**Determining the Dimensionality of Causal networks Based on Growth Derived from Sequential Substitution Systems**

**Amy Joy Beard**

Department of Physics

Southern Adventist University

Collegedale, TN 37315

amybeard@southern.edu

**Abstract**

In this paper, we introduce a method for determining the dimensionality of the growth for a causal network. The purpose is to provide a way of measuring the complexity of a causal network. In this process, we explain our method for deriving causal networks from sequential substitution systems. We also catalog the types of causal networks discovered and discuss their dimensionality. Networks of exact dimensionality of one and two have been found while only approximations of three dimensional and exponential networks have been discovered thus far.

1. INTRODUCTION

In the past ten years, scientists have been refreshed by Wolfram’s program *Mathematica,* which makes mathematical tools readily available and easy to use. What is not commonly known is that *Mathematica* was actually developed for the purpose of exploring what Stephen Wolfram calls a “new kind of science” [1]. Science observes phenomena and then creates models based on predominantly continuous functions in order to explain the phenomena. According to Gray [2], Wolfram suggests that time and space are discrete making it difficult to model phenomena on continuous models. The proposal for a new kind of science is to take the opposite approach by observing a large set of models via cellular automata, which can be created by a simple computer program. From these models, we then search for characteristics which resemble phenomena in the natural world. The nature of mathematical science is that we create a complicated rule or formula to model a simple behavior. Using this new kind of science, it is possible to describe both simple and complex behavior with a simple rule using cellular automata [1]. The causal networks that arise out of generating cellular automata using a simple program are of particular interest. Causal networks are defined by Wolfram as “acyclic graph[s] arising from the evolution of a substitution system.”[3] These networks are the models for which we are seeking to describe the real world.

The focus of our research is the dimensionality of causal networks produced from sequential substitution systems. In this paper, we will introduce what a sequential substitution system is and show how a causal network is generated. We will also discuss our method for generating sequential substitution systems and the types of causal networks that are found. Finally, we will discuss what we mean by dimensionality and what methods we have developed to determine this characteristic for any given causal network. The backbone of all this research is the sequential substitution system.

1. SEQUENTIAL SUBSTITUTION SYSTEMS

Our attention is turned to causal networks that arise from a specific type of substitution system. A sequential substitution system (SSS) is a substitution system in which a string is scanned from left to right in an attempt to recognize a specific pattern to apply a rule. The rules are applied in a specific order or sequence. That is, we start with considering the first rule for application scanning the string from left to right. If the pattern for the first rule is matched, then the corresponding pattern is substituted for the original pattern, which produces a new string. Then the process is repeated using the first rule. If an application to the first rule does not exist in the string under consideration, then the second rule is considered then the third and so on. This process is repeated on each newly created string every time a substitution occurs [1]. Each new string represents one substitution that has been applied to the previous one. No two rules are applied to the same string at the same time. This process may continue indefinitely or it may stop if no application is found for any substitution rule in the set. An example of a SSS is shown in Figure 1.

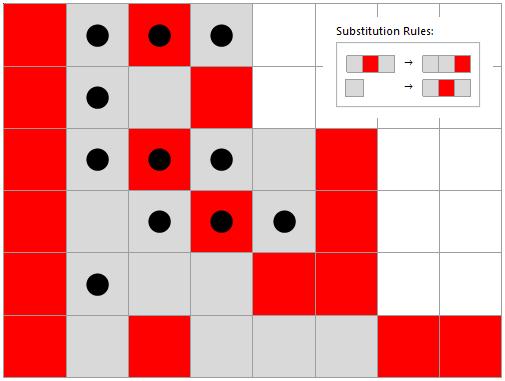


Figure Five iterations of SSS rule set “ABA”->”AAB” and “A”->”ABA” with the initial string “BABA”. The bold black dots represent matches to a specific rule which is then replaced by the substitute on the next line.

In this example, we see how a SSS evolves. We start with a beginning string which can be thought of as our initial condition. This can be seen on the top row of Figure 1 which is {Red, Gray, Red, Gray}. The first step is to scan the initial string from left to right looking for a match to the first rule. The three black dots on the first row represent a match to the first rule. The substitution is then made by replacing the {Gray, Red, Gray} portion with {Gray, Gray, Red}. The process then continues with the string on the second row. No match is found for the first rule, so the second rule is applied. This process is continues indefinitely or until there are no longer any applications to any rule in the set. This particular SSS is growing because a rule set exists that substitutes a string larger than itself.

1. DERIVATION OF CAUSAL NETWORKS FROM SEQUENTIAL SUBSTITUTION SYSTEMS

The causal networks derived from the sequential substitution systems are the main topic of our research. We are more interested in studying the characteristics of the causal networks as they may be used to describe natural phenomena. We have chosen a specific way to represent a SSS as a causal network. As in Wolfram’s book, our network is a set of nodes which are connected a certain way [1]. Every iteration performs a substitution which creates and destroys cells. Each node represents one event or substitution. The connections represent the “life” of a cell, tracing where the connection begins with the node that creates the cell and ends at the node where the cell is destroyed. Wolfram also describes a tagging system [1]. In order to keep track of creations and destructions for the connections, a tagging system was needed. In our system, every tag (in this case, the numbers naming each cell) represents a cell. It begins by starting with the initial string and numbering them from left to right. When a match is found, the matched cells are destroyed and replaced by cells as specified by the rule set. Cells replaced by the same type of cell are not considered to be the same and each receives a different tag. The newly created cells continue the numbering from left to right and so on. An example is in Figure 2 using the same SSS as in Figure 1.

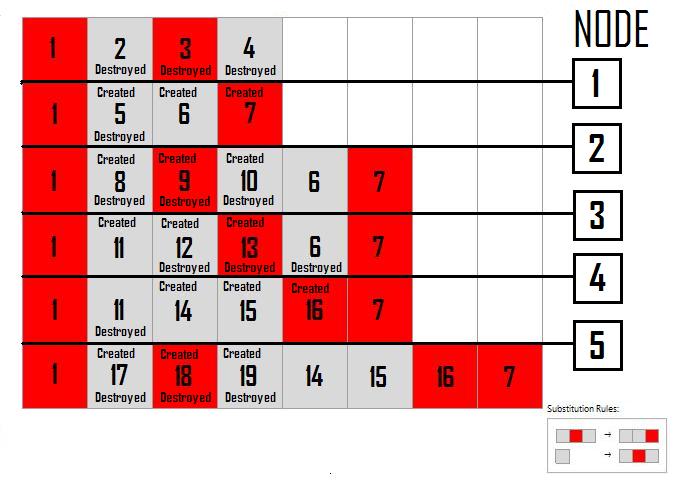


Figure A representation of the tagging system, node occurrences and the birth and death of individual cells which represent connections for five iterations applied to the SSS rule set “ABA” -> “AAB” and “A” -> “ABA” with the initial string “BABA”.

