SIMULATING FINANCIAL MARKETS USING PHYSICAL MODELS

by

Russell Jay Steed

A senior thesis submitted to the faculty of

Brigham Young University - Idaho

in partial fulfillment of the requirements for the degree of

Bachelor of Science

Department of Physics

Brigham Young University - Idaho

April 2016

BRIGHAM YOUNG UNIVERSITY - IDAHO

DEPARTMENT APPROVAL

of a senior thesis submitted by

Russell Jay Steed

This thesis has been reviewed by the research committee, senior thesis coordinator, and department chair and has been found to be satisfactory.

Date	Stephen Turcotte, Advisor
Date	Todd Lines, Senior Thesis Coordinator
Date	Richard Datwyler, Committee Member
Date	Lance Nelson, Committee Member
Date	Stephen Mcneil, Department Chair

ABSTRACT

SIMULATING FINANCIAL MARKETS USING PHYSICAL MODELS

Russell Jay Steed

Department of Physics

Bachelor of Science

Physicists use a two-state model (+1 and -1) called the Ising Model to study a variety of physical phenomena, including the behavior of magnetic materials. A material is broken up into a lattice of spins where it is then possible to model how each individual spin, as well as the entire lattice, is effected by different external and internal factors. I am able to adapt the same model to financial markets. By simplifying a market into a similar two-state model, I can treat each market agent or stockholder as a spin in the lattice. Each agent is able to buy (+1) or sell (-1). The agents will be influenced by their nearest neighbors as well as an overall magnetization of the market. Using computer software I am able to implement the modified Ising Model to create hypothetical stock data. In order to know if the computer simulations are correctly modeling what actually happens in the stock market, I analyze characteristics found in actual stock data; namely a fat-tailed distribution of returns, slowly decaying

autocorrelation of returns, and volatility clustering. The purpose of this research is to show that the Ising Model is able to effectively model financial markets and identify some of the key factors that influence financial markets. I will use data collected from the Standard and Poors 500 Index over the past 30 years to cross analyze with the data created by the Ising Model.

ACKNOWLEDGMENTS

Thank you to my research mentor, Brother Turcotte, for your insights, support, and initial push. To Brother Lines, Brother Datwyler, and Brother Nelson, for guiding me in the writing of this thesis. To Brother Hansen for helping me in the research process, the BYU-Idaho physics department, and to all my fellow physics majors for helping me with syntax, grammar, and motivation.

Thank you to my parents for always encouraging me to reach my true potential.

My biggest appreciation is to my wife, Alyssa, for putting up with the long nights, hectic days, and stressful weeks of research, homework, and writing. She is my greatest supporter.

Contents

List of Figures

Chapter 1

Introduction

Complex systems, such as financial markets, have been studied for decades. Economists and financial enthusiasts alike try to better understand the underlying factors that drive the behavior of these markets. Some scientists have adapted computational models, normally used in physics, to financial markets. This interdisciplinary field is called econophysics.

Models that are used regularly in understanding the physical world around us are designed to take very complex problems and simplify them so that they can be better understood. One such model is called the Ising Model. The Ising Model is commonly used in modeling magnetic materials. It can also be adapted to financial markets, rush-hour traffic, and other semi-random systems.

In this thesis I will study the ability of a two-state Ising Model to effectively produce results similar to a real financial market. I will also give background on the statistical characteristics present in financial markets. These characteristics will be compared with data generated by the modified Ising Model to see if similar characteristics are present.

By adding in different terms and varying their strengths I will be able to determine

which terms most accurately cause the model to act as a real financial market. I will include two main terms; a term for herding behavior and another that influences agents to go against the crowd.

I have focused more on the physics aspect for this thesis since it to fulfill the requirements of a Bachelors of Physics. However, for completeness I will give an overview of financial markets as well as the Ising Model.

I began my research with a literary search of people who have done similar computations. Stefan Bornholdt wrote a paper on expectation bubbles using a spin agent model. I have used a few of his equation modifications in my computations [1]. I also used studies that use three state models instead of two [2] [3] [4].

I also had to do quite a bit of literature search on financial markets since I was less familiar with that subject. I began by reading into the Efficient Markets Theory which establishes basic assumptions used when modeling financial markets [5]. In order to better understand volatility clustering in financial markets I read a few papers that give an introduction to agent-based models and volatility clustering [6] [7]. I also used a few other sources such as "An introcution to Econophysics" provided by Cambridge University [8] [9].

Chapter 2

Financial Markets

This chapter focuses on the statistical characteristics of financial markets; specifically fat-tailed distribution of returns, volatility clustering, and autocorrelation of returns. I will also compare how these attributes are not present in purely stochastic systems. These characteristics will make it possible to analyze the data created in a modified Ising Model simulation in later chapters.

The Standard and Poors 500 (S&P 500) is a stock index that is a market composite of 500 American companies. It is regarded as being one of the best reflection of the overall stock market. I will mainly use the S&P 500 to demonstrate the characteristics found in general financial markets. I will be using the stock data provided by Yahoo Finance [10].

2.1 Volatility Clustering

Financial markets demonstrate volatility clustering as one of their most visible attributes. Volatility clustering is how a fluctuation in the market is most likely followed by another fluctuation of the same kind. In other words, it is the pattern of large

fluctuations followed by large fluctuations, and small fluctuations followed by small fluctuations [6].

Using the S&P 500 as an example, I take the daily stock values and subtract the previous days value to find the daily return value.

$$r(t) = v(t) - v(t-1) \tag{2.1}$$

Where $v(t)$ is the value of the market at a given time t. Graphing these returns gives a qualitative way of determining if volatility clustering is present. A plot of the S&P daily returns is shown in Figure 2.1.

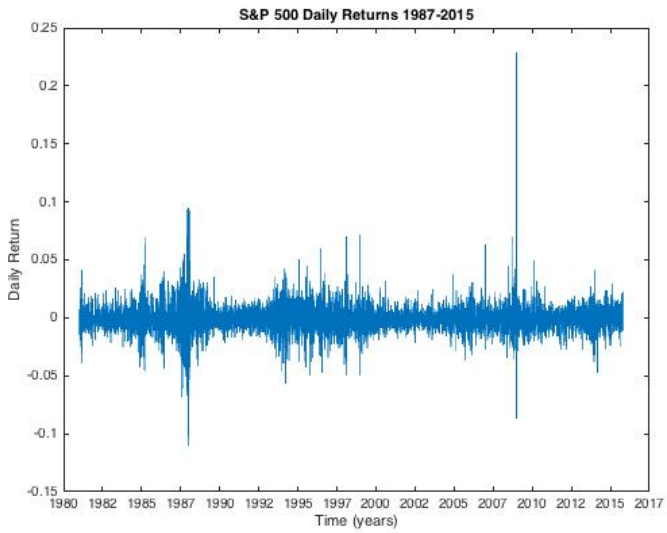

Figure 2.1 Return value versus time
from January 1, 1981 to October 15, 2015. This is a visual representation of volatility clustering.

At first this plot may seem obvious or uninteresting. In order to understand its importance it is helpful to think of a completely random process, such as a random walk. If we calculate the daily returns of a random walk we would get a returns graph that looks very uniform such as in Figure 2.2.

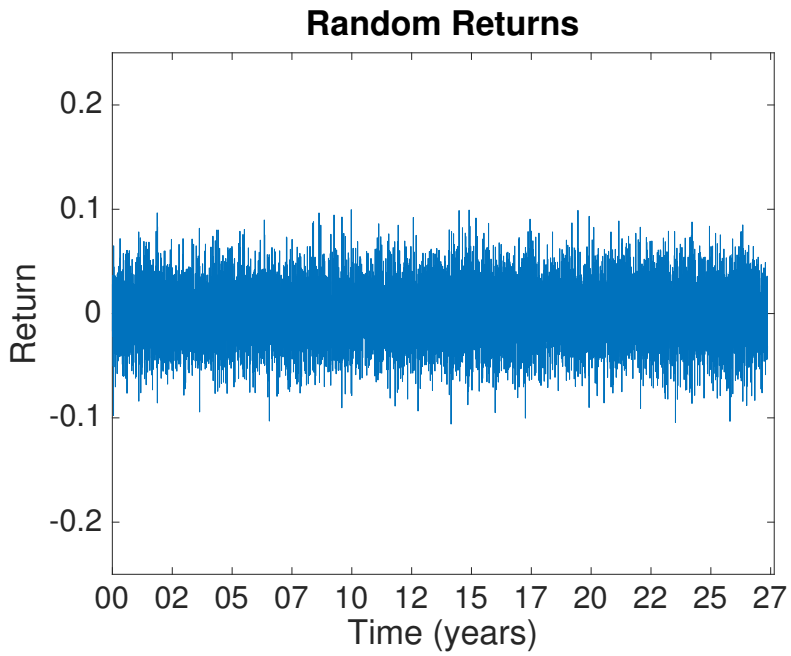

Figure 2.2 Returns versus time plot in a purely random system

After looking at Figure 2.2 it is very noticeable that Figure 2.1 shows periods of small returns and periods of large returns. Figure 2.2 does not show any dependence on time and does not fluctuate far away from the most probable return value. The presence of volatility clustering in Figure 2.1 shows that the data, although seemingly random, is affected by other factors.

