ABSTRACT

Title of Thesis: SECOND WAVE MECHANICS

Anthony Fabbri, Master of Science in Reliability Engineering, 2024

Thesis Directed By: Professor Jeffrey W. Herrmann, Dept. of Mechanical Engineering

The COVID-19 pandemic experienced very well-documented "waves" of the virus's progression, which can be analyzed to predict future wave behavior. This thesis describes a data analysis algorithm for analyzing pandemic behavior and other, similar problems. This involves splitting the linear and sinusoidal elements of a pandemic in order to predict the behavior of future "waves" of infection from previous "waves" of infection, creating a very long-term prediction of a pandemic. Common wave shape patterns can also be identified, to predict the pattern of mutations that have recently occurred, but have not become popularly known as yet, to predict the remaining future outcome of the wave. By only considering the patterns in the data that could possibly have acted in tandem to generate the observed results, many false patterns can be eliminated, and, therefore, hidden variables can be estimated to a very high degree of probability. Similar mathematical relationships can reveal hidden variables in other underlying differential equations.

SECOND WAVE MECHANICS

by

Anthony Fabbri

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2024

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 PURPOSE

The problem of predicting the progress of epidemics has been studied, however much of the base numerical data underlying the actual real-world numbers can be missing, or contains junk data fabricated by intermediaries. This provides challenges that are not present in more theoretical mathematics. This thesis documents a way of overcoming this problem by measuring not only the data, but connections and correlations the data represents, and therefore maximizing the amount of useful actionable information that can be derived from the data, even in ultra-sparse or very noisy data sets.

A series of data points that begins {1, 2, 3, 4, 5, 6, 7} does not gain much additional explanatory power by also then including data points {8, 9, 10}. The pattern recognition has been achieved, and increasing amounts of data do not necessarily lead to similarly increasing amounts of conceptual understanding of that data.

Pandemics, and other mathematical structures, have similar hidden variables which influence the underlying function, but are not directly obtainable from that function. Previous attempts to derive such 'paradigm variables,' or meta-variables, rely on numerical methods, statistical techniques, machine learning, and/ or Bayesian inference to estimate the values, where such values cannot be exactly calculated analytically. These methods frequently rely on intense computational calculation. However, such 'brute force' methods are wasteful, as they do not take into account that only some explanations for the data have any feasible likelihood of providing contributory power to the model. By first analyzing the data for such bottlenecks, the solution space can be greatly reduced beforehand, putting less burden on the later computational methods, especially by separating useful from non-useful information, before going through with the effort of deriving the non-useful information, which is especially useful for such noisy data sources.

Historically, the measurement of systems of flows has assumed some type of conservation law, implying a source or sink of some finite value, that can be depleted, consumed, or refilled. However, some flows, such as pandemics, do not obey a conservation law; a second person getting sick does not somehow cause the first person to become less sick, so the flow of the pandemic is not conserved. This requires a different flow methodology than the classical differential equations involving water pipe flows, electrical current flows, traffic flow, and other systems that do obey a conservation law. While people move about, the general “contagiousness” of a population can vary wildly, for example during the winter, and cannot be tracked by calculating in-minus-out, as a conserved flow would.

This expands the solution set drastically. For example, an infinite number of sine waves of different amplitudes can be described that all have the same wavelength. Sine waves of that wavelength may be described by an amplitude of 5 or an amplitude of 1,000,000; the size of the amplitude of that source does not in this way necessarily aid in the measurement of that wavelength. If the size of multiple synchronous wave sources are completely unknown, such wavelengths would be difficult, if not impossible, to solve for analytically, using classical methods. New methods that economize on the obtainable information are necessary.

1.2 OVERVIEW

This thesis describes a brief introduction and summary background in the relevant points of virology and virus data retrieval. And how these new methods can be used to solve such a problem.

This work describes three methodologies operating in tandem to solve certain types of problems that have historically been difficult to solve. While prior works touch on portions of the three methods, by combining all three in tandem, each method can “gold mine” nuggets useful information units, in a vast field of nonuseful information. The different methodologies tend to find completely different units of information first, so by uniting all three methodologies simultaneously, most of the “essence” of the pandemic’s underlying function can be ascertained more elegantly. The methods are as follows: 1. the similar waves method, 2. The similar flow method, and 3. the reverse chaos method, each of which will be discussed in turn separately, and then unified into one algorithm. Further, this describes previous attempts to solve similar problems mathematically, including the predator-prey model and previous flow methods, and what the current algorithm takes from these methods and where they diverge.

There is an emphasis of the wave behavior of the COVID-19 virus as a case study, which has a good conceptual model, as well as a great deal of numerical data documenting its spread; but these methodologies can also be used for other types of differential equations that behave in similar ways, such as physics simulations that include real-world noise, the propagation of pricing and consumer behavior data, tourism flow, and other nonconservative flows.

The COVID-19 virus's behavior occurs with a wave-like pattern. The number of infected people rises to some peak, and then declines to some trough, before rising in a second peak, and then declining in a second trough, etc. An instance of the virus stopping the decline, rising to an arbitrary peak, and then sliding down to the arbitrary trough as commonly termed a "second wave" of the virus, similar to parlance for ocean waves. As such, other terms used in waves can also be used, for example each individual wave instance has an amplitude, which is the height of the peak for that one wave instance. And a given wave instance has a wavelength, which is the time it takes to go from trough, to the peak, and then back to through again. A single wave instance of one wavelength from trough to peak amplitude, back to trough will be called a single "wave" of the virus. A pandemic or outbreak, then, would consist of many waves. Note that the individual waves/wave instances will each have slightly different amplitudes and lengths of time, even if they occur in the same area.

1.3 USE OF COVID-19 VIRUS DATA

The COVID-19 virus's spread is exceptionally well-documented in several countries, to a very high degree of precision. The CDC in the United States, for example, has publicly available data sets for populations and sub-populations of various kinds, which provided a robust publicly-accessible data source. The currently used data source is the CDC report of weekly COVID-19 infections by state or region. The regions the CDC tracked include the 50 States of the US, The District of Columbia as a 51st "state", New York City as its own region, and eight additional territories and principalities that are similarly tracked, forming a time-series data for all 60 regions that are so tracked by the CDC. This tracks the pandemic from the 1st week the pandemic was declared until the 173rd week when the pandemic was officially declared resolved.