The numbers in the center of each cell represents the tags given internally. When the first match is made by the first rule, the tags two, three, and four are destroyed and replaced by the tags five, six, and seven. Nodes represent substitutions. In node one, three cells are destroyed and three cells are created. Had there been any previous nodes, a connection would have been formed with the nodes that created the tags two, three, and four with node one. Similarly, an arrow will be drawn from node one to any node that destroys any cell which was created in node one. For example, cell five was created in node one then was destroyed by node two. Therefore, in the causal network there will be a connection from node one to node two. Similarly, cell six was created in node one and destroyed in node four. In Figure 3, we can see that there is a connection from node one to node two and from node one to node four and that each connection is labeled by the cell which was created and destroyed by events. There are at most three direct connections possible from node one to any other node. Cell seven was created in node one but is not destroyed in the five iterations of the network. It is possible that a cell could be created in a relatively early node then destroyed hundreds of nodes away or the nature of the SSS may prevent the destruction of certain cells from every occurring. The causal network for the SSS in Figure 2 is shown in Figure 3 over five iterations.

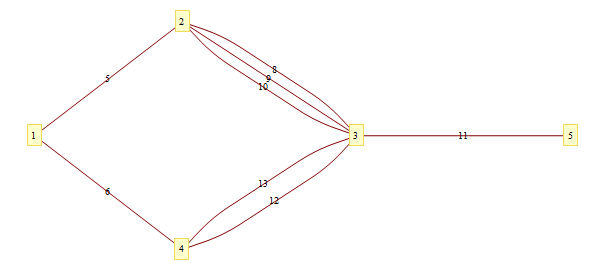


Figure The network generated from five iterations of SSS rule set “ABA” -> “AAB” and “A” -> “ABA” with the initial string “BABA” where each connection is labeled according to the tagging system for each cell created and destroyed.

Every network that we research is created in this manner. The network created is not unique to the SSS. There have been instances where two different sequential substitution systems created the same network as will be shown in a later section of this paper. Thus we do not see it as necessary to include the tagging information as it will vary depending on the specific SSS and the initial string. In addition, the causal networks have the property of progressing forward in events. The nature of causal networks guarantees that a cell can never be destroyed before it is created. Thus we always know the direction of a connection and have chosen to leave out directional arrows in our figures. Also there are cells which are never destroyed which do not affect the network even though it may affect the way the SSS appears. Our goal is to analyze the changes in these sequential substitution systems.

1. METHODS OF ANALYZING CAUSAL NETWORKS

The number of possible sets of rule sets for sequential substitution systems is infinite. We can increase the number of cell types (the size of the alphabet), the number of cells used in the rule set (basically the length of the strings), and the number of rules in the rule set. Fortunately, this set is countable. An enumeration of these rule sets was needed in order to create a systematic method of analyzing the sequential substitution systems without leaving any out. The enumeration we decided to use was created by K.E. Caviness and was detailed in his paper [4]. In this enumeration, we consider nothing ““ to be a valid match as there are substitutions that create three cells from two cells and so on. This is the enumeration that we use to systematically go through and analyze each rule set.

Table Every causal network can be produced by an infinite number of different rule sets. The rule sets that we use in the paper are given with the corresponding enumeration index along with an alternate rule set that could have been used.

|  |  |  |
| --- | --- | --- |
| Enumeration Index | SSS Rule Set | Alternate SSS Rule Set |
| 35,521,928 | “ABA” -> “AAB”, “A” -> “ABA” | “BAB”-> “BBA”, “B”-> “BAB” |
| 3 | “A”->””,”” -> “A” | “A”-> “B”, “”-> “BA” |
| 6 | “A” -> “A” | “A” -> “BA”,”” -> “A” |
| 137,679 | “AB” -> “BA”, “A” -> ””, ”” -> “AA” | “AB” -> “BA”, “A” -> ””, ”” -> “AAA” |
| 137,703 | “AB” -> “BA” , ”” -> “BA” | “BA” -> “AB” , “” -> “AB” |
| 33,157 | “AAB” -> “BA”, “” -> “A” | “AAA” -> “AAB”, ”” -> “A” |
| 135,400,946,080 | “AAA” -> “AB”, “BB” -> “BA”, “B” -> “BAA” | “BBB” -> “BA”, “AA” -> “AB”, “A” -> “ABB” |
| 530,557 | “AAB” -> “ABAA”, “” -> “A” | “AAB” -> “ABAA”, “” -> “AA” |
| 43,052,276,128 | “BAB” -> “ABA”, “A” -> “B”, “B” -> “BAA” | “ABA” -> “BAB”, “B” -> “A”, “A” -> “ABB” |
| 2,123,767 | “AAB” -> “BAAA”, “” -> “AA” | “AAB” -> “BAAA”, ”” -> “AAA” |
| 2,203,521 | “AB” -> “BAA”, “” -> “AAB” | “BA” -> “ABB”, “” -> “BBA” |

Each SSS is initiated with a sufficiently complicated string which is guaranteed to have at least a few matches. In order to save time, we created a program that would skip certain sequential substitution systems in the enumeration in order to arrive at more interesting causal networks faster. In going through the enumeration, we noticed that there were a large number of sequential substitution systems that essentially “die” out. That is, the nature of the rule sets is such that eventually the rules would no longer be applicable. Although all possible rule sets appear somewhere in the enumeration, twenty-five percent of the rule sets generated are duplicates and are immediately discarded. Others can be eliminated from consideration without actually following the evolution of the SSS and creating the associated causal network. In addition to the exact duplicates, we have found a number of rule sets that produce the same causal network due to the nature of the rule sets. Duplicate network rule sets are given for each rule set we use in this paper in Table 1.

One case occurs when a non-solo identity rule occurs. An identity rule is a rule in which a match is replaced by itself, such as “AB” -> “AB”. If the rule is ever applied, then it will be applied indefinitely. In this case, when this rule is applied, it forms a chain. Any of these instances are skipped unless it is the only rule in the rule set. If it occurs in a set of rules, then its end behavior will either be a chain or the rule will never be used. These two cases are already considered in the enumeration for either the identity rule case or the SSS that occurs without the identity rule. The cases for which it will never be used are already accounted for in the enumeration. Another instance for which we skip a rule set is when the rules themselves conflict with each other. For instance, if the rule set contains the rules “A” -> “B” and “A” -> “C”, then the second rule will never be applied since they are the same match. These are but a few ways in which we make our search for unique interesting causal networks more efficient.