2.2 Autocorrelation of Returns

In order to quantify volatility clustering, I used what is called an autocorrelation of returns (ACR) function. Autocorrelation is a mathematical tool used to find patterns in data. It is used often in sonar and radar and other areas that use signal analysis. Autocorrelation is a way of cross-examining the data with itself using varying time steps. This helps to filter out the background noise and find correlations within the actual data.

When calculating and plotting the returns, I use a time-step of one day. The time-step for the ACR changes. The function begins with the smallest time-step, one day, and incrementally increases the amount of days between the values of interest. I use one day as the smallest time step since it is the smallest data available in the archived data available through Yahoo Finance [10]. By doing this I can see if the relationship between two consecutive days is the same or different than the values of two days that have weeks between them. This draws the link between volatility clustering and the autocorrelation of returns giving the volatility clustering a more quantifiable form.

In order to understand why there is a curve present (Figure 2.3) in the ACR function of financial markets and absent in purely random systems we need a better understanding of the equations used. One way to calculate the ACR is similar to using expectation values in quantum mechanics:

$$C(x_t, x_{t+\tau}) = \frac{\langle (x_t - \langle x_t \rangle)(x_{t+\tau} - \langle x_{t+\tau} \rangle) \rangle}{\sqrt{\langle x_t^2 \rangle - \langle x_t \rangle^2}\sqrt{\langle x_{t+\tau}^2 \rangle - \langle x_{t+\tau} \rangle^2}} \tag{2.2}$$

The numerator subtracts the expectation value of the returns, or the average return value $\langle x_t \rangle$, from the current value x_t. Then, using the current time step τ, look at the next value of interest $x_{t+\tau}$ and subtract the expectation value of returns with the time-step included $\langle x_{t+\tau} \rangle$. Then take the product of those top two terms. In the denominator, square the current value x_t before finding its expectation value $\langle x_t^2 \rangle$ and subtract from that the expectation value of the same term squared $\langle x_t^2 \rangle$. Then do the same procedure adding in the current time-step τ and square rooting both terms in the denominator.

Since the idea is the same but the application is different it is necessary to convert equation 2.2 into terms applicable to financial markets. Rewriting the expectation

values as average return values and using a summation to add up all the return values we get equation 2.3,

$$\rho_A(\tau) = \frac{\sum_{t=\tau+1}^{T}(|r_t| - |\bar{r}|)(|r_{t-\tau}| - |\bar{r}|)}{\sum_{t=1}^{T}(|r_t| - |\bar{r}|)^2} \tag{2.3}$$

where τ is the time-step, r is the current value of the returns, and $|\bar{r}|$ is the average return value. As τ increases it is possible to see how the return value compares to those at varying times in the future. I begin the summation at $t = 1$, which is just a time-step of one day between measurements. I can then recalculate the autocorrelation, $\rho_A(\tau)$, by increasing τ and plotting the results.

When graphed, the autocorrelation function shows a slow decay as seen in Figure 2.3. This slow decay is present when volatility clustering is present in the data. For data sets that are completely random the same function returns a flat trend line (see Figure 2.4). The difference between a purely random process and the financial data is visibly present.

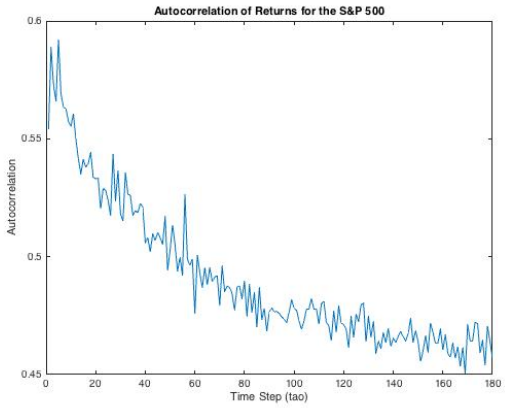

Figure 2.3 The autocorrelation of returns of the S&P 500

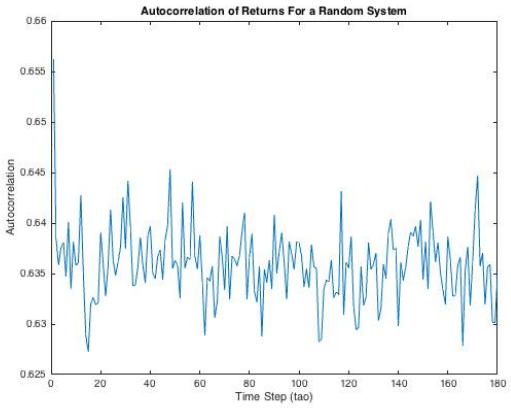

Figure 2.4 The autocorrelation of returns of a completely random system

In a completely random system it is expected to see little to no curve in the graph trend-line. Inversely, we would expect to see a time-step dependence of the returns generated by a system that has more than just random factors in it. For example, if volatility clustering is present there will be large fluctuations followed by other large fluctuations in the returns. This means that there are times when there are quite a few of these fluctuations bunched together and times when there are very small fluctuations bunched together. After putting these values in to an autocorrelation function, the presence of these bunched fluctuations show up as the slow decay.

2.3 Fat-Tailed Distribution of Returns

The probability distribution of the daily returns demonstrates another characteristic specific to financial markets. By plotting the frequency of values calculated in equation 2.1 the probability distribution can be visually formed.

In the stock market there can be return fluctuations close to 20 standard deviations away from the average return value. For example, in October of 2008 the housing bubble burst and sent the United States into an economic recession. On October

6th, 7th, and 9th the daily return values were greater than -50 which is close to 20 standard deviations away from the average return of the S&P 500 [2].

With a Gaussian distribution having returns that are close to 20 standard deviations away is so unlikely that we can almost count on it never happening. Instead, markets show a similar distribution with heavier tails, sometimes called Fat Tails or Long Tails. This distribution demonstrates the possible fluctuations of the market. Figure 2.5 shows the comparison of a Gaussian distribution and the fat-tailed distribution calculated from the S&P 500.

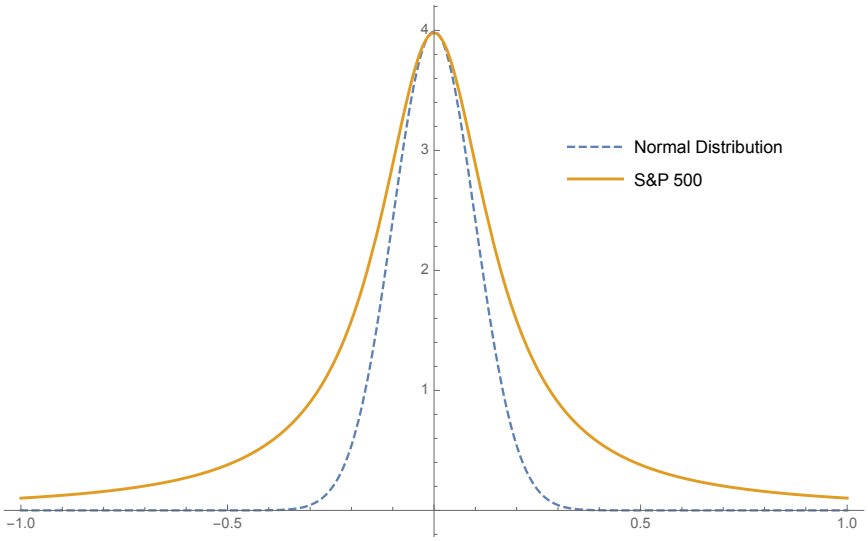

Figure 2.5 Normal distribution compared with the fat-tailed distribution present in the S&P 500

Chapter 3

The Ising Model

I will be using the Ising Model to simulate and analyze financial markets. It is necessary to understand the basics of the model in order to better understand how this can be done. This chapter focuses on the basic Ising Model as well as a few example applications in various fields.

3.1 What is the Ising Model?

Physicists use statistical and stochastic models to simulate and understand physical systems. We will be looking specifically at the Ising Model.

The Ising Model is used in various fields of physics as a method of looking at a system where each site has two possible states, +1 and -1. Each position in the state lattice is evaluated in relation to its nearest neighbors. Each neighbor has a direct effect on the value of the state and can influence it to align with it's positive or negative value. By adding up the "energy" states of each neighbor and adding an element of randomness it is determined if the current state will flip signs or stay the same. The randomness is introduced by taking the calculated energy and comparing

it with a previously determined probability distribution. The random attribute of the Ising Model is important in replicating how natural systems work, since nature is random more often than not. This process is repeated for however many iterations and states desired.

By allowing each state to be influenced by its neighbors, as well as the probability distribution, it is possible to study how a given system will act over a certain time period. This becomes extremely useful in understanding ferromagnetism, quantum spins, and even every day systems such as traffic and stock markets. Since the applications are fairly broad, I will give a brief overview of them in the following section.

