2N}\right)+\frac{x^2}{6N^2}+\frac{x^2}{12N^2}csc^2\left(\frac{k\pi}{2N}\right)\right]^{-1/2} - \left(\frac{x}{\sinh x}\right)^{1/2}\right)}{\left(\frac{x}{\sinh x}\right)^{1/2}}$$

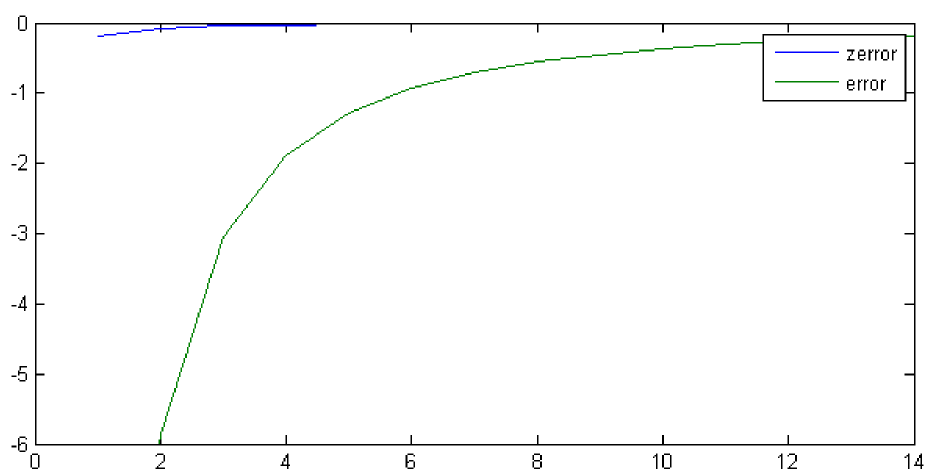
then after these equations are derrived we can put them into the code as seen in section 2. Then, when running the code, the only areas I cared to see is the error plots, or the zerror and error plots. Hopefully seeing the apparent limit as N increases go to zero, showing convergence of the two equations. After analyses of the output in the computer, I can come to the conclusion if their approximations are good or not. Then from there I can go on to the next step of my problem and go into the code and change the value of z and compare the zerror to the error to see how the changing value of x effects the results. Then the final part of the problem is to check analytically if the approximation should result in the propagator equation. This should prove very challenging, checking the limits of each aspect of the equation as N increases, We will see if that step is acheived.

4 Progress

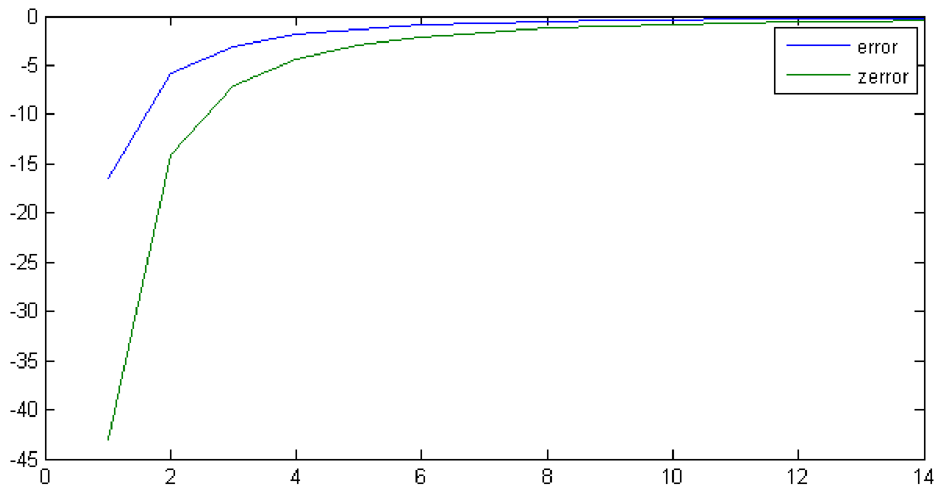
So far I have completed almost everything aside from the analyses of the limit of the second function. Through the outputs of the program I have found the basic $x = \pi$ error result comparing equation 22 to equation 20. The graph is shown here:



This shows that as N increases the approximation gets drastically closer to the actual value of the propagator. this is in the case of $x = \pi$ but when $x = 1$ we get this result,