1. TYPES OF CAUSAL NETWORKS AND DIMENSIONALITY

Using the enumeration provided by Caviness, we systematically went through and analyzed each causal network beginning with one and continuing through increasing indexes. Many causal networks, even large blocks of sequential cases, were skipped in the enumeration due to reasons analyzed in the previous section. There is a need to take record of the different types of causal networks that are generated from sequential substitutions systems. What we mean by dimensionality will also be explored in this section from an intuitive stand point. Everyone considers a straight horizontal line to be one-dimensional since there is only one direction where a change takes place. So in a sense, it can only grow in a single direction. So we view dimensionality as how many directions a network grows. This can be challenging as we normally view causal networks on a two-dimensional plane and only have the capability of displaying it in three dimensions. In addition, a general sense of what we mean by “dimension” may be accomplished by intuitively analyzing the different casual networks that we found. In our analysis, we found four basic types of nontrivial causal networks.

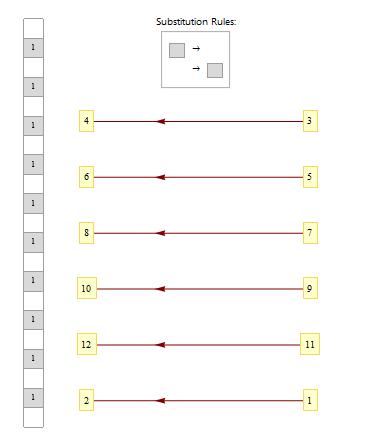


Figure First detached causal network and SSS Enumeration index 3. Derived from SSS “A” -> ””,”” -> “A” with initial string “”.

The first type of causal network that we noticed in the enumeration was a detached network shown in Figure 4. This figure shows a periodic detachment which means that a node is not connected in any way with the previous nodes. In this particular example, the detachments occur every two nodes. We define a connected network to be a network that only contains a finite number of detachments and it is well-connected if it contains no detachments. In our definition, we specify a finite number of detachments as there are some networks that may contain a single detachment but afterward seem to grow in a connected way after that detachment. It is difficult in some cases to classify a network as connected or well-connected as we do not always know the end behavior of a causal network. In Figure 4, we can see the cause of the detachments and can prove that the detachments are periodic and thus infinite. We would expect the dimensionality of this network to be zero as it does not seem to grow in any direction.

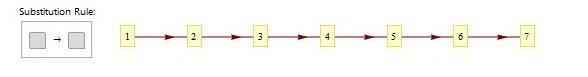


Figure First chain causal network enumeration index six. It was derived from SSS “A” -> “A” with initial string “A”.

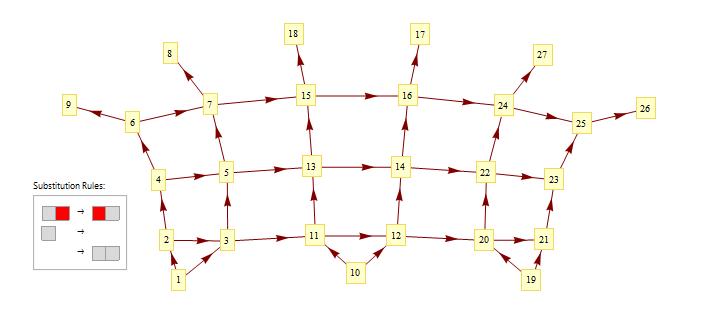
We now turn our attention to well-connected causal networks. The next type of causal network cataloged as we traverse the enumeration was the chain shown in Figure 5. This is number six in the enumeration. This causal network is infinite and grows only in one direction, here shown growing from left to right, and can be considered as one dimensional. In this example, the nodes are linked together by a single connection. We refer to this as a single-linked chain. If there are two connections, then it is double-linked chain and so on. Anything that has the basic structure of chains with nodes coming off of it is also considered a chain. Also, there are bands which are also categorized as one dimensional. This is when the network grows in one direction but still has more than one node in another perpendicular direction. An example of a one-dimensional band is shown above in Figure 6. This band initially grows in two dimensions but then attains and maintains a constant width of four and a half in one direction, continuing to grown only one dimensionally.

Figure Causal network for SSS “AB”->”BA”,”A”->””,””->”AA” with initial string “BBB”, enumeration 137679; this is known as a band.

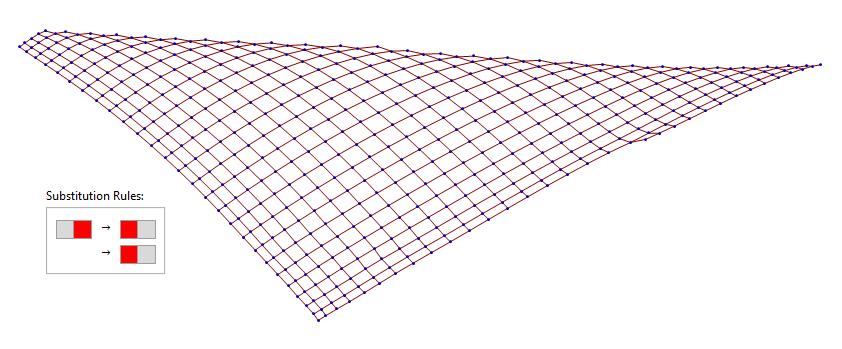
 We also come across causal networks in the enumeration that appeared to be growing in two directions as shown in **Error! Reference source not found.**. This was the first occurrence of what we call the fishnet set of causal networks. The figure shows a network that is growing in two directions. Upon analysis, we find more rows are being added to the figure as the iterations increase and that each row is increasing in number. This is exactly what we would intuitively call a two-dimensional causal network.

Figure Fishnet Causal network derived from the SSS 137703 rule set “AB”->”BA”, “”->”BA”. This describes a two-dimensional case.

We usually view two-dimensional images of these causal networks, and indeed the method of construction guarantees that a two-dimensional layout with no intersecting edges can always be found, perhaps justifying the intuitive expectation that causal networks of sequential substitution systems may be limited to dimension two or less. Is there any sense in which such a network may be considered to have higher dimensionality? Yes, if the growth occurs at a rate greater than that found in two-dimensional space. There are some causal networks that exhibit this kind of behavior. The first in the enumeration is in Figure 8. These are the causal networks that we are interested in. We are looking for complex behavior arising out of a simple set of rules. After analysis, some of the causal networks have been discovered to be two dimensional while others have displayed promise for a higher dimension. These causal networks of dimension that appear to be more than two will be treated in a future section.