3.2 The Ising Model in Physics

The Ising Model was originally tused to model ferromagnetism and was soon adapted to apply to quantum mechanics using the quantum spin of electrons as the two states [11]. Soon after its conception in the 1920s, the Ising Model was disregarded by many physicists as being an oversimplification of natural systems. With the advent of computers and other technological breakthroughs the Ising Model began to prove its usefulness among physicists. John Ziman, a solid-state theoretical physicist, said this about simplified physical models:

> "The real art of the theoretical physicist may lie in his skill at devising models that are mathematically soluble and that still contain the key to the behavior of the system that he is studying. The justification of a model is not that it is an approximate solution of the equations [describing the system by the laws of nature]... but that it provides qualitative or semi-quantitative results that are in agreement with observation. It provides

> explanation and insights, it can tell us what is going on, even though it cannot be anything like as complicated as events in the real world" [12].

It wasnt until the 1960s that physicists began to recognize the Ising Model as a useful tool in statistical mechanics. The model was able to accurately determine constants, and other physical values, when modeling phase transitions [11]. In the following decades the Ising Model became more widely used among physicists.

3.3 Examples of the Ising Model

The most common example of the Ising Model is its application to ferromagnetism, as most often taught to students in computational physics. I will walk through the basic setup, computations, and analysis of this basic Ising Model. I have used Matlab to code the Ising program but will relate the process using pseudocode and equations so that it can be applied to any computing language. The entire Matlab code is included in Appendix A.

This Ising Model computation will look at a simple lattice of a ferromagnet there are two polarizations for each lattice point. I will use a Boltzmann distribution which depends on the temperature of the system. I begin by creating a square lattice of size N, where N is any integer value. Specific initial conditions can be applied to the lattice or randomly fill with values of -1 and +1. The +1 values represent a spin up state and the -1 represents the opposite, spin down, state. With the lattice and initial conditions created I set up three *for* loops as follows. The first loop is the overall iteration, or time, loop. The second and third loops will move through the columns and rows 1 to N respectively.

For t = 1- total time

For i = 1-N rows

For j = 1-N columns

(main Ising calculations)

end loop

end loop

end loop

As previously mentioned in this chapter, the Ising Model looks at each position in the lattice and, by looking at its nearest neighbors, determines the energy at that point. The energy is calculated as summing up all the +1s and -1s. So as I move through the rows and columns of our lattice I will use the following equation to determine the energy:

$$E_{flip} = -J \sum_{\text{neighbor}=1}^{n} s_{\text{neighbor}} \tag{3.1}$$

where J is the exchange energy, which is a constant, n is the number of nearest neighbors, and $s_{neighbor}$ is the spin value of a neighbor. For a 1 dimensional system the sum would be of the two points directly adjacent to the point, for two dimensions there would be the neighbors to the right and left as well as above and below, and so on.

An issue arises when we look at the nearest neighbors of those positions on the edges of the lattice. I have used periodic boundary conditions to solve this problem. In other words, the nearest neighbors to an edge are the positions on the side directly opposite. Once I have set up my code to check the energy at each location I then have to determine what to do with that energy. If the energy is less than zero then

This is wrong way to calculate, we are suppose to check the difference in the energy and check if the new enery is of lower energy i.e it is more stable so we accept the flip and then we change the spin, else we will check the probability with with which it will flip

the spin will automatically flip. If the energy is greater than zero then I use that energy to calculate a probability and compare it to a random number between 0 and 1 to decide whether to flip the spin or not.

$$e^{-E_{flip}/kT_c} \tag{3.2}$$

Where T_c is the critical temperature of the system and k is Boltzmanns constant. Note that I am using the Boltzmann probability distribution in this example since it is the most common. In the following chapter I will use a different distribution that applies more to financial markets.

With the main calculations in place, I am able to run the code and plot the lattice. By changing the constants J and T_c, I am able to observe how the system reacts. For example, Figure 3.1 shows a system that is in an unstable state whereas Figure 3.2 has converged into a stable state. A stable state is where pockets of the same value have developed.

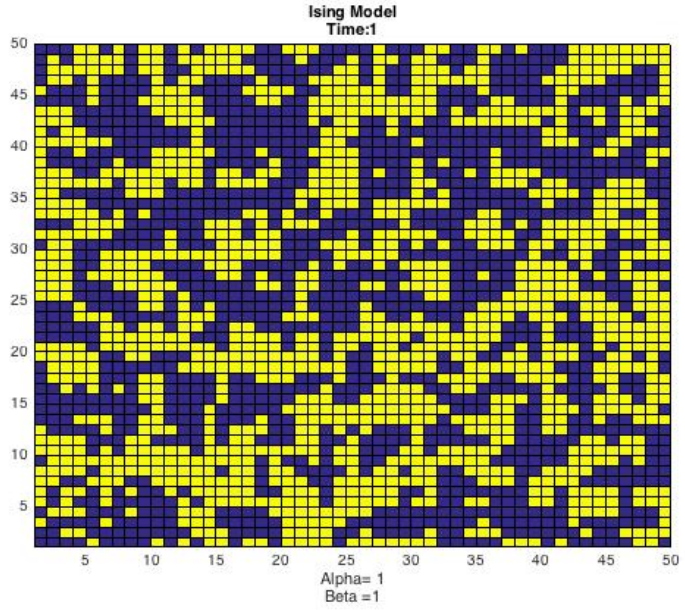

Figure 3.1 Representation of an unstable state.
T = 4, k = 1, t = 500
for simplicity all constants have no units

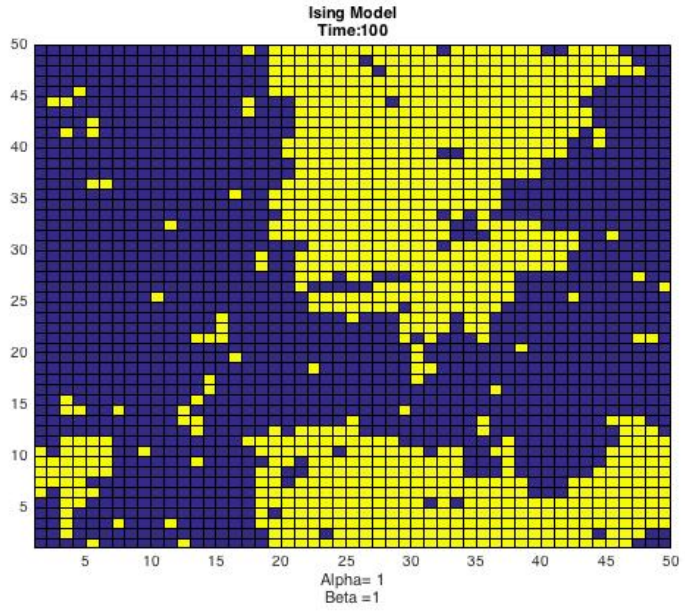

Figure 3.2 Represenation of a stable State.
T = 1, k = 1, t = 1,000
for simplicity all constants have no units

Observing how the system changes as I manipulate the constants gives me insight into what key attributes the system has. For example, as I change T from 0.5 to 2, I observe that the system goes from converging into a stable state fairly quickly to not converging at all. We can call this point the critical temperature, or T_c, since it is where the system transitions from disorder to order.

These same principles apply in any other Ising Model application to help us understand the key components that effect the system. Other terms may be added to the energy calculations such as an external magnetic field that attracts the spins to align in that direction. Using the above example as basis, I modified the Ising Model to apply to financial markets in the following chapter.

Chapter 4

Adapting the Ising Model to Financial Markets

In this chapter I will go over how I adapted the Ising Model to work with financial markets. I will also go over different approaches to coding the Ising Model and how they work with markets. Lastly, I will present an overview of the code that I created using Matlab.

4.1 Behavioral influences on Financial Market

The Ising Model is a simplification of very complex systems, which is what makes it a possible fit for modeling financial markets. In order to make the Ising Model act as a financial market I need to account for the major behaviors that influence agents working in a financial market; volatility clustering, autocorrelation of returns, and fat-tailed distribution of returns.

4.1.1 Human Rationality and The Efficient Capital Markets Theory

Financial markets are the product of a human system, which introduces even more complexity into the system. The rational behavior of humans is a difficult thing to assume since such behavior is dependent on deductive reasoning [13]. In other words, there are few situations where all the facts are known and a conclusion can be reached deductively. Instead, humans must reason by induction. Basically, we fill in unknown gaps of information in order to come up with an executable decision. This phenomenon is described computationally in the El Farol problem [13] and will be described in more detail in section 4.1.2.

In order to resolve the complexity issue of rationality and inductive reasoning I use the Efficient Markets Hypothesis (EMH). This theory lists basic assumptions made to simplify the market. One such assumption states that stock prices instantly reflect all available knowledge, even hidden or insider information [5]. The EMH simplifies the idea of markets and gives us a serious implication, that no agent can beat the market. In simple terms this means that the market reflects all information, thus making it impossible for an agent to make decisions that beat the system. This takes care of the difficulty of trying to account for human rationality.

