1995

Modeling stock market returns with local iterated function systems

Douglas E. Galarus

The University of Montana

Let us know how access to this document benefits you.
Follow this and additional works at: https://scholarworks.umt.edu/etd

Recommended Citation
https://scholarworks.umt.edu/etd/4825

This Thesis is brought to you for free and open access by the Graduate School at ScholarWorks at University of Montana. It has been accepted for inclusion in Graduate Student Theses, Dissertations, & Professional Papers by an authorized administrator of ScholarWorks at University of Montana. For more information, please contact scholarworks@mso.umt.edu.
The University of MONTANA

Permission is granted by the author to reproduce this material in its entirety, provided that this material is used for scholarly purposes and is properly cited in published works and reports.

** Please check "Yes" or "No" and provide signature **

Yes, I grant permission
No, I do not grant permission

Author's Signature ____________________

Date ________________________

Any copying for commercial purposes or financial gain may be undertaken only with the author's explicit consent.
Modeling Stock Market Returns with Local Iterated Function Systems

By

Douglas E. Galarus

B.A., University of Montana, 1989

M.A.T., University of Montana, 1993

Presented in partial fulfillment of the requirements for the degree of Master of Science University of Montana

1995

Approved by
Chairman, Board of Examiners

Dean, Graduate School

November 22, 1995

Date
Modeling Stock Market Returns with Local Iterated Function Systems

Director: Alden Wright

Fractal Image Compression is a lossy image compression technique developed by Michael Barnsley and others in the past decade. It is used commercially and is comparable to if not better than other methods. Fractal Structure in capital markets was discovered by Benoit Mandelbrot and lead to the work that established him as the father of Fractal Geometry, which allows us to describe the shapes of nature.

Data compression can be thought of as a form of data modeling and data that exhibits certain fractal properties can be modeled with Fractal Image Compression techniques using Local Iterated Function Systems. Such models can then be used to interpolate / extrapolate data. The performance of these types of data prediction is examined with a degree of skepticism due to the inherent unpredictability of chaotic systems.
Table of Contents

INTRODUCTION
Providence

FROM FRACTAL IMAGE COMPRESSION TO DATA MODELING
A New Application of a New Application

CHAOS AND FRACTALS
The Tools of the Model

ITERATED FUNCTION SYSTEMS AND FRACTAL IMAGE COMPRESSION
A Method for Describing and Constructing Fractals

MODELING FINITE SEQUENCES WITH LOCAL ITERATED FUNCTION SYSTEM CONTRACTION MAPPINGS
Background on Metric Spaces and Contraction Mappings
The Domain - Range Method for Finding Contraction Mappings (Local Iterated Function Systems)

THE METHOD AT WORK
Some Simple Examples
A More Complicated Example
A Closer Look and Some Problems with the Model
Refinements of the Method

JUSTIFICATION FOR THE METHOD
Fractal Structure and the Use of Logarithmic Returns with Stock Market Data
Fractal Structure in Dow Returns

TESTING THE METHOD
CONCLUSION

Where has Providence Lead Me?
List of Tables and Figures

STORAGE REQUIRED FOR 300 DPI 4 IN X 6 IN IMAGES BY TYPE 5
THE CHAOS GAME 9
INITIAL GAME POINTS FOR A RUN OF THE CHAOS GAME 10
GAME POINTS FROM THE CHAOS GAME 11
THE SIERPINSKI TRIANGLE 12
ITERATED FUNCTION SYSTEM FOR THE SIERPINSKI TRIANGLE 13
ITERATION VALUES FOR A LOCAL IFS MODEL OF <5,8,11,14,17,20,23,26> 33
ITERATION VALUES CONVERGING TO THE SEQUENCE <1,2,3,4,5,6,7,8> 33
ITERATION VALUES, SHOWN IN SEPARATE GRAPHS, CONVERGING TO THE SEQUENCE <5,8,11,14,17,20,23,26> 34
INTERPOLATING AND EXTRAPOLATING VALUES 35
IBM CLOSING VALUES - MARCH 4, 1991 THROUGH JANUARY 4, 1994 41
IBM CLOSING VALUES: 3/4/91-1/4/94 42
DOMAIN VALUES (EVERY 5TH CLOSING VALUE) 43
MAPPINGS PARAMETERS AND ERRORS 44
MODEL OF IBM DATA 45
DATA VS. MODEL 46
ERROR 46
THE FIRST SUBMAPPING 47
FIRST DOMAIN-RANGE PAIR 48
DOMAIN TO RANGE MAPPING 49
DOMAIN TO RANGE MAPPING AND MODEL VALUES 49
RETURNS COMPOUNDED ANNUALLY AT 10% 51
RANDOM WALK TIME-SERIES 54
INCREMENTS AND SCALED INCREMENTS 55
CUMULATIVE INCREMENTS AND SCALED CUMULATIVE INCREMENTS 56
LEFT TAIL CUMULATIVE FREQUENCIES: LAG=1,2,4 57
RIGHT TAIL CUMULATIVE FREQUENCIES: LAG=1,2,4 58
COMPUTATION OF FRACTAL DIMENSION OF TIME SERIES 59
LOG-LOG PLOT OF VALUES FOR COMPUTING FRACTAL DIMENSION - RANDOM WALK 60
DOW VALUES: 1/2/30 - 12/29/89 62
LOGARITHMS OF DOW VALUES: 1/2/30 - 12/29/89 62
SCALING IN LEFT TAILS OF DOW RETURNS: LAG=1,2,4,8 63
SCALING IN RIGHT TAILS OF DOW RETURNS: LAG=1,2,4,8 64
LOG-LOG PLOT OF VALUES FOR COMPUTING FRACTAL DIMENSION - DOW RETURNS 64
COMPARISON OF LEFT AND RIGHT TAILS: LAG=1 65
COMPARISON OF LEFT AND RIGHT TAILS WITH NORMAL DISTRIBUTION 66
DOW LEFT TAILS: LAG=1,2,4,8,16,32 67
DOW RIGHT TAILS: LAG=1,2,4,8,16,32 67
SCRAMBLED DOW LEFT TAILS: LAG=1,2,4,8,16,32 68
SCRAMBLED DOW RIGHT TAILS: LAG=1,2,4,8,16,32 68
CORRELELOGRAM FOR 1000 NORMAL INCREMENTS 69
DOW DAILY INCREMENTS VERSUS SUCCESSIVE DAILY INCREMENTS 70
CORRELELOGRAM FOR DOW DAILY INCREMENTS 71
CORRELELOGRAM FOR DOW WEEKLY INCREMENTS: LAG=5 71
CORRELELOGRAM FOR DOW MONTHLY INCREMENTS: LAG=20 71
BIASED RANDOM WALK: NORMAL(0.1,0) 72
LEFT TAILS: NORMAL(0.1,0) 73
RIGHT TAILS: NORMAL(0.1,0) 73
LOGARITHMS OF 16,000 DOW VALUES, 1930-1989 75
MODEL OF LOGARITHMIC VALUES: 20 DAY RANGES 75
MODEL ERROR: 20 DAY RANGES 76
AX+B VALUES AND RANGE VALUES 76
GRAPH OF AX+B VALUES AND LAST RANGE VALUES 77
EXTRAPOLATED VALUES AND ACTUAL RANGE DATA VALUES 78
GRAPH OF EXTRAPOLATED VALUES VERSUS ACTUAL RANGE DATA VALUES 79
LORENZ'S DIVERGING WEATHER PATTERNS (THE BUTTERFLY EFFECT) 79
MODEL OF LOGARITHMIC VALUES: 100 DAY RANGES 80
MODEL ERROR: 100 DAY RANGES 81
GRAPH OF AX+B VALUES AND RANGE VALUES: 100 DAY RANGES 82
PREDICTED VALUES USING MODIFIED LEAST-SQUARES METHOD 83
PREDICTIONS MADE USING LEAST SQUARES MATCHES WITH NO RESTRICTIONS 85
USING MAPPINGS TO PREDICT NEXT DAY UP OR DOWN 86
TESTING THE METHOD AGAINST RANDOM PREDICTIONS 87
Acknowledgments

I owe a world of thanks to my mom and dad for their encouragement. They grew up in hard times and were never given the opportunities that they've given me. Special thanks to them for the work ethic they have taught me.

To my little dog and companion, I promise to spend more time with you now.

To my friends and coworkers, thanks for putting up with me and my moods while trying to work full-time and finish this project.

To my late brother Mike, who passed away last year while I was in the middle of this, I wish I could have done more for you. You taught me perseverance in difficult situations. Rest in Peace.

Finally, thanks to Dr. Alden Wright and others at the University of Montana who have helped and encouraged me through the years. I've know Dr. Wright and Dr. George McRae since I was in high school and both have always been there when I needed help.
Introduction

Providence

When two full years had passed, Pharaoh had a dream: He was standing by the Nile, when out of the river there came up seven cows, sleek and fat, and they grazed among the reeds. After them, seven other cows, ugly and gaunt, came up out of the Nile and stood beside those on the riverbank. And the cows that were ugly and gaunt ate up the seven sleek, fat cows. Then Pharaoh woke up. He fell asleep again and had a second dream: Seven heads of grain, healthy and good, were growing on a single stalk. After them, seven other heads of grain sprouted—thin and scorched by the east wind. The thin heads of grain swallowed up the even healthy, full heads. Then Pharaoh woke up; it had been a dream.

In the morning his mind was troubled, so he sent for all the magicians and wise men of Egypt. Pharaoh told them his dreams, but no one could interpret them for him. Then the chief cupbearer said to Pharaoh, "Today I am reminded of my shortcomings. Pharaoh was once angry with his servants, and he imprisoned me and the chief baker in the house of the captain of the guard. Each of us had a dream the same night, and each dream had a meaning of its own. Now a young Hebrew was there with us, a servant of the captain of the guard. We told him our dreams, and he interpreted them for us, giving each man the interpretation of his dream. And things turned out exactly as he interpreted them to us: I was restored to my position, and the other man was hanged."

So Pharaoh sent for Joseph, and he was quickly brought from the dungeon. When he had shaved and changed his clothes, he came before Pharaoh. Pharaoh said to Joseph, "I had a dream, and no one can interpret it. But I have heard it said of you that when you
hear a dream you can interpret it." "I cannot do it," Joseph replied to Pharaoh, "but God will give Pharaoh the answer he desires." Then Pharaoh said to Joseph, "In my dream I was standing on the bank of the Nile, when out of the river there came up seven cows, fat and sleek, and they grazed among the reeds. After them, seven other cows came up—scrawny and very ugly and lean. I had never seen such ugly cows in all the land of Egypt. The lean, ugly cows ate up the seven fat cows that came up first. But even after they ate them, no one could tell that they had done so; they looked just as ugly as before. Then I woke up. "In my dreams I also saw seven heads of grain, full and good, growing on a single stalk. After them, seven other heads sprouted—withered and thin and scorched by the east wind. The thin heads of grain swallowed up the seven good heads. I told this to the magicians, but none could explain it to me."

Then Joseph said to Pharaoh, "The dreams of Pharaoh are one and the same. God has revealed to Pharaoh what he is about to do. The seven good cows are seven years, and the seven good heads of grain are seven years; it is one and the same dream. The seven lean, ugly cows that came up afterward are seven years, and so are the seven worthless heads of grain scorched by the east wind: They are seven years of famine. "It is just as I said to Pharaoh: God has shown Pharaoh what he is about to do. Seven years of great abundance are coming throughout the land of Egypt, but seven years of famine will follow them. Then all the abundance in Egypt will be forgotten, and the famine will ravage the land. The abundance in the land will not be remembered, because the famine that follows it will be so severe. The reason the dream was given to Pharaoh in two forms is that the matter has been firmly decided by God, and God will do it soon. "And now let Pharaoh look for a discerning and wise man and put him in charge of the land of Egypt. Let Pharaoh appoint commissioners over the land to take a fifth of the harvest of Egypt during the seven years of abundance. They should collect all the food of these good years that are coming and store up the grain under the authority of Pharaoh, to be
kept in the cities for food. This food should be held in reserve for the country, to be used during the seven years of famine that will come upon Egypt, so that the country may not be ruined by the famine."

The plan seemed good to Pharaoh and to all his officials. So Pharaoh asked them, "Can we find anyone like this man, one in whom is the spirit of God?" Then Pharaoh said to Joseph, "Since God has made all this known to you, there is no one so discerning and wise as you. You shall be in charge of my palace, and all my people are to submit to your orders. Only with respect to the throne will I be greater than you." So Pharaoh said to Joseph, "I hereby put you in charge of the whole land of Egypt." Then Pharaoh took his signet ring from his finger and put it on Joseph's finger. He dressed him in robes of fine linen and put a gold chain around his neck. He had him ride in a chariot as his second-in-command, and men shouted before him, "Make way!" Thus he put him in charge of the whole land of Egypt. Then Pharaoh said to Joseph, "I am Pharaoh, but without your word no one will lift hand or foot in all Egypt."

I've chosen the story of Joseph and his interpretation of the Pharaoh's dreams, found in Genesis 40:1-44, as an introduction to my work for several reasons. Joseph was able to interpret the dreams that foretold the future while none of Pharaoh's "magicians" could do so. My goal is similar, though certainly not as divine. I want to predict the stock market. There is no shortage of Wall Street "magicians" and amateurs like me with similar aspirations. Some are successful from time to time. But, none are successful all the time and their failure is demonstrated in the loss of their own money or, worse, the loss of money entrusted to them. My goal is admittedly a lofty one. Nevertheless, I will try.

One of the first books I read that had more text than pictures was given to me as a gift from my second grade teacher and her husband. It is titled "Joseph and the Coat of Many Colors" by Lavinia Derwent and is dear to me both because it was a gift and because of the story it told. It seems fitting that I use part of that story, giving it a place at a concluding point in my education since it was so important to me at the start of
my formal education. This reminiscing may seem irrelevant but a recurring theme in my work is the search for events in the past that influence and even foretell the future.

My association of the story of Joseph with the stock market is not original. Mandelbrot made that connection some time ago and made another biblical reference to the story of Noah in his work with fractals and the stock market [Mandelbrot, 1983]. But as mentioned, this particular story was dear to me long before I began working on my thesis and I like to think the connection is more than coincidence.

This summer a friend’s wife, Becky, asked me to describe my thesis. Since she is not a mathematician or math teacher I proceeded to ask her if she knew what fractals were. Her husband, Paul, a mathematics professor, had described them to her before so she knew of some connection to nature. As I described that my worked depended on a relationship between fractals and the stock market, Paul said that he believes fractals are the language that God uses to create nature.

Several months earlier at a convention in Boston he made a similar intriguing statement. I decided to ask if he would like to meet a colleague and me at a popular restaurant. It was close to dinner time and we expected that the restaurant would soon be full. Fortunately, and unknown to me, he was staying at the hotel that housed the restaurant and proceeded to make reservations for us. After about a 15 minute walk from our hotel, my colleague and I arrived and were greeted by Paul who said our table had just become available. Later, as we noticed a crowd gathering and the waiting line growing longer, we discussed how fortunate our timing was. Paul said it was neither luck nor coincidence, but providence.

After reflecting on this and other experiences, I’ve decided that providence may be an underlying order hidden in the chaos of life. Providence is defined as “the care or benevolent guidance of God or nature.” Certainly the story of Joseph is a story of providence. My work is a story of the search for order in the stock market which is described as chaotic by many [Gleick, 1987]. I can only hope that I am able to recognize and interpret that order if I find it, as Joseph interpreted Pharaoh’s dreams when others failed. Perhaps providence will guide me.
From Fractal Image Compression to Data Modeling

A New Application of a New Application

Michael Barnsley and Alan Sloan published an article called “A Better Way to Compress Images” in the January 1988 edition of Byte magazine. The subtitle, “Mathematics is providing a novel technique for achieving compression ratios of 10,000 to 1 -- and higher” was intriguing and hard to believe [Barnsley and Sloan, 1988]. A brief discussion of image storage and compression may clarify why this claim was both intriguing and astounding.

First consider the amount of storage necessary for a 4 in x 6 in image at 300 x 300 dpi (dots per inch) resolution. Such an image contains 2,160,000 dots (pixels) and each pixel requires an amount of storage that depends on the type of image. Several possibilities are shown below.

<table>
<thead>
<tr>
<th>type of image</th>
<th>bits per pixel</th>
<th>storage required (bits)</th>
<th>storage required (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>monochrome</td>
<td>1</td>
<td>2,160,000</td>
<td>270,000</td>
</tr>
<tr>
<td>16 shade greyscale</td>
<td>4</td>
<td>8,640,000</td>
<td>1,080,000</td>
</tr>
<tr>
<td>256 shade greyscale</td>
<td>8</td>
<td>17,280,000</td>
<td>2,160,000</td>
</tr>
<tr>
<td>$2^{24} = 16,777,216$ colors</td>
<td>24</td>
<td>51,840,000</td>
<td>6,480,000</td>
</tr>
</tbody>
</table>

In the extreme case (full color), this small image requires the equivalent of five 1.44 MB floppy disks for storage. Compare this to the amount of storage for larger images or motion pictures with 30 frames per second where each frame is an image. The amount of storage required becomes overwhelming and soon surpasses the capacity of the largest disk drives. The usefulness and necessity of compression should now be clear. If Barnsley’s and Sloan’s claim of 10,000 to 1 compression ratios is accurate, then the example image could fit on a single floppy disk. Even as many as 2000 such images could be stored on a single floppy so the task of storing images could become more manageable.

A technique is described vaguely in the Byte article by Barnsley and Sloan. The technique used in practice is different from what they describe. The distinctions will not be made here though details of a method that
can be used in practice will be given later. Louisa Hanson provided an update and further description of Fractal Image Compression in a 1993 Byte magazine article [Hanson, 1993]. She gives some of the following information about image compression in general. Fractal Image Compression belongs to a category of methods known as lossy. The other category of compression methods is lossless. These terms can apply whether the data is image data or some other type (sound, etc.) A lossless compression method always produces decompressed data identical to the original. An example of lossless data compression is the method used in PKWare’s PKZip. Similar lossless data compression methods are used by disk compression utilities for obvious reasons. But lossless methods achieve small compression ratios, usually about 2 to 1.