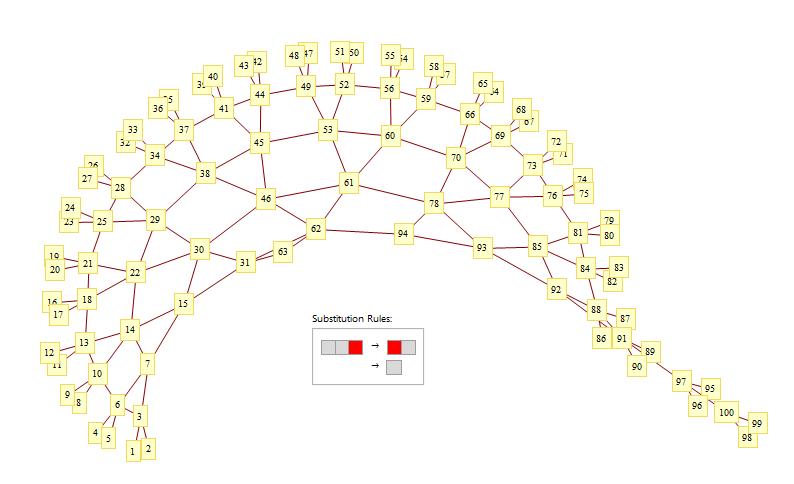


Figure Causal network derived from SSS 33181 rule set “AAB” -> “BA”, “” -> “A” with initial string “BBBBB” known as the first ruffled.

1. Method of Determining Growth of a SSS to Determine Dimensionality

Casual networks produced from sequential substitution systems contain many interesting characteristics. Several well-connected sequential substitution systemsproduce a network with myriad properties to analyze. We turned our attention to the rate of growth of a SSS which we will term dimensionality. The whole idea of a new kind of science is to find complicated systems for which physical phenomena can be modeled after, with a few relatively simple rules. The dimensionality gives an indication of the complexity of a system. Similarities in dimensionality can provide a way to equate certain sequential substitutions systemswith currently explained phenomena. Others which are unexplained may be modeled by networks that do not seem to have an ordered type of behavior which may be effective in modeling unexplained phenomena.

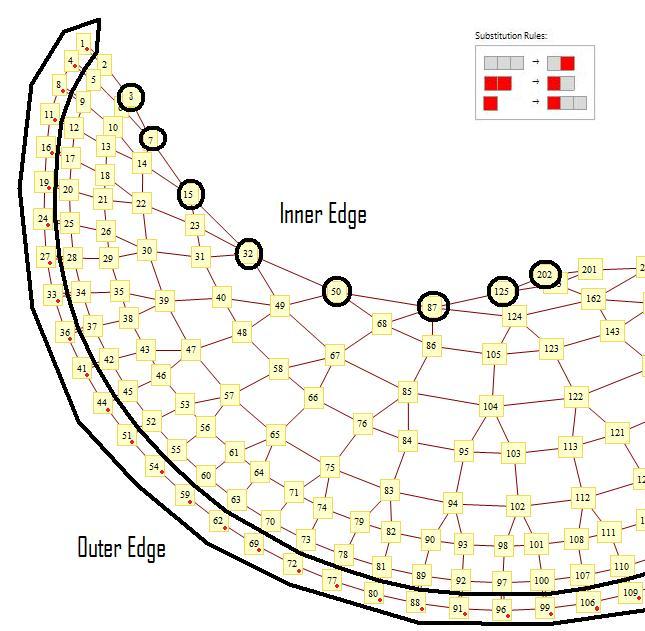


Figure Cropped view of SSS Rule Set “AAA” -> “AB”, “BB” -> “BA”, and “B” -> “BAA” which initial string “BABA” over two hundred and two iterations with emphasis on the Inner and Outer Edges.

In our first approaches, we focused on sequential substitution systems that were fairly complicated such as in the example in **Error! Reference source not found.**. It was thought that it would be profitable to analyze the inner and outer edges as shown in the **Error! Reference source not found.**. Analyzing the way that each SSS grows is important in this process. It is not immediately apparent as to which nodes will be part of the permanent inner edge. As one follows the evolution of these inner edge nodes, it is clear that node two hundred and one is not a part of the inner edge although this is not apparent unless you view the way the SSS has been growing. The goal was to find some sort of pattern. The outer and inner edges in this specific example do produce some complicated formulas but only after the first few nodes in the inner edge and outer edge sequences are dropped. There were some cases where the inner edge and outer edge appeared to be predictable. In practice, however, it was very cumbersome to employ this method. We have no algorithm that would automatically produce the needed information of the node numbers along these edges. Every inner edge and outer edge sequence would have to be determined manually. As a result, there was a higher probability of human error and the time needed to analyze each SSS was wasteful. In addition, not every SSS had an easily distinguishable inner or outer edge. A more elementary and general way of classifying a sequential substitution system was needed to determine or approximate dimensionality.

We developed a more productive method for measuring the dimensionality of a network. We define distance to the least number of connections from node one of the network to the node in question. We assert that the dimensionality of any given sequential substitution system can be determined by analyzing a function f(d) which is defined to be the number nodes that are a given distanced ,d , from the origin which is node one. For example, if there are three nodes that are distance five away from the origin, then f(5)=3. Not every SSS has a clearly definable inner or outer edge to analyze. The distance-from-origin concept provides a simpler and more automatable way to determine the dimensionality of any causal network that is well-connected (as defined in the catalog section above). Each distance can be considered a radius. Ifone is considering two dimensions, the definition of the number of nodes or length a given distance from the origin is the definition of circumference. For three dimensions it is considered the surface area and so on. Notice that the degree of the polynomial of any surface function is one less than the dimension. By analyzing the surface function, we can determine the dimension of a given network. This method is thus more universal.

A few examples may serve to illustrate this method. First, dimensionality of growth of a given SSS makes sense only in the context of a well-connected SSS. That is, a node cannot be periodically detached from the origin as in Figure 4. In this case, our program denotes the detached node’s distance from the origin as ∞. If we have a case in which the SSS is not periodically detached but simply appears to have a finite number of detachments, generating the SSS from a different initial string may very well fix the problem. We have found that while any SSS generally behaves in the same way from a sufficiently complicated initial string, changing this initial string may reproduce the same network in a different stage that may make it easier to quantify dimensionality.

The string example in Figure 5 is just a chain. We would intuitively expect this to be a one-dimensional network as it appears to only grow in a single direction. In general for chains, we get that f(d)=c where c is just some constant. This means that the number of nodes a distance d from the origin is always the same. As expected, we determined that fishnet example in **Error! Reference source not found.** to be two dimensional. As complexity increases, we would expect the dimensionality of the SSS to increase. We would also expect the dimensionality of **Error! Reference source not found.** to exhibit a higher dimension than two. More analysis tools are needed to effectively analyze the dimensionality of any SSS having a higher approximate dimensionality.

The actual procedure of determining the number of nodes that are distance d away requires a more involved approach. It is important to have generated enough nodes of the SSS to determine a point for which we adequately trust our data. For example, there may be some SSS that appears to have a given behavior at the beginning only to find that it changes at some point. For example, Figure displays a portion of the pattern that occurs for the first seventy-six nodes of a particular SSS. This portion would lead you to believe that it continues on forever as a modified chain. However, Figure shows five hundred iterations which reveal a more complicated structure. This example shows why each SSS should be analyzed individually. In examples such as this, it is more efficient to begin the SSS at the string for which the new pattern emerges. In this case we restart the SSS where the chain pattern ends and the ruffle pattern starts at node one hundred and sixteen. Analyzing this SSS from the new starting point, a clear surface formula appears to be . If this equation was true as d approaches infinity, then we could say that the SSS grows exponentially and has an infinite dimension. However, this pattern is broken after 3000 iterations. This happened because a few nodes which were created at the beginning were destroyed after thousands of iterations. This can be predicted by viewing the pattern of the SSS. Information on all possible connections for a particular node is helpful in trusting the reliability of the data. This is why an algorithm to analyze these is difficult as it requires a detailed analysis of each particular SSS.