4.1.2 The Minority Game

The El Farol problem is a simplification of the minority game, which appears in financial markets. Agents often find the best returns when they are in the minority [1]. Thus, it is advantageous to be among the minority agents. In other words, sell when everyone is buying or buy when everyone is selling. Since agents have no way of exactly knowing what the minority will be, the probability of them being in the

minority is fairly low.

4.1.3 Herding Behavior

On the other hand, humans have the desire to follow the crowd. This is known as the herding behavior. Even though the possible return is greatest when the agent is in the minority there is still a large part of human behavior that makes us more comfortable making a decision that we see many other people making.

4.2 Equation Changes

A simple Ising Model moves through every point in the lattice and calculates the energy at that position. This energy is calculated by adding up the values of its nearest neighbors.

$$E_{flip} = -J \sum_{\text{neighbor}=1}^{n} s_{\text{neighbor}} \tag{4.1}$$

The energy is then compared with a probability distribution to determine if the value at that position will flip to its inverse or not.

$$P = e^{-E_{flip}/kT_c} \tag{4.2}$$

The value calculated with a Boltzmanns equation is then compared with a randomly generated number between 0 and 1. If the random value is lower than the energy, then the values sign flips. Inversely, if the value is higher than the energy, the values sign stays the same.

Taking equations 4.1 and 4.2 I am able to modify them to account for the two opposing behaviors that I want present in this computation. The first, the herding

behavior, and second, the minority game. The herding behavior is taken into account with the basic Ising Model calculations shown in equation 4.1. Going through and looking at the spins, or actions, of each agents nearest neighbors and then making a decision to flip or not flip based on that data is essentially what agents do in real life.

In order to account for the minority game behavior, I have to add in another term. This term will obviously oppose the herding behavior. In order to capture this behavior, I add up all the values of the entire lattice and take the average. If the average is closer to sell (-1) or buy (+1) this term will entice the agent to do the opposite. This is similar to adding in a magnetization term into a ferromagnetic Ising Model calculation. This idea was originally proposed by Stefan Bornholdt in his 2001 paper *Expectation Bubbles in a Spin Model of Markets* [1]. The modifications to the Ising equation, although they use the ideas of Bornholdt, resemble the equations used by Pavel Dvorak [2]. Equation 4.3 reflects the changes made to the simple Ising Model calculation.

$$h_i j(t) = \sum_{i,j=1}^{k} J_{ij} s_{ij}(t) - \alpha s_{ij}(t)|M(t)|. \tag{4.3}$$

The first term is the same as the simple Ising Model. The second term subtracts the magnetization of the entire system where α is a constant multiplier that determines the influence of the magnetization. The magnetization can be calculated by finding the average value of the lattice.

$$M(t) = \frac{1}{k^2} \sum_{i,j=1}^{k} s_{ij}(t) \tag{4.4}$$

The last change I need to make is on the probability distribution. Instead of using a Boltzmann distribution I will use the probability presented by Pavel Dvorak [2].

$$p = \frac{1}{1 + e^{-2\beta h_i(t)}} \tag{4.5}$$

where β determines the slope of the distribution. The probability distribution plot is given in Figure 4.1 with a few different values of β.

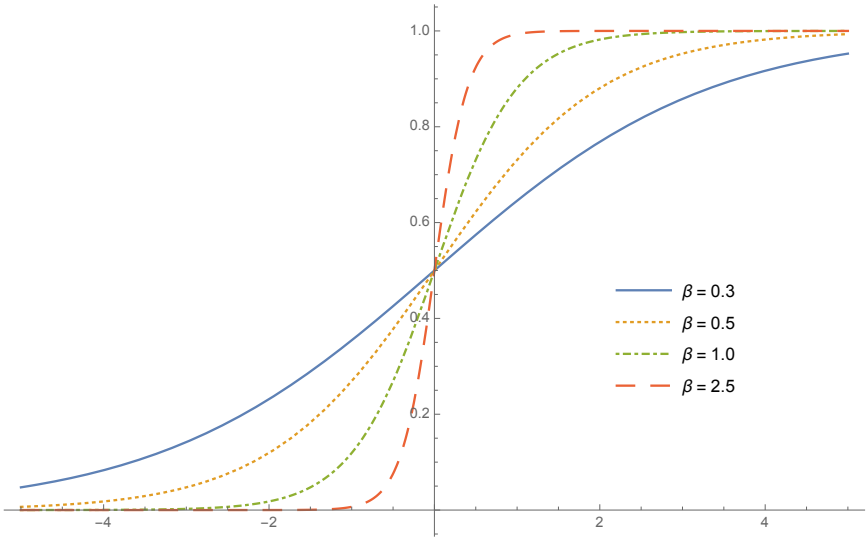

Figure 4.1 The Cumulative Distribution Function (CDF), given in equation 4.5, plotted with various values of β to show how β effects the output values.

4.3 Serial Update Vs. Parallel Update

There are two main ways of running through the Ising Model. The first way that I tried was using parallel update. This is basically running through each row and column consecutively and determining if it should flip or not. While this method works for many applications including atomic spins, it does not work very well with financial markets.

The reason it does not work is because it forces the system to reach an equilibrium state very quickly and does not provide enough randomness to vary outside of that state. This caused a lot of confusion when the data returned from my early code did not resemble the rising and falling of financial markets. Instead, I had to use a serial update method.

Serial update means that instead of moving consecutively through the lattice, a

random point is chosen and updated. This provides enough randomness to the system that it does not get stuck in a single equilibrium state [2].

4.4 Code Overview

There are a few parts of the code that are key to in building and analyzing this model. I will briefly explain those few parts in this section. Please refer to Appendix A for the complete Matlab code.

I have written two pieces of code. The first is the Ising Model and the second analyzes the data output by the Ising Model as well as the S&P 500 stock data.

4.4.1 Ising Model Code

In the Ising Model I had to include three *for* loops. The first is the time loop. For most of my calculations I used 10,000 iterations. The second loop runs through each row of the lattice. I used a 70x70 matrix in my calculations. The last main loop runs through each column of the lattice.

At the beginning of the time loop I calculate the overall magnetization of the lattice, as shown in equation 4.4. This value will be used in the middle of the three main loops to calculate the flipping energy (equation 4.3).

The flipping energy, $h_i(t)$, is put into a probability distribution function (line 4). If the value of $h_i(t)$ is less than zero than the sign of the value is automatically flipped. If it is greater than zero then it is compared to a randomly generated number between 0 and 1. This is shown in line 13 of the following code segment.

```
1 %Main Ising Calculation
2  h(n,m) = J*(Top+Bottom+Left+Right)*Center-alpha*abs(mag(t));
```

```
3
4           %calculate the probability given by Pavel Dvorak
5           prob(t) = 1/(1+exp(-1*beta*h(n,m)));
6           %prob(t) = exp(-h(n,m)/(kb*T)); %Calculate Botzmann Factor
7
8           if h(n,m) <= 0    %if "energy" is less than zero
9               spinArray(n,m) = -spinArray(n,m);   %Change spin
10          else
11             %generate a random number between 1 and 0
12             r(t) = rand(1);
13             %evealuate r and prob to set the spin
14             if r(t) >= prob(t)
15               spinArray(n,m) = -spinArray(n,m);     %change spin
16             end
17          end
```

The next step is to calculate the returns generated by the main Ising calculations shown above. This is done at the end of the time iteration loop. I use the overall magnetization calculated during the current time step subtracted by the magnetization of the previous time step. I use the overall magnetization since it is a measure of the current *value* of the market.

4.4.2 Analysis Code

This code imports data saved to Excel spreadsheets of the Ising Model simulation as well as the S&P stock data from January 1^{st} 1981 to October 15^{th} 2015. It then calculates the daily returns, returns distribution, plots the stock data versus time, and calculates the autocorrelation of returns. The more complex ACR calculation is as follows:

```
1  for tau = 1:tauMax
2      numerator = 0;   %initializing the numerator
3      denominator = 0; %initializing the denominator
4      for m = 1+tau:dataSize - 1 %move through data by tau
5          numerator = numerator + ((abs(returns(m))-abs(average))...
6              *(abs(returns(m-tau))-abs(average)));
7          denominator = denominator + ...
8              ((abs(returns(m-tau))-abs(average))^2);
9      end
10     %put the top and bottom together to get the AC Returns Value
11     ACReturns(1,n) = numerator/denominator;
12     n = n + 1; %increase iteration variable
13 end
```

Since the autocorrelation of returns has to run through the entire data array from $\tau = 1$ to $\tau = \tau_{max}$, it is necessary to have two for loops. The first to iterate τ and the second to apply the autocorrelation function to the data (lines 4-9).

I also used Mathematica to analyze the distributions and autocorrelation or returns of all the data sets. The Mathematica code will not be included in this paper, although the results of the code will be shown in the following chapter.

Chapter 5

Results and Analysis

This chapter covers the results of the computational Ising Model. I will compare these results with the S&P 500 control data. Conclusions will be drawn as to the usefulness of the Ising Model in understanding financial markets and which aspects of the calculations determine how the system acts.