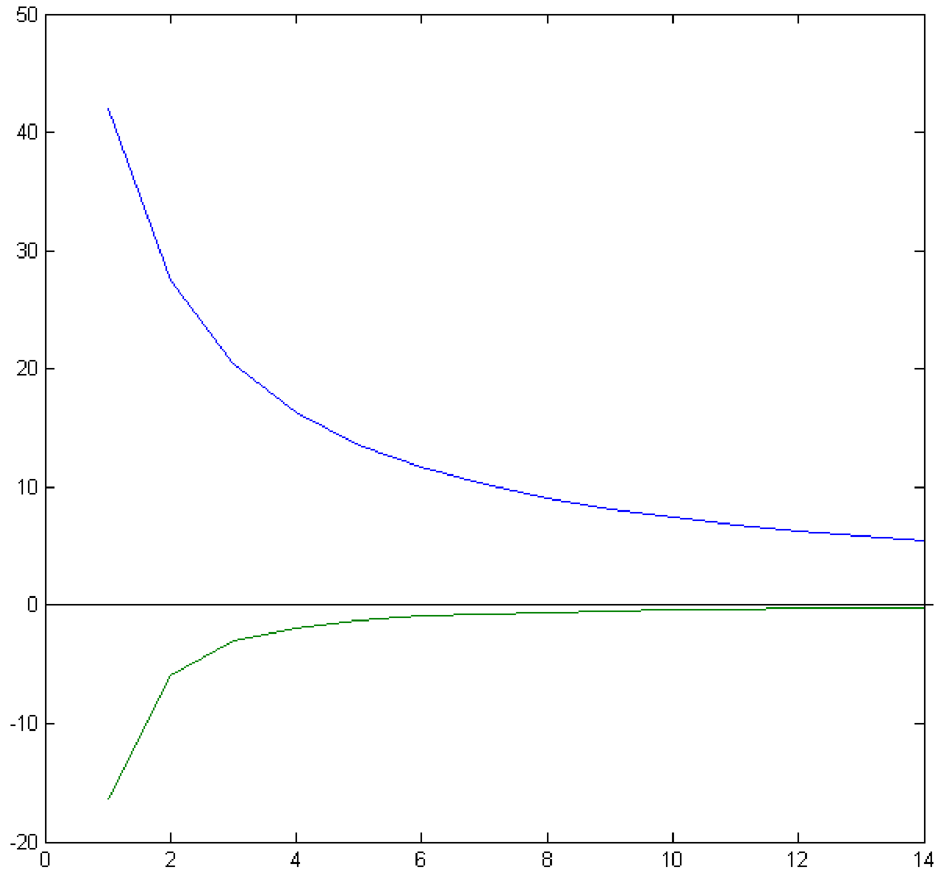


the zerror is the line that represents x=1 and the error line is the $x = \pi$ as you can see the error starts very low and goes quickly towards 0 with the x=1. this is the same for most x values below pi, so we can assume that as x decreases the convergence rate increases.



that is a graph comparing the convergence of $x = \pi$ (blue) and $x = 4$ (green) showing clearly that 4 starts farther away and converges slower than the other. From more results I have come to the observation that as x goes up, the convergence slows down and starts farther away.

The final thing I did with my code was compare the result of taking the initial term out of equation 22 leaving just the product series and seeing the effect that it had on the resulting convergence. The graph I got was:



As can be observed this did not help much, the blue line represents the new equation without the coefficient on equation 22 and the green line represents the original.

5 CSUMS

CSUMS for me was a good expirience, I enjoyed the freedom and ability to set my own pace (sometimes for the better, sometimes for the worse) but it definitley allowed me to become more independent.
Best of:

1. Great with all the presentations, really allowed students to show how hard they have been working and to see if they really understood the problem they were dealing with.
2. The students who did work and showed up in class were a pleasure to be around and gave a good environment.
Cons:
1. Some students kind of put a damper on the expirience.
2. My adviser wasnt exactly accessible all the time, but that is not a problem with CSUMS.

6 References

1. Makri, Nancy and Miller, William. "CORRECT SHORT TIME PROPAGATOR FOR FEYNMAN PATH INTEGRATION BY POWER SERIES EXPANSION IN Δt" Chemical Physics Letters 151, No. 1.2(1988)