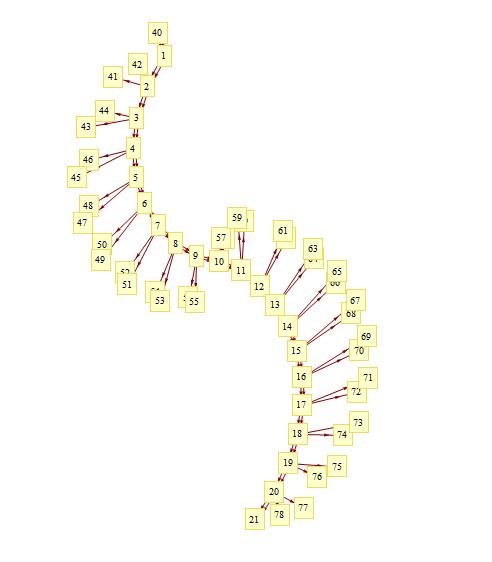


Figure 10 SSS rules set “AAB” -> “ABAA” and “” -> “A” with initial string “BBABABABABABABAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBA”. A portion of this SSS, including seventy eight iterations

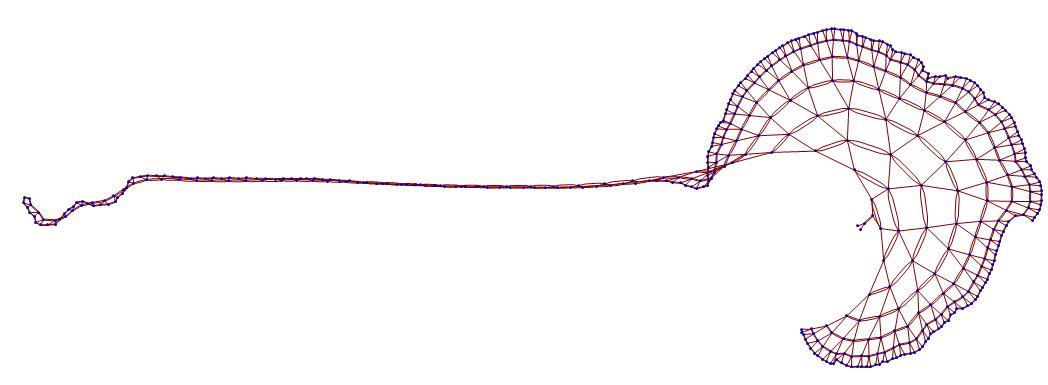


Figure 11 SSS rules set “AAB” -> “ABAA” and “”-> “A” with initial string “BBABABABABABABAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBA”. A portion of this SSS, including five hundred iterations

In some cases, we see a much more solid pattern and trust the data due to the nature of the growth and/or rule. Using these methods, we have found cases in which the dimensionality of an SSS is two. Dimensions of higher than two and sequential substitution systems with a fractal dimensions and exponential growth are of particular interest. Unfortunately, no perfect examples have been found as of yet. However, there are several sequential substitution systems that do not produce an exact formula but still seem to grow on the order of a certain dimension. Although these sequential substitution systemsmay be harder to express, they encourage the possibility of finding an SSS that is easily described and grows at the exact dimension desired.

1. APPROXIMATELY HIGHER THAN TWO DIMENSIONS

Our search for more complex behavior has produced a few examples which appear to have an approximate dimension of more than two. Each of the cases found are unique and through the process of analyzing them, we will show what we mean when we say that we trust data. And introduce a few tools to determine the dimensionality of a given causal network. We will start with examples that are less complicated and progress to more sophisticated ones.

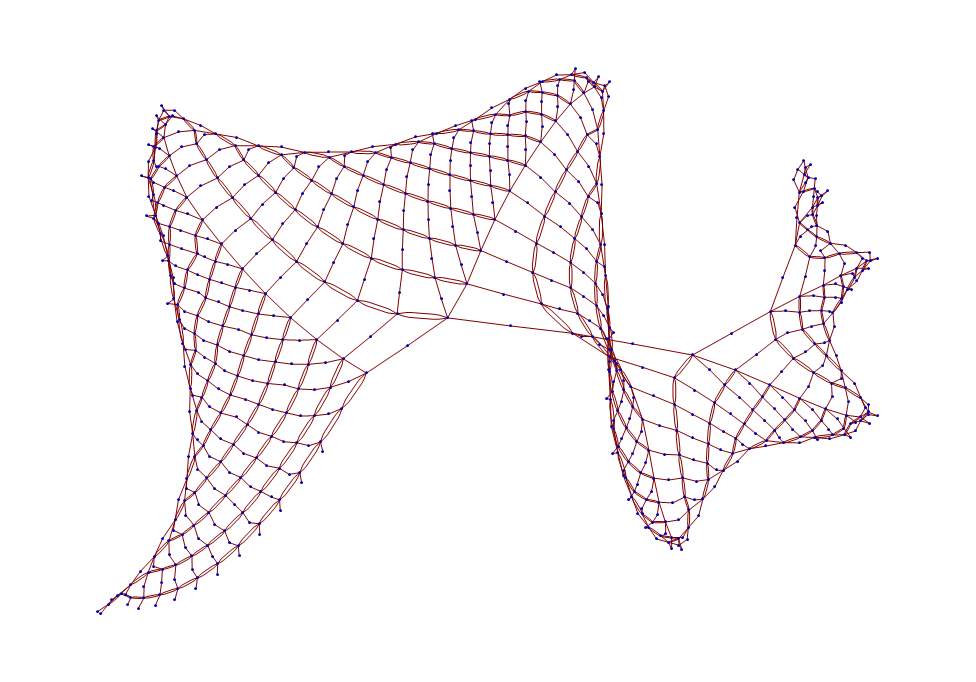
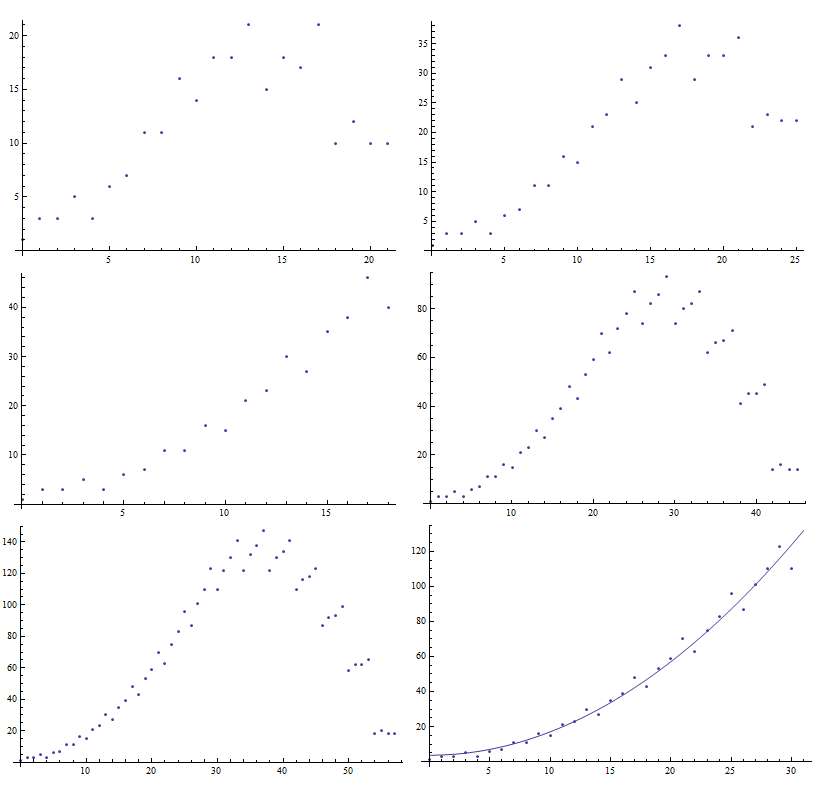