5.1 Volatility Clustering

The first characteristic that I used to test if my model is acting like a real financial market is volatility clustering. Volatility clustering is qualitatively apparent in a returns versus time graph of the S&P 500 as shown in Figure 5.1. For comparison Figure 5.2 also shows how a completely random system does not show any signs of volatility clustering. Figure 5.3 is a plot of the returns calculated in the Ising Model calculations.

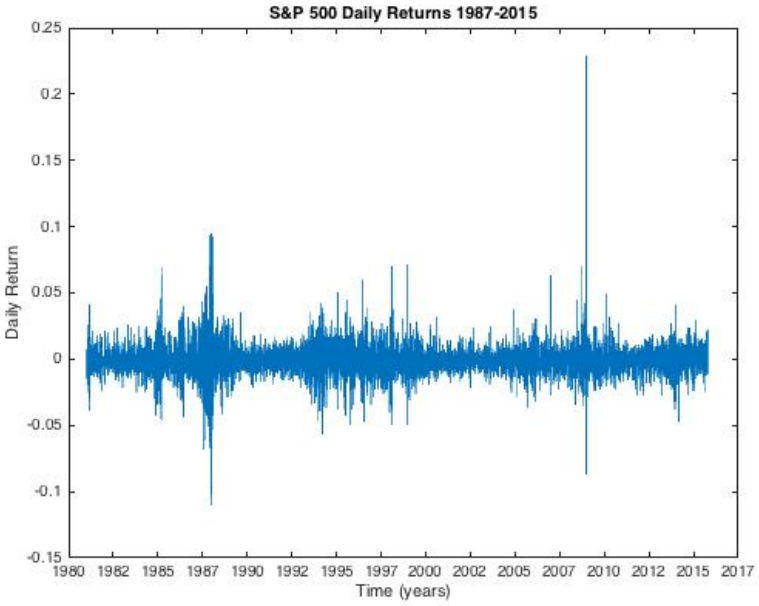

Figure 5.1 Return value versus time
from January 1, 1981 to October 15, 2015. This is a visual representation of volatility clustering.

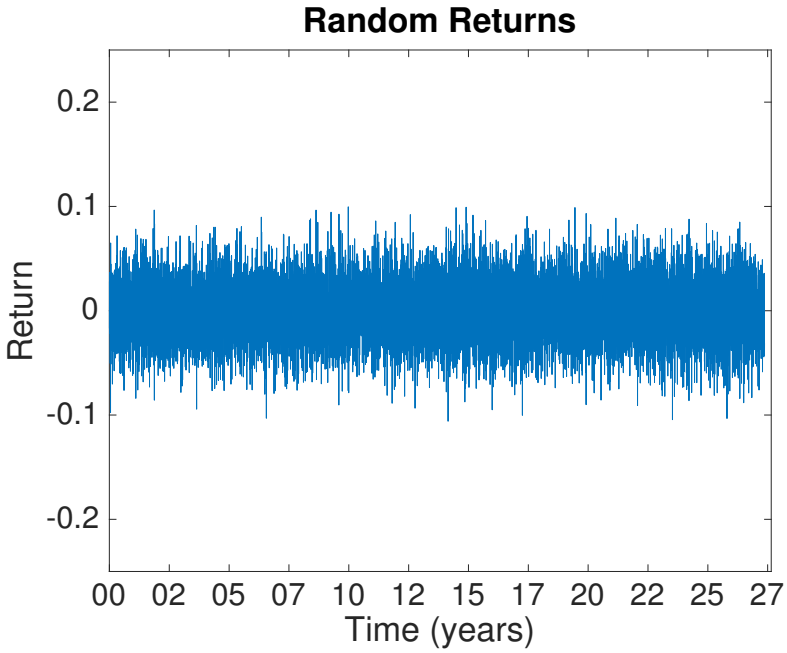

Figure 5.2 Returns versus time plot in a purely random system

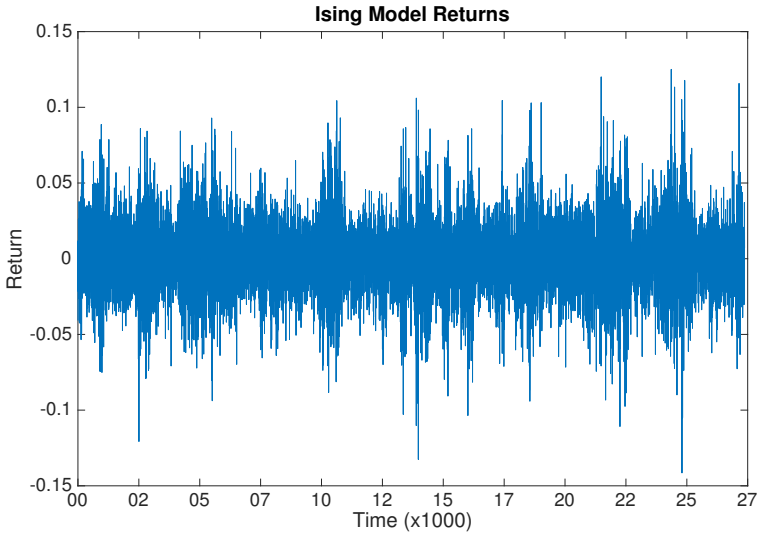

Figure 5.3 Returns versus time plot produced by an Ising Model simulation (top) and the market value versus time graph (bottom)

By looking at the returns plot in Figure 5.3 it can be concluded that it is somewhere between Figure 5.2 and 5.1. Even though volatility clustering is not as apparent as in the S&P 500 data, it still shows signs of being effected by more than just a random process.

5.2 Autocorrelation of Returns

In order to better quantify volatility clustering an analysis or the autocorrelation of returns is necessary. Figure 5.4 and 5.5 show the difference between the S&P 500 and a completely random process. Figure 5.6 shows the results of the Ising Model calculations.

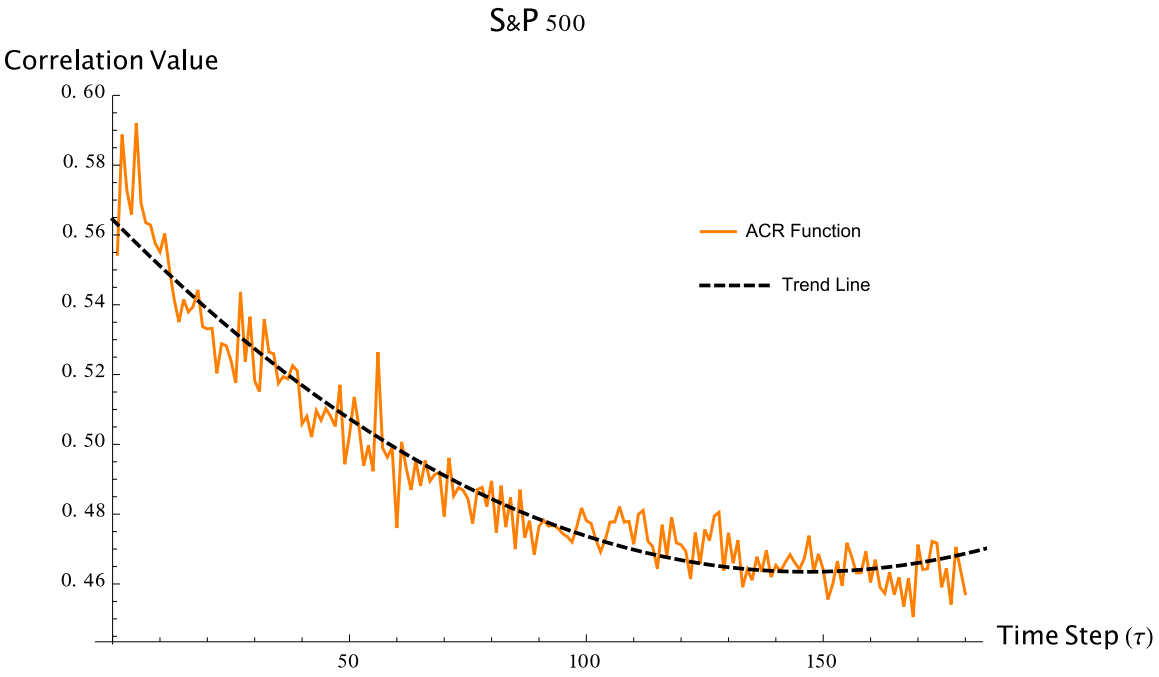

Figure 5.4 The autocorrelation of returns of the S&P 500 with a fitted trend line