Lossy compression methods yield greater compression ratios but generally do not produce decompressed data identical to the original data. Instead, the decompressed data differs from the original data by some error and, depending on the type of data, such an error may be acceptable. For example, slight differences in the color or intensity of pixels between an original image and a decompressed image may be unnoticeable. Generally, allowing greater error results in a greater compression ratios. In other words, the cost of greater compression is poorer quality of the decompressed data.

I'm not sure when I first heard of Fractal Image Compression. I did not read the 1988 Byte Magazine article at the time of publication. Several years ago, I read or heard that Fractal Image Compression would be used in a new product from Microsoft called Encarta, a complete Encyclopedia with pictures, maps, sound, and full motion video on CD-ROM. Though I don't recall when I first heard of this method, I do remember that the idea intrigued me and involved my major academic areas, Mathematics and Computer Science. I wanted to learn more about this "hot" new topic and was sure I could find an aspect of it that was suitable for research in a thesis.

When I first proposed the idea of studying Fractal Image Compression to my advisor, Dr. Alden Wright, he was receptive but wanted to make sure I could define my objectives clearly and narrow the topic of my thesis so it would be manageable. Neither of us knew much about Fractal Image Compression and I knew it would be necessary to do preliminary research in chaos theory and fractals while also researching data.
compression. This was necessary since detailed references on Fractal Image Compression were not available at the time but were anticipated -- they were advertised by publishers but did not come into print at the promised times. Barnsley soon published a book on the subject [Barnsley and Hurd, 1993] and Yuval Fisher followed [Fisher, 1995].

So far I've made no mention of the stock market which will supply the data analyzed in my research. This is where Fractal Image Compression leads to stock market modeling. At some point, Dr. Wright made the following observation which ultimately defined the topic of my thesis. He remarked that in general, data compression is a type of data modeling where the compressed form of the data is a model of the original data. This was not clear to me since I thought of compressed data as encrypted information that could only be deciphered through the decompression process. I guess I pictured in my mind a .zip file that was the compressed form of a text file. While the original text file could be meaningfully read in a text editor or in print, the compressed .zip file was meaningless when viewed the same way. Its use as a model seemed restricted only to the compression application.

Dr. Wright added that the techniques of Fractal Image Compression or variations of them could possibly be applied to model some type of data in a meaningful way. It is widely known through the work of Mandelbrot, Barnsley, and others, that fractals can model objects in nature like trees, mountains, clouds, and coastlines that cannot be described well using traditional Euclidean Geometry [Mandelbrot, 1983], [Barnsley, 1988]. My preliminary goal was now to find a way to use a fractal model to gain insight into the workings of some object or system. The object became time-series data exhibiting chaotic properties; in particular, stock market data.

As mentioned, I didn't see a connection between data compression and data modeling. My skepticism was fueled further by what I had read and learned about chaotic systems. If the purpose of the model is to predict rather than just describe, then by the very nature of a chaotic system, such a goal is doomed to failure. I will call the stock market a chaotic system without further explanation or justification at this time. For a similar characterization see [Gleick, 1987].
At the time, I argued the following to justify my skepticism. It is widely known that weather predictions are accurate at most one to two weeks into the future. A phenomena called the Butterfly Effect was discovered by Lorenz in the 1960s and roughly says that a disturbance as small as a butterfly flapping its wings could affect future global weather patterns [Gleick, 1987]. Another description of this is “sensitivity to initial conditions.” Furthermore, short-term predictions are often wrong -- it rains when the weather service predicts little or no chance of precipitation, despite extensive data collection and modeling. Likewise, earthquakes have been studied for years and areas at greatest risk have been identified. Yet, short term predictions of earthquakes can not be made accurately. Predictions generally take the form of “The probability of an earthquake of magnitude ___ or greater in the ___ area is ___ in the next ___ years.” where the region is relatively large, the probability is moderate, and the time frame is long. It is difficult and perhaps impossible to predict an earthquake in a certain region within a number of days. This may mean that the only types of predictions that can be made are related to risk.

Armed with skepticism and lack of understanding, I proceeded to refine the goal of my thesis to that described above -- to model stock market data using fractals and apply the techniques of Fractal Image Compression. With this in mind, I will now describe the details of the work I have done.
Chaos and Fractals

The Tools of the Model

A brief introduction to chaos and fractals seems appropriate at this time. This introduction will neither be thorough or precise. It is meant to provide enough information for the basics of Fractal Image Compression. Defining the words “fractal” and “chaos” is a problem. Example seems best to give an intuitive idea of what they are and how they are related.

Barnsley [Barnsley, 1988] describes a process similar to the one below as the Chaos Game. To those who have never seen it, the result is surprising and shows a relationship between chaos and fractals.

The Chaos Game

Setup:

- Choose three non-collinear points and identify them as 0, 1, and 2. They form the vertices of a triangle.
- Choose an arbitrary point in the plane. This will be the starting game point.
- Have a die or random number generator to generate the outcomes 0, 1, and 2 randomly with equal probabilities. (Equal probabilities are not necessary but work best.) The outcomes correspond to the vertices of the triangle.

The Game:

- Roll the die. Suppose the outcome is 1 corresponding to vertex 1. Then the next game point is the midpoint between the current game point and vertex 1. If 0 or 2 had been rolled, the new game point would have been found similarly using vertex 0 or vertex 2.
- Continue rolling the die generating and recording game points as described.
Example:

Triangle vertices 0, 1, and 2 are chosen as (0,0), (1,0), and (0,1) respectively with (0.5,0.5) as the starting game point. The resulting game points for the sequence of random digits 2, 0, 1 and 0 are shown below. Only the game points would normally be shown but the triangle vertices and segments that connect the game points are shown to help demonstrate the procedure.

**Initial Game Points for a Run of the Chaos Game**

One might attempt to predict what would happen if this process continued. But there are several types of predictions that could be attempted. Predicting the location of the next game point or any specific game point in the future is impossible since it depends on chance, the current game point, and consequently, all past game points. There are three possible locations for the next game point, given the current game point, each with equal probability. Looking further into the future, the number of possibilities increases exponentially. But as the process continues a pattern emerges.
This is called the chaos game because there is randomness and unpredictability in the location of specific game points. However, there is a pattern formed by the game points that does not appear random at all. In fact, the same pattern will be formed regardless of the chosen starting point and the sequence of random digits, though it may be necessary to disregard early game points. In one sense the results of the game are unpredictable while in another they appear very predictable.

The game points begin to form a fractal called the Sierpinski Triangle. It is also called a chaotic attractor because the points generated by the chaotic process are attracted to the points that form the pattern regardless of the starting point and sequence of random numbers. Again note that the pattern is formed from an arbitrary initial point and a random sequence. Each new point depends on all previous points. But, the same pattern emerges regardless of the starting point and the random sequence. While local prediction (like the specific location of particular game points) are not possible, global prediction is possible using the fractal pattern that emerges.
The Sierpinski Triangle is a set of points with a triangle-shaped outline containing three miniature (similar) copies of itself with sides exactly half the length of those of the original. Objects that contain smaller similar copies of themselves are called self-similar.

![The Sierpinski Triangle](image)

Since it is self-similar and contains more than one point, the Sierpinski Triangle must contain an infinite number of points and an infinite number of miniature copies of itself of varying sizes as can be seen above. Magnifying these miniature copies reveal the same details as found in the Sierpinski Triangle though at a smaller scale. This is, by definition, self-similarity.

Certainly the term fractal has not been formally defined here. But, an important characteristic of fractals (self-similarity) is given in place of a formal definition. Many books on the topic avoid defining fractals. In general, they have self-similarity of some kind. They contain miniature copies of themselves or miniature copies with similar characteristics to the whole. More detailed definitions use such terms as fractal dimension and exclude figures such as lines and squares with their interiors which are also self-similar.

Rigorous definition is not necessary for my purposes. Note that self-similarity has been defined using strict geometric similarity here. Later, another form of similarity and self-similarity will be presented and used.
Iterated Function Systems and Fractal Image Compression

A Method for Describing and Constructing Fractals

Barnsley and others use systems of mappings that describe and generate figures like the Sierpinski Triangle. Such systems are closely related to the process used to find the game points in the Chaos Game. Recall the vertices in the example and suppose the current game point is the point \((x, y)\). The mappings below determine the coordinates of the next game point depending on the outcome of the roll of the die.

**Iterated Function System for the Sierpinski Triangle**

<table>
<thead>
<tr>
<th>outcome</th>
<th>mapping giving the next game point</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(w_0(x,y) = (0.5x, 0.5y))</td>
</tr>
<tr>
<td>1</td>
<td>(w_1(x,y) = (0.5x + 0.5, 0.5y))</td>
</tr>
<tr>
<td>2</td>
<td>(w_2(x,y) = (0.5x, 0.5y + 0.5))</td>
</tr>
</tbody>
</table>

Such a collection of mappings is called an iterated function system or IFS. Barnsley describes iterated function systems in his work so they will not be discussed in great detail here [Barnsley, 1988]. Simply stated, they are collections of affine transformations that when applied as in the chaos game, result is an attractor which may be a fractal like the Sierpinski Triangle. Note that no limit is placed on the number of mappings in the system and no vertices are given as were for the Chaos Game since they are implicit in the mappings.

Barnsley vaguely describes the use of IFSs for image compression in the 1988 Byte magazine [Barnsley and Hurd, 1988]. The problem of implementation is determining IFS mappings that match an image (if they exist), which is sometimes called the inverse problem. Barnsley does not describe implementation in the article so there may be cause for skepticism. As mentioned earlier, the method he alludes to is not the
method that is commercially used and there is doubt that an automated method for finding such IFSs even exists. A related method will be described and used later.

Consider the IFS for the Sierpinski Triangle as a compressed representation of that triangle. The IFS can be rewritten as

\[
\begin{align*}
    w_0(x,y) &= (0.5x + 0, 0.5y + 0) \\
    w_1(x,y) &= (0.5x + 0.5, 0.5y + 0) \\
    w_2(x,y) &= (0.5x + 0, 0.5y + 0.5)
\end{align*}
\]

so each transformation is of the form \( w_n(x,y) = (a_n x + b_n, c_n y + d_n) \). It suffices to know that there are three mappings and the coefficients \((a_n, b_n, c_n, d_n)\) associated with each are \((0.5, 0, 0.5, 0)\), \((0.5, 0.5, 0.5, 0)\), and \((0.5, 0, 0.5, 0.5)\) so the Sierpinski Triangle is defined completely by 12 values. Note that the mappings could take more complicated forms requiring more coefficients.

Now consider a finite approximation of the Sierpinski Triangle displayed or printed at a fixed resolution. This is only an approximation since the Sierpinski Triangle contains infinitely many dimensionless points as compared to a finite number of pixels that are rectangles or dots with length and width. Such an image can be generated using the IFS above, the chaos game, or another iterative method and a change in coordinates to match a pixel coordinate system. (This is how the images shown earlier were generated.) It could be printed on a laser printer in a square inch at 300 x 300 dpi where up to 90,000 dots (pixels) could be used. The storage required for 90,000 monochrome pixels is 90,000 bits or 11,250 bytes. Alternatively the parameters defining the IFS for the Sierpinski Triangle can be stored far more efficiently. Assuming 8 bytes are used to store a floating point number then 96 bytes suffice and the resulting compression ratio is 11,250:96 or about 117:1 using the IFS to "compress" the individual pixel data.

Now suppose the image was printed on the useable portion of letter sized paper at 720 x 720 dpi. Then there would be 6,750,000 dots (pixels) requiring 6,750,000 bits or 843,750 bytes. The compression ratio is now 843,750:96 or about 8789:1, which is near the 10,000:1 claim made by Barnsley and Sloan. It can be
seen that the compression ratio can be made arbitrarily large choosing higher resolutions and / or larger output sizes.

Such compression ratios are deceiving since they depend on the resolution. But, since maximizing compression is not the major objective of my work, this is a minor observation. More important is the fact that the IFS is resolution independent. As resolution is increased, the IFS yields greater detail while requiring a fixed amount of storage (96 bytes for the example). The IFS contains all necessary information about the Sierpinski Triangle independent of resolution. It also contains information about the self-similarity in the Sierpinski Triangle.

Consider the vertices of the original triangle from the Chaos Game: (0,0), (1,0), and (0,1). The images of the vertices under the mappings in the IFS reveals the self-similarity of the figure.

\[
\begin{align*}
w_0(0,0) &= (0,0) & w_0(1,0) &= (0.5,0) & w_0(0,1) &= (0,0.5) \\
w_1(0,0) &= (0.5,0) & w_1(1,0) &= (1,0) & w_1(0,1) &= (0.5,0.5) \\
w_2(0,0) &= (0,0.5) & w_2(1,0) &= (0.5,0.5) & w_2(0,1) &= (0,1) \\
\end{align*}
\]

\(w_0\) maps the original triangle to the lower left triangle:

\(w_1\) maps the original triangle to the lower right triangle:
$w_2$ maps the original triangle to the upper left triangle:

These three triangles outline the three portions of the Sierpinski Triangle that are similar to the whole with sides half the length of the original. The IFS reveals a property that is resolution independent, self-similarity, and can be used to generate detail at any resolution.

In contrast, consider a naive comparison with the result of compression using PKZip of a discrete approximation of the Sierpinski Triangle at 256 x 256 resolution. A text file was created with 256 rows and 256 columns containing X’s to represent points in the triangle and O’s to represent points not in the triangle. The file as printed would appear similar to the 16 x 16 version shown below:

Disregarding end-of-line characters, etc. there are 65,536 characters in the file. 6561 of which are X’s and 58975 are O’s. The total size of the actual file is 66,048 bytes. Certainly this is an inefficient representation, using a byte for each “monochrome” pixel. A bit per pixel would be much more efficient. But, the point is
to determine how well this lossless compression method works on the data, and PKZip operates on bytes, not bits. The resulting compressed file using PKZip v.2.04 is 1,548 bytes in length giving a compression ratio of 66,048:1,548 or approximately 43:1. This seems good -- far better than 2:1. But a good compression ratio should be anticipated since there are far more O's than X's, like symbols tend to cluster together, and there are only two different symbols.

To demonstrate a reason for my early misunderstanding of the comparison of compression to modeling, the figure below shows what the .zip file might look like in a text editor.

The file looks like gibberish -- unreadable without the details of file format, etc. Even if the format was given, the compressed file would probably give little useful information about the geometric structure the figure represents, just the characters within it. Definitely it only describes the figure at the given resolution with no information useful for other resolutions. The IFS clearly appears far more useful for describing (modeling) the figure and provides better compression.

The use of the Sierpinski Triangle as an example serves for little more than demonstrative purposes since an IFS models it perfectly. Furthermore, the IFS is known. No procedure was given for finding an IFS for an arbitrary image. Nor is there evidence that such an IFS will always exist. The Sierpinski Triangle has perfect self-similarity. An arbitrary image will most likely not have self-similarity. Even if it does, detecting it is not an easy task. These problems are addressed in critiques of Barnsley's original claims. Again, they
are not the main concerns of my work though, so I present little more than an awareness of these difficulties
is presented.

Instead, what follows is a general description of the background for and a method similar to what Barnsley
calls the Fractal Transform method, a method that is used commercially since it can be automated without
the problems mentioned above. This method was discovered by a graduate student of Barnsley's [Jacquin,
1993]. My description is presented partly for simplicity for the one-dimensional version of the problem.
Images would be considered at least two-dimensional while motion video would be at least three-
dimensional. Time-series data is one-dimensional, so it fits the one-dimensional representation and can be
covered in detail. What follows should provide a minimal mathematical background for explaining this
method and why it works.
Modeling Finite Sequences with Local Iterated Function System Contraction Mappings

Background on Metric Spaces and Contraction Mappings

Time series data or other ordered sets of data can be represented as finite sequences of the form

\[ s = (s_1, \ldots, s_n) \]  

A real-valued sequence can be represented as an ordered n-tuple \( s = (s[1], \ldots, s[n]) \) in \( R^n \) which is a metric space when associated with a metric (distance function). The following definitions and background will be used for a method to model such n-tuples representing ordered finite sequences.

These proofs and theorems can be found in most analysis texts and [Barnsley, 1988].

A metric space is a set \( X \) and a function (metric) \( d : X \times X \to R \) such that for all \( x, y, z \in X \),

(a) \( d(x, x) = 0 \)

(b) If \( x \neq y \), then \( d(x, y) > 0 \).

(c) \( d(x, y) = d(y, x) \)

(d) \( d(x, z) \leq d(x, y) + d(y, z) \)

There are several common metrics associated with \( R^n \).
\((\mathbb{R}^n, d_\infty)\) is a metric space where \(d_\infty(x, y) = \max_{k=1}^{\infty} \{|x[k] - y[k]|\}\).

Proof: Let \(x, y, z\) be arbitrary elements of \(\mathbb{R}^n\).

\[
d_\infty(x, x) = \max_{k=1}^{\infty} \{|x[k] - x[k]|\} = 0.
\]

If \(x \neq y\), then there exists an \(i\) where \(1 \leq i \leq n\) and \(x_i - y_i \neq 0\).

Then \(|x[i] - y[i]| > 0\) and \(d_\infty(x, y) > 0\).

\[
d_\infty(x, y) = \max_{k=1}^{\infty} \{|x[k] - y[k]|\} = \max_{k=1}^{\infty} \{|y[k] - x[k]|\} = d_\infty(y, x).
\]

\[
d_\infty(x, z) = \max_{k=1}^{\infty} \{|x[k] - z[k]|\} = \max_{k=1}^{\infty} \{|x[k] - y[k] + y[k] - z[k]|\}
\]

\[
\leq \max_{k=1}^{\infty} \{|x[k] - y[k]| + |y[k] - z[k]|\} \leq \max_{k=1}^{\infty} \{|x[k] - y[k]|\} + \max_{k=1}^{\infty} \{|y[k] - z[k]|\}
\]

\[
= d_\infty(x, y) + d_\infty(y, z)
\]

\((\mathbb{R}^n, d_1)\) is a metric space where \(d_1(x, y) = \sum_{k=1}^{n} |x[k] - y[k]|\).