Figure 12 Causal network derived from SSS rule set "BAB" -> "ABA", "A" -> "B", and "B" -> "BAA" with initial string "BAB" over seven hundred and fifty iterations.

We begin by viewing the causal network in Figure which is three dimensional. Since we are observing these representations on a two-dimensional plane, it can be difficult to ascertain the actual dimensionality of the network. However, the unique structure of the network is highly suspect of displaying three-dimensional behaviors. We analyze the dimensionality of the network by generating the surface function. In this process, the evolution of the causal network is important. We can make an educated guess as to how the network will continue to grow by looking at the progression. This can be visualized in Figure . The plot on the top left represents this progression after two hundred and fifty iterations. So far, no pattern seems to be emerging except for a general trend for the points to increase as one traverses the x axis. On the top right, a pattern becomes a bit more distinct. As we traverse through the iterations, we see that some points toward the origin seem to have stabilized while there



4000 Iterationserations

4000 Iterationserations

2000 Iterationserations

1000 Iterationserations

500 Iterationserations

250 Iterations

Figure 13 Progression of surface data for causal network produced by SSS rule set "BAB" -> "ABA", "A" -> "B", and "B" -> "BAA" with initial string "BAB". The number of iteration is doubled every consecutive figure. (Top left) 250 iterations, (Top right) 500 iterations, (Middle left) 1000 iterations, (Middle right) 2000 iterations, (Bottom Left) 4000 iterations, (Bottom right) 4000 iterations with trusted data fitted to the parabola . Note that the scaling in different on both the x and y axis per figure in the grid.

are some trailing terms which have not settled yet. The expectation is that these trailing terms will stabilize as the number of iterations increase. After four thousand iterations, it appeared that all of the points distance thirty away from the origin or less stabilized. We then took this data and tried to fit it to a function using *Mathematica*’s tools. The best fit produced the parabola .It is also possible to find the best fit of a third degree polynomial; however, the coefficient for the highest term was small enough to be negligible. We rule out higher dimension surface equations by analyzing the highest coefficients. Since this the surface function is of degree two for this particular causal network, it can be concluded that the causal network is approximately three dimensional. We also show another three-dimensional causal network displayed in **Error! Reference source not found.** and the best fit with the surface data in Figure .

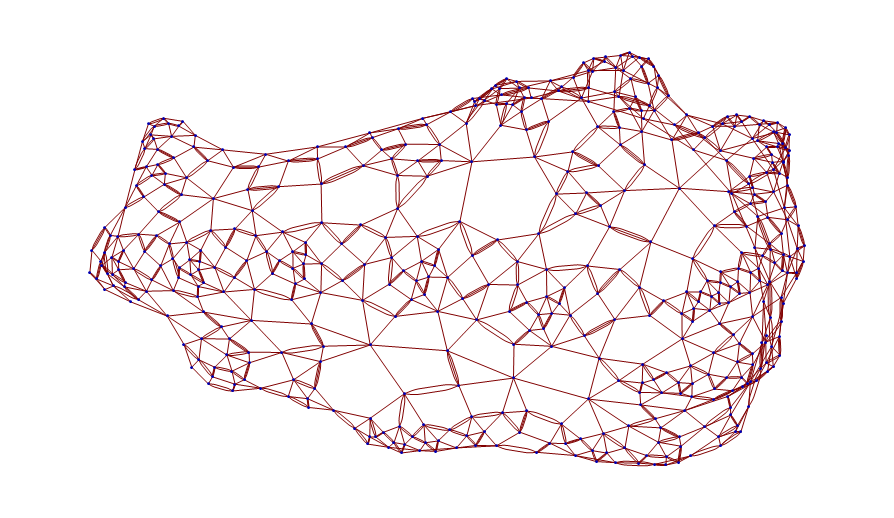


Figure 14 Casual network produced from the SSS rule set “BAA” -> “ABBB”, “BABB” -> “ABA”, “A” -> “AABB” with initial string “BAB” over five hundred iterations.

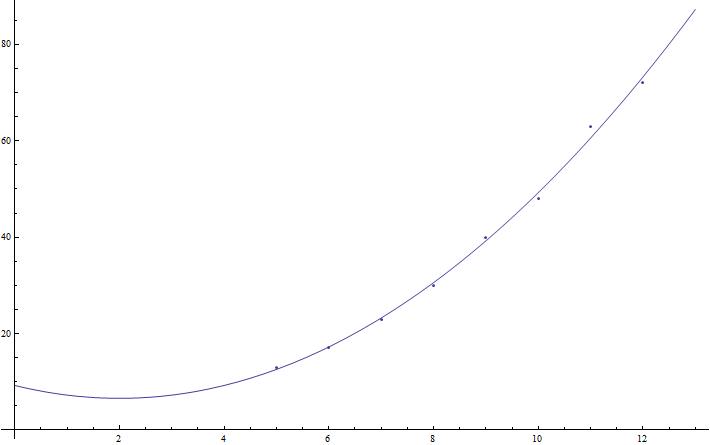


Figure 15 The surface data of the casual network produced from the SSS rule set “BAA” -> “ABBB”, “BABB” -> “ABA”, “A” -> “AABB” with initial string “BAB” over five hundred iterations plotted with . The data was fitted after ignoring the first four terms and securing reliable data.