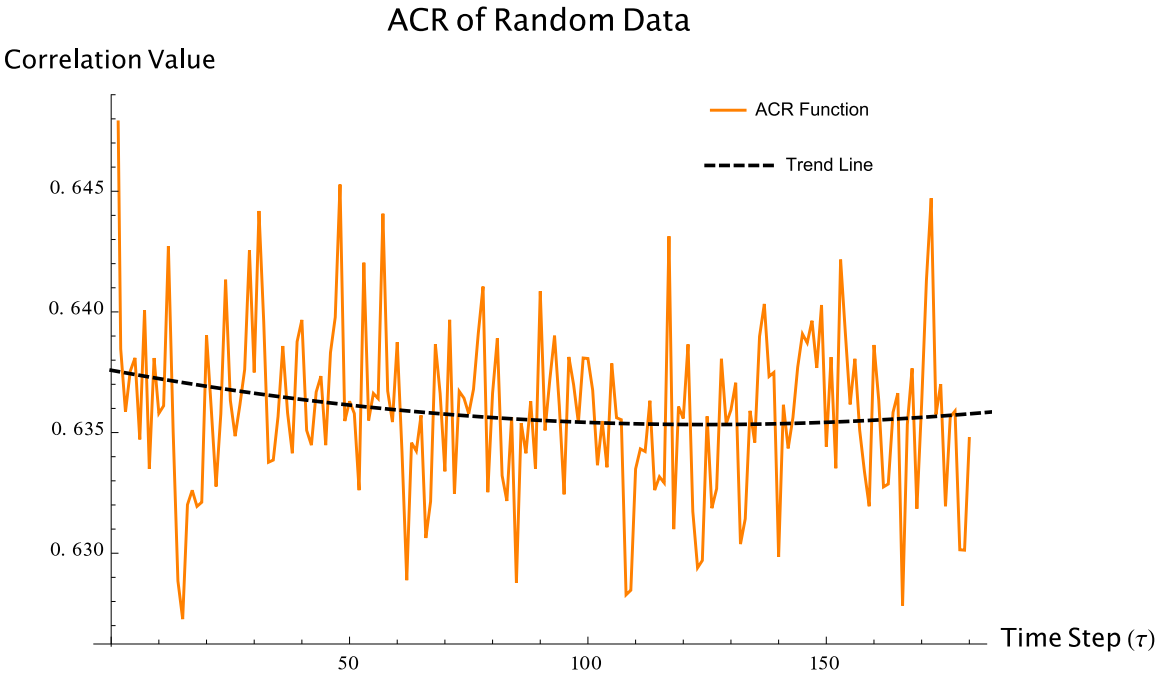

Figure 5.5 The autocorrelation of returns of a completely random system along with a fitted trendline

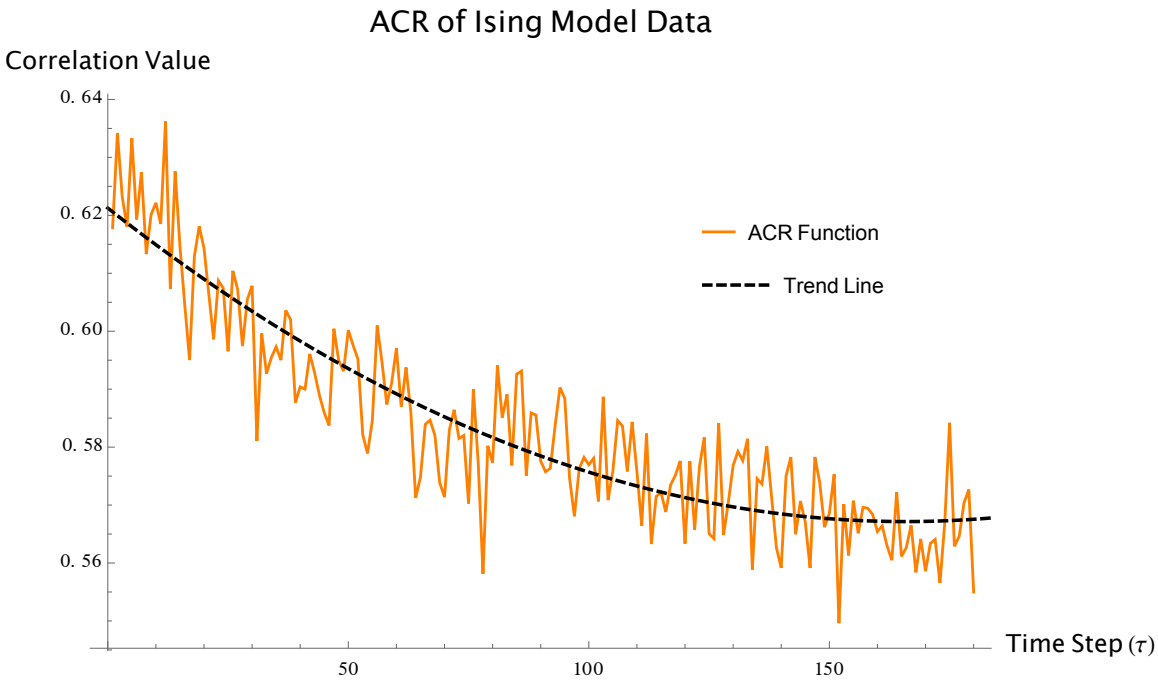

Figure 5.6 The autocorrelation of returns generated by an Ising Model computation along with a fitted trendline

Again the graph of the Ising Model data lies in between that of the S&P 500 and a purely random system. A slow power-law decay is present in Figure 5.6, showing that there is a correlation of return values and time.

Figure 5.7 shows the trend lines of all three data sets relative to each other. We see that the Ising Model data is approaching the power-law decay visible in the S&P 500 ACR data.

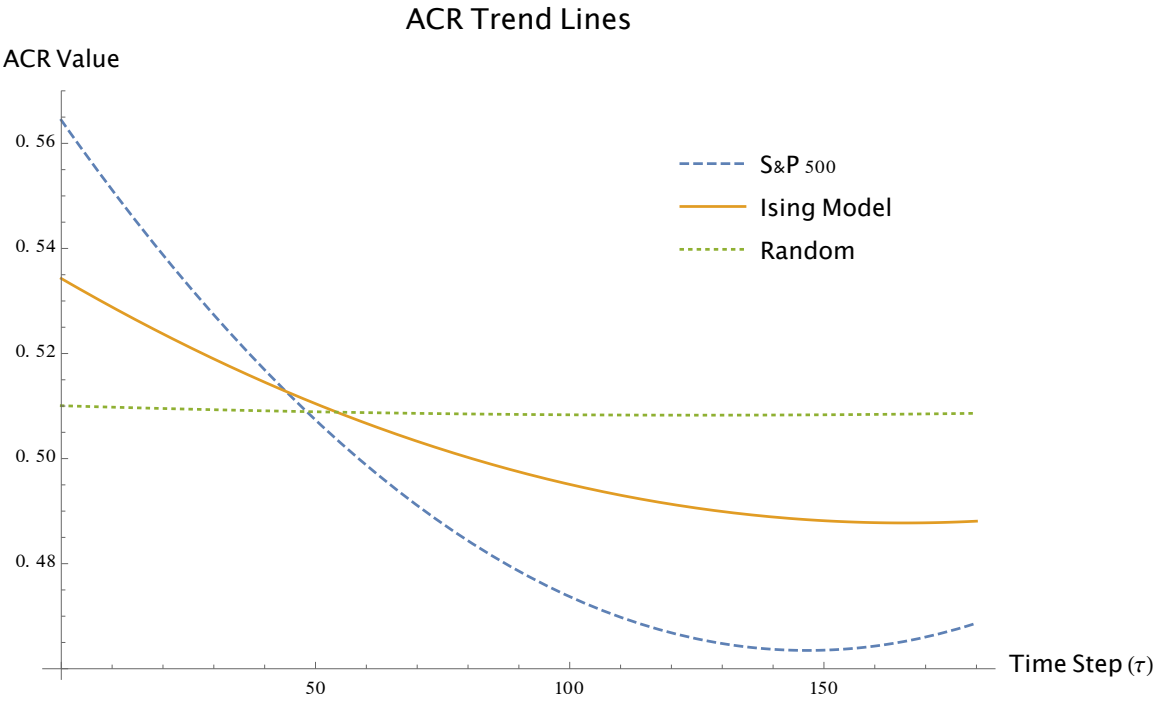

Figure 5.7 A comparison of the three ACR trend lines relative to each other

5.3 Distribution of Returns

Lastly I analyze the distribution of returns. Figure 5.8 shows the distribution of the S&P 500 and Figure 5.9 contrasts that with the distribution of a purely random system. Figure 5.10 displays the distribution of returns generated by the Ising Model.

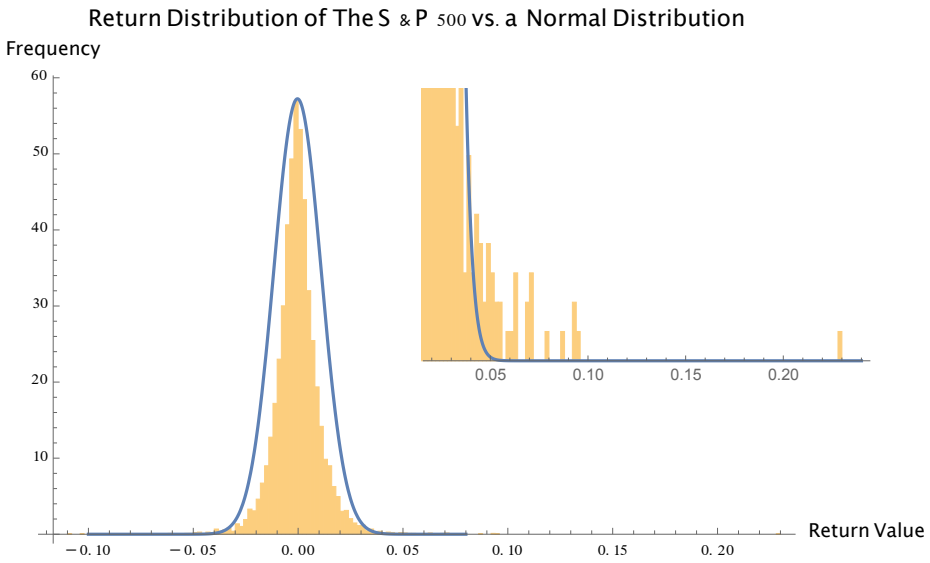

Figure 5.8 Distribution of returns of the S&P 500 compared with a normal distribution. The graph in the upper right is a zoomed in portion of the distributions right wing to show how there are fluctuations well beyond that predicted by the normal distribution.

