



# Simulating Quantum Wave Packets Using Schrödinger's equation

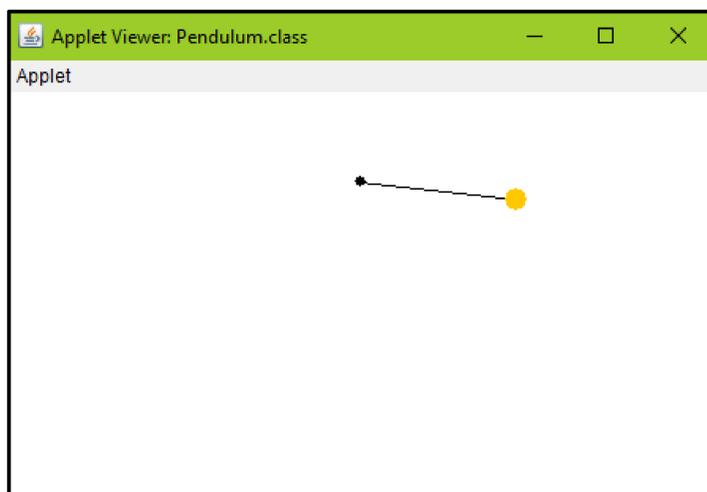
Vikram Sharma, Neel Saswade, Luis Martinez and Joel Yuen-Zhou

The Department of Chemistry & Biochemistry, University of California San Diego, 9500 Gilman drive, La Jolla, California

## Abstract

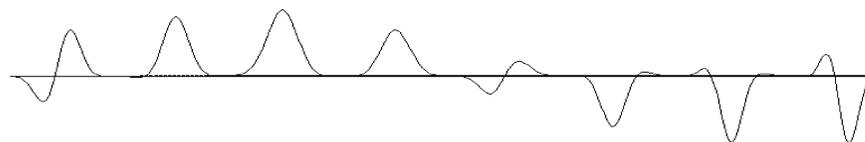
The purpose of this project was to construct a videogame that properly simulates quantum wave packets and Fast Fourier Transformation (FFT). In quantum mechanics, a wave packet represents the dispersion of energy particles after coming in contact with a potential. A probability amplitude is interpreted to determine the density of particles at certain states of time given the velocity. The respective equation for this wave function is represented through Schrödinger's equation. Our videogame which is a classic top down shooter applies this equation in the form of a two-dimensional wave packet. This means that the dispersion of particles has to be represented on an x-y plane, thus requiring Schrödinger's equation to be squared. Our game represents the dispersion of these particles by launching the wave packet through a propagator. Then, we deduct health from the enemies based on the probability density respective to the time and their location. To be able to run these calculations, we created a Quantum Library in Java which is composed of a vast series of methods.

## Basic Gravity Engine / Pendulum Simulation



In order to construct a complete one-dimensional wave propagation simulation, we first had to create a program that could model a harmonic oscillation. This oscillation would be based off of a single object (the wave) using the Earth's gravitational constant. This was in accordance with the pre-defined necessary features of a propagation simulation (as defined by Schrödinger). To create a visual representation of our understanding of harmonics oscillators, we created a simple pendulum program.