Using this data, the virus's propagation can be analyzed to predict future wave behavior that would be in line with previous waves' behavior. Future waves statistically have features in common with past waves. In addition, sudden deviational wave shape shifts, that were later found to be the result of mutations, can be found embedded in real-time data to predict mutations that must have recently occurred, but have not yet become popularly known. This allows unearthing of hidden information, that can be used to predict the different future outcome of the wave as a result of this new information, even if the hidden information itself can not be known.

The basic idea of the analysis is being able to tell what hidden variables are present, and whether or not the variables are changing over time. If the shape function remains exactly the same, but the intensity factor changes dramatically after a known mutation, this provides further confidence in several derivative conclusions: For example, when the shape itself of the function is not changing as greatly, but the intensity factor is changing greatly, one can determine a value of the intensity factor that matches very well to the old data, and a higher intensity factor that matches very well to the new data; this allows information that is derived from the previous wave features to be applied to predictions of the next wave without having to explicitly calculate them. This can be seen as analogous to similar triangles in geometry; by maintaining such a two-system correlation, this allows each cross-correlation to inform, but not contaminate the results of, the other. Additionally, good-quality leading indicators, for example of a mutation, can be used to predict with good accuracy indicators of a new infection.

1.4 OUTLINE

Chapter 2 describes previous work in virology and in epidemiology, as well as the conceptual basis for later improvements. Chapter 3 discusses the logic and algorithms of a new method that can predict sudden changes in outbreaks with far less time-series data than previous methods, allowing problems to be discovered much earlier in the chain. Chapter 4 discusses actual results with current COVID-19 data at the CDC. Chapter 5 discusses uses of the method with other types of ultra-sparse problems such as physics simulations, or adversarial math problems where noise has been introduced, such as pricing data; for example, gas prices are “contagious” to other, nearby gas stations. And customer service improves when a nearby competitor drastically increases its customer service.

2. Prior work

2.1 VIROLOGY BACKGROUND

Under the prevailing Viral Load theory of virus propagation [1], a given infection requires the presence of some approximate number of virite particles to achieve symptomatic infection. If a few virite particles slip into the body, the immune system has a high probability of destroying the virites before too much damage is done. However, if a “critical mass” beachhead is established by the virus, it is much more difficult for the body to prepare a response. Those with compromised immune systems and/or elderly persons thus have a lower viral load required, whereas young children can have a much higher viral load necessary, to achieve symptomatic infection.

The model of how much virus a person sheds to those around them is a function of the amount of virus present in the person, as well as the level of contact between that person has with other people. Those with a high number of virite particles tend to shed more virus, and those who interact frequently with another person tend to have a higher number of virus-shedding events with that other individual. This provides for a differential equation between the two individuals, or any other groups that the first individual happens to meet. However, the number of virite particles a person possesses at a particular time is not directly measurable. The fact that the person contains ANY virite particles only becomes revealed when the person starts showing symptoms and gets tested for the virus, having already exceeded the viral load threshold, when it is far too late to do anything about it.

In addition, the viral load ceiling to achieve symptomatic infection is, itself, always changing; as the person is further exposed to non-symptomatic numbers of virite particles. As the person experiences symptomless infection, the ceiling goes up slightly, depending upon the immune system of that one individual, the viral load that person currently possesses, and other factors outside of the experimenters' control. As the ceiling goes up, the "average" apparent breakthrough infection rate in society proportionally goes down, as fewer and fewer people now have a near-100% viral load rate, due to the increasing ceiling required to achieve symptomatic infection. As the threshold is now much higher, their current viral load is no longer sufficient to achieve symptomatic infection, with the now-higher threshold. This provides a dampening effect on the visibility of the virus, even though the number of virites in circulation has not diminished. The result of this effect is a second-order differential equation; as the number of virites goes up, the number of virites needed to achieve visibility also goes up, so the number of apparent infections goes down, causing people to interact more, causing people to spread an even higher number of virite particles, which finally exceeds the new, higher threshold, causing the number of apparent infections to go up again. This explains the "wave" nature of the infection.

Additionally, the virus can suddenly, and unexpectedly, mutate, suddenly crashing the viral load threshold to a much, much lower value. This causes a large number of breakthrough cases to simultaneously suddenly become visibly emergent. Or, similarly, the virus can drastically increase the replication rate, leading to a similar outcome from the other direction, by more quickly achieving the current ceiling.

The time between outbreaks can tell us the "depth" of each wave in a locality, which tells us hidden variables, such as the number of virite particles a given sub-population must necessarily have as a fraction of threshold, regardless of what that threshold's value actually is; we can determine how much of the wavelength is being consumed per day. This predicts the time to next outbreak, and its severity, giving us access to previously invisible information. This also informs how much the population is shedding, which can give a good idea of what facilities should be open or closed in key places in the wave. Additionally, the long-term trends of the virus's waves tells us when the virus suddenly deviates, for example, during mutation. This provides for early notification of the mutation, before it becomes readily visible, and allows steps to be taken to mitigate and observe the new strain before the new strain becomes even more problematic, leading to advance notice of problems.

2.2 FLOW METHOD (PRECURSOR TO SIMILAR FLOW METHOD)

The flow method works for measuring non-conserved flows moving from one location to another to create a flow diagram of the flow of contagion from one node to another. There are numerous ways of measuring human contact and travel flows in this way, however, Schlundt et al. [7], teaches that zoonotic origins are the prime spreader in foodborne illnesses, which means that current methods that primarily track human contagion will not provide much insight when the primary contagion process is caused by animal-to-animal transmission, outside of observation. New methodologies are required to offset this.

Nohara et al. [2] teaches using an effective distance approach combined with time-to-onset for each graph node. This allows for correlation even among 'invisible' nodes, such as would be encountered for a zoonotic origin or contagion, as is described by Schlundt et al. [7]

Brockmann et al. [3] shows that a distance/correlation constant can be used to simplify the n! graph edges, overcoming overcomplexity and allowing for a pulling of the actionable information out of the graph as a series of outward flow functions. Nodes with a high correlation to each other, but are separated in time, are likely logically nearby on the flow graph, but with several invisible nodes in between, taking up time in propagation delay.

Jacobs et al. [4] suggests that a node’s flux can be exceptionally well documented by a number of biggest contributors. If there are n neighbor nodes, branches with flux much less than 1/n of the total flux are usually not applicable. The analogy is used that in electric circuits, “the smallest resistor is the biggest contributor to the current” [4]. This lowers the solution set significantly. In real life processes, the flow tends to be mostly determined by a few biggest contributors, and the remaining n! nodes can safely be ignored.