Proof: Let \(x, y, z\) be arbitrary elements of \(\mathbb{R}^n\).

\[
d_1(x, x) = \sum_{k=1}^{n} |x[k] - x[k]| = 0.
\]

If \(x \neq y\), then there exists an \(i\) where \(1 \leq i \leq n\) and \(x[i] - y[i] \neq 0\).

Then \(|x[i] - y[i]| > 0\) and \(d_1(x, y) > 0\).

\[
d_1(x, y) = \sum_{k=1}^{n} |x[k] - y[k]| = \sum_{k=1}^{n} |y[k] - x[k]| = d_1(y, x).
\]

\[
d_1(x, z) = \sum_{k=1}^{n} |x[k] - z[k]| = \sum_{k=1}^{n} |x[k] - y[k] + y[k] - z[k]|
\]

\[
\leq \sum_{k=1}^{n} \{|x[k] - y[k]| + |y[k] - z[k]|\} = \sum_{k=1}^{n} |x[k] - y[k]| + \sum_{k=1}^{n} |y[k] - z[k]|
\]

\[
= d_1(x, y) + d_1(y, z)
\]
\((\mathbb{R}^n, d_2)\) is a metric space where 
\[
d_2(x, y) = \sqrt{\sum_{k=1}^{n} (x[k] - y[k])^2}.
\]

Proof: Let \(x, y, z\) be arbitrary elements of \(\mathbb{R}^n\).

\[
d_2(x, x) = \sqrt{\sum_{k=1}^{n} (x[k] - x[k])^2} = 0.
\]

If \(x \neq y\), then there exists an \(i\) where \(1 \leq i \leq n\) and \(x_i - y_i \neq 0\).

Then \((x[i] - y[i])^2 > 0\) and \(d_2(x, y) > 0\).

\[
d_2(x, y) = \sqrt{\sum_{k=1}^{n} (x[k] - y[k])^2} = \sqrt{\sum_{k=1}^{n} (y[k] - x[k])^2} = d_2(y, x)
\]

\[
d_2(x, z) = \sqrt{\sum_{k=1}^{n} (x[k] - z[k])^2} = \sqrt{\sum_{k=1}^{n} (x[k] - y[k] + y[k] - z[k])^2}
\]

\[
\leq \sqrt{\sum_{k=1}^{n} (x[k] - y[k])^2} + \sqrt{\sum_{k=1}^{n} (y[k] - z[k])^2} \leq \sqrt{\sum_{k=1}^{n} (x[k] - y[k])^2} + \sqrt{\sum_{k=1}^{n} (y[k] - z[k])^2}
\]

\[
= d_2(x, y) + d_2(y, z)
\]

\(d_1\) is often called the lattice metric, the Manhattan metric, or the taxicab metric. \(d_2\) is the Euclidean metric.

Though these metrics and \(d_\infty\) seem quite different, they are related.
Two metrics $d$ and $D$ on a set $X$ are equivalent if there exist positive numbers $m$ and $M$ such that for all $x, y \in X$, \( m \cdot D(x, y) \leq d(x, y) \leq M \cdot D(x, y) \). This definition is symmetric since it follows that for all $x, y \in X$, \( \frac{1}{M} \cdot d(x, y) \leq D(x, y) \leq \frac{1}{m} \cdot d(x, y) \).

The metrics $d_1(x, y) = \sum_{k=1}^{n} |x[k] - y[k]|$, $d_2(x, y) = \sqrt{\sum_{k=1}^{n} (x[k] - y[k])^2}$, and $d_\infty(x, y) = \max_{k=1,n} \{|x[k] - y[k]|\}$ are equivalent on $\mathbb{R}^n$.

Proof: Let $x, y$ be arbitrary elements of $\mathbb{R}^n$.

\[
de_\infty(x, y) \leq d_1(x, y) \leq n \cdot d_\infty(x, y)
\]

\[
de_\infty(x, y) \leq d_2(x, y) \leq \sqrt{n} \cdot d_\infty(x, y)
\]

\[
\frac{1}{n} \cdot d_1(x, y) \leq d_2(x, y) \leq d_1(x, y)
\]

Following are several definitions that apply to metric spaces.

A mapping $f$ from a metric space $(X, d_X)$ to a metric space $(Y, d_Y)$ is continuous at a point $a \in X$ if for all $\varepsilon > 0$, there exists a $\delta > 0$ such that when $d_X(x, a) < \delta$ then $d_Y(f(x), f(a)) < \varepsilon$.

$f$ is continuous on $X$, or simply continuous, if it is continuous at every point in $X$.

If $\{x_k\}$ is a sequence in metric space $(X, d)$, then $\{x_k\}$ converges to $l$, written $\lim_{k \to \infty} x_k = l$ or $x_k \to l$, if for all $\varepsilon > 0$, there exists an $N$ such that if $k > N$ then $d(x_k, l) < \varepsilon$.

If the limit does not exist, the sequence diverges.

A sequence $\{x_k\}$ in a metric space $(X, d)$ is a Cauchy Sequence if for all $\varepsilon > 0$, there exists an $N$ such that if $i, j > N$ then $d(x_i, x_j) < \varepsilon$. 

22
Equivalent metrics act similarly on metric spaces.

If two metrics are equivalent then they produce the same Cauchy sequences, the same convergent sequences, and the same continuous functions.

Proof: Let \( d \) and \( D \) be equivalent metrics on set \( X \). Let \( m \) and \( M \) be positive numbers such that for all \( x, y \in X \), \( m \cdot D(x, y) \leq d(x, y) \leq M \cdot D(x, y) \).

Let \( \{x_k\} \) be a sequence that converges to \( l \) in metric space \( (X, D) \).

Then for all \( \varepsilon > 0 \), there exists an \( N \) such that if \( i > N \) then \( D(x_i, l) < \frac{\varepsilon}{M} \).

Then \( d(x_i, l) \leq M \cdot D(x_i, l) < M \cdot \frac{\varepsilon}{M} = \varepsilon \) and \( \{x_k\} \) converges to \( l \) in \((X, d)\).

Let \( \{x_k\} \) be a Cauchy sequence in \((X, D)\).

Then for all \( \varepsilon > 0 \), there exists an \( N \) such that if \( i, j > N \) then \( D(x_i, x_j) < \frac{\varepsilon}{M} \).

Then \( d(x_i, x_j) \leq M \cdot D(x_i, x_j) < M \cdot \frac{\varepsilon}{M} = \varepsilon \) and \( \{x_k\} \) is a Cauchy sequence in \((X, d)\).

Let \( f \) be a continuous function from \( X \) to \( X \) in \((X, D)\).

Then for all \( a \in X \), and \( \varepsilon > 0 \), there exists a \( \delta > 0 \) such that if \( D(x, a) < \delta \)
then \( D(f(x), f(a)) < \frac{\varepsilon}{M} \). Then \( d(f(x), f(a)) \leq M \cdot D(f(x), f(a)) < M \cdot \frac{\varepsilon}{M} = \varepsilon \).

Thus, \( f \) is a continuous function from \( X \) to \( X \) in \((X, d)\).

Since the definition of equivalent metrics is symmetric, this proof is also symmetric in \( D \) and \( d \).

In a metric space, every convergent sequence is a Cauchy sequence.

Proof: Let \( \{x_k\} \) be a sequence in metric space \((X, d)\) such that \( x_k \to l \), and let \( \varepsilon > 0 \) be given.

Then there exists an \( N \) such that when \( i > N \) \( \Rightarrow \) \( d(x_i, l) < \frac{\varepsilon}{2} \). Let \( i, j > N \).

Then \( d(x_i, x_j) \leq d(x_i, l) + d(l, x_j) < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon \) and \( \{x_k\} \) is a Cauchy sequence. However, the converse is not necessarily true. When it is true for a metric space, the metric space is said to be complete.

A metric space is complete if every Cauchy sequence converges.
In a complete metric space, certain mappings called contraction mappings have properties that make them suitable for modeling points in $\mathbb{R}^n$.

A mapping $f : X \to X$ on a metric space $(X, d)$ is called a \textbf{contraction mapping} if there is a constant $0 \leq s < 1$, such that for all $x, y \in X$, $d(f(x), f(y)) \leq s \cdot d(x, y)$.

Any such number $s$ is called a \textbf{contractivity factor} for $f$.

\begin{center}
\begin{tabular}{l}
\textbf{Contraction mappings are continuous:} \\
Proof: Let $f$ be a contraction mapping on metric space $(X, d)$ with contractivity factor $s$. Let $\varepsilon > 0$ be given and let $\delta = \frac{\varepsilon}{s}$. If $d(x, a) < \delta = \frac{\varepsilon}{s}$ then $d(f(x), f(a)) \leq s \cdot d(x, a) < s \cdot \frac{\varepsilon}{s} = \varepsilon$. So $f$ is continuous.
\end{tabular}
\end{center}

Contraction mappings will be used to model points in $\mathbb{R}^n$ that represent finite real-valued sequences (time-series data). If such mappings can be described using less parameters than the number of data items in the dataset modeled, then compression is achieved. Certainly such mappings should be chosen. But the most important measure of effectiveness of these mappings as models is the error between the model and the actual data. The following theorem indicates how these mappings can model data and also provides an estimate of the error.
In a complete metric space, certain mappings called contraction mappings have properties that make them suitable for modeling points in $\mathbb{R}^n$.

A mapping $f: X \to X$ on a metric space $(X, d)$ is called a **contraction mapping** if there is a constant $0 < s < 1$, such that for all $x, y \in X, d(f(x), f(y)) \leq s \cdot d(x, y)$.

Any such number $s$ is called a **contractivity factor** for $f$.

**Contraction mappings are continuous:**

Proof: Let $f$ be a contraction mapping on metric space $(X, d)$ with contractivity factor $s$.

Let $\varepsilon > 0$ be given and let $\delta = \frac{\varepsilon}{s}$. If $d(x, a) < \delta = \frac{\varepsilon}{s}$ then $d(f(x), f(a)) \leq s \cdot d(x, a) < s \cdot \frac{\varepsilon}{s} = \varepsilon$. So $f$ is continuous.

Contraction mappings will be used to model points in $\mathbb{R}^n$ that represent finite real-valued sequences (time-series data). If such mappings can be described using less parameters than the number of data items in the dataset modeled then compression is achieved. Certainly such mappings should be chosen. But the most important measure of effectiveness of these mappings as models is the error between the model and the actual data. The following theorem indicates how these mappings can model data and also provides an estimate of the error.
The Contraction Mapping Theorem with Error Estimate:
Let \( f : X \to X \) be a contraction mapping with contractivity factor \( s \) on a complete metric space \((X, d)\). Then \( f \) possesses exactly one fixed point \( x_f \in X \) and, for any point \( x \in X \), the sequence \( \{ f^n(x) \mid n = 0, 1, 2, \ldots \} \) converges to \( x_f \). Furthermore, for all \( x \in X \),
\[
d(x, x_f) \leq \frac{d(x, f(x))}{1 - s}.
\]

Proof: First it will be shown that for all \( x \in X \) and whole numbers \( k \),
\[
d(x, f^k(x)) \leq \left( 1 + s + \ldots + s^{k-1} \right) \cdot d(x, f(x)) \leq \frac{d(x, f(x))}{1 - s}.
\]
Let \( x \in X \) and prove by induction on \( k \).
If \( k = 0 \),
\[
d(x, f^0(x)) = d(x, x) = 0 \leq 1 \cdot d(x, f(x)) \leq \frac{d(x, f(x))}{1 - s}.
\]
Now suppose
\[
d(x, f^k(x)) \leq \left( 1 + s + \ldots + s^{k-1} \right) \cdot d(x, f(x)) \leq \frac{d(x, f(x))}{1 - s}.
\]
Then
\[
d(x, f^{k+1}(x)) \leq \left( 1 + s + \ldots + s^{k-1} \right) \cdot d(x, f(x)) + s \cdot d(x, f(x)) = \left( 1 + s + \ldots + s^{k-1} + s^k \right) \cdot d(x, f(x)) \leq \frac{d(x, f(x))}{1 - s}.
\]
Then, by the axiom of induction, \( d(x, f^k(x)) \leq \frac{d(x, f(x))}{1 - s} \) for all whole numbers \( k \). Let \( x \in X \).

Then \( d(f^i(x), f^j(x)) \leq s^{\min(i,j)} \cdot d(x, f^{i-j}(x)) \) for all \( i, j = 0, 1, 2, \ldots \). Also, \( d(x, f^k(x)) \leq \frac{d(x, f(x))}{1 - s} \). Substituting gives \( d(f^i(x), f^j(x)) \leq s^{\min(i,j)} \cdot \frac{d(x, f(x))}{1 - s} \). The right hand side can be made arbitrarily small by choosing sufficiently large \( i \) and \( j \). Thus,
\[
\{ f^k(x) \}_{k=0}^{\infty}
\]
is a Cauchy sequence. Since \( X \) is complete, the Cauchy sequence converges to a limit \( x_f \in X \) where
\[
\lim_{k \to \infty} f^k(x) = x_f.
\]
Since \( f \) is contractive, it is continuous. Thus,
\[
f \left( \lim_{k \to \infty} f^k(x) \right) = \lim_{k \to \infty} f^{k+1}(x) = x_f,
\]
and \( x_f \) is a fixed point. Now suppose \( y_f \in X \) is a fixed point. Then \( d(x_f, y_f) = d(f(x_f), f(y_f)) \leq s \cdot d(x_f, y_f) \), so \( (1 - s) \cdot d(x_f, y_f) \leq 0 \) which implies that \( d(x_f, y_f) = 0 \) and \( x_f = y_f \). So \( x_f \) is a unique fixed point.

Then for all \( x \in X \),
\[
d(x, x_f) = d \left( x, \lim_{k \to \infty} f^k(x) \right) = \lim_{k \to \infty} d(x, f^k(x)) \leq \frac{d(x, f(x))}{1 - s}.
\]

The Contraction Mapping Theorem indicates how a contraction mapping can be chosen to model a point in a complete metric space. Suppose a point \( x \in X \) is to be modeled by a contraction mapping \( f \) with contractivity factor \( s \) and unique fixed point \( x_f \). The measure of how well \( f \) models \( x \) is \( d(x, x_f) \) which is...
called the error. Since $d(x, x_f) \leq \frac{d(x, f(x))}{1 - s}$, then the best choices for $f$ are the contraction mappings with small contractivity factors and/or those for which $d(x, f(x))$ is small.

Further criteria for choosing a contraction mapping to model a point may include minimizing the complexity of the mapping's specification. This may conflict with minimizing error. Data compression can be accomplished if the mapping requires less data for its specification than the modeled point. But nothing guarantees that such mappings have fixed points satisfactorily close to the modeled point. More complicated mappings may be necessary for a better model at the cost increased storage for mapping specification.

**The Domain - Range Method for Finding Contraction Mappings (Local Iterated Function Systems)**

The problem is to find a contraction mapping on $\mathbb{R}^n$ to model a point $s = (s[1], \ldots, s[n])$ in $\mathbb{R}^n$ that can be specified efficiently while modeling the point within an acceptable error. There are many ways such a mapping can be chosen. The mappings considered will be restricted to mappings $M$ that fit the following criteria.

A mapping $M$ on $\mathbb{R}^n$ will consist of a set of submappings $W = \{w_1, \ldots, w_m\}$ with associated "domains" $D = \{D_1, \ldots, D_m\}$ and "ranges" $R = \{R_1, \ldots, R_m\}$. The domains and ranges are ordered sets of indices chosen from $\{1, \ldots, n\}$ such that $\bigcup_{j=1}^{m} R_j = \{1, \ldots, n\}$ and for all $i, j$ where $1 \leq i, j \leq n$, if $i \neq j$ then $R_i \cap R_j = \emptyset$. In other words, the ranges are non-overlapping and cover $\{1, \ldots, n\}$. Furthermore, $M[D_j] = w_j$.

The domains and ranges will be chosen in the following way. The ranges will consist of consecutive indices. For example, $R_i = (R_i[1], R_i[2], \ldots, R_i[|R_i|])$ where $R_i[n + 1] = R_i[n] + 1$. The ranges need not all be
the same size. The domains will consist of indices that form a subsequence of an arithmetic sequence.

For example, \( D_i = \{ D_i[1], D_i[2], \ldots, D_i[|D_i|] \} \) where \( D_i[n+1] = D_i[n] + d \) and \( d \) is a natural number called the lag. Furthermore, for all \( i \), \( |D_i| = |R_i| \).

The submappings will be of the form \( w_j : R^{|R_j|} \to R^{|R_j|} \) where \( w_j = w_j \left( \left( x[1], \ldots, x[|R_j|] \right) \right) = \left( a_j \cdot x[1]+b_j, \ldots, a_j \cdot x[|R_j|]+b_j \right) \), \( a_j \) and \( b_j \) are constants.

Then \( M : R^n \to R^n \) and \( M(x) = y = (y[1], y[2], \ldots, y[n]) \) where \( y[i] = y_j[k] = w_j[x_j[k]] = a_j \cdot x_j[k]+b_j \) such that \( R_j \) is the unique range where \( i \in R_j \) and \( k \) is the unique index such that \( i = R_j[k] \). The uniqueness of \( j \) and \( k \) for a given \( i \in \{1, \ldots, n\} \) guarantees that \( M \) is single-valued and everywhere defined. \( M \) must be a contraction mapping with a fixed point \( M^\ast \) where the error, \( d(M^\ast, s) \), is within a chosen bound for a metric \( d \). Furthermore, the mapping may be chosen to minimize \( |W| \), the number of submappings that make up \( M \).

In other words, the submappings will be affine mappings on the vector of values corresponding to the indices in the domains. A method for choosing the parameters \( a_j \) and \( b_j \) of the submappings follows. Conditions guaranteeing that the overall mapping \( M \) will be a contraction and that the error is minimized to a certain degree will be primary considerations. Examples of the implementation of the method will be given later to help clarify it.
Conditions that make $M$ a contraction can be determined. Consider the metric space $(\mathbb{R}^n, d_{\infty})$ where $d_{\infty}(u, v) = \max_{i=1,m}|u[i] - v[i]|$ for all $u, v \in \mathbb{R}^n$. Let $i \in \{1, \ldots, n\}$. Then, there exists a unique $j \in \{1, \ldots, m\}$ such that $i \in R_j$.