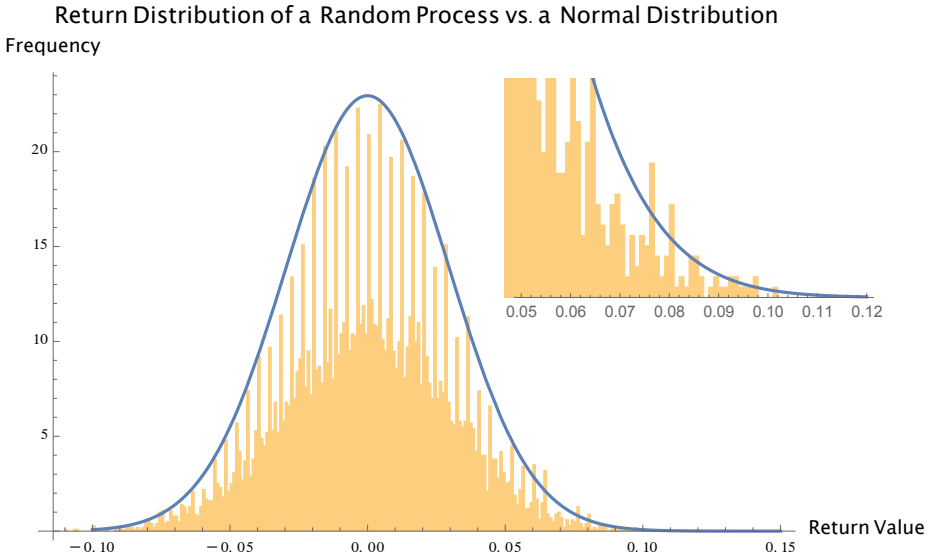

Figure 5.9 Distribution of returns for a completely random system compared with a normal distribution. The graph in the upper right is a zoomed up portion of the distribution to show how the there are no return fluctuations when the normal distribution probability goes to zero.

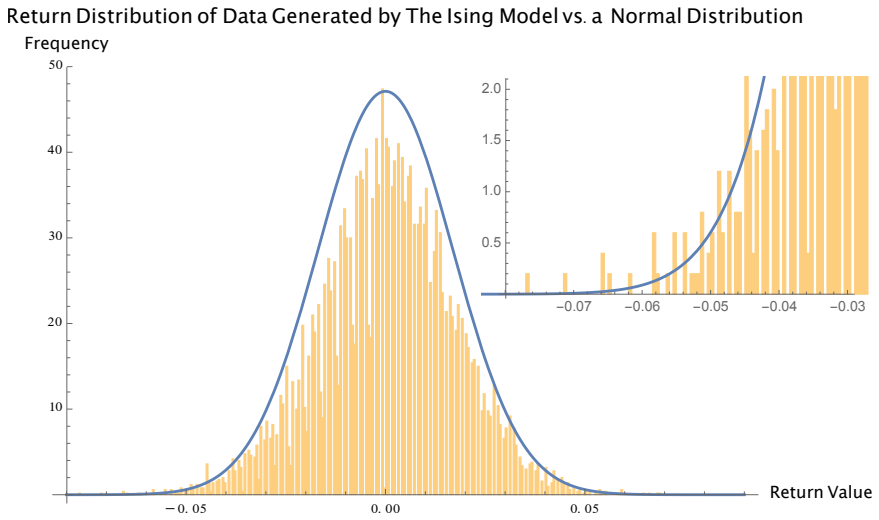

Figure 5.10 Distribution of returns for an Ising Model computation plotted with a normal distribution. The upper right graph shows a zoomed up portion of the left wing of the distribution. This shows how there are return fluctuations beyond what is expected with a normal distribution, which is what we see in real stock market data.

The distribution of returns given in Figure 5.9 shows a mix between the previous two figures. Even though it is not dramatically skinny and tall like Figure 5.7 it is apparent that it is not as widely distributed as Figure 5.8. The return distribution of the completely random process displays a Gaussian, or normal, distribution and Figure 5.7 demonstrates a fat-tailed distribution.

Chapter 6

Conclusion

The data generated by the Ising Model demonstrates the same characteristics that are present in the S&P 500. However, these characteristics are not as apparent or dominating as in the real stock data. Taking into account the two main behavoirs included in this study, herding and minority game, I have begun to model how an actual financial market acts.

This brings us to the conclusion that the Ising Model is able to capture some of the behaviors present in a real financial market and is a useful tool in understanding how each behavior effects the results of the simulation. Thus, for an extremely simplified version of a very complex system, the Ising Model is a valuable tool. However, more behaviors and phenomena should be taken into account to more accurately model a financial market.

Future Work

In this research I only considered a two-state Ising Model. Agents trading on a real stock market not only have the option to buy and sell, but also to do nothing. Adding

in another state where the agent makes no change would add another level of accuracy to the model explained in this thesis [1].

Another area of further research would be to more fully quantify the results of the computed data. By comparing the computed data with both the random data and the S&P 500 data a stronger argument may be formulated as to the effectiveness of the computational model.

Bibliography

[1] Bornholdt, S., 2001. "Expectation bubbles in a spin model of markets: Intermittency from frustration across scales." *International Journal of Modern Physics*(5), 16 May, pp. 1–5.

[2] Dvorak, P., 2012. Ising model in finance from microscopic rules to macroscopic phenomena.

[3] Zhou, W., and Sornette, D., 2008. "Self-fulfilling ising model of financial markets." *APS*, 2 Feb, pp. 1–4.

[4] Sieczka, P., and Holyst, J., 2007. "A threshold model of financial markets." pp. 525–530.

[5] Fama, E. F., 1970. "Efficient capital markets: A review of theory and empirical work." *The Journal of Finance,* **25**, Dec, pp. 383–391.

[6] Cont, R., 2007. "Volatility clustering in financial markets: Empirical facts and agent-based models." *Long Memory in Economics*, pp. 1–5.

[7] Eom, C., and Kim, T., 2007. "Measuring volatility clustering in stock markets." pp. 1–4.

[8] Mantegna, . N., and Stanley, H. E., 2000. *An Introduction to Econophysics: Correlations and Complexity in Finance*. The Press Syndicate of the University of Cambridge.

[9] Tseng, J.-J., and Li, S.-P., 2010. "Quantifying volatility clustering in financial time series." pp. 2–5.

[10] Yahoo! http://finance.yahoo.com/q/hp?s=^GSPC+Historical+Prices Accessed: 11-17-2015.

[11] Niss, M., 2009. "History of the lenz-ising model 1950-1965: from irrelevance to relevance." *Archive for History of Exact Sciences,* **63**(3), May, pp. 243–287.

[12] Ziman, J. M., 1965. "Mathematical models and physical toys." *Nature,* **206**, pp. 1187–1192.

[13] Arthur, W. B., 1994. "Inductive reasoning and bounded rationality." *American Economic Review,* **84**, pp. 2–7.

Appendix A

Matlab Code

This appendix includes the Matlab code. I made two files that are included in their entirety. Each code imports data saved in Excel spreadsheets. For brevity these Excel sheets are not included in this paper.