Pastore et al. [5] shows that a least spanning tree can be used to parse out a path to the origin of contamination from a nodal network. This can be combined with the above references to form a back-in-time path, and extrapolate a forward-in-time path from a given snapshot state of the nodes.

Pinto et al. [6] describes accounting for invisible nodes along a flow trajectory. It does this by estimating a source, by sorting by propagation time, and then filling in invisible nodes, as needed.

The disadvantage of the previous methods is the assumption of burst behavior on the part of the virus; that the progression of the virus is monotonically increasing, and all the nodes gradually go from 0% virus to 100% virus over some period of time. However, long-term trends don't support this assumption. Even factoring in mutations, the relative prevalence of different strains go up and down several times over the course of several years, accounting for things such as population movement, local holidays, and seasonal variations. The downforce of virus must be accounted for along with upwards force in virus propagation.

2.3 PREDATOR-PREY MODEL (PRECURSOR TO SIMILAR WAVES)

The predator-prey model is a classic two-population model in differential equations: predator and prey. [9] A population of predators (Y) can be supported by a given population of prey (X). If the prey becomes more numerous, it can support more predators. If the population of prey becomes less numerous, it can now support a lower population of predators, putting a downward pressure on the predator population; for some constants C_1, C_2, time t:

$$Y'(t) = C1\, Y(t) + C2\, X(t) \qquad 1$$

However, the new, lower population of predators catches less of the prey, causing more prey to escape and reproduce, causing the population of prey to rebound in the absence of predators; for some constants C_3, C_4, time t:

$$X'(t) = C3\, X(t) - C4\, Y(t) \qquad 2$$

The new, higher population of prey can support a higher population of predators, leading the predators to also then rebound at a later time t, which, in turn, causes the population of prey to crash again. This combination forms a second-order differential equation. The population of prey is continuously rising and falling.

To determine the exact equation of the populations of predator and prey, a number of snapshots of the estimated predator and prey population are captured, and the values for C_1, C_2, X(t=0) and Y(t=0) can be solved for, using differential equations. However, this methodology requires knowing the populations of X and Y under consideration, and also doesn't account for predators or prey moving on to a different location not under observation. This type of methodology quickly breaks down when the number of unknowns drastically outnumbers the amount of knowns. This makes it difficult to "mine" sparse data, or data that may contain erroneous entries.

2.4 CURRENT CDC METHOD

The CDC currently analyzes long-term virus propagation using R_0 and Rt. [10] Rt can also be notated as R(t) as a function of t. The value R_0 represents, generally speaking, how many new people a given patient should infect. Rt is used similarly, but is a better estimate because it is time-varying, and therefore generally represents whether the population of infected is increasing or decreasing. The formula for the number infected at any given time is therefore a straightforward exponential of R_0 (or Rt) raised to the power of t, representing time; the Number infected N at time t, is therefore:

3

$$N(t) = R_0^{\,t}$$

While Rt makes for a really good paradigm variable, it necessarily limits the amount of information it can represent. The CDC also acknowledges that it is a trailing indicator, in that this information only becomes evident after a given process has already completed. Whereas a more helpful function, such as the one being proposed, represents a leading indicator, which enables better predictive methodology of what is going to happen in the immediate future. It is an improvement to acknowledge not only that something has happened, but as a result of the new difference, exactly what new changes will occur, as a consequence.

During the COVID-19 pandemic, many other models were experimented on to attempt to determine a long-range planning model. However, the further out from the training data the models went, the more wildly divergent they became. "some early models predicted millions of deaths during the first few months in the United States … while others expected an end to the pandemic by May 2020" (Rahmandad et al.) [18]. Further, Cramer et al. [16] describe a "naive baseline" with which to compare models. In this study, only a single, complex ensemble model performed better than the naive baseline in all such cases. Cramer additionally discusses other problems the further out from the training data a prediction attempts to infer, leaving most of the models incapable of predicting a great degree of time in one go, instead continuously updating with recent data to get predictions distant in time by slowly catching up to them. The naïve baseline in Cramer was the predicted number of reported deaths in the next week is equal to the number of reported deaths in the previous week. Similarly, in the current study, the number of reported cases' naïve baseline is based on a function of the number of expected cases in the previous week. This similar naïve baseline prediction, for comparison, is being made using the CDC's methodology of $R_0{}^t$ by taking the number of infected in the previous week times R_0, taking the number of infected in the training data and interpolating it forward in time. This creates an exponential growth, which registers as a line on a logarithmic plot. This created an empirical R(t=53) for each state, and a sum combined R(t=53) for the USA as a whole, and interpolates it forward using $R_0{}^t$. This baseline is compared to the new methodologies in a manner similar to the CDC comparison. Using the current R(53) data for reference, the R_0 is determined to be

1.005 per week, extended forward in time from week 53's value R(53), multiplying this value by 1.005 every week forward after this time.

2.5 CHAOTIC METHODS

Many systems in nature are Chaotic, for example the famous Mandelbrot set, and other fractally-nested problems. A sine wave can be nested within another sine wave, creating Beat Notes. That second sine wave can then be nested within a third, fourth, etc. indefinitely. For example, the following equation (See figure 1 for plot) iterates the next step by taking a Rate of growth R times the sine of the previous iteration. For a rate of growth R (especially with R > pi) :

$$X_{(n+1)} = R * \mathrm{Sin}(X_n) \tag{4}$$

Nested sine waves must inherit some of the properties of the parent sine waves, regardless of the viewing scale; for example, the ratio of the second to the third is the same as the ratio of the third to the fourth; the ratio of the 1st and 10th is necessarily the same as the 10th and 19th, by the properties of logarithms. The use of such nested fractals is already well understood, even if difficult to decode. [8]

A solving methodology commonly used to break very strong encryption problems is the use of Rainbow Tables, [19] discussed next. In a strongly one-way encryption algorithm or hash function, a problem that is easy to solve one way but difficult to reverse is used, for example multiplying two numbers is trivial, however factoring large numbers is numerically difficult by brute force. If a hash method is sufficiently punishing to reverse, it would be easier to perform multiple iterations of the original, easy function. For example, encryption of the number 1, then the number 2, then 3, … through one million, using the easy half of the one-way function, and outputting the results in a table of the hashes of the first million integers, called a Rainbow Table. Then, the target’s encrypted traffic is scanned over a large period of time. If any of the transmitted numbers provided by the scanned target match any of the hashed numbers in the Rainbow Table, the original key that was used to encrypt that message might be revealed.