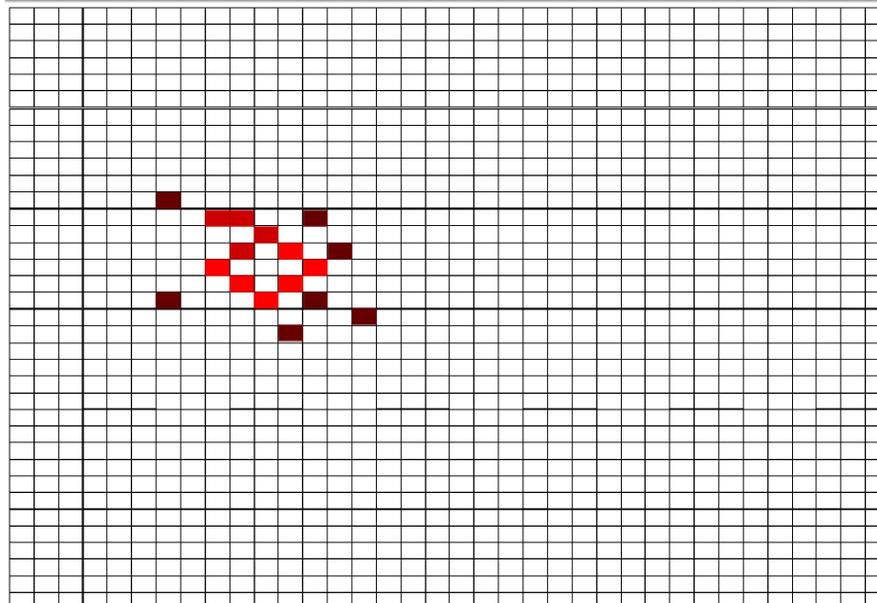
## One Dimensional Wave Simulation



X Values: -10.0-9.9-9.8-6.0000000000000001-9.4000000000000002-9.2000000000000003-9.0000000000000004-8.8000000000000004-8.6000000000000004-8.4000000000000004-8.2000000000000004-8.0000000000000004-7.8000000000000004-7.6000000000000004-7.4000000000000004-7.2000000000000004-7.0000000000000004-6.8000000000000004-6.6000000000000004-6.4000000000000004-6.2000000000000004-6.0000000000000004-5.8000000000000004-5.6000000000000004-5.4000000000000004-5.2000000000000004-5.0000000000000004-4.8000000000000004-4.6000000000000004-4.4000000000000004-4.2000000000000004-4.0000000000000004-3.8000000000000004-3.6000000000000004-3.4000000000000004-3.2000000000000004-3.0000000000000004-2.8000000000000004-2.6000000000000004-2.4000000000000004-2.2000000000000004-2.0000000000000004-1.8000000000000004-1.6000000000000004-1.4000000000000004-1.2000000000000004-1.0000000000000004-0.8000000000000004-0.6000000000000004-0.4000000000000004-0.2000000000000004-0.0000000000000004-0.2000000000000004-0.4000000000000004-0.6000000000000004-0.8000000000000004-1.0000000000000004-1.2000000000000004-1.4000000000000004-1.6000000000000004-1.8000000000000004-2.0000000000000004-2.2000000000000004-2.4000000000000004-2.6000000000000004-2.8000000000000004-3.0000000000000004-3.2000000000000004-3.4000000000000004-3.6000000000000004-3.8000000000000004-4.0000000000000004-4.2000000000000004-4.4000000000000004-4.6000000000000004-4.8000000000000004-5.0000000000000004-5.2000000000000004-5.4000000000000004-5.6000000000000004-5.8000000000000004-6.0000000000000004-6.2000000000000004-6.4000000000000004-6.6000000000000004-6.8000000000000004-7.0000000000000004-7.2000000000000004-7.4000000000000004-7.6000000000000004-7.8000000000000004-8.0000000000000004-8.2000000000000004-8.4000000000000004-8.6000000000000004-8.8000000000000004-9.0000000000000004-9.2000000000000004-9.4000000000000004-9.6000000000000004-9.8000000000000004-10.000000000000000

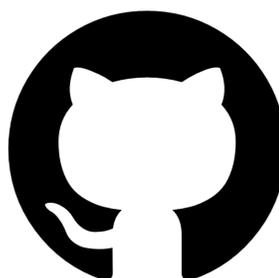
The next step was to use a basic understanding of Fast Fourier Transformation and harmonic oscillation to graph a single dimensional wave packet simulation. The single dimensional program was then displayed in a Java Applet to compare with accepted propagation values. Once these were verified as correct, it was possible to create higher level systems, such as two or three dimensional propagators.