Let $u, v \in \mathbb{R}^n$. Then, $(M(u))[i] = a_j \cdot u_j[k] + b_j$ and $(M(v))[i] = a_j \cdot v_j[k] + b_j$, where $k$ is the unique index such that $i = R_j[k]$ and $u_j, v_j$ are found for $u$ and $v$ as $x_j$ was for $s$.

So, $|(M(u))[i] - (M(v))[i]| = |(a_j \cdot u_j[k] + b_j) - (a_j \cdot v_j[k] + b_j)| = |a_j| |u_j[k] - v_j[k]|$

$= |a_j| |R_j[k] - R_j[k]| \leq |a_j| d_{\infty}(u, v)$

So, $|(M(u))[i] - (M(v))[i]| \leq |a_j| d_{\infty}(u, v)$.

Since this is true for any $i \in \{1, \ldots, n\}$ and the corresponding unique $j \in \{1, \ldots, m\}$ where $i \in R_j$, then $M$ is a contraction if for all $j \in \{1, \ldots, m\}$, $|a_j| < 1$ and the contractivity factor for $M$ will be $\max_{j=1,m} |a_j|$. Again note that this is for the $d_{\infty}$ metric. With the Euclidean or Manhattan metrics it is more difficult to determine conditions guaranteeing a contraction mapping. Such a mapping will be called a Local Iterated Function System since the submappings are not mappings on the whole space $\mathbb{R}^n$ but on subspaces of $\mathbb{R}^n$. In an IFS the submappings map on the entire space.

Now assume that $D_j$ and $R_j$ are given and the point to be modeled is $s = (s[1], \ldots, s[n])$. Then let $x_j = s[D_j]$ be the vector of data values corresponding to the indices in $D_j$, and $y_j = s[R_j]$ be the vector of data values corresponding to the indices in $R_j$. The values for $a_j$ and $b_j$ can be determined to give the least-squares-error when mapping $x_j$ to $y_j$. Only values such that $|a_j| < 1$ will be considered so the result will be a contraction mapping. Other mappings than the least-squares-error mapping could be considered, but it is a natural first choice since the necessary computations can be easily derived.
To simplify notation, let \( n = |R_j| \) and \( x = \{x_1, \ldots, x_n\} = x_j \), \( y = \{y_1, \ldots, y_n\} = y_j \) and determine \( a \) and \( b \) to give the least-squares mapping from \( x \) to \( y \). Then \( a_j = a \) and \( b_j = b \).

Let \( \text{Error} = \sum_{i=1}^{n} (ax_i + b - y_i)^2 \)

Suppose \( x \) is constant. Then \( ax \) will be constant for any \( a \), and \( b \) can be chosen to minimize

\[
\text{Error} = \sum_{i=1}^{n} (b - y_i)^2 = \sum_{i=1}^{n} (b^2 - 2by_i + y_i^2) = nb^2 - 2b \sum_{i=1}^{n} y_i + \sum_{i=1}^{n} y_i^2.
\]

Then \( \frac{\partial \text{Error}}{\partial b} = 2nb - 2 \sum_{i=1}^{n} y_i \) and setting \( \frac{\partial \text{Error}}{\partial b} = 0 \) gives \( b = \frac{\sum_{i=1}^{n} y_i}{n} \).

So, \( a = 0 \) and \( b = \frac{\sum_{i=1}^{n} y_i}{n} \).

Suppose \( y \) is constant. Let \( a = 0 \) and \( b = \frac{\sum_{i=1}^{n} x_i}{n} \). Then \( \text{Error} = 0 \).

Suppose neither \( x \) nor \( y \) is constant.

\[
\text{Error} = \sum_{i=1}^{n} (ax_i + b - y_i)^2 = a^2 \sum_{i=1}^{n} x_i^2 + 2ab \sum_{i=1}^{n} x_i - 2a \sum_{i=1}^{n} x_i y_i + nb^2 - 2b \sum_{i=1}^{n} y_i + \sum_{i=1}^{n} y_i^2.
\]

Then \( \frac{\partial \text{Error}}{\partial b} = 2a \sum_{i=1}^{n} x_i + nb - 2 \sum_{i=1}^{n} y_i \). Setting \( \frac{\partial \text{Error}}{\partial b} = 0 \) gives \( b = \frac{\sum_{i=1}^{n} y_i}{n} - a \frac{\sum_{i=1}^{n} x_i}{n} \).

Also, \( \frac{\partial \text{Error}}{\partial a} = 2a \sum_{i=1}^{n} x_i^2 + 2b \sum_{i=1}^{n} x_i - 2 \sum_{i=1}^{n} x_i y_i \) and setting \( \frac{\partial \text{Error}}{\partial a} = 0 \) gives \( a = \frac{\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i^2 - \left( \sum_{i=1}^{n} x_i \right)^2} \).

So, \( a = \frac{\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n \sum_{i=1}^{n} x_i^2 - \left( \sum_{i=1}^{n} x_i \right)^2} \) and \( b = \frac{\sum_{i=1}^{n} y_i}{n} - a \frac{\sum_{i=1}^{n} x_i}{n} \).
In summary, when matching a domain $D_j$ and range $R_j$, $x_j$ is the vector of values corresponding to the indices in the domain, and $y_j = s[R_j]$ is the vector of values at the range indices. $a_j$ and $b_j$ are constants chosen to minimize $Error_j$, the sum-of-squares error when mapping $x_j$ to $y_j$ by $a_j \cdot x_j + b_j$. The optimal values are given below. Since $|a_j|$ is the contraction factor for the submapping, $|a_j|$ should be less than 1. The goal is to choose a collection of domains and ranges so the submappings are contraction mappings, the overall error is within a given bound, and the number of submappings is minimized.

$$a_j = \frac{\sum_{i=1}^{|R_j|} y_j[i]}{|R_j|},$$

If $x_j$ is constant or $y_j$ is constant then $a_j = 0$, $b_j = \frac{\sum_{i=1}^{|R_j|} x_j[i]}{|R_j|}$. If neither $x_j$ is constant nor $y_j$ is constant then $a_j = \frac{\left| R_j \right| \sum_{i=1}^{|R_j|} x_j[i] y_j[i] - \left( \sum_{i=1}^{|R_j|} x_j[i] \right) \left( \sum_{i=1}^{|R_j|} y_j[i] \right)}{\left( \sum_{i=1}^{|R_j|} x_j[i] \right)^2 - \left( \sum_{i=1}^{|R_j|} x_j[i] \right)^2}$,

$$b_j = \frac{1}{|R_j|} \left( \sum_{i=1}^{|R_j|} y_j[i] - a_j \sum_{i=1}^{|R_j|} x_j[i] \right).$$
The Method at Work

Some Simple Examples

Consider what should be an easy sequence to model, an arithmetic (linear) sequence with eight values, \( 5, 8, 11, 14, 17, 20, 23, 26 \), so \( s = (5, 8, 11, 14, 17, 20, 23, 26) \) and \( n = 8 \).

Suppose \( m = 2 \) with \( D_1 = (1, 3, 5, 7) \), \( R_1 = (5, 6, 7, 8) \), and \( D_2 = (2, 4, 6, 8) \), \( R_2 = (1, 2, 3, 4) \). This choice of domains and ranges is consistent with the described method since \( |D_1| = |R_1| = 4 \), \( |D_2| = |R_2| = 4 \), \( \bigcup_{i=1}^{2} R_i = \{1, \ldots, n\} \), \( R_1 \cap R_2 = \emptyset \), the indices in \( R_i \) and \( R_i \) are sequential, and the indices in \( D_1 \) and \( D_2 \) are arithmetic sequences with lag=2. The parameters for the least squares submappings are \( a_1 = 0.5 \), \( b_1 = 14.5 \), \( a_2 = 0.5 \), and \( b_2 = 1 \). So, \( w_1 : \mathbb{R}^4 \rightarrow \mathbb{R}^4 \) where \( w_1(x) = 0.5x + 14.5 \) and \( w_2(x) = 0.5x + 1 \).

Since \( w_1(s(D_1)) = w_1((5,11,17,23)) = (17,20,23,26) = s(R_1) \) and \( w_2(s(D_2)) = w_2((8,14,20,26)) = (5,8,11,14) = s(R_2) \) then the error for each mapping is 0. Furthermore, each submapping has a contraction factor of 0.5.

Then \( M : \mathbb{R}^8 \rightarrow \mathbb{R}^8 \) and \( M(x) = M((x[1], \ldots, x[8])) = (w_1(x[2]), w_2(x[4]), w_3(x[6]), w_4(x[8]), w_5(x[1]), w_6(x[3]), w_7(x[5]), w_8(x[7])) \) so \( M(s) = M((5,8,11,14,17,20,23,26)) = (5,8,11,14,17,20,23,26) \) and there is no error. The contraction factor is \( \min(0.5, 0.5) = 0.5 \) so the mapping is a contraction.

The algorithm proceeds as follows:

- Start with an arbitrary sequence \( x \).
- Iterate \( M \) starting with the input \( x \) until the iterations converge. Since \( a_1 = 0.5 \) and \( a_2 = 0.5 \) then \( |a_1| < 1 \) and \( |a_2| < 1 \) so \( M \) is a contraction and will converge to a fixed point (sequence).
Start with \( x = <0,0,0,0,0,0,0,0> \). Successive iterations of \( M \) on \( x \) are shown in the following table and graph. Notice how the values converge to values of the sequence being modeled. After 16 iterations there is convergence to the nearest thousandths place.

**Iteration Values for a Local IFS Model of \(<5,8,11,14,17,20,23,26>\)**

<table>
<thead>
<tr>
<th>index</th>
<th>( x )</th>
<th>( M(x) )</th>
<th>( M^2(x) )</th>
<th>( M^3(x) )</th>
<th>( M^4(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1.5</td>
<td>1.75</td>
<td>3.563</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1.5</td>
<td>1.75</td>
<td>3.563</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1.5</td>
<td>1.75</td>
<td>3.563</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1.5</td>
<td>1.75</td>
<td>3.563</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1.5</td>
<td>1.75</td>
<td>3.563</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Iteration Values Converging to the Sequence \(<1,2,3,4,5,6,7,8>\)**

![Graph showing convergence](image)
Iteration Values, Shown in Separate Graphs, Converging to the Sequence
<5, 8, 11, 14, 17, 20, 23, 26>
This linear sequence is modeled well by a Local Iterated Function System which should be expected since it comes from a self-similar object. Recall that a line is self-similar.

Not all sequences are modeled well by Local Iterated Function Systems, for example, a sequence where the values are the squares of the corresponding indices. Intuitively, such a sequence should not be modeled well since it is not self-similar under affine mappings.

Using the sequence <1, 4, 9, 16, 25, 36, 49, 64> and choosing the domains and ranges as before, \( D_1 = <1, 3, 5, 7> \), \( R_1 = <5, 6, 7, 8> \), and, \( D_2 = <2, 4, 6, 8> \), \( R_2 = <1, 2, 3, 4> \), then the least-squares mappings are \( w_1(x) = 0.786x + 27 \) and \( w_2(x) = 0.25x \). Again, let \( M = \{ w_1, w_2 \} \) which is a contraction mapping.

The model of the data, the fixed point of the mapping, is <1.022, 4.086, 8.4, 16.345, 27.803, 33.6, 48.845, 65.378> with values accurate to the nearest thousandth after 18 iterations. Notice the error in the values when compared to the original sequence. The greatest pointwise error, 2.4, occurs at the eighth entry. A different choice of domain-range pairs might result in a better fit, but this dataset was chosen only for illustrative purposes. Even though there is an error, this model is somewhat accurate to some degree and can be used to illustrate methods for interpolating or extrapolating missing values.

**Interpolating and Extrapolating Values**

Suppose a sequence of samples is collected at uniform time intervals from a data source. Then it may be desirable to interpolate unsampled values directly or through use of the model. The intent of modeling in this paper is prediction and not compression, so it is desirable for the model to assist in this process. There are several ways such predictions can be made and several types of predictions that can be made.

First, the easiest method for interpolating intermediate values does not require the model but just the original data. Simple linear interpolation can be used and will obviously result in errors for non-linear data. For the quadratic example, the model differs from the original data so it is expected that greater errors would occur if the model data was used instead of the original data so the model would be of little use.
A second method for interpolating intermediate values directly uses the mappings and either the original values (better) or the model values (probably with greater error.) Consider the original data as samples from times in the intervals shown in the diagrams below, where domain-range interval pairings are also indicated.

By extending the mappings from points to intervals, \( w_1 \) not only maps the value corresponding to index 1 to a value at index 5, but it maps values corresponding to the interval \([1,3)\) to values corresponding the interval \([5,6)\) and so on for other pairs of intervals indicated as well as for the interval pairs shown for \( w_2 \). Then, the value of the function at 5.5 can be interpolated as \( w_1(2) = 0.786(2) + 27 = 28.572 \). The actual value should be \((5.5)^2 = 30.25\) so there is some error.

Values can be interpolated similarly at 6.5, 7.5, and 8.5 using \( w_1 \) and at 1.5, 2.5, and 3.5 using \( w_2 \). Notice that a value cannot be interpolated at 4.5 since the value at 9 is not known. This leaves a gap in the values that can be found and is the consequence of selecting a second domain that includes indices at the end of the original sequence. Choosing domains that don’t have indices at the end of the sequence would eliminate such gaps.

Notice that the values beyond the end of the original sequence can also be extrapolated on the interval \((8,9)\). This could be a way of predicting future values if the data is time-series data.
A third method, similar to the second, also uses mappings over intervals. In this method entries are included in the iterated sequences corresponding to intermediate values. The same mappings are used. Recall that the indices for the original mapping(s) were <1, 2, 3, 4, 5, 6, 7, 8> and the domains and ranges were: \( D_1 = <1, 3, 5, 7>, R_1 = <5, 6, 7, 8> \), and, \( D_2 = <2, 4, 6, 8>, R_2 = <1, 2, 3, 4> \). Consider the following "indices" which are the originals and those following in the middle of each interval as well as 9: <1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9>. Then the domains and ranges corresponding to the originals are: \( D_1 = <1, 2, 3, 4, 5, 6, 7, 8, 9>, R_1 = <5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9> \), and, \( D_2 = <2, 3, 4, 5, 6, 7, 8, 9>, R_2 = <1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5> \).

After 48 iteration the mapping converges to the following values, accurate to the nearest thousandth place:

(1, 1.022), (1.5, 2.1), (2, 4.087), (2.5, 6.951), (3, 8.401), (3.5, 12.213), (4, 16.35), (4.5, 31.542), (5, 27.803), (5.5, 30.213), (6, 33.603), (6.5, 39.851), (7, 48.853), (7.5, 53.412), (8, 65.399), (8.5, 78.403), (9, 126.167). Notice several things about the model data. First, the corresponding values fall close to the original fixed point which was <1.022, 4.086, 8.4, 16.345, 27.803, 33.6, 48.845, 65.378>. The value interpolated for 5.5 is 30.213 which is much closer to the actual value (30.25) than the interpolated value 28.572 from method two. But, the values for 4.5 and 9 are not close to what they should be. This might be expected since the data is non-linear and does not have the type of self-similarity best modeled by this method. When the same method is applied to the linear data set <1, 2, 3, 4, 5, 6, 7, 8>, the resulting fixed point <1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9> occurs after 15 iterations and models the original data correctly as well as the unknown data.

Another more general way for finding intermediate values and values beyond those in the original data is the escape time algorithm. This is a general technique used to generate fractal images [Barnsley, 1988]. Its advantage is that it can be used to find single values without iterating entire sequences. Its disadvantage is that it can be computationally intensive. This method can be applied when none of the domains are "too close" to the end intervals as discussed earlier.
Consider the sequence \(<0, 1, 2, 3, 4, 5>\) with domains \(D_1 = <0, 2>, D_2 = <1, 3>, D_3 = <2, 5>\) and ranges \(R_1 = <0, 1>, R_2 = <2, 3>, R_3 = <4, 5>\). For simplicity sake, the data is chosen identical to the indices, this time starting with index 0. Then \(w_1(x) = 0.5x, w_2(x) = 0.5x + 1.5, w_3(x) = 0.5x + 3\) and these mappings can be extended to map on intervals rather than indices. Then, \(w_1\) would map values corresponding to the interval \([0,4)\) to values corresponding to the interval \([0,2)\). Likewise, \(w_2\) would map values corresponding to the interval \([1,5)\) to values corresponding to the interval \([2,4)\) and \(w_3\) would map values corresponding to the interval \([2,6)\) to values corresponding to the interval \([4,6)\).