For example, say 65 hashed to 1,234,567,890,123,456,789. Reversing the function of such a huge number back to 65 might be nearly impossible. However, that very long number would appear in our Rainbow Table as f(65) = 1,234,567,890,123,456,789, right between f(64) and f(66). Then, at a later time, if the number "1,234,567,890,123,456,789" is observed in a target, one can recognize the values from the table, and realize the original password, before encryption, was "65". This is commonly how very unsecure passwords are mined from exposed password databases, even if the passwords were stored in an encrypted format.

One can make a similar Rainbow Table for the nested sine waves, knowing that below the resolution of the current scale of observation, the underlying mini-waves must be following the same pattern as the ones under observation by the rules of logarithms, above. One can then find the chaotic value of the function by using the Rainbow Table previously calculated for the coarsest level of precision, then adding to this a multiple of the logarithm of the next closest level of precision, and repeating to the desired level of precision. This is similar to how long division divides, for example 1/7 by first dividing by 10 and finding the next level of precision to the first numeral after the decimal point, (1) then dividing by 10 (30) and using a multiplication table to get the next numeral after the decimal point (4, because 4*7=28), then dividing by 10 and getting the next numeral (20, and 2*7 = 14) and so forth to a desired level of precision. Similarly, one can solve a fractal wave to an arbitrary level of precision by finding the outermost wave, and then the next outermost wave, and then so forth to an arbitrary level of precision. After all, any estimation of an outbreak wave on a ridiculously fine resolution, say from a second-to-second basis, is below the threshold of observation and is likely empirically meaningless.

This process is more numerically-based than the other two, rather than being as empirically based. Snapshots of all of the waves can be compared for key features, which can, in other embodiments, include machine learning. The first, second, and third derivatives of the running average can be analyzed as features to determine whether hidden patterns can be found that affect the remainder of the wave, forming a table of new, unexpected paradigm variables of the form ‘when X is observed to happen, Y is also observed to happen’. A given wave can be divided into a number of wavelets, wherein each wavelet’s behavior predicts certain aspects of the behavior of the remainder of the wave.

Validation is critical to this method because spurious correlations and machine hallucination is common. Even among human beings, superstitious behavior can easily arise when things happen around other things, by mere chance. However, sub-wavelet behavior that does predict future events can be analyzed, and the exact connection between the wavelet behavior and the future even can be registered and future events can be decomposed to a linear combination of the contributions of each sub-behavior. As each sub-behavior is deducted from the noisy whole, it is expected that future patterns might be found that are orthogonal to the current patterns, which can also be used to predict.

2.6 NEXT

Chapter 3 discusses how this method may be improved to predict sudden changes in outbreaks with far less time-series data than previous methods, allowing problems to be discovered earlier in time, and therefore reacted to, before the predicted consequences fully occur.

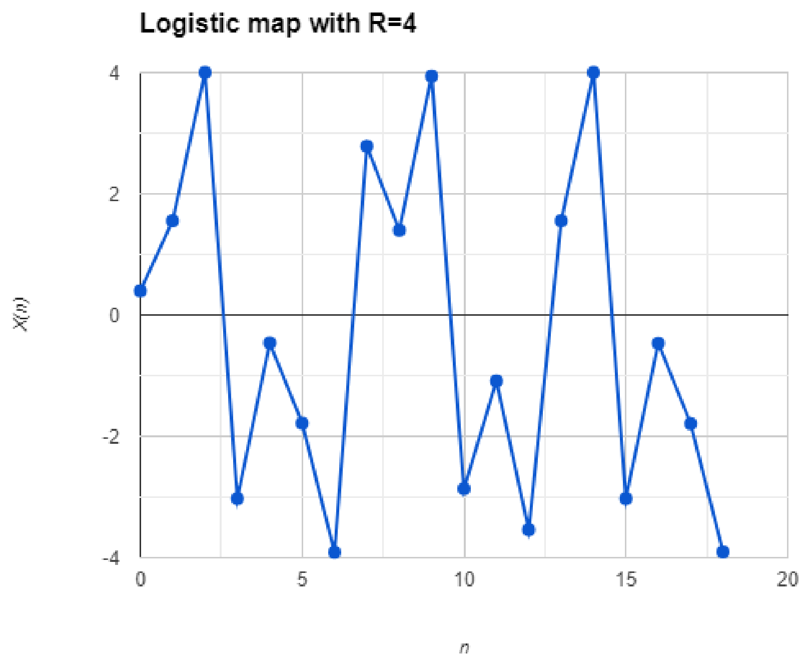


Figure 1, $X_{(n+1)} = R * \mathrm{Sin}(X_n)$, (Note the nested wavelets inside a larger wave)