## Two Dimensional Wave Packet Propagation



To create the two-dimensional wave packet simulation, it is necessary to square the equation used. We implemented this in Java by utilizing two-dimensional arrays. By creating a 100 by 100 grid, we were able to accurately display every value we used in propagation. In locations where certain percentages are evident, we classified as a "used grid," and colored it red. The tone and hue defines how much energy is displayed

## Quantum Mechanics Library



Our library to model several mechanical/ quantum models Includes the following methods and classes:

FFT.java ~ From Princeton CS intro site, to manage Fast Fourier Transformation Complex.java ~ From Princeton CS intro site, to aid in management of Complex numbers as Java is unable to handle imaginary concepts

QuantumMethods.java ~ All methods which can be used to create simulations

propagation ~ given the inputs, it can be used to calculate wave propagation

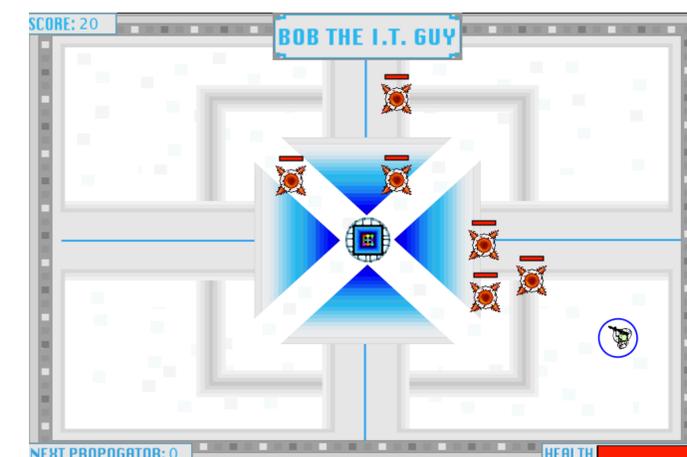
getAngle ~ given an angle of bounce  $\theta$ , this will give the angle the ball will bounce from reference angle  $\theta$

getEnergy ~ given parameters, calculates the amount of energy an object has

calculateVelocity ~ Will calculate the velocity of the object given the inputs

predictChance ~ given an array of previous answers, and the prime answer choice, this method will tell you the % chance of getting the prime value

## Bob The IT Guy – Final Product Video Game



Our video game is titled Bob the I.T. Guy and takes place in a High-Tech Science Facility. He is hired to remove a reported virus from the central database. He also knows that the facility holds a powerful weapon called the Propagator. By the time he catches the virus, it is too late and it has released all of the cyborg zombie specimens from their holding capsules. They begin to storm the room and he fights them to protect himself and the Propagator.

The player is able to freely navigate in the given area and must decisively evade and shoot the zombies. The base weapon represents classical mechanics with a the ball projectile following a linear trajectory. Every thirty seconds, the player has the ability to pick up the Propagator which powers up your weapon into a wave blasting cannon. Once fired, it deals massive damage to every enemy in the area and deducts the health of the enemies by using Schrödinger's equation. The objective of the game is to stay alive for as long as possible, until the player's health has depleted.

## Discussion and Conclusions

After the complete creation of both our game and our Quantum Mechanics Java library, we went through thorough further testing to make sure they were accurate to the concepts covered in Quantum mechanics. The first step was to define as many variables as we could that corresponded with the algorithms we created. We tested our complete one dimensional wave packet propagation algorithm in Java by experimenting with tested and confirmed propagation values. We initialized and set all of our values to valid and information and then used a simple graph program to display the values and make sure they were within accepted limits. Once that was finished, we moved on to the game and the integration of the new library into the game. Our conclusion with this was that we could use one dimensional wave packet algorithms to create two dimensional planar grids. Essentially, this allows us to check each values of a grid using a custom built algorithm to simulate many different types of quantum models. We displayed the dimensional expansion of this algorithm in our video game. The propagation gun weapon lets a player run a program to take health points from every enemy on the screen in accordance with our algorithm.

## References

1. 9.7 "Data Analysis." *Data Analysis*. Princeton University, n.d. Web. 27 July 2016. <http://introcs.cs.princeton.edu/java/97data>
2. "Fast Fourier Transform." -- from *Wolfram MathWorld*. N.p., n.d. Web. 27 July 2016.
3. <http://mathworld.wolfram.com/FastFourierTransform.html>

## Acknowledgments

Sincerest thanks to the Academic Connections program and Professor Komives for directing the Research Scholars program, to Professor Joel Yuen-Zhou for welcoming me into his lab, and to Luis Martinez for helping me with everything.