Let \(M = \{ w_1, w_2, w_3 \} \) which is a contraction. Since \(M\) is a contraction it will map, through iteration, starting values (sequences or functions on intervals) to a fixed point, which is \(f(x) = x\) for \(0 < x < 6\) or some sequence of sample indices. Furthermore, the fixed point of \(M\) will map to itself by definition. The inverse of \(M\), \(M^1\), is not a contraction mapping but has an unstable fixed point identical to the fixed point of \(M\). All other points will be "repelled" or moved away from that fixed point by \(M^1\). The rate at which points are repelled depends on how close they are to the fixed point and the fixed point can be determined by the rate at which points are repelled and a search algorithm. For the example above,

\[
M^{-1}(\begin{pmatrix} x \\ y \end{pmatrix}) = \begin{cases} 
\begin{pmatrix} 2x \\ 2y \end{pmatrix} & \text{if } x \in [0,2), \\
\begin{pmatrix} 2x - 3 \\ 2y - 3 \end{pmatrix} & \text{if } x \in [2,4), \\
\begin{pmatrix} 2x - 6 \\ 2y - 6 \end{pmatrix} & \text{if } x \in [4,6).
\end{cases}
\]

To determine if a point is on the fixed point, \(M^1\) is iterated on that point and the \(y\) value is examined. If the \(y\) value stabilizes, the point is on the fixed point. Otherwise, it is not on the fixed point and will be "repelled" at a rate depending on its original distance from the fixed point.
\begin{align*}
\begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{ is on the fixed point since } f(0) = 0. \quad (M^{-1})^{10} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.
\end{align*}

\begin{align*}
\begin{bmatrix} 0 \\ 0.01 \end{bmatrix} & \text{ is not on the fixed point since but close. } \quad (M^{-1})^{10} \begin{bmatrix} 0 \\ 0.01 \end{bmatrix} = \begin{bmatrix} 0 \\ 10.24 \end{bmatrix}.
\end{align*}

Also, \((M^{-1})^{10} \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0 \\ 102.4 \end{bmatrix}.

\begin{align*}
\text{Notice that } \begin{bmatrix} 0 \\ 0.01 \end{bmatrix} & \text{ is repelled, though less than } \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} \text{ since it is closer to the value on the fixed point. Also, } \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ is fixed. A search algorithm can be devised to determine } y \text{ values on the fixed point for a given } x \text{ by examining the "escape" rates. Other interesting results for values iterated with } M^{-1} \text{ are shown below:}
\end{align*}

\begin{align*}
(M^{-1})^{10} \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, & \quad (M^{-1})^{10} \begin{bmatrix} 5 \\ 5 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \quad (M^{-1})^{10} \begin{bmatrix} 5.5 \\ 5.5 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \\
(M^{-1})^{10} \begin{bmatrix} 5.999 \\ 5.999 \end{bmatrix} = \begin{bmatrix} 4.976 \\ 4.976 \end{bmatrix}, & \quad (M^{-1})^{10} \begin{bmatrix} 3.14 \\ 3.14 \end{bmatrix} = \begin{bmatrix} 2.36 \\ 2.36 \end{bmatrix}, \\
(M^{-1})^{100} \begin{bmatrix} 5.999 \\ 5.999 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, & \quad (M^{-1})^{100} \begin{bmatrix} 3.14 \\ 3.14 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \text{ and } (M^{-1})^{10} \begin{bmatrix} 10 \\ 10 \end{bmatrix} = \begin{bmatrix} 4102 \\ 4102 \end{bmatrix}.
\end{align*}

Notice some values on the fixed point are attracted to other fixed point values. \(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \text{ and } \begin{bmatrix} 2 \\ 2 \end{bmatrix}\) appear to be attractors on the fixed point. Also notice that intermediate values like \(\begin{bmatrix} 5 \\ 5 \end{bmatrix}\) and \(\begin{bmatrix} 5.5 \\ 5.5 \end{bmatrix}\) are shown to be on the fixed point. Some values like \(\begin{bmatrix} 5.999 \\ 5.999 \end{bmatrix}\) and \(\begin{bmatrix} 3.14 \\ 3.14 \end{bmatrix}\) take longer to converge though they are never repelled by the fixed point. \(\begin{bmatrix} 10 \\ 10 \end{bmatrix}\) is repelled since 10 goes beyond the intervals in the mappings. The escape time algorithm may be helpful in using Local Iterated Function System models to determine intermediate values or values beyond the given sequence (time series), given appropriately chosen ranges.

These data sets were used for illustrative purposes only. Beyond showing how this sort of modeling works and how interpolation and extrapolation can be accomplished an important observation has been made. Not
all data is suitable for this type of modeling. It will later be necessary to show that the method is suitable for time-series stock data. Another method for extrapolating data will be given then.
A More Complicated Example

The method can now be applied to a larger and more complex dataset. No predictions will be made from the data at this time. The objective will be to simply fit the model to the data, compare the model to the original data, and determine the error. The data used is 720 consecutive daily closing values for IBM stock from March 4th, 1991 to January 4th, 1994. 720 values were chosen since 720 divides evenly in many ways which is useful when choosing ranges. The modeling method has only been described and demonstrated for several small datasets and there can be many choices involved in its implementation for larger datasets. The data values used are shown below.

IBM Closing Values - March 4, 1991 through January 4, 1994

<table>
<thead>
<tr>
<th>Index</th>
<th>+0</th>
<th>+1</th>
<th>+2</th>
<th>+3</th>
<th>+4</th>
<th>+5</th>
<th>+6</th>
<th>+7</th>
<th>+8</th>
<th>+9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>130</td>
<td>133</td>
<td>132</td>
<td>132</td>
<td>131</td>
<td>129</td>
<td>127</td>
<td>129</td>
<td>128</td>
<td>129</td>
</tr>
<tr>
<td>1</td>
<td>277</td>
<td>277</td>
<td>277</td>
<td>277</td>
<td>277</td>
<td>277</td>
<td>277</td>
<td>277</td>
<td>277</td>
<td>277</td>
</tr>
<tr>
<td>2</td>
<td>113</td>
<td>113</td>
<td>113</td>
<td>112</td>
<td>112</td>
<td>113</td>
<td>112</td>
<td>112</td>
<td>112</td>
<td>112</td>
</tr>
<tr>
<td>4</td>
<td>103</td>
<td>104</td>
<td>105</td>
<td>103</td>
<td>103</td>
<td>103</td>
<td>103</td>
<td>103</td>
<td>103</td>
<td>103</td>
</tr>
<tr>
<td>5</td>
<td>106</td>
<td>107</td>
<td>106</td>
<td>106</td>
<td>106</td>
<td>106</td>
<td>106</td>
<td>106</td>
<td>106</td>
<td>106</td>
</tr>
<tr>
<td>6</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>104</td>
</tr>
<tr>
<td>7</td>
<td>101</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>8</td>
<td>98</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>11</td>
<td>99</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
</tr>
<tr>
<td>12</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>94</td>
</tr>
<tr>
<td>13</td>
<td>99</td>
<td>101</td>
<td>101</td>
<td>101</td>
<td>101</td>
<td>101</td>
<td>101</td>
<td>101</td>
<td>101</td>
<td>101</td>
</tr>
<tr>
<td>14</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>104</td>
</tr>
<tr>
<td>15</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
</tr>
<tr>
<td>16</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>17</td>
<td>98</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
</tr>
<tr>
<td>18</td>
<td>96</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
</tr>
<tr>
<td>19</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
</tr>
<tr>
<td>20</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
</tr>
<tr>
<td>21</td>
<td>89</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>22</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
</tr>
<tr>
<td>23</td>
<td>91</td>
<td>91</td>
<td>91</td>
<td>91</td>
<td>91</td>
<td>91</td>
<td>91</td>
<td>91</td>
<td>91</td>
<td>91</td>
</tr>
<tr>
<td>24</td>
<td>92</td>
<td>90</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>25</td>
<td>87</td>
<td>87</td>
<td>87</td>
<td>87</td>
<td>87</td>
<td>87</td>
<td>87</td>
<td>87</td>
<td>87</td>
<td>87</td>
</tr>
<tr>
<td>26</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>27</td>
<td>84</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
</tr>
<tr>
<td>28</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>88</td>
<td>88</td>
<td>88</td>
<td>88</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td>29</td>
<td>88</td>
<td>88</td>
<td>88</td>
<td>88</td>
<td>88</td>
<td>88</td>
<td>88</td>
<td>88</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td>30</td>
<td>93</td>
<td>93</td>
<td>93</td>
<td>93</td>
<td>93</td>
<td>93</td>
<td>93</td>
<td>93</td>
<td>93</td>
<td>93</td>
</tr>
<tr>
<td>31</td>
<td>91</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>32</td>
<td>91</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>33</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
</tr>
<tr>
<td>34</td>
<td>96</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
<td>97</td>
</tr>
<tr>
<td>35</td>
<td>93</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
</tr>
<tr>
<td>Index</td>
<td>Closing Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>---------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>150</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>140</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>130</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>120</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>110</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>80</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**IBM Closing Values: 3/4/91-1/4/94**

- **Index:** The index runs from 0 to 800.
- **Closing Value:** The closing value is shown in dollars.

The graph illustrates the trend of IBM's closing values from March 4, 1991, to January 4, 1994.
Given such a dataset, the following choices must be made to determine the Local Iterated Function System that will model the data.

- Choose a contraction factor so the mapping will be a contraction mapping, guaranteeing a fixed-point. If $|a_j| < 1$ for all $w_j$, the mapping will be a contraction mapping. (Note that it is possible to have some $a_j$ values where $|a_j| \geq 1$ and the mapping is still a contraction due to "mixing" that occurs through iteration of the submappings. This creates a mapping that is called "eventually contractive." [Fisher, 1995] Such mappings will not be dealt with here and an arbitrary bound $|a_j| < 0.75$ will be used.)

- A pool of ranges and potential domains must be chosen. The ranges must cover the dataset but may not overlap. They need not be the same size and neither do the domains (though matching domains and ranges must be the same size.) To simplify computations by reducing the number of possible comparisons, ranges will be chosen as sets of 20 consecutive indices (corresponding to 20 consecutive days or roughly 4 weeks / one month of trading.) The pool of domains will be restricted to the set including every fifth trading day, roughly the start of each trading week. Overlapping domains are considered and each domain consists of 20 consecutive "weekly starting values" as described.

- There will be 36 ranges and 36 corresponding mappings. There were 144 "weekly" data values giving 125 potential domains. For each range, a corresponding mapping will be found by determining the domain and mapping giving the least sum-of-squares error. A graph of the pool of domain values is shown below. Naturally it looks like a smoothed trace of the daily data.

```
Domain Values (Every 5th Closing Value)
```

```

closing
value

0  10  20  30  40  50  60  70  80  90  100  110  120  130  140
index
```

43
The resulting mapping is defined by the following parameters.

<table>
<thead>
<tr>
<th>range</th>
<th>domain</th>
<th>a</th>
<th>b</th>
<th>sum-sq-err</th>
<th>max error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>360</td>
<td>0.709</td>
<td>68.701</td>
<td>163.338</td>
<td>5.989</td>
</tr>
<tr>
<td>20</td>
<td>375</td>
<td>0.171</td>
<td>97.973</td>
<td>17.756</td>
<td>2.56</td>
</tr>
<tr>
<td>40</td>
<td>490</td>
<td>0.252</td>
<td>117.051</td>
<td>25.105</td>
<td>2.237</td>
</tr>
<tr>
<td>60</td>
<td>370</td>
<td>0.175</td>
<td>89.367</td>
<td>15.365</td>
<td>1.765</td>
</tr>
<tr>
<td>80</td>
<td>500</td>
<td>0.388</td>
<td>118.711</td>
<td>20.215</td>
<td>2.182</td>
</tr>
<tr>
<td>100</td>
<td>320</td>
<td>0.248</td>
<td>77.25</td>
<td>8.023</td>
<td>1.679</td>
</tr>
<tr>
<td>120</td>
<td>385</td>
<td>0.294</td>
<td>118.814</td>
<td>16.616</td>
<td>1.796</td>
</tr>
<tr>
<td>140</td>
<td>140</td>
<td>0.424</td>
<td>61.275</td>
<td>39.984</td>
<td>2.896</td>
</tr>
<tr>
<td>160</td>
<td>65</td>
<td>0.403</td>
<td>58.63</td>
<td>9.201</td>
<td>2.045</td>
</tr>
<tr>
<td>180</td>
<td>355</td>
<td>0.335</td>
<td>66.104</td>
<td>33.71</td>
<td>3.893</td>
</tr>
<tr>
<td>200</td>
<td>265</td>
<td>0.523</td>
<td>41.564</td>
<td>24.038</td>
<td>2.18</td>
</tr>
<tr>
<td>220</td>
<td>370</td>
<td>0.165</td>
<td>80.737</td>
<td>13.258</td>
<td>1.613</td>
</tr>
<tr>
<td>240</td>
<td>410</td>
<td>0.154</td>
<td>79.573</td>
<td>7.275</td>
<td>1.337</td>
</tr>
<tr>
<td>260</td>
<td>400</td>
<td>0.21</td>
<td>72.294</td>
<td>14.136</td>
<td>2.088</td>
</tr>
<tr>
<td>280</td>
<td>515</td>
<td>-0.666</td>
<td>122.545</td>
<td>19.969</td>
<td>1.769</td>
</tr>
<tr>
<td>300</td>
<td>0</td>
<td>0.105</td>
<td>80.06</td>
<td>6.292</td>
<td>1.382</td>
</tr>
<tr>
<td>320</td>
<td>395</td>
<td>-0.231</td>
<td>109.449</td>
<td>11.039</td>
<td>1.801</td>
</tr>
<tr>
<td>340</td>
<td>415</td>
<td>0.28</td>
<td>79.469</td>
<td>28.753</td>
<td>3.496</td>
</tr>
<tr>
<td>360</td>
<td>5</td>
<td>0.19</td>
<td>67.642</td>
<td>23.224</td>
<td>2.053</td>
</tr>
<tr>
<td>380</td>
<td>370</td>
<td>0.161</td>
<td>73.445</td>
<td>20.164</td>
<td>2.361</td>
</tr>
<tr>
<td>400</td>
<td>355</td>
<td>0.473</td>
<td>37.672</td>
<td>44.531</td>
<td>3.256</td>
</tr>
<tr>
<td>420</td>
<td>255</td>
<td>-0.432</td>
<td>104.699</td>
<td>11.302</td>
<td>1.965</td>
</tr>
<tr>
<td>440</td>
<td>350</td>
<td>0.612</td>
<td>12.169</td>
<td>78.999</td>
<td>4.137</td>
</tr>
<tr>
<td>460</td>
<td>5</td>
<td>0.156</td>
<td>32.23</td>
<td>15.994</td>
<td>2.245</td>
</tr>
<tr>
<td>480</td>
<td>225</td>
<td>-0.317</td>
<td>79.396</td>
<td>7.523</td>
<td>1.288</td>
</tr>
<tr>
<td>500</td>
<td>220</td>
<td>-0.301</td>
<td>81.491</td>
<td>10.852</td>
<td>1.692</td>
</tr>
<tr>
<td>520</td>
<td>530</td>
<td>0.313</td>
<td>35.758</td>
<td>13.533</td>
<td>2.191</td>
</tr>
<tr>
<td>540</td>
<td>190</td>
<td>0.122</td>
<td>37.707</td>
<td>3.954</td>
<td>0.854</td>
</tr>
<tr>
<td>560</td>
<td>425</td>
<td>-0.204</td>
<td>63.158</td>
<td>21.227</td>
<td>2.203</td>
</tr>
<tr>
<td>580</td>
<td>535</td>
<td>0.278</td>
<td>35.172</td>
<td>7.914</td>
<td>1.273</td>
</tr>
<tr>
<td>600</td>
<td>100</td>
<td>0.154</td>
<td>28.412</td>
<td>11.18</td>
<td>1.705</td>
</tr>
<tr>
<td>620</td>
<td>270</td>
<td>0.292</td>
<td>17.368</td>
<td>3.872</td>
<td>0.808</td>
</tr>
<tr>
<td>640</td>
<td>295</td>
<td>-0.215</td>
<td>62.656</td>
<td>5.259</td>
<td>0.886</td>
</tr>
<tr>
<td>660</td>
<td>335</td>
<td>-0.231</td>
<td>65.255</td>
<td>15.797</td>
<td>1.61</td>
</tr>
<tr>
<td>680</td>
<td>355</td>
<td>-0.283</td>
<td>79.633</td>
<td>9.287</td>
<td>1.489</td>
</tr>
<tr>
<td>700</td>
<td>365</td>
<td>-0.421</td>
<td>76.6</td>
<td>17.481</td>
<td>2.529</td>
</tr>
</tbody>
</table>

- **range** indicates the starting index of a range.
- **domain** indicates the starting index of a domain.
- **a** and **b** are the least squares coefficients (|a| is the contraction factor).
- **sum-sq-err** is the sum of squares error for mapping original domain values to range values and is minimal for the given bound on the contraction factor.
- **max error** is the maximum error between original mapped domain values and original range values.

The contraction factor(s) and the sum-of-squared errors can be used to compute an upper bound on the error between the model and the actual data using the Contraction Mapping Theorem. In general, such a
bound will be much higher than the actual error so it serves of little use. The computation is shown below and it will be compared to the actual error later.

$$\text{sum sq error} \leq \frac{d(x, M(x))}{1 - s} = 796.166 \div (1 - 0.709) = 2735.966.$$  

For the submapping $w_1$, the range is $<0, 1, 2, ..., 19>$ and the domain is $<360, 365, 370, ..., 455>$. The submapping parameters are $a = 0.709$ and $b = 68.701$ so the contraction factor is 0.709. $w_0$ maps the value corresponding to index 360 to a value corresponding to index 0, the value at index 365 to a value at index 1, and so on.

The "sum-of-squares error" and "max error" are greatest for this particular submapping. This may have been anticipated by observing the sharp decline between the 11th and 12th day. Such drastic declines between daily values are rare in this dataset and harder to match. Notice another sharp decline in values for the range $<440, ..., 459>$ causing a relatively large error in the mapping $w_{22}$.

A graph of the fixed-point model produced by the mapping is shown below:

Model of IBM data
Visually compare this graph with the graph of the original data. They are shown together below. Also shown is a graph of the pointwise errors.

![Data vs. Model](image)

The sum-of-squares error for the model is 865.658 and the maximum difference between a model value and its corresponding data value is 5.969 at index 10. The sum-of-squares error is considerably less than 2735.966, the bound given by the Contraction Mapping. Considering the limited amount of data, particularly in the pool of domain values, the model seems to fit the data relatively well.
A Closer Look and Some Problems with the Model

The first range (indices 0 through 19) and its corresponding values, domain values, mapping values, and model values are shown below. The $ax + b$ values are found using the original domain values. Notice that they differ from the model values, a consequence of iteration and mixing.