Our next objective was to determine whether causal networks existed that have an approximate dimension higher than three. While we have not found any causal network with a finite dimension higher than three, we have found two approximately exponential causal networks whose dimension is always increasing. The first is shown in Figure along with its best fit after dropping the first four terms in Figure . The challenge with exponential causal networks is determining whether it is actually exponential or whether it is just a much higher dimension polynomial. This problem is to be expected since an exponential can be written as an infinite series of polynomials [5]. One solution we have found is to plot the surface data with the y axis scaled logarithmically. Linear behavior in the log plot indicates exponential behavior. This is shown in Figure after the first four terms.

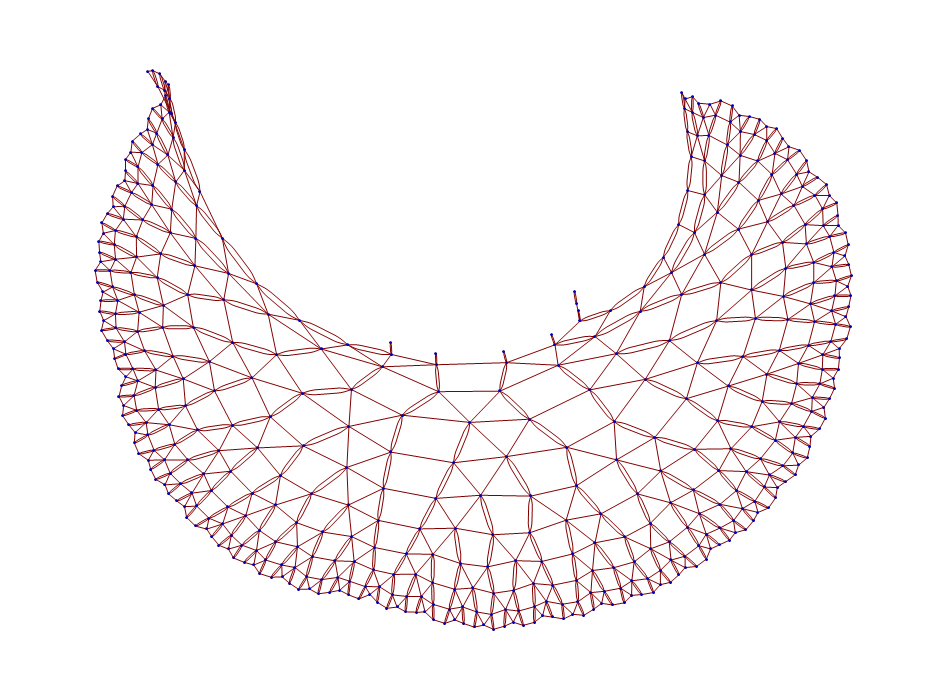


Figure 16 The causal network produced by the SSS rule set “AAB”-> “BAAA” and “”-> “AA” produced from the string “BBBBBABBABABABAAAAAAAAAAAAAAAAAAAAA” over 500 iterations.

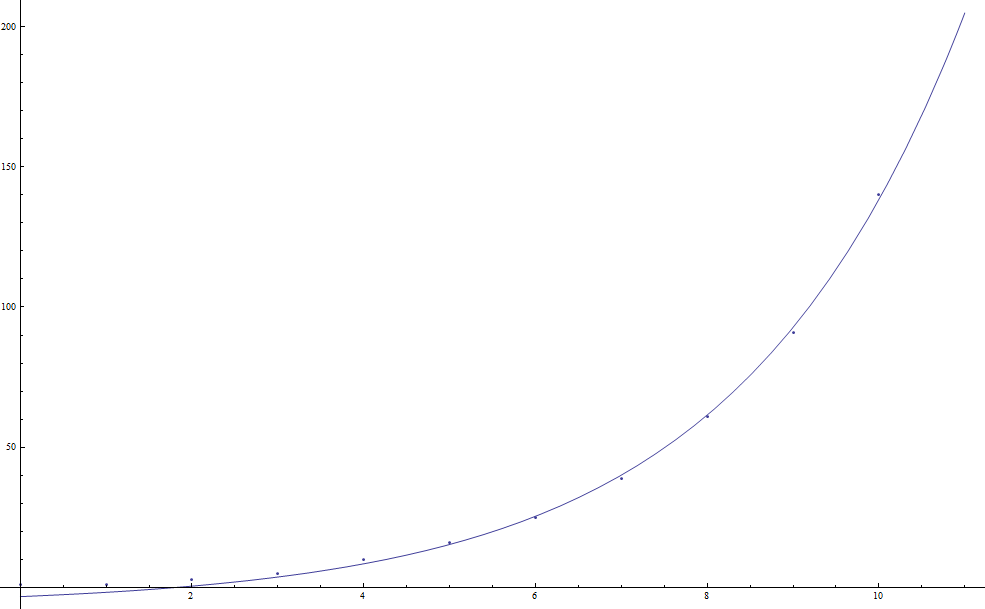


Figure 17 Plot of the surface data for the causal network produced from the SSS rule set “AAB”-> “BAAA” and “” -> “AA” produced from the string “BBBBBABBABABABAAAAAAAAAAAAAAAAAAAAA” fitted to using data after the first four terms.

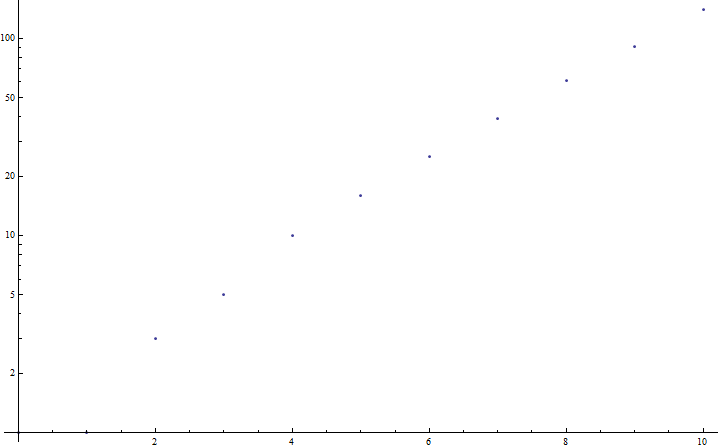


Figure 18 Plot of the surface data for the causal network produced from the SSS rule set “AAB” -> “BAAA” and “”-> “AA” produced from the string “BBBBBABBABABABAAAAAAAAAAAAAAAAAAAAA” using a logarithmic scale for the y axis.

Our second of the two exponential causal networks is shown in Figure along with the best fit plot in Figure . The exponential networks that we have found have the characteristic of branching out. It is very easy to mistake these networks as two dimensional since there is not much twisting or turning in the figures.