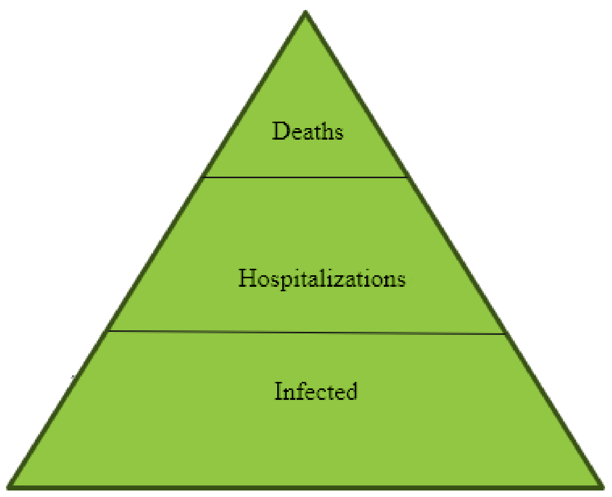


Figure 2, Pyramid of infectivity

3. Approach

3.1 BIG PICTURE

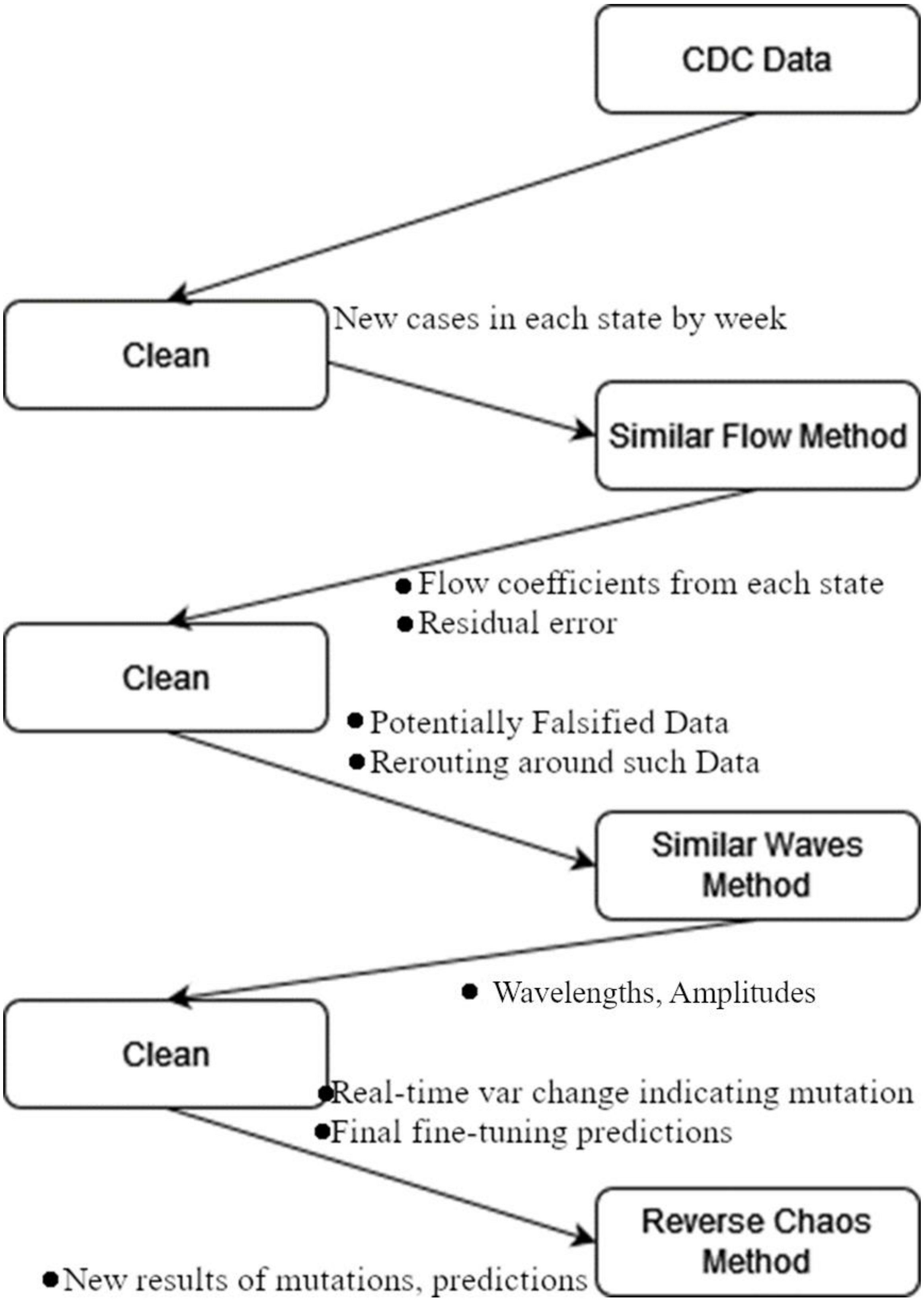

CDC Data
Clean
New cases in each state by week
Similar Flow Method
● Flow coefficients from each state
● Residual error
Clean
● Potentially Falsified Data
● Rerouting around such Data
Similar Waves Method
● Wavelengths, Amplitudes
Clean
● Real-time var change indicating mutation
● Final fine-tuning predictions
Reverse Chaos Method
● New results of mutations, predictions

Figure 3, Flowchart of method steps

This algorithm evolves from an input data source (such as CDC data) to establish further data that is invisible from a single data measurement, by collating together several snapshots of data from the same locations over time, and several times at the same location. While random variation would, of course, cause each iteration of the wave to be slightly different, the shape of the wave, in general, should be similar, in many ways, from one wave to another, since they were caused by certain initial conditions. Therefore, we treat observations from the same place in multiple different waves as observations of a similar event. Additionally, we can gather observations from several different places at similar times as observations of what is occurring at that moment in time. This allows for interpolation of multiple dimensions of data. This can generate knowledge of movement through space vs. movement through time, which allows for the mapping of the "contagiousness" of an outbreak, and the overall orthogonality of the generic data.

Each state's flow mechanics (in this case) is based on internal infection, with a modifier for importing of virus from the global society. Other methodologies presume infection increases to be a function of the current local or global infection rate, but that doesn't take into account external infection "popping" a bubble of an uninfected city, sparking a local massive uptick at a given arbitrary time. It is the combination of multiple aspects of the method that lends its strength.

Figure 3 is a flowchart describing the overall procedure of the method. The CDC (Center for Disease Control) has pandemic data available to a very high degree of precision. It has daily population and heat maps available for various states, cities, ethnic groups, and age partitions, telling us the number of documented, visible positive cases on a day-to-day basis. The current analysis is using time-series data from the 50 US states as well as Washington, DC to create a 51-element array. Additionally, the CDC added NYC as a 52nd "state", as well as 8 territories or regions, making 60 total locations. I added an additional trivial, fictional state of "Outside" to estimate inflow from the global population in that step. This creates a time-series array with 61 location entries, and weekly data from each state for time entries.

First, trailing indicators can be used to verify the veracity of data from municipal agencies. For example, the "excess mortality" is a very unalterable measure of the disease's progression. One takes the number of deaths from any cause, whatsoever, in a community over several years running. Then, this 'average mortality' is subtracted from the mortality in that community that occurred during the pandemic. The result is the 'excess mortality' that is assumed to be caused by the pandemic. If one area, for example Florida, suddenly shows a quite large number of deaths simultaneously with a precisely zero reported covid rate, this provides a strong indicator that the numbers being reported by Florida have perhaps been massaged too heavily, and should be relied upon with lesser confidence than data from more verifiable sources. Especially for the first few iterations, only highly trusted data is used in the method. Optionally, less-trusted data can be used later to interpolate future results once the baseline values are discovered analytically. However, less-trusted data is preferably not used to derive those values as this could skew the analysis. Further description of the data cleaning and filtering protocol will be discussed in the relevant section. This limits the data under review to a smaller number of ‘trusted’ states along one axis, currently 51, and a number of entries representing weeks along the time axis, creating a 2-dimentional array. Further details of this cleaning of data are defined in section 3.7, below.