### The First Submapping

<table>
<thead>
<tr>
<th>index</th>
<th>range value</th>
<th>index</th>
<th>domain value</th>
<th>ax+b value</th>
<th>model value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>130.625</td>
<td>360</td>
<td>94.25</td>
<td>135.493</td>
<td>134.201</td>
</tr>
<tr>
<td>1</td>
<td>133.5</td>
<td>365</td>
<td>88</td>
<td>131.064</td>
<td>131.391</td>
</tr>
<tr>
<td>2</td>
<td>132.875</td>
<td>370</td>
<td>88</td>
<td>131.064</td>
<td>130.697</td>
</tr>
<tr>
<td>3</td>
<td>132.5</td>
<td>375</td>
<td>87.125</td>
<td>130.444</td>
<td>129.874</td>
</tr>
<tr>
<td>4</td>
<td>131.25</td>
<td>380</td>
<td>87.625</td>
<td>130.799</td>
<td>130.739</td>
</tr>
<tr>
<td>5</td>
<td>129.125</td>
<td>385</td>
<td>86.875</td>
<td>130.267</td>
<td>130.332</td>
</tr>
<tr>
<td>6</td>
<td>127</td>
<td>390</td>
<td>83.125</td>
<td>127.61</td>
<td>128.386</td>
</tr>
<tr>
<td>7</td>
<td>129.125</td>
<td>395</td>
<td>83.625</td>
<td>127.964</td>
<td>128.168</td>
</tr>
<tr>
<td>8</td>
<td>128.125</td>
<td>400</td>
<td>80.75</td>
<td>125.926</td>
<td>127.041</td>
</tr>
<tr>
<td>9</td>
<td>126.625</td>
<td>405</td>
<td>78.375</td>
<td>124.243</td>
<td>124.743</td>
</tr>
<tr>
<td>10</td>
<td>127.875</td>
<td>410</td>
<td>78</td>
<td>123.978</td>
<td>121.906</td>
</tr>
<tr>
<td>11</td>
<td>115.125</td>
<td>415</td>
<td>68.5</td>
<td>117.245</td>
<td>117.191</td>
</tr>
<tr>
<td>12</td>
<td>114.125</td>
<td>420</td>
<td>67.125</td>
<td>116.271</td>
<td>116.095</td>
</tr>
<tr>
<td>13</td>
<td>111.875</td>
<td>425</td>
<td>68.25</td>
<td>117.068</td>
<td>116.458</td>
</tr>
<tr>
<td>14</td>
<td>111.625</td>
<td>430</td>
<td>65</td>
<td>114.765</td>
<td>114.775</td>
</tr>
<tr>
<td>15</td>
<td>113.5</td>
<td>435</td>
<td>63.125</td>
<td>113.436</td>
<td>113.978</td>
</tr>
<tr>
<td>16</td>
<td>113.875</td>
<td>440</td>
<td>65.375</td>
<td>115.031</td>
<td>117.753</td>
</tr>
<tr>
<td>17</td>
<td>112.875</td>
<td>445</td>
<td>67.25</td>
<td>116.359</td>
<td>114.742</td>
</tr>
<tr>
<td>18</td>
<td>113.875</td>
<td>450</td>
<td>61.75</td>
<td>112.462</td>
<td>113.009</td>
</tr>
<tr>
<td>19</td>
<td>112.25</td>
<td>455</td>
<td>53</td>
<td>106.261</td>
<td>106.536</td>
</tr>
</tbody>
</table>
The graph below shows the values for this range and the matching domain.

First Domain-Range Pair

![Graph showing range and domain values](image)

Notice the similarities between the two datasets, in particular the decline between the 11th and 12th data items. Both show a downward trend though the domain shows a greater decline than the range. The statistical range of the domain is greater than that of the range data implying that the least squares mapping from the domain to the range should be a contraction.

Notice too that there are dissimilarities between the domain and range values. While the range values start with a moderate increase, the domain starts with a decrease nearly as abrupt as that between the 11th and 12th elements. Furthermore, there is another abrupt steady decline between the 18th and 20th elements of the domain while the corresponding range values show little change at all.

Matching this domain-range pair as similar is now questionable, even though this domain gives the least-squares mapping. This pair gives the greatest error which may be a consequence of too few domains in the domain pool. Yet, further refinements may be necessary to get a more acceptable and meaningful model.

The following graph shows the range, domain, and "ax+b" values. The "ax+b" values are the values for the least-squares mapping from the domain values to the range values. The differences noted earlier appear in this graph.
Finally, the range, "ax+b", and model values are shown together. Again, iteration and mixing cause the model values to differ from the "ax+b" values though they do not appear to differ by much.

Further complications can occur when looking for meaning in the mapping and its fixed point. Thirteen of the thirty-six mappings have negative values for the parameter $a$. These values satisfy the condition that $|a| < 0.75$ set earlier but are questionable since they map increases to decreases and decreases to increases. The $a$ parameter not only rescales the data, but it "flips it over" when negative. It is questionable whether such matchings should be made. This problem could be "fixed" by further requiring that all $a_i > 0$. 

49
If the goal of this "modeling" was compression, then this would not be a concern and would be ignored entirely for the sake of greater compression ratios. This is done in image compression. Portions of images that look similar (or are inverses of each other) can be matched with no relationship beyond similar appearance under some transformation. Since my goal is to find a meaningful modeling method, such concerns must be addressed.

The IFS description of the Sierpinski Triangle was resolution independent — detail could be determined at any resolution. This was a consequence of the Sierpinski Triangle's self-similarity being described exactly by an IFS. The Fractal Transform method used to create Local Iterated Function Systems is necessary since few images (datasets) can be described precisely by an IFS, or, even if an image can be described well by an IFS, finding that IFS is difficult if not impossible. These observations are valid for other types of data, including time-series data.

As mentioned, this method can match subsets of images (data) that are similar but have no meaningful relationship. So interpolation of greater detail or extrapolation of missing data becomes questionable.

Consider an image of a person's face. This image could be compressed, with some degree of error, with the Fractal Transform Method [Barnsley and Hurd, 1993]. Methods similar to those given for the one-dimensional case (sequences) can be used to interpolate "features" at higher resolutions. But it is doubtful that these features will be realistic. If you view a human face at greater resolution (closer and closer) you soon see pores in the skin, individual cells, and (using a microscope) eventually molecules and atoms. These features surely would not appear in the "zoomed in" decompressed image since there is no evidence of these features in the original image. Mappings that make up Local Iterated Function Systems do not appear to have great predictive potential for images.

At this point, it may seem that modeling time-series stock data may be doomed to similar failure. But, it may still be possible to modify the method, and find a meaningful and useful modeling method.
Refinements of the Method

Returns from stocks and other investments are commonly given as interest rates from one period to another. A simple example is an investment with a compounded fixed interest rate. Stock market return rates fluctuate as the value of stocks change in short time frames. Simple fixed-rate compounding interest situations will be examined first for insight into potential changes in the model. Consider an investment with a 10% yield compounded annually. The table below shows the balance at the end of each year starting with a balance of 1. Shown too are the logarithms of the balance which will also be discussed.

<table>
<thead>
<tr>
<th>year</th>
<th>balance</th>
<th>log(balance)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>1</td>
<td>1.100</td>
<td>log(1.1) = 0.041</td>
</tr>
<tr>
<td>2</td>
<td>(1.1)^2 = 1.210</td>
<td>2 log(1.1) = 0.083</td>
</tr>
<tr>
<td>3</td>
<td>(1.1)^3 = 1.331</td>
<td>3 log(1.1) = 0.124</td>
</tr>
<tr>
<td>4</td>
<td>(1.1)^4 = 1.464</td>
<td>4 log(1.1) = 0.166</td>
</tr>
<tr>
<td>5</td>
<td>(1.1)^5 = 1.611</td>
<td>5 log(1.1) = 0.207</td>
</tr>
<tr>
<td>6</td>
<td>(1.1)^6 = 1.772</td>
<td>6 log(1.1) = 0.248</td>
</tr>
<tr>
<td>7</td>
<td>(1.1)^7 = 1.949</td>
<td>7 log(1.1) = 0.290</td>
</tr>
</tbody>
</table>

The balance here grows exponentially. Recall the attempt to model the function \( f(x) = x^2 \) using Local Iterated Function Systems. Since an exponential function grows much faster than the quadratic function it is certain that the same method will fail with this type of data, especially for predictive purposes (interpolation / extrapolation). But, notice that the logarithms of the data are linear and the method is very successful for modeling and predicting with linear data including predictions beyond the last data value. So, the logarithms of the data could be modeled with no error and the original data could then be retrieved by taking the exponential of the model (fixed point) values.

This motivates two potential modifications. First, it may be desirable to model the logarithms of stock data rather than the original data. Furthermore, domain values should be mapped to range values so their initial values correspond. By making the initial values correspond, comparisons of yield or rate of return over the
intervals corresponding to the domain and range is easier and more meaningful. The computation of the parameters $a$ and $b$ for a least-squares mapping that maps the first domain value to the first range value is shown below.

Determine the parameters $a$ and $b$ for the least-squares map $y = ax + b$ from $(x_1, x_2, \ldots, x_n)$ to $(y_1, y_2, \ldots, y_n)$ such that $x_1$ maps to $y_1$, or $y_1 = a_1 \cdot x_1 + b$.

Then $b = y_1 - ax_1$. Let $\text{Error} = \sum_{i=1}^{n} (ax_i + b - y_i)^2 = \sum_{i=1}^{n} (ax_i + (y_1 - ax_1) - y_i)^2$

Then $\frac{d \text{Error}}{da} = \sum_{i=1}^{n} 2(ax_i + (y_1 - ax_1) - y_i)(x_i - x_1) = \sum_{i=1}^{n} 2[a(x_i - x_1) + (y_1 - y_i)](x_i - x_1)$

$= 2a \sum_{i=1}^{n} (x_i - x_1)^2 - 2 \sum_{i=1}^{n} (y_i - y_1)(x_i - x_1)$

Setting $\frac{d \text{error}}{da} = 0$ to find the value of $a$ that gives the minimum, gives the equation

$2a \sum_{i=1}^{n} (x_i - x_1)^2 - 2 \sum_{i=1}^{n} (y_i - y_1)(x_i - x_1) = 0$

Then $a = \frac{\sum_{i=1}^{n} (y_i - y_1)(x_i - x_1)}{\sum_{i=1}^{n} (x_i - x_1)^2}$ and $b = y_1 - ax_1$, if $x$ is not constant.

If $x$ is constant, then set $a = 0$ and $b = y_1$.

Further justification for the refinements seems necessary.
Justification for the Method

Fractal Structure and the Use of Logarithmic Returns with Stock Market Data

The original method presented in the last chapter worked well in a simple situation with linear data which happens to be self-similar. It was shown to be less useful for non-linear data (quadratic), though exponential data could be modeled by transforming it into linear data using logarithms. In this chapter, justification will be given for using similar techniques with stock market data. To do this, it will be shown that the distribution of logarithmic returns from stock market data is statistically self-similar. Self-similarity may justify the model, but won’t guarantee that prediction of future events are possible. Predictions will be attempted and studied later.

Statistical self-similarity must be distinguished from the strict geometric self-similarity presented earlier. This can be done by example. Consider a time series where increments are independent identically distributed normal random variables. Let $x_0 = 0$ and $x_i = x_{i-1} + \text{Norm}(0,1)$ where $\text{Norm}(0,1)$ is a normal random variable with mean 0 and variance 1. By definition, $x_{i+1} - x_i \sim \text{Norm}(0,1)$. Furthermore, $x_{i+\text{lag}} - x_i \sim \text{Norm}(0,\text{lag})$ since the variance of the sum of independent normal random variables is the sum of the variances of those random variables, or $1 \times \text{lag}$ in this case, and the mean is the sum of the means which is 0.

Thus, $x_{i+\text{lag}} - x_i$, can be scaled using its standard deviation multiplying by \[ \frac{1}{\sqrt{\text{lag}}} \] to give a distribution identical to that of $x_{i+1} - x_i \sim \text{Norm}(0,1)$, or, equivalently, $\frac{x_{i+\text{lag}} - x_i}{\sqrt{\text{lag}}} \sim \text{Norm}(0,1)$. For this time series, differences between data sampled at a given frequency (indicated by $\text{lag}$) will have the same distribution as differences between data sampled at different frequencies when rescaled by an appropriate factor. This leads to the conclusion that domain and range values will be identically distributed when scaled by that factor.
The similarity described here is in distributions and thus, there is statistical self-similarity. Then finding matching subsequences becomes a matter of probability. Similar results will be guaranteed by the Central Limit Theorem if the increments are independent identically distributed random variables from a distribution with finite mean and variance. The scale factor, which depends on the "lags" of the subsequences, may give an indication of the expected contraction factors for the submappings of a Local IFS. It may be possible to model such a time series with a Local IFS, but predictions of future events can not be accurate since the increments are random with no dependence on past increments. Hopefully, stock data will be more favorable in this regard. This time series is a type of random walk and has a visual similarity to stock market data. There may even appear to be trends, though they are only illusions since the increments are random and independent.

Statistical self-similarity can be observed graphically in several ways. First, increments can be plotted and compared for different lags. Since they are independent (for this time series), no trends should be present. Dispersion is the key feature of these plots which are shown for the first 100 increments with lag=1, the first 100 increments with lag=2, and the first 100 increments with lag=4. The same plots are shown alongside with increments scaled so they appear as samples from a Norm(0,1) distribution.
Dispersion increases as the lags increase as shown in the first column. The rescaled plots in the second column visually show roughly the same dispersions as for lag=1. This could be verified quantitatively to show evidence of self-similarity in the sample distributions.
Similar comparison of dispersion can be made with plots of the cumulative increments. The rescaled plots
appear to have similar dispersion and appearance regardless of the lag whereas the unscaled plots in the first
column clearly have different dispersions and appearances for different lags.
Comparing plots of increments or cumulative increments seems to show statistical self-similarity where rescaling compensates for different lags. But, quantitative methods would be necessary to verify this. Other graphical methods may be more useful since, as may be discussed later, standard quantitative methods are not appropriate for other time-series data if distributions do not have "nice" properties like those of the normal distribution.

The following plots show tail frequencies for the increment distributions. The left tail is taken to be the negative data and the right tail is the positive data. Cumulative frequencies are determined starting at the extremes so increments near 0 should correspond approximately to cumulative frequency 0.5 since their distribution is symmetric. Log-log plots are used because multiplicative scaling will be indicated if the plot of cumulative tail frequencies for a given lag is a translation of the plot for another lag. (Multiplication by a constant becomes addition of a constant after taking the logarithm and addition of a constant is recognized in a plot as a translation.) In terms of distributions, if \( F_1(x) \) is the cumulative frequency function for lag 1 and likewise for \( F_2(x) \) and lag 2 then this scaling means that there is a constant \( a \) such that \( F_2(ax) = F_1(x) \) for all appropriate values of \( x \). Note too that a log-log plot will show the frequencies at the extremes in more detail than a non-logarithmic plot would.

**Left Tail Cumulative Frequencies: \( \text{lag}=1,2,4 \)**

![Graph of left tail cumulative frequencies with log-log scale for different lags.]
Notice that the plot of frequencies for the lags appear to be horizontal translations of each other indicating multiplicative scaling. Notice too that the plots for the right tails and left tails appear nearly identical indicating the symmetry of the distribution. (Plotting frequencies on a log scale is not necessary since the translations will be strictly horizontal for this data.)

This sort of visual comparison is more useful than those presented earlier since scaling can be recognized without the quantitative comparisons that seem necessary for the other types of plots. A note of caution should be made. This comparison method does not account for the order of the increments. For this dataset, consecutive increments are independent. For other datasets, that may not be true. In fact, if predictions are to be made then it is hoped that there will be some sort of dependence between consecutive data items. Taking this into account, this comparison method seems to be useful and was used by Mandelbrot for showing scaling properties in distributions [Mandelbrot, 1983], [Peters, 1995].

Another method for finding self-similarity is both graphical and quantitative. The fractal dimension of the time-series can be determined. The method is identical to the one used for the classic problem of determining the length of the coastline of Britain. A detailed description of this method is in [Peitgen, Jurgens, and Saupe, 1992]. The fractal dimension is computed as $1 + d$ where $d$ is the coefficient in the
fitted equation \( \log u = d \cdot \log \frac{1}{s} + b \) where estimates, \( u \), of the “length” of the time series plot are computed using different lags \( (s) \). The values for \( d \) and \( b \) are the slope and intercept for the least-squares equation for \( \log \frac{1}{s} \) versus \( \log u \). The \( s, u \), and corresponding logarithm values are shown below.

### Computation of Fractal Dimension of Time Series

<table>
<thead>
<tr>
<th>lag ((s))</th>
<th>( \log \frac{1}{s} )</th>
<th>length ((u))</th>
<th>( \log u )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000</td>
<td>13055.48</td>
<td>4.116</td>
</tr>
<tr>
<td>2</td>
<td>-0.301</td>
<td>9301.56</td>
<td>3.969</td>
</tr>
<tr>
<td>4</td>
<td>-0.602</td>
<td>6537.08</td>
<td>3.815</td>
</tr>
<tr>
<td>8</td>
<td>-0.903</td>
<td>4608.52</td>
<td>3.664</td>
</tr>
<tr>
<td>16</td>
<td>-1.204</td>
<td>3444.56</td>
<td>3.537</td>
</tr>
<tr>
<td>32</td>
<td>-1.505</td>
<td>2437.57</td>
<td>3.387</td>
</tr>
<tr>
<td>64</td>
<td>-1.806</td>
<td>1723.50</td>
<td>3.236</td>
</tr>
<tr>
<td>128</td>
<td>-2.107</td>
<td>1183.12</td>
<td>3.073</td>
</tr>
<tr>
<td>256</td>
<td>-2.408</td>
<td>859.61</td>
<td>2.934</td>
</tr>
<tr>
<td>512</td>
<td>-2.709</td>
<td>605.18</td>
<td>2.782</td>
</tr>
<tr>
<td>1024</td>
<td>-3.010</td>
<td>552.46</td>
<td>2.742</td>
</tr>
</tbody>
</table>

The value of \( d \) is 0.472 so the fractal dimension is approximately 1.472. Peitgen, Jürgens, and Saupe [Peitgen, Jürgens, and Saupe, 1992] show that for such a time-series (an unbiased random walk), the theoretical fractal dimension is 1.5 which is not far from the computed value. A plot of the logarithmic values is shown below.
The plot shows that least-squares regression is appropriate for computing a value for $d$ because the points fall very close to a straight line with the exception of the point corresponding to $s=1024$. (Without that point, the slope would be closer to $0.5$, the theoretical value.) The linearity of the data in this plot shows that the time-series has a fractal structure. This gives a scaling relationship that takes into account the order of the data. Mandelbrot gives hope for finding self similarity in stock market data in both the increment distributions and the traces with several statements that followed his own research.

He called the following statement the “Scaling Principle of Price Change.”