Ising Model Code

```
% Ising model applied to financial markets
% Russell Steed
% Senior Physics Research - BYU Idaho
% November 2015

clc; close all; clear;

%% Constants and Arrays
k = 70;                          %Dimensions of the array
iterations = 3000;
spinArray = round(rand(k))*2-1;%fill array with +1 or -1 randomly
J = 1;                  %The exchange constant
alpha = 3;             %Global coupling constant (changes magnetization)
beta = 1;               %Determines slope of prob. dist.
mag = zeros(iterations,1);
returns = zeros(iterations,1);
dt = 100;                        %time step for returns
stock = ones(iterations,1);
```

```
parallel = 0;

%% Ising Calculations


for t = 1:iterations %How many times you want to go through the array
    %calculate the overall "magnetization" of the system
    for l = 1:k
        for p = 1:k
            mag(t) = mag(t)+(1/k^2)*spinArray(l,p);
        end
    end

    %actual Ising Model Calculations
    for n = 1:k           %Columns
        for m = 1:k       %Rows
            if parallel == 0;
                n = round(rand*(k-1))+1;
                m = round(rand*(k-1))+1;
            end

 %this is the spin point of interest in the lattice
                Center=spinArray(n,m);

                %this find the value of the spin to the right of
                %the spin of interest, on the boundry it loops
                %the matrix and uses the value of the spin on
                %the opposite boundry
                if n==1
                    Top=spinArray(k,m);
                else
                    Top=spinArray(n-1,m);
                end

%this find the value of the spin to the right of the spin of
                %interest, on the boundry it loops the
                %matrix and uses the value of the spin on
                %the opposite boundry
                if  n==k
                    Bottom=spinArray(1,m);
                else
                    Bottom=spinArray(n+1,m);
                end

%this finds the value of the spin to the right of the spin of
                %interest, on the boundry it loops the
                %matrix and uses the value of the spin on the
                %opposite boundry
                if m==k
                    Right=spinArray(n,1);
                else
```

```
70                      Right=spinArray(n,m+1);
71                  end
72
73  %this finds the value of the spin to the right of the spin of
74                  %interest, on the boundry it loops the matrix
75                  %and uses the value of the spin on the opposite
76                  %boundry
77                  if m==1
78                      Left=spinArray(n,k);
79                  else
80                      Left=spinArray(n,m-1);
81                  end
82
83              %Main Ising Calculation
84              h(n,m) = J*(Top+Bottom+Left+Right)*Center-alpha*abs(mag(t));
85
86              %calculate the probability given by Pavel Dvorak
87              prob(t) = 1/(1+exp(-1*beta*h(n,m)));
88              %prob(t) = exp(-h(n,m)/(kb*T)); %Calculate Botzmann Factor
89
90              if h(n,m) <= 0    %if "energy" is less than zero
91                  spinArray(n,m) = -spinArray(n,m);   %Change spin
92              else
93                 %generate a random number between 1 and 0
94                 r(t) = rand(1);
95                 %evealuate r and prob to set the spin
96                 if r(t) >= prob(t)
97                   spinArray(n,m) = -spinArray(n,m);     %change spin
98                 end
99              end
100         end
101     end
102
103     %calculate returns
104     if t > 1
105         returns(t) = mag(t) - mag(t-1);
106         stock(t) = stock(t-1) + returns(t);
107     end
108
109
110     figure(1)
111     pcolor(spinArray)
112     title(strcat({['Ising Model'];['Time:',num2str(t)]}))
113     xlabel(strcat({['Alpha= ',num2str(alpha)];...
114         [strcat('Beta = '),num2str(beta)]}))
115     set(gca,'Fontsize',20)  %Set font size to 20
116     drawnow;
117
118 end
119
120 figure(2)
```

```
121     subplot(2,1,1)
122     plot(returns)
123     title('Returns')
124     xlabel(strcat({['Alpha= ',num2str(alpha)];...
125         [strcat('Beta = '),num2str(beta)]}))
126     set(gca,'Fontsize',20)  %Set font size to 20
127     drawnow;
128     subplot(2,1,2)
129     plot(stock)
130     title('Stock Price')
131     xlabel('Time')
132     set(gca,'Fontsize',20)  %Set font size to 20
133     drawnow;
```

Analysis Code

Code used to analyze the S&P 500 stock data as well as the data generated by the Ising Model simulation:

```
1  % Data Analysis of S&P 500
2  % Senior Research OCT 2015
3  % Russell Steed
4
5  clear; close all; clc;
6
7  %% Import Data
8  IsingModel = 1; %0: Stock Data, 1: Ising Data
9
10 if IsingModel == 0
11     filename = 'SP Data.xlsx'; %specify file name and location
12     sheet = 1; % which sheet inside the excel file
13
14     % Import data from Excel sheet
15     dailyValue = transpose(xlsread(filename,sheet,'G2:G8879'));
16 else
17     %specify file name and location
18     filename = 'Ising Model Data Sheet';
19     sheet = 1; % which sheet inside the excel file
20
21     % Import data from Excel sheet
22         % G,I,Q,S,U = Ising data, K = Completely random
23     dailyValue = transpose(xlsread(filename,sheet,'U6:U12006'));
24 end
25
26 %% Index from 1987 - 2015
27 [x,dataSize] = size(dailyValue(1,:)); %get amount of data points
```

```
28
29 % Setup the dates to change our x-axis values to respective dates
30 startDate = datenum('01-02-1981');
31 endDate = datenum('10-20-2015');
32 xData = linspace(startDate,endDate,dataSize);
33
34 % Make a plot of the daily values of the index
35 if IsingModel == 0
36     subplot(2,2,1)
37     plot(xData, flip(dailyValue(1,:)))
38     datetick('x','yy') % Chage x axis values so that it shows date
39     title('S&P 500 Index 1981-2015')
40     xlabel('Time (years)')
41     ylabel('Stock Price (USD)')
42     set(gca,'Fontsize',20)  %Set font size to 20
43 else
44     figure(2)
45     plot(xData, flip(dailyValue(1,:)))
46     datetick('x','yy') % Chage x axis values so that it shows date
47     title('S&P 500 Index 1981-2015')
48     xlabel('Time (years)')
49     ylabel('Stock Price (USD)')
50     set(gca,'Fontsize',20)  %Set font size to 20
51 end
52
53 %% Daily Returns
54 returns = zeros(1,dataSize-1); % Initialize returns array
55
56 % Calculate the daily returns using matrix operations
57 returns(:) = log(dailyValue(1,2:dataSize))-...
58     log(dailyValue(1,1:dataSize-1));
59
60 % Plot dialy returns
61 if IsingModel == 0
62     subplot(2,2,2)
63     plot(xData(1,1:dataSize-1), returns)
64     datetick('x','yy') % Change x axis values so that it shows date
65     title('S&P 500 Daily Returns 1987-2015')
66     xlabel('Time (years)')
67     ylabel('Daily Return')
68     set(gca,'Fontsize',20)  %Set font size to 20
69 else
70     subplot(1,3,1)
71     plot(returns)
72     datetick('x','yy') % Change x axis values so that it shows date
73     title('Ising Model Returns')
74     xlabel('Time (years)')
75     ylabel('Daily Return')
76     set(gca,'Fontsize',20)  %Set font size to 20
77 end
78
```

```
79 %% Distribution
80
81 %First we need to create a normal distribution line to compare
82 %to our actual distribution
83 nmax = size(returns);
84 distribution = zeros(nmax(1,2),1);
85 numMax = size(find(returns == nmax(1,2)));
86 C = numMax(1,2);
87 sigma = .2;
88 mu = 0;
89 n = 0;
90 dn = 1;
91
92 for j = 1:nmax(1,2)/dn-1
93     distribution(j,1) = C*exp(-(n-mu)^2/((2*sigma)^2));
94     distribution(j,2) = n;
95     n = n + dn;
96 end
97
98 if IsingModel == 0
99     subplot(2,2,3)
100    histogram(returns)
101    title('Returns Distribution of S&P 500')
102    xlabel('Daily Return Value')
103    ylabel('Probability')
104    set(gca,'Fontsize',20)  %Set font size to 20
105 else
106    subplot(1,3,2)
107    histogram(returns)
108    title('Returns Distribution of Ising Model Data')
109    xlabel('Daily Return Value')
110    ylabel('Probability')
111    set(gca,'Fontsize',20)  %Set font size to 20
112 end
113
114 %% Autocorrelation of Daily Returns
115 average = 0;
116
117 % Find the average of the returns
118 for n = 1:dataSize-1
119     average = average + returns(n);
120 end
121 average = average / dataSize;
122
123 % Autocorrelation function
124 tauMax = 180;
125 ACReturns = zeros(1,tauMax);
126 n = 1;
127
128 % Using normal loop methods
129 for tau = 1:tauMax
```

```
130     numerator = 0;   %initializing the numerator
131     denominator = 0; %initializing the denominator
132     for m = 1+tau:dataSize - 1 %move through data by tao
133         numerator = numerator + ((abs(returns(m))-abs(average))...
134             *(abs(returns(m-tau))-abs(average)));
135         denominator = denominator + ...
136             ((abs(returns(m-tau))-abs(average))^2);
137     end
138     %put the top and bottom together to get the AC Returns Value
139     ACReturns(1,n) = numerator/denominator;
140     n = n + 1; %increase iteration variable
141 end
142
143
144 if IsingModel == 0
145     subplot(2,2,4)
146     plot(ACReturns)
147     title('Autocorrelation of Returns')
148     xlabel('Time Step (tau)')
149     ylabel('Autocorrelation')
150     set(gca,'Fontsize',20)  %Set font size to 20
151 else
152     subplot(1,3,3)
153     plot(ACReturns)
154     title('Autocorrelation of Returns')
155     xlabel('Time Step (tau)')
156     ylabel('Autocorrelation')
157     set(gca,'Fontsize',20)  %Set font size to 20
158 end
159
160 figure(3)
161 plot(returns)
162     datetick('x','yy') % Change x axis values so that it shows date
163     title('Ising Model Returns')
164     xlabel('Time (years)')
165     ylabel('Return')
166     set(gca,'Fontsize',20)  %Set font size to 20
167     axis([0 10100 -.25 .25])
```