The data is first fed into the Similar Flow method discussed in section 3.2, next section. The Similar Flow method’s algorithm is directed towards discovering the inflow of each state based upon the current values of all the other states at a given moment in time. The increase due to external contagion in a state is based not only on the neighboring states, but can also include one-time events, such as Mardi Gras, as well as the current infectivity of major population centers, such as New York City. The similar flow method aims to discover by how much a given state increases its infectivity by including importing infection from the other states in a given time period. The internal flow data from the Similar Flow method can then be used in the next algorithm.

The Similar Flow method takes in the 2-dimentional array from the cleaning step. Each state has an amount of “contagiousness” assigned to every week, which is number of newly infected from one week. Those people, who caught the virus, received the virus from one of two sources: internal or external infection. If they received it from internal infection, the amount by which the infection has increased from the previous week should be roughly proportional to the contagiousness of that state, as it represents some function of the spread inter-among the state of people who were previously infected. If they have received it from external infection, that portion of new infection must have come from contagiousness of another state or from the “outside” state. (The Outside state simply means it has come from outside the system as it has been measured, whether from outside the US, a heretofore unobserved pocket inside the US, or from a noncompliant state whose data was removed from consideration.)

The Similar Flow method determines what fraction of the contagiousness comes from within the state, which should be based on the previous contagiousness of that particular state, and what fraction comes from other places, which is the portion which is not based on the previous contagiousness of that particular state, but should be based on the contagiousness of each of the other states and the Outside state. The Similar Flow method’s algorithm is set up to determine this partition, and thereby outputs a connected network graph that identifies the inward/outward flow between the various states, and from each state to itself; see section 3.2 for details on the algorithm.

The data from the first iteration of the Similar Flow method is then fed into a second algorithm, the Similar Waves method, that aims to show purely internal infection, independent of outward importation of infection. Recall the concept of the viral load ceiling upon which an individual shows symptomatic infection, creating a second-order differential equation, that can be solved for.

The resulting data points of purely internal infection, derived from the previous method, can be placed in a Markov Chain, differential equation solver, or other, similar methodology, for example using the Python programming language. This allows multiple waves to be evaluated for shape parameters and magnitude parameters, where the shape parameters should be similar between waves of the same virus in the same location, however magnitude parameters can be wildly divergent. By measuring the behavior of the virus under different types of stresses, an acceleration factor can be derived to determine that types of forces that determine the wavelength of the wave and the amplitude of the resulting wave under various historically-known initial conditions, and how they affect the shape of the wave. The exact algorithm for finding this differential equation is described in section 3.3, further below.

Additionally, this information would allow the determination of a given wave's new hidden variables, such as historical knowledge of a mutation, by recognizing the sudden change in shape very early in the wave, as soon as it occurs, using the Reverse Chaos method, mentioned in section 3.4, below. Once the new variables are determined, whether through Reverse Chaos or Similar Waves, the algorithm can thereby extrapolate the remainder of the wave. In addition, if the wave's behavior changes suddenly mid-wave, this provides new information that the underlying hidden variables have suddenly changed and what they might now be; for example, a mutation causing increased replication and/or evasion of the patients' immune systems. The new hidden variables will allow for a new interpolation of how the remainder of the wave will now play out, accounting for this mutation's change in parameters.

For example, a large increase in replication or evasion will both cause the wave's amplitude to increase, however if evasion lowers the inhibitory threshold, the change becomes more obvious sooner, and makes changes to the wavelength that may cause it to burn out sooner. This changes the shape and wavelength of the wave in ways that a simple increase in replication does not. If the increase in replication is not caused by, for example, Christmas travel, this may provide early warning of mutation, especially when accompanied by new evasion markers. Otherwise, the Christmas increase in replication can be added into the model as an acceleration factor, and predict hospital occupancy in the weeks following Christmas based on the rates immediately preceding it and the estimated acceleration factor generated by holiday travel in previous years, thereby modifying the current wave’s trajectory by that holiday multiplier/acceleration factor, allowing multiple factors to be extrapolated simultaneously.

The amount of internal infection can be fed back into the Similar Flow algorithm to adjust the amount of external infection variables, and the loop repeats until the change in the variables so derived no longer changes by a significant amount. The resulting predictions can then be extrapolated forward in time to “future” data and compared to the CDC’s estimation of the future and actual data that occurred later in the epidemic to validate the methodology.

3.2 SIMILAR FLOW METHOD

The Similar Flow method works for measuring the effects of a node on its neighbors, creating a “ripple in a pond” effect. For each change made to a node, the "ripple" of that change will create measurable propagation results on the remaining nodes. The change the other nodes experience can be triangulated to compute the location and intensity of the original "splash".

A dripping faucet takes longer to fill up a cup than one going full blast. In the same way, the in-flow rate to any node (cup) is proportional to the time until a given threshold is infected in the node. In this way, the flow rate into the node can be calculated. Once calculated, the inflow of a node is equal to the flow input from all nodes. Therefore, once the flow rate for a node is calculated as a delta, a contribution flow can be calculated from its neighbors (and itself) as shares of the incoming flow. ("inflow" from a node’s self represents internal spread of the virus.)

Each “link” between given nodes A and node B has a value denoting the population travel dispersal/flux along that path from A to B. This is basically the contagion flow from A to B, and is roughly proportional to Pr(B is infected), given Pr(A is infected). Each link is then assigned a link weight depending on the flow along that link. The weight of each link can be used to calculate a minimum spanning tree using well-known methods of solving the resulting linear equations. It can then be extrapolated forward into the future to determine the outcome of the outbreak if nothing is done to contain it. This methodology can also be used as a validation technique for the method; can be used with "future" data from historical outbreaks to verify whether the predicted outcome matches the actual, historical outcome. Similarly, the original point of the spanning tree can extrapolated backward into the past to identify the origin point of outbreaks to see if it agrees with the generally accepted value for historical outbreaks, and then again as a practical use, especially to track the origin of foodborne illnesses.

An example step-through of the process will now be performed with arbitrary data.