**SCALING PRINCIPLE OF PRICE CHANGE**

*When $X(t)$ is a price, $\log X(t)$ has the property that its increment over an arbitrary time lag $d$, $\log X(t+d)-\log X(t)$, has a distribution independent of $d$, except for a scale factor.*

[Mandelbrot, 1983]

This principle indicates that scaling distributions should be expected for the logarithms of price increments. This will be investigated. Note that logarithmic differences give more useful information for returns than absolute differences in price. Logarithmic differences are equivalent to percentage changes (and are thus, independent of value) whereas differences in price can be misleading. Suppose the value of a stock increases
$20. If the original value was $20, then this represents a 100% increase. If the original value was $100, then it represents a 20% increase, a far less profitable return on original value. But, since

$$\log X(t + d) - \log X(t) = \log \frac{X(t + d)}{X(t)}$$

equal logarithmic differences correspond to equal percentage rates of return. This gives further motivation using logarithmic returns, though scaling must still be checked.

With regard to time-series stock data, the following statement is encouraging.

One starts with the distribution of daily price changes over a period of five years of middling price variability. And one finds that if this distribution is extrapolated to monthly price changes, its graph goes right through the data from various recessions, depression, etc. It accounts for all the most extreme events of nearly a century in the history of an essential and most volatile commodity factor.

[Mandelbrot, 1983]

This seems to say that stock data appears similar despite the time frame (lag) used, given an implied scale factor. Since the data has self similarity (and thus a fractal structure) there is hope for finding nearly identical matches of subsequences within or across time scales. It is from these matchings that meaningful predictions can be attempted.
Fractal Structure in Dow Returns

16,047 daily closing values for the Dow from January 2nd, 1930 to December 29th, 1989 were used as a dataset. The values and their logarithms (base 10) are shown below. The logarithms give a better indication of changes relative to current values as discussed earlier. Notice that the decline in 1987 is more noticeable in the original data values though other declines over slightly longer time periods were more drastic relative to the value of the Dow at those times as seen in the logarithmic plot.
The following log-log graphs of tail frequencies for increments with lags 1, 2, 4, and 8 seem to show scaling that Mandelbrot described. The first 16,000 data values were used to find these frequencies. This scaling is similar to that found for time series with independent identically distribute normal increments. But notice that for the left and right tails, unusual features appear at the extremes. Even though there are 16,000 data points for lag=1, 8000 for lag=2, etc., this may be a consequence of lack of data. For the left tails, 0.111 is an extreme value corresponding to the 508 point drop on “Black Monday,” October 19th, 1987 which is the greatest single day absolute and relative decrease. This value should not be disregarded. It causes the “jumps” in the plots for lag=1 and lag=2. In the right tails, the plots for lag=2 and lag=4 unexpectedly cross, though at the extremes. Nevertheless, scaling seems likely since horizontal translations give roughly the same plots for a majority of the data.
The fractal dimension of the trace can also be computed and used as evidence of scaling. The fractal dimension is computed as 1.45, not much different than what was found for a random walk. Scaling seems likely since the data shown is linear.
Notice that the left and right tails appear nearly identical, though right tail frequencies appear slightly lower near 0, greater for intermediate values, and lower for the most extreme values. Since the Dow increased in general over the period studied, there is a bias of some sort toward increases which should be indicated by right tail values. There is a noticeable discrepancy at the extremes, which may be the consequence of insufficient data. Even though 6 decades of data is used, this may still be insufficient to indicate true theoretical frequencies for the extremes. Note that the discrepancies occur after frequency 0.001 or one one-thousandth so there are 16 or less associated data values.

The following plot again shows the left and right tails and includes a plot for the normal distribution. The result should be surprising because even if the increments were not normal but were still independent, identically distributed, and had a finite mean and variance, then the sum of returns (which would be seen for increasing lags) would approach a normal distribution. This plot shows a distinct difference in appearance of the frequencies for Dow returns (increments) and for normal increments. Dow increments appear to have higher frequencies at the extremes.
The Central Limit Theorem, given these assumptions, would predict that Dow frequencies approach normal frequencies for greater lags. But, as shown earlier, Dow frequencies appear to scale and therefore do not approach normal frequencies indicating that at least one of the assumptions is incorrect. Either the Dow returns are not independent, not identically distributed, do not have finite mean or variance, or some combination of these. This was studied in detail by Mandelbrot who attributes it to what he calls “Infinite Variance Syndrome,” and when there is infinite variance the Central Limit Theorem will not apply. Mandelbrot describes a class of scaling distributions called “Stable Paretian Distributions” [Mandelbrot, 1983] that account for such behavior in non-normal distributions.

Recall that scaling and self-similarity are desirable features for the type of modeling that will be attempted. The normal distribution scales, but is not desirable since increments are independent. There is no correlation between consecutive increments. Dow increments scale and it is hoped that there is a correlation between successive increments so predictions of future returns will be possible. If there is no correlation, then predictions will not be likely. One way to test for such a correlation is to use a technique that Edgar Peters uses in another type of analysis of returns [Peters, 1993].

He scrambles (randomizes the order of) the original data and compares the scrambled data to the original data. This seems to be a useful technique here for checking for sequential correlation of data. It is possible
that sequential correlation could cause Dow increments to differ from normal increments (since it would violate the independence assumption.) Then, if the increments are scrambled, the frequencies would approach normal frequencies. Frequencies for the tails with lags 1, 2, 4, 8, 16, and 32 are shown below for the original data and for scrambled data.

**Dow Left Tails: lag=1,2,4,8,16,32**

![Dow Left Tails Graph]

**Dow Right Tails: lag=1,2,4,8,16,32**

![Dow Right Tails Graph]
For lag=1, the plots will be identical for the original data and the scrambled data. For increasing lags, changes would be expected if there is sequential correlation. Notice though that the scrambled frequencies still seem to scale and are nearly identical to those for the original data, with the exception of the extremes. The frequencies do not appear to approach normal frequencies. This may mean that prediction of future events using the model may not be possible. Note that in the original data extreme jumps appeared for
lags=1 and 2 but disappeared for greater lags. This could indicate isolated dependence and behavior related to the jumps (recovery from the decreases). In the scrambled data a jump appears for each lag, showing the absence of such behavior. This may give some hope for dependence.

Another way of checking for sequential correlation is a plot called a correlogram that shows the Pearson Correlation Coefficient for consecutive increments with different lags.

The following plot shows the Pearson Correlation Coefficient is computed for all data pairs \((X_i, X_{i+l})\) from the sequence for \(i=1, 2, 3, ..., 100\). Notice that the correlations are all close to 0 with some variation as would be expected.

![Correlogram for 1000 Normal Increments](image)

Again, what is shown in these plots is the Pearson Correlation Coefficient for subsequences of increments where the subsequences are offset by a given value. For example, the correlation between consecutive increments can be computed. A plot of daily increments versus increments that immediately follow is shown below. There appears to be no correlation between successive daily increments as is indicated in the correlogram and the correlation coefficient is 0.051 which is very close to 0. This contradicts an expectation that positive increases would generally be followed by positive increases and likewise for negative increments yielding trends if there was a sequential dependence in the data.
Following are correlelograms for daily, weekly, and monthly increments. They seem to show no correlation between successive increments for any of the time frames. Notice that the range of correlations increases when going from daily to weekly and then to monthly increments. Recalling the correlations for the independent normal increments where the sample size was 1000, this increase in correlation is most likely a consequence of decreasing sample size. There are approximately 16,000 pairs for daily data, 3000+ pairs for weekly data, and about 800 pairs for monthly data.
Differences between the Dow frequency plots and normal frequency plots could also be attributed to bias.

The Dow average increased from 1930 to 1989 while a time-series with normal increments with mean 0
neither steadily increases or decreases over the whole time period. Dow returns increase and are biased toward increase, though periods of decrease do occur. A biased random walk can be made with Normal returns having a non-zero mean. Using \( \text{Norm}(0.1, 1) \) increments, a series with relatively steady increases can be formed. Frequencies for the tails can then be compared to those for a Standard Normal Distribution.

The following plot shows the trace of such a biased random walk. Notice the steady increase. There are places where decreases occur but they are dominated by increases since the increments are biased in the positive direction.

**Biased Random Walk: \( \text{Normal}(0.1, 0) \)**

The frequencies for the tails of the increment distribution follow. In each, a plot is given for the standard normal distribution (mean=0, variance=1) shown as a dotted curve. Notice that there is little difference in the appearance of the plots. This reinforces the distinction between normal increments and Dow increments. Theoretically it should be expected that frequencies at the extremes will correspond.
In summary, log differences in Dow returns (increments) appear to have scaling distributions that are non-normal. This is a desirable feature for the modeling method described earlier. Yet, there is no evidence of sequential correlation between increments. So, such a model may "fit" the data for stock returns but may be of little use in predicting future returns. This will now be tested.
Testing the Method

Modeling the Dow

As a first test, a Local IFS will be found for the first 16,000 daily Dow values from the dataset. This model will then be used to predict the remaining 47 data values from the original dataset. The modeling method will be identical to that used earlier. Domains will be chosen from weekly returns (every 5th value) and ranges will be roughly one month's data (20 days). The method for extrapolating the missing 47 values will be slightly different from those described earlier. The mapping for the last range and its corresponding domain will be applied to the 47 weekly values that follow the domain values. The results will then be compared to the original 47 values. The motivation for this is that the chosen domain best fits that range (by the modified least-squares computation) and that fit may extend to values beyond the domain and range values. Given the lack of evidence of correlations between successive increments, success is not anticipated. However, there may be a weak correlation between successive increments but a stronger correlation between sequences of increments and successive increments.

Following is the plot of logarithmic returns for the first 16,000 days from 1930 to 1989. Then, the plot of the Local IFS model (fixed point) is given. There is little noticeable difference between the two plots, so the model seems to fit well. Since there are 16000 values in the dataset and domains and ranges have length 20, there are 800 mappings in the local IFS. Each mapping has associated floating point $a$ and $b$ values and must include information about the domain (an index) and the range (which is implied.) So, the compression ratio is less than 10:1. The model's purpose is not compression, but if it was, this is not a very high compression ratio. A higher ratio can be achieved by making the domains and ranges longer which may have an affect on the accuracy of predictions. A plot is also given for the pointwise error in the model. Corresponding summary error values comparing the model to the original data are: root mean square error
= 0.004, average error = 0.002 (the average of the pointwise errors), and maximum error = 0.086 (the maximum of the pointwise errors). This occurs, as expected, on Black Monday in October, 1987.

**Logarithms of 16,000 Dow Values, 1930-1989**

**Model of Logarithmic Values: 20 Day Ranges**
Model Error: 20 Day Ranges

For the model, indices start at 0 so the last data value has index 15999. The last range includes values from indices 15980 to 15999 and the corresponding domain has values from indices 13400, 13405, ..., 13495.

The mapping from the domain to the range has parameters $a = 0.6405$ and $b = 1.555$ and the domain, range, and $y = ax + b$ values are summarized below.

### $ax + b$ Values and Range Values

<table>
<thead>
<tr>
<th>domain value</th>
<th>range value</th>
<th>$ax + b$ value</th>
<th>error</th>
<th>index</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.921</td>
<td>3.426</td>
<td>3.426</td>
<td>0.0003</td>
<td>15980</td>
</tr>
<tr>
<td>2.918</td>
<td>3.427</td>
<td>3.424</td>
<td>0.003</td>
<td>15981</td>
</tr>
<tr>
<td>2.924</td>
<td>3.431</td>
<td>3.428</td>
<td>0.003</td>
<td>15982</td>
</tr>
<tr>
<td>2.927</td>
<td>3.43</td>
<td>3.43</td>
<td>0.0002</td>
<td>15983</td>
</tr>
<tr>
<td>2.938</td>
<td>3.434</td>
<td>3.437</td>
<td>0.003</td>
<td>15984</td>
</tr>
<tr>
<td>2.946</td>
<td>3.44</td>
<td>3.442</td>
<td>0.002</td>
<td>15985</td>
</tr>
<tr>
<td>2.945</td>
<td>3.443</td>
<td>3.441</td>
<td>0.002</td>
<td>15986</td>
</tr>
<tr>
<td>2.948</td>
<td>3.443</td>
<td>3.443</td>
<td>0.0003</td>
<td>15987</td>
</tr>
<tr>
<td>2.943</td>
<td>3.445</td>
<td>3.44</td>
<td>0.005</td>
<td>15988</td>
</tr>
<tr>
<td>2.945</td>
<td>3.446</td>
<td>3.441</td>
<td>0.004</td>
<td>15989</td>
</tr>
<tr>
<td>2.947</td>
<td>3.445</td>
<td>3.443</td>
<td>0.002</td>
<td>15990</td>
</tr>
<tr>
<td>2.941</td>
<td>3.443</td>
<td>3.439</td>
<td>0.004</td>
<td>15991</td>
</tr>
<tr>
<td>2.946</td>
<td>3.441</td>
<td>3.442</td>
<td>0.001</td>
<td>15992</td>
</tr>
<tr>
<td>2.92</td>
<td>3.41</td>
<td>3.425</td>
<td>0.015</td>
<td>15993</td>
</tr>
<tr>
<td>2.908</td>
<td>3.424</td>
<td>3.418</td>
<td>0.007</td>
<td>15994</td>
</tr>
<tr>
<td>2.908</td>
<td>3.421</td>
<td>3.417</td>
<td>0.004</td>
<td>15995</td>
</tr>
<tr>
<td>2.91</td>
<td>3.422</td>
<td>3.419</td>
<td>0.003</td>
<td>15996</td>
</tr>
<tr>
<td>2.915</td>
<td>3.429</td>
<td>3.422</td>
<td>0.007</td>
<td>15997</td>
</tr>
<tr>
<td>2.911</td>
<td>3.43</td>
<td>3.42</td>
<td>0.01</td>
<td>15998</td>
</tr>
<tr>
<td>2.917</td>
<td>3.425</td>
<td>3.423</td>
<td>0.002</td>
<td>15999</td>
</tr>
</tbody>
</table>

76
There seems to be a good fit between the $ax+b$ values and the data though at index 15993 there is an error of 0.015. The following graph shows the fit.

Graph of $ax+b$ Values and Last Range Values

Now the domains and ranges can be extended and $ax+b$ values can be computed to extrapolate values for indices 16000 through 16046. The results are shown in the following table.
### Extrapolated Values and Actual Range Data Values

<table>
<thead>
<tr>
<th>domain value</th>
<th>range value</th>
<th>$ax+b$ value</th>
<th>error</th>
<th>index</th>
<th>domain value</th>
<th>range value</th>
<th>$ax+b$ value</th>
<th>error</th>
<th>index</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.916</td>
<td>3.425</td>
<td>3.423</td>
<td>0.002</td>
<td>16000</td>
<td>2.933</td>
<td>3.432</td>
<td>3.434</td>
<td>0.002</td>
<td>16024</td>
</tr>
<tr>
<td>2.921</td>
<td>3.424</td>
<td>3.426</td>
<td>0.002</td>
<td>16001</td>
<td>2.926</td>
<td>3.43</td>
<td>3.429</td>
<td>0.003</td>
<td>16025</td>
</tr>
<tr>
<td>2.924</td>
<td>3.417</td>
<td>3.428</td>
<td>0.01</td>
<td>16002</td>
<td>2.937</td>
<td>3.432</td>
<td>3.436</td>
<td>0.003</td>
<td>16026</td>
</tr>
<tr>
<td>2.923</td>
<td>3.414</td>
<td>3.427</td>
<td>0.013</td>
<td>16003</td>
<td>2.944</td>
<td>3.432</td>
<td>3.441</td>
<td>0.002</td>
<td>16027</td>
</tr>
<tr>
<td>2.914</td>
<td>3.416</td>
<td>3.421</td>
<td>0.006</td>
<td>16004</td>
<td>2.943</td>
<td>3.44</td>
<td>3.44</td>
<td>0.002</td>
<td>16028</td>
</tr>
<tr>
<td>2.934</td>
<td>3.422</td>
<td>3.434</td>
<td>0.012</td>
<td>16005</td>
<td>2.941</td>
<td>3.438</td>
<td>3.438</td>
<td>0.005</td>
<td>16029</td>
</tr>
<tr>
<td>2.936</td>
<td>3.423</td>
<td>3.436</td>
<td>0.013</td>
<td>16006</td>
<td>2.953</td>
<td>3.437</td>
<td>3.446</td>
<td>0.009</td>
<td>16030</td>
</tr>
<tr>
<td>2.944</td>
<td>3.42</td>
<td>3.441</td>
<td>0.021</td>
<td>16007</td>
<td>2.956</td>
<td>3.435</td>
<td>3.449</td>
<td>0.014</td>
<td>16031</td>
</tr>
<tr>
<td>2.942</td>
<td>3.42</td>
<td>3.44</td>
<td>0.02</td>
<td>16008</td>
<td>2.968</td>
<td>3.436</td>
<td>3.456</td>
<td>0.019</td>
<td>16032</td>
</tr>
<tr>
<td>2.947</td>
<td>3.412</td>
<td>3.443</td>
<td>0.031</td>
<td>16009</td>
<td>2.971</td>
<td>3.436</td>
<td>3.458</td>
<td>0.022</td>
<td>16033</td>
</tr>
<tr>
<td>2.951</td>
<td>3.414</td>
<td>3.445</td>
<td>0.031</td>
<td>16010</td>
<td>2.972</td>
<td>3.44</td>
<td>3.459</td>
<td>0.019</td>
<td>16034</td>
</tr>
<tr>
<td>2.939</td>
<td>3.419</td>
<td>3.437</td>
<td>0.019</td>
<td>16011</td>
<td>2.977</td>
<td>3.441</td>
<td>3.462</td>
<td>0.021</td>
<td>16035</td>
</tr>
<tr>
<td>2.936</td>
<td>3.416</td>
<td>3.436</td>
<td>0.02</td>
<td>16012</td>
<td>2.976</td>
<td>3.44</td>
<td>3.461</td>
<td>0.021</td>
<td>16036</td>
</tr>
<tr>
<td>2.914</td>
<td>3.419</td>
<td>3.421</td>
<td>0.002</td>
<td>16013</td>
<td>2.975</td>
<td>3.438</td>
<td>3.46</td>
<td>0.023</td>
<td>16037</td>
</tr>
<tr>
<td>2.909</td>
<td>3.419</td>
<td>3.418</td>
<td>0.0009</td>
<td>16014</td>
<td>2.977</td>
<td>3.431</td>
<td>3.462</td>
<td>0.031</td>
<td>16038</td>
</tr>
<tr>
<td>2.895</td>
<td>3.417</td>
<td>3.409</td>
<td>0.007</td>
<td>16015</td>
<td>2.974</td>
<td>3.431</td>
<td>3.46</td>
<td>0.029</td>
<td>16039</td>
</tr>
<tr>
<td>2.891</td>
<td>3.42</td>
<td>3.407</td>
<td>0.014</td>
<td>16016</td>
<td>2.981</td>
<td>3.429</td>
<td>3.464</td>
<td>0.035</td>
<td>16040</td>
</tr>
<tr>
<td>2.886</td>
<td>3.421</td>
<td>3.403</td>
<td>0.018</td>
<td>16017</td>
<td>2.98</td>
<td>3.43</td>
<td>3.464</td>
<td>0.034</td>
<td>16041</td>
</tr>
<tr>
<td>2.895</td>
<td>3.424</td>
<td>3.409</td>
<td>0.015</td>
<td>16018</td>
<td>2.974</td>
<td>3.433</td>
<td>3.46</td>
<td>0.027</td>
<td>16042</td>
</tr>
<tr>
<td>2.88</td>
<td>3.42</td>
<td>3.4</td>
<td>0.02</td>
<td>16019</td>
<td>2.982</td>
<td>3.433</td>
<td>3.465</td>
<td>0.032</td>
<td>16043</td>
</tr>
<tr>
<td>2.906</td>
<td>3.421</td>
<td>3.416</td>
<td>0.005</td>
<td>16020</td>
<td>2.982</td>
<td>3.435</td>
<td>3.465</td>
<td>0.029</td>
<td>16044</td>
</tr>
<tr>
<td>2.912</td>
<td>3.424</td>
<td>3.42</td>
<td>0.004</td>
<td>16021</td>
<td>2.973</td>
<td>3.437</td>
<td>3.459</td>
<td>0.023</td>
<td>16045</td>
</tr>
<tr>
<td>2.906</td>
<td>3.427</td>
<td>3.416</td>
<td>0.011</td>
<td>16022</td>
<td>2.963</td>
<td>3.44</td>
<td>3.453</td>
<td>0.013</td>
<td>16046</td>
</tr>
<tr>
<td>2.92</td>
<td>3.431</td>
<td>3.425</td>
<td>0.006</td>
<td>16023</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notice that the errors are generally greater for the extrapolated values than for the values in the original domain. The method does not appear to work well for predicting this data. This can be seen more clearly from the graph of data values, which shows that the errors are relatively large. Recall that the fit data corresponds to indices 0-19 and the extrapolated data follows.
This graph is reminiscent of one that demonstrates the Butterfly Effect for Lorenz's data. Recall that the Butterfly Effect is also referred to as "sensitivity to initial conditions." In his graph, which shows a simulation of weather conditions, "weather patterns" correspond closely for a short period of time and soon diverge to a point where there is no correspondence as shown below.