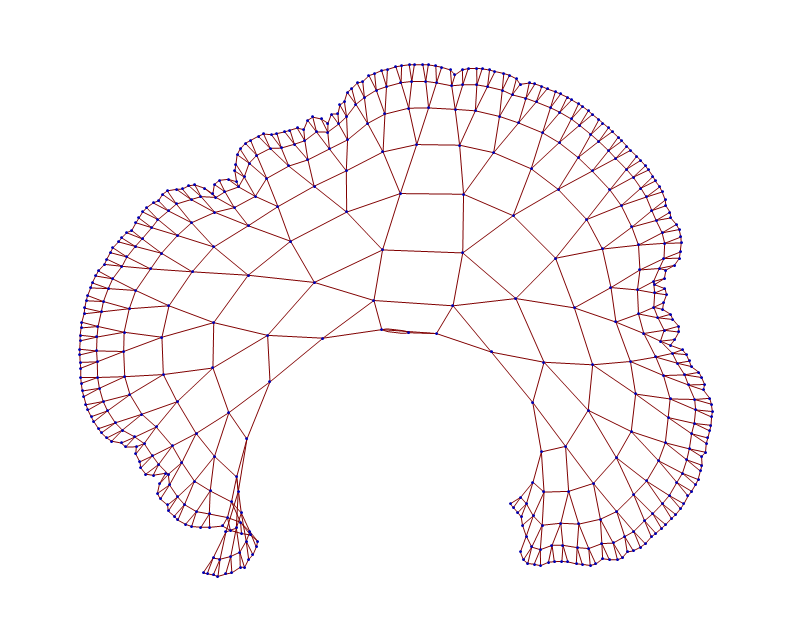


Figure 19 Causal network produced from SSS rule set “AB” -> “BAA”,”” -> “AAB” from with initial string “BBBBBBB” over 500 iterations.

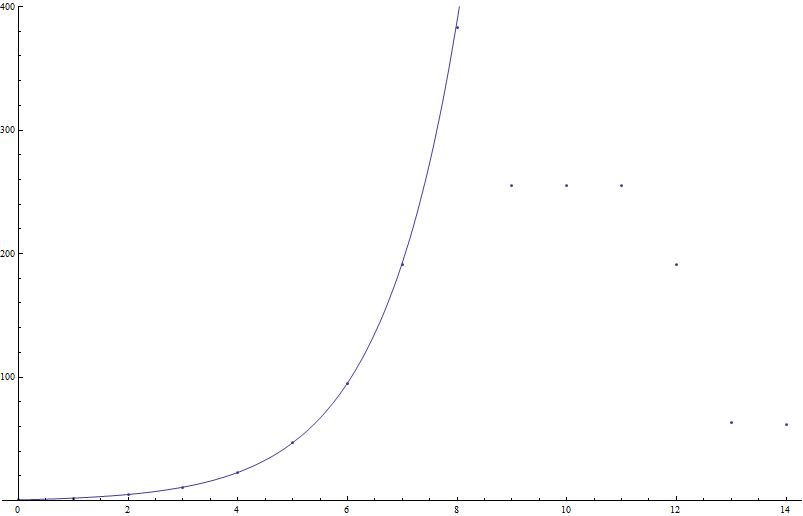


Figure 20 Plot of the best fit equation with th surface data of the causal network produced from SSS rule set “AB”->”BAA”,””->”AAB from with initial string “BBBBBBB”.

1. CONCLUSIONS

Causal networks derived from sequential substitution systems have many characteristics to explore. Knowledge of how the causal network is created from a sequential substitution system, an enumeration of all possible combinations of rule sets, and methods to reduce the number of duplicate sequential substitution systems analyzed are all essential in analyzing and determining the dimensionality of a causal network. The dimensionality is determined by creating a function which outputs the number of nodes a certain distance away with the input being the given distance. These formulas may be determined by analyzing each SSS individually to determine the number of iterations needed to create a set of trusted data. Then finally we analyze these functions and determine the dimension of them using normal mathematical procedure of analyzing the highest exponent or pronouncing infinite by finding an exponential. We have been unable to find a causal network generated from an SSS whose dimension exceeds two. However, we have several examples whose dimension is approximately exponential or greater than two. There is still much research to be done in modifying the program, developing new tools to continue analyzing causal networks in the order of the enumeration, and equating causal network to physical phenomena.

Part of our goal is to modify the program in order to skip less interesting causal networks in order to identify the unusual causal networks faster. Methods of how to skip everything that has a dimension of two or less is in progress. This can be hard as not every SSS grows in exactly the same way to produce the simple behavior. Our efforts are focused on analyzing the sequential substitution systems in order to discard uninteresting networks. In addition, there is work to be done in creating an efficient way of skipping more duplicate causal networks without slowing the program down. We want to create more tools that are geared toward analyzing the dimensionality of the causal networks. This may be an ongoing process as more ideas of what would make it easier to analyze the networks.

One modification to the program that we are working on is to generate the information of how many possible connections a given node has as compared with how many already exist. This will tell us how prone the network is to spontaneously changing its pattern. The surface formula is reliable unless a node is randomly generated near the origin. This is another way of measuring how much we trust our data. In addition, we are always searching for more methods of analyzing sophisticated causal networks as we continue with the enumeration. We have encountered a very small number of the different sequential substitution systems that are possible. We will also continue traversing through the enumeration in search of complex causal networks in order to analyze them. The whole essence of this project is to discover causal networks that can be used to explain phenomena in the physical world. We will continue to catalog interesting causal networks in order that someday someone may use them. Actual analyzing of the networks which resemble the real world will forever be the study of anyone involved in a new kind of science.

1. ACKNOWLEDGEMENTS

The author is very grateful to Ken Caviness for the meaningful guidance and instruction that he has provided. We would also like to acknowledge Breanna Bowden-Green and Kelsey Dobbs for selection and preliminary analysis of the data. The author would like to thank Dr. Chris Hansen of Southern Adventist University for supervising the compilation of this paper for the Scientific Writing and Presentation course. Finally, we would like to acknowledge the generous support of Southern Adventist University in providing funding for this project.

REFERENCES

1. S. Wolfram, *A New Kind of Science* (Wolfram Media, Champaign, IL, 2002) pp. 1-22, 82-94, 433-503.
2. L. Gray, “A Mathematician Looks at Wolfram’s New Kind of Science”, Notices of the American Mathematical Society **50** (2), 200-211 (2003).
3. Wolfram Research Inc., 100 Trade Center Dr., Champaign, IL 61820-7237,

http://www.wolframalpha.com.

1. K.E. Caviness, “Indexing Strings and Rulesets: An Exploration Leading to an Enumeration”, The Mathematica Journal **13** (2011).
2. J. Stewart, *Calculus: Concepts & Contexts, Third Edition* (Bob Pirtle, 2006) pp 610.