Table 1: Arbitrary data used for flow example step-through

Infected	**Week 1**	**Week 2**	**Week 3**	**Week 4**
Node 1	4.5	14.5	21	22
Node 2	3.2	5.1	6.4	7
Node 3	2	2.5	2.7	2.8
Node 4	1.7	2.2	3.7	7

Node1(1,2) is defined as change in Node 1's infected value from Week 1 to Week 2, so the change is Week 2's value minus Week 1's value.. In this case, it is 14.5 – 4.5 = 10. Now that we know the incoming flow is 10 at this time, we need to find what share of that 10 should be attributed to each of the incoming links. Note that this includes a node pipe to itself, for contagion spreading within a place node. Node 1's internal infection is denoted x_1. Node 1's inflow from node 2 is denoted x_2. Node

1's inflow from node 3 is denoted x_3, and node 1's inflow from node 4 is denoted x_4. We can then solve for x_1 through x_4 using linear algebra, with a line of best fit for overconstrained values: (these Xn are other day 1 values for node1’s potential neighbors)

5

$$\begin{aligned} Day\,1:&\quad 10 = x1 \times 4.5 + x2 \times 3.2 + x3 \times 2 + x4 \times 1.7 \\ Day\,2:&\quad (21 - 14.5) = x1 \times 14.5 + x2 \times 5.1 + x3 \times 2.5 + x4 \times 2.2 \\ Day\,3:&\quad (22 - 21) = x1 \times 21 + x2 \times 6.4 + x3 \times 2.7 + x4 \times 3.7 \end{aligned}$$

Now with these equations, node 1’s weights to nodes 2, 3, and 4 can be calculated. However, this is only for node 1. The process repeats for nodes 2, 3, and 4 to find their weights to the other nodes.

Once each nodal weight is calculated, this immediately describes some patterns. Some places may have mostly internal infection, especially at earlier or later stages of the outbreak, however some high-tourist/ high-travel areas may have inflow infection all the way through the outbreak. Places with high internal infection have more internal disease travel than places with low internal infection, and may not be behaving appropriately. Similarly, if a line of best fit for a node is exceptionally poor, whereas all of its neighbors are not, this raises concerns that this specific node may be fabricating data, since it does not appear to be correlated with any travel from any of its neighbors in a consistent way. However, the system can 'repair' around this. Since the node with fabricated data is not contributing consistent flow to its neighbors, the flow contribution from second-most-nearby neighbors will show a better line of best fit than the immediate neighbors, thereby drawing some information out of the flow network despite one misbehaving node. For example, if Node3 is fabricating data, Node4’s input from x_1 and x_2 for Node1 and Node2 would be different in the line of best fit, as they would have a more consistent correlation with the results of node4 than the fictional data from Node3.

The final diagram can provide value, in that the effects of one area can then be extrapolated to other areas forward in time, also proving validation for the method by comparing predictions to “future” data that was recorded after.

Overall, this routine takes in 61 sets of weekly time series data and outputs a link diagram yielding each state's average contagion to each other state, and an additional link to self-contagion in the next week. This predicts that the contagion in a given state in the next week is a linear function of: the contagion of each of the states this week, and a global "Outside" parameter. The self-contagion will be further tweaked in the next functions, but for the first iteration will be assumed to be linear. After the first iteration of all three functions is complete, the process can also repeat this routine with a time-based independent function, instead of a purely linear self-contagion parameter, yielding more precise results, until the results are not changed significantly.

A number of states and/or metropolitan areas can be connected together in a flow diagram to determine how the effects of one influences the spread to the remainder. For example, during Spring Break and holiday travel, the effects on each state is very dependent on the effects of key other states that are common travel destinations, and states and municipalities that are close to each source. For example, a high number of virites in Chicago can affect the outbreak in Michigan, Indiana, and greater Illinois in the immediate future, even during non-holiday times. This causes high population areas to radiate contagion to all of the other nodes to a greater or lesser degree and to absorb contamination to a greater or lesser degree. Additionally, nearby areas may radiate and absorb non-trivial amounts of contagion even if they are not high-travel or high-population areas, for example again North Dakota to South Dakota. Originally, all possible locations can be made nodes, and then notes with trivial contagion links can be eliminated as inefficient. Historically, most of the contagion comes from a finite number of links that are either culturally important or geographically nearby, so that the final network can be solved with much, much fewer links. A perfectly naive way would be to try every node and every day connecting with every other node in every week, along every possible chain through every possible week for 100 nodes and 100 weeks, crating 100! links. Thankfully, only a handful are actually nontrivial, at least for the states of the US. It is cautioned however, that locations that have different geographical needs may experience a much different final flow diagram, for example nations that have most of their population in a single capital city, with a very sparse population in other districts. Those populations may wind up looking more similar to a hub-and-spoke diagram when trivial connections are severed.

3.3 SIMILAR WAVES METHOD

The immunity ceiling continuously increases by an amount proportional to (the current viral load * time), excepting artificial increases, such as vaccination, suddenly increasing the ceiling drastically. Therefore, the ceiling an individual possesses right now is necessarily proportional to the sum of all previous viral loads through all of time to t = 0, because on Day 1 the ceiling was increased by some x based on day 1's viral load, and then on Day 2, it was increased again by an amount proportional to Day 2's viral load, etc. A given individual will start somewhere at their 'natural immunity' point, and then travel 'uphill' based on the new viral load received, granting them additional immunity. Someone with absolutely no immunity has a probability of 1 of showing symptoms (ceiling = 0 virites); regardless of the probabilistic 'roll of the dice' of the number of virites obtained, the number of virites will always be above such a low ceiling (for example, those with compromised immunity).

The number of VISIBLE cases is therefore proportional to the number of virites, minus the immunity ceiling of the sum/integral of the number of virites in the past through time. The result is a differential equation that is solved to use the number of visible cases to derive the average amount of virites through finding the probability of infection (number of visible cases/population = probability of that population being visibly infected at that time), and from that deriving the average amount of virites in circulation, taking into account that some people, especially the elderly, have a very low amount of natural immunity to infection.

First will be the mathematical background of the solving of the underlying equations, followed by a more procedural description outlining the method steps.

The visible data of breakthrough-threshold cases from a single wave's window of time can be analyzed. The rate of rise and fall in the wave informs how "deep" the wave is allowed to be. The replication of the virus, reduced by the population's immunity raising the threshold ceiling, must necessarily limit the options of the shape and scale parameters of the resulting equation. Other waves have similar shape parameters/ disease models of propagation, but may have a vastly different scale/ intensity factor, from viral load/evasion mutation, as well as replication on the part of the virus.