Lorenz's Diverging Weather Patterns (The Butterfly Effect)

How two weather patterns diverge. From nearly the same starting point, Edward Lorenz saw his computer weather produce patterns that grew farther and farther apart until all resemblance disappeared. (From Lorenz's 1961 printouts.)

[Gleick, 1987]
Similar results for stock data should not be surprising if the stock market is chaotic. This concern was expressed when I started my research and I was well aware of it before pursuing development of this model. This seems to provide more evidence that predictions of this type are difficult to make accurately. Several more attempts will be made in support of this conclusion. First, the same data will be modeled using domains and ranges of length 100. The corresponding Local IFS model will be found, compared to the original data, and then the mapping for the last range will be used to extrapolate values for the 47 data values that were left out. Perhaps results will be better since there will be an established fit for 100 data values preceding the 47 values to be extrapolated as compared to a fit of just 20 values. But, results similar to the Butterfly Effect shown in Lorenz's graph should not be a surprise.

![Model of Logarithmic Values: 100 Day Ranges](image)

For this model, root mean square error = 0.01022, average error = 0.00658, and maximum error = 0.12732. These values are worse than those for 20 day ranges as would be expected. But, the model still seems to fit the data well. Again, there is a large error on Black Monday in October, 1987 though it is not the greatest error. That occurs for data in the 1930's.
The range corresponds to indices 15900, 15901, ..., 15999 and the domain corresponds to indices 830, 835, 840, ..., 1325. The mapping parameters are $a = 0.14584$ and $b = 3.13895$. The following plot shows the mapped $ax + b$ values in comparison to the corresponding range values for the original data along with extrapolated values for the missing data values and the actual data values. Notice that the fit between the mapping and the range values isn’t as close as the fit was for the 20 day range. In general, the extrapolated values appear to fit the actual data values for the remainder of the data no better than data fit to the range values. But the error between the extrapolated values and the actual values is less than that for the earlier data. No conclusions can be made from this graph. It could be read as a somewhat reasonable prediction or as a poor fit. The actual value for index 16046, 47 days “into the future”, is 3.44 and the predicted value from the model is 3.431 which is pretty close. When using the 20 day returns the predicted value was 3.453, a little further from the actual value. For other indices, errors are greater.
Graph of $ax+b$ Values and Range Values: 100 Day Ranges

In a final attempt to see if this method can work a disheartening observation must be made. The model has been shown to fit given data reasonably well. So far though, the model has been of questionable use in predicting future data values. Beyond that, it should be noted that most of the information from the Local IFSs is not used in making the predictions. The only information that is used is the specification of the mapping that corresponds to the last range in the portion of the data that is modeled, which is then extended to extrapolate future values. The model (fixed point) values are never used. Recall that these values will likely differ from the $ax+b$ values found by mapping domain values directly. So, the only relevant information is the one mapping. Then it would make sense for prediction purposes to omit the Local IFS altogether and simply search for mappings that best fit a range at the end of the data. This would be more efficient allowing more time to search for a better match for the end data. In the examples, the domain pool consisted of “weekly” data only. This was done to make the process of finding the model more efficient though it may have sacrificed matching better fitting domains to ranges. Despite this, the model seemed to fit the data well even though that seems to be of little use for the prediction application. It simply verifies that there is self-similarity in the data.
By focusing on end data alone, more time can be spent finding better matches by expanding the pool of potential domains. All of the data can be searched for domains using different lags and a set of the best fits can be chosen and used for prediction purposes in the same way as has been presented.

For this test, the range length is chosen to be 50 and corresponds to indices 15950 through 15999 and values will be extrapolated again for indices 16000-16046. This time the domains will not be limited to weekly data. Instead, all domains will be checked for lags 1, 2, 3, 4, and 5 and the 10 that give the best fit will be chosen provided they satisfy criteria concerning the scale factor. Recall that it may be desirable to choose domains that roughly correspond to the range in terms of return rate. So the modified least-squares method that maps first elements to each other was used. Furthermore, for lag=1, the scale factor was limited to 0.9<α<1.1 so return rates nearly correspond. Similarly, for lag=2 it is restricted to 0.45<α<0.55 and so on for lag=3, 4, and 5. Certainly this restricts the domains but provides more “meaningful” matches. The results are shown as follows. Errors are given for fits to the range and not the extrapolated data.

**Predicted Values Using Modified Least-Squares Method**

- **domain=7505, lag=4, a=0.26547, b=2.71692, sumsqerr=0.00166, maxerr=0.02097**
- **domain=6714, lag=2, a=0.53438, b=2.12064, sumsqerr=0.00166, maxerr=0.02109**
- **domain=7506, lag=4, a=0.27412, b=2.69338, sumsqerr=0.00167, maxerr=0.02031**
- **domain=9741, lag=3, a=0.35185, b=2.39403, sumsqerr=0.00169, maxerr=0.0218**
Notice that six of the mappings have lag=4 and domains that start near 7500. It shouldn’t be surprising that for moderate to large lags, domains in the same time frame will be similar. Maybe only the best of these six should have been shown since all give roughly the same result. None of predictions for any of the matchings are perfect or near perfect. The one corresponding to domain=11173 and lag=1 is by far the worst. The predicted values immediately deviate from the actual values after the range indices. Other predictions are
somewhat close but do not capture the shape of the true data, especially in the extrapolated portions.

Perhaps better predictions could have been found by placing no restrictions on the mappings and their parameters. This can be attempted next.

Again, a range length of 50 is used. All potential domains with lags 1, 2, 3, 4, and 5 were considered and mappings were determined to minimize the sum-of-squares error with no restrictions on the scaling factor. It can be positive or negative and first values from domains need not be mapped exactly to first values from ranges. Three examples are given below. The resulting predictions appear no better than earlier ones with almost immediate deviation from the actual data following the range indices.

**Predictions Made Using Least Squares Matches with no Restrictions**

<table>
<thead>
<tr>
<th>Domain</th>
<th>Lag</th>
<th>A</th>
<th>B</th>
<th>Sumsqerr</th>
<th>Maxerr</th>
</tr>
</thead>
<tbody>
<tr>
<td>8343</td>
<td>4</td>
<td>0.47922</td>
<td>2.08465</td>
<td>0.00099</td>
<td>0.02126</td>
</tr>
<tr>
<td>9697</td>
<td>1</td>
<td>1.83607</td>
<td>-1.97108</td>
<td>0.00107</td>
<td>0.01122</td>
</tr>
<tr>
<td>15126</td>
<td>2</td>
<td>0.46512</td>
<td>1.9142</td>
<td>0.00113</td>
<td>0.01391</td>
</tr>
</tbody>
</table>

Obviously this method fails to give consistent and accurate moderate to long term predictions. This conclusion can be made from observation of the prediction graphs versus graphs of the actual data. Before
dismissing the method entirely, a final more rigorous test can be performed on a weaker type of prediction. The method can be modified to predict whether or not the market will go up (or down) on a given day. The method is modified in the following way. A sequence of returns preceding the day for which the prediction will be made is chosen as the range. From a pool of domains, the domain giving the least-squares error mapping is chosen. The mapping is extended using the next value beyond the domain to predict the value that would follow the range. If the predicted value exceeds the mapped value for the last day in the range, then it is predicted that the market will go up on that day.

Using Mappings to Predict Next Day Up or Down

In the figure shown above, the model is found using the mapping and domain giving the least-squares error for values in the range corresponding to actual data values in the range for indices 0-19. The mapping is extended using the domain value for index 20 and there is a slight decrease from the 19th and 20th values in the model which leads to a prediction of a decrease in the market on the 20th day. The actual data also shows a decrease between the 19th and 20th data elements so, in this case, the prediction is accurate.

The overall accuracy of this type of prediction can be compared with that of two random (guessing) methods. First, the change in the market could be predicting by guessing up or down with equal probabilities. A slightly more sophisticated method would involve randomly guessing an increase or decrease using probabilities based on the historical distribution of increases and decreases. If the described
method shows results that are better than those for the random methods and the difference is statistically significant, then there is hope that this method can be used to make accurate predictions.

Sets of predictions were made, each with different lags and domain lengths. Each set consisted of 151 predictions. Each prediction was made using a range of a given length and a domain pool consisting of all possible domains chosen from the 1000 data elements preceding the range.

For example, a set of predictions was made using lag=1 and len=20. The first prediction was made with the range corresponding to data elements 1001 through 1020. The best domain (the domain giving the least-squares error) was chosen from data elements 1 through 1000 and was extended and used to predict whether the market would go up or down on day 1021 as described earlier. The prediction was then compared to the actual change in the market. The second prediction was made using the range corresponding to data indices 1101 through 1120 and domains from indices 101 through 1100, and so on resulting in 151 predictions.

Once the accuracy ratio for a set of predictions was known, it was compared to what would be expected for chance predictions and tested to see if it was significantly different. Results are given below. lag=1..4 indicates that the domain pool included domains with lags 1, 2, 3, and 4. The null hypothesis $p=0.519$ corresponds to the historical percentage of increases in the market.

<table>
<thead>
<tr>
<th>lag</th>
<th>len</th>
<th>ratio of correct predictions</th>
<th>null hypothesis: $p=0.519$</th>
<th>null hypothesis: $p=0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>z-value</td>
<td>z-value</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>80/151=0.530</td>
<td>0.278</td>
<td>0.732</td>
</tr>
<tr>
<td>1..4</td>
<td>20</td>
<td>81/151=0.536</td>
<td>0.441</td>
<td>0.875</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>85/151=0.563</td>
<td>1.092</td>
<td>1.546</td>
</tr>
<tr>
<td>1..4</td>
<td>10</td>
<td>84/151=0.556</td>
<td>0.929</td>
<td>1.383</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>76/151=0.503</td>
<td>-0.374</td>
<td>0.081</td>
</tr>
<tr>
<td>1..4</td>
<td>5</td>
<td>71/151=0.470</td>
<td>-1.188</td>
<td>-0.732</td>
</tr>
</tbody>
</table>

None of the z-values fall above 1.96 or below -1.96 so neither null hypothesis can be rejected. So, there is no evidence that the prediction method is better than (or worse than) guessing. And this is for what may be
the weakest type of prediction that can be made, whether or not the market will be up or down tomorrow.

So, both long term and short term methods seem to fail.
Conclusion

Where has Providence Lead Me?

The application of fractals and chaos theory in data compression and economics are both hot topics when treated separately. What I've tried to do is combine aspects of both into a new and meaningful modeling technique. I can't take credit for this idea since it was my Advisor's, Dr. Alden Wright. Had I followed my inclinations I would have focused on Fractal Image Compression and its implementation. There is still room for research in that area and practical results would be useful in applications that involve both the storage and transmission of images and other types of data which is important in the telecommunications field. I'm glad I didn't follow that course though.

Even though the results of my prediction method are inconclusive, which is not surprising considering the nature of chaotic systems, I think there may be other ways to apply similar ideas to economic systems. I've learned a great deal from my research and could have continued indefinitely. It seems as though every time I tried a new idea I would come across another reference or other ideas to try and could have spent years in research. I'd like to mentioned several of these references which were particularly interesting and useful.

Benoit Mandelbrot is known as the father of Fractal Geometry. Something I didn't know was that much of his work in Fractal Geometry was born in research of cotton prices. Perhaps providence guided him from economics to discoveries that allow us to describe nature's objects: clouds, trees, and mountains. In the words of my friend Paul, this is the language of God. If that is true, then my work is evidence that though we may recognize God's language, few are able to interpret it.

Mathematicians and economists seem to speak different languages in terms of the technical jargon they use and their ways of thinking. Mandelbrot had difficulty expressing his discoveries to economists and even those who understood were more than reluctant to accept them over 30 years ago and there still is reluctance. A crucial feature of much of modern economic theory is the assumption of normal distributions.
This allows the use of the standard well-developed statistical analysis. Mandelbrot showed that this assumption is false for most market return values. Old theory is still accepted and used but is giving way to new methods that conform to Mandelbrot’s discoveries. I will discuss these shortly.

Mandelbrot used biblical references when coining the phrases “Joseph Effect” and “Noah Effect,” which describe certain characteristics of market returns. The Noah Effect describes the presence of sudden great increases or decreases and appropriately associates them with the great flood in the story of Noah. These increases and decreases are evident in the tail frequencies of the distributions of Dow returns. These are the catastrophes of the system like the fall of the market on Black Monday in October, 1987.

The Joseph Effect, which I’ve associated with my own past, describes the presence of a long-term memory or persistence in a system. It was so named because of the persistence (trends) of the seven years of plenty and then the seven years of famine in the story of Joseph. I searched for a correlation between successive Dow returns but was unable to find one. The Joseph Effect may refer to a memory different from the type I hoped to find. Though it would seem to have been of use for my model and method, it did not help me.

The notion intrigues me and I still find the story of Joseph appropriate as a theme for my research.

The story of Joseph took place in Egypt and, surprisingly, research of the Nile River from the first half of this century reappears in the study of the structure of market returns. Mandelbrot mentioned the work of H.E. Hurst, a hydrologist and dam builder who was interested in the flow of the Nile which had to be taken into account when determining the capacity of a dam. As was common for complex systems with many variables (such as the stock market), river influxes were considered to be normally distributed. But after examining over 800 years of data he found that influxes for the Nile were not normally distributed and even far from it. He proceeded to examine other rivers and natural phenomena finding similar results.

I will not describe the analysis he used, though I’ve done some research into it. I have found several papers and two particularly interesting books by an investment manager name Edgar Peters who has revived Hurst’s and Mandelbrot’s work [Peters, 1991], [Peters, 1995]. Peters is practitioner of these methods using real money and is not an academician, so I take his interest in the topic and his work seriously.
He uses and extends Hurst's analysis techniques to study volatility of stocks and commodities and search for periodic behavior in markets. He develops what he calls the Fractal Market Hypothesis which he claims corrects the failings of earlier hypotheses that assume or make use of normal distributions. Assessment of risk, a major component in Modern Portfolio Theory makes use of standard deviations computed assuming normal distributions. Peters applies Hurst analysis, which is related to fractal dimension, as an alternative way to assess risk or volatility. 30 years after Mandelbrot’s work, this is still a radical way of thinking in economics but it seems to be gaining wider acceptance as indicated by reviews of Peter’s work, at least those given on the books’ jackets.

Assessing risk seems to be a more global analysis than attempting to predict individual returns, as I’ve tried. Recall my example of the Sierpinski Triangle and the Chaos Game, which Peters uses, though for somewhat different reasons. Global predictions are possible for the results of the Chaos Game in the structure of the collection of points generated by the process -- the Sierpinski Triangle results every time. Local predictions (short term) are not possible due to randomness in the process -- it is not possible to predict the exact location of a particular game point. It is certain they will fall on (or very close to) the Sierpinski Triangle which is the attractor for the system. But where they fall on the triangle can not be determined in any sense other than probabilistic. Therefore, global analysis may be more practical. This seems to be indicated by Peters’ work.

Maybe this global structure is the long term memory described as the Joseph Effect. If that is the case, then perhaps providence has led me to that realization through the long and arduous process of researching and writing this thesis. Too bad I didn’t read Peters’ and Mandelbrot’s works earlier. They may have saved time and duplication of earlier work. But, sometimes providence leads you to learn through experience. Joseph endured the hatred of his brothers and slavery before becoming governor of Egypt. I guess I can complain about a several year long learning experience.
Bibliography