Additionally, the release of various vaccines will drastically raise the viral threshold, whereas a documented mutation will drastically lower it. The effects of these numbers on the resulting waves provide valuable insights into how changing one parameter alone changes the values, independent of other parameters, such as how people travel. Similarly, tracking the waves that occur after major travel holidays provide additional insights of the increase in replication, independent of mutation, if a major mutation was not simultaneously documented in that area. This allows multiple paradigm variables to be analyzed independent of the others, in a more controlled environment.

A single wave has an amplitude A and a wavelength lambda. The first iteration can be solved empirically by taking the ratio of children, basic adults, and elderly in the population of, for example, a state, and the number proven infected (over threshold ceiling) in each sub-population of the state. Similarly, a running average of total hospitalizations, releases, and excess mortality can cross-verify the wave data as likely to be verifiable. The states with least opaque data make a good benchmark for the first iteration. The wavelength and amplitude of several waves adjacent in time and adjacent in space can then be measured, for example, North Dakota vs. South Dakota for the same time period, and a given wave compared to the wave immediately before it and after it.

A number of paradigm variables can be determined once a good baseline is found across many areas and time periods. The baseline variables are then validated by taking data early in the pandemic and then extrapolating them into the “future” to historical data that was recorded later in the pandemic. The later-pandemic data matches those that would be expected from the paradigm variables observed earlier in the pandemic. Further, when the virus has been observed to mutate at known dates, the paradigm variables change drastically coming off of those dates. Similarly, when patients have disobeyed no-travel orders during popular holiday times such as Christmas (and early in the pandemic, Mardi Gras especially), the paradigm variable for contagion increases drastically for a limited period afterwards. Even if some of these paradigm variables aren’t directly alterable (one cannot make people travel or not travel), the predictive capacity still provides value. For example, many hospitals are again seeing a resurgence around the holiday season, and foresight that contagion is likely to increase by around a known percentage can inform staffing levels and possible postponement of elective surgeries. Whereas, especially early in the pandemic, panicking hospitals postponed elective surgeries in some areas over and over, anticipating a resurgence that never came. Knowing when to predict the resurgence and capping a high likelihood upper and lower limit on the strength of the resurgence allows such resources to be redeployed elsewhere.

Once the effects of changes on paradigm variables are better known, combined paradigm variables can be used to predict and interpolate multiple variables simultaneously, for example a mutation that occurs during Christmastime. The Mutation meta-variable can be deduced from the change in wave behavior immediately preceding the holidays, and the Christmastime travel behavior can be inferred from previous and next years’ data. The interaction therebetween can then be deduced from each contribution separately, and the additional contribution to each other. Eventually, the values of several paradigm variables can be deduced, for example those that are unique to one geographical area such as Mardi Gras, or a particularly unlucky random event, or New York City’s first outbreak, as a deviation from a standardized wave for that time period. The inertia of purely exponential growth is

$x(t+1)=R_0*x(t)$ + Neighboring add-ins(t) from Outside (denoted O(t)), for some infectivity increase R_0, and the neighboring add-ins from phase 1 are deducted to calculate purely internal spread. But this is not a constant R_0; it goes up and down. So we have the goal of minimizing epsilon, ε, in:

$$x(t+1) = \varepsilon + O(t) + R_0 \times x(t) - Ceiling(Zpop) \qquad 6$$

The R_0 can be calculated from very early in the pandemic, when virtually no one is hitting the ceiling naturally. The ceiling ratio can be best calculated when the second derivative switches from positive to negative, meaning that more people are being saved by the ceiling than getting new cases. Zpop is a normal distribution of the natural immunity ceiling, ex. 1,000 virites for adults, 500 for elderly, 1500 for children. This can be modeled loosely as a normal distribution. In addition, the immunity for each step is indicative of a higher level of infection. Ex. People who are very affected are hospitalized, and people that are even further infected cannot be saved. This means the ratio of cases of infections to hospitalization to deaths provides a good indicator of the "depth" of the curve, whereas the gain or loss over time gives a good indication of the "width" of the curve.

The curve can be difficult to solve analytically, however, a table can be made for varying height and width values, and the table entry that most closely matches the curve can be used to define close height and width vales that must have created a similar curve to the one observed.

This is especially true as Zpop changes over time proportionally with the integral of all the asymptomatic virites an individual has ever come into contact with, so both the mean and standard deviation go up over time, resulting in an expanded Zpop of the original Zpop + integral(x(t)) dx :

$$x(t+1) = \varepsilon + O(t) + R_0 \times x(t) - Ceiling \times Z_{pop} \int x(t)\, dx \qquad 7$$

Turning the difference equation into a differential equation as follows:

$$x'(t) = \varepsilon + x(t) + O(t) + R_0 \times x(t) - Ceiling \times Z_{pop} \int x(t)\, dx \tag{8}$$

Taking the derivative of the whole equation with respect to x gives us the following second order differential equation:

$$x''(t) = x'(t) + O'(t) + R_0 \times x'(t) - C\,(Z_{pop} + x(t)) \tag{9}$$

Rearranging gives us a very solvable equation, for some constants C_1 ,C_2, where C_1 is 1 + the R_0 and C_2 the old C times Z_{pop} :

$$x''(t) - C_1\, x'(t) + C_2\, x(t) = O'(t) \tag{10}$$

C_1 and C_2 can be solved for by the computer; in this embodiment, by the Python Solve and Fast Fourier Transform functions, and the results can be used to modulate the internal contagion from a node to itself in the Similar Flow method, fine-tuning the parameters for O(t), which can then, in turn, fine-tune the parameters for the Similar Waves/Reverse Chaos methods.

The procedure of the method steps will now be described. The first step is to take the purely internal contagion from the states' time-series data, which is the amount of contagion each state has to itself in the future, after the outside contagion has been subtracted, so there will be 60 differentials between each state at week t and week t+1 that cannot be explained by external effects. This sets the "outside" of the above equation (10) to zero.

We can then take the grid of the 61 states under consideration by the number of weeks under observation for analysis. In this embodiment, since we are attempting to measure R_0 especially, the first wave is measured in the first year. The resulting 61 state by 53 week grid is run through a differential equation solver in Python (although other solvers can be used such as MATLAB or R) to solve for the constants in equation (10). The result is a single sinusoid up and down with $1 + R_0 = C_1$ and C_2 being the mean of the distribution of the viral load ceiling.

If the wave under consideration contains no sub-waves, for example in the first wave ever of the COVID-19 pandemic, this completely describes the internal contagion. The internal numbers of infection from each state to itself is replaced by the numbers from the differential equation and recorded.

However, sometimes a single wave contains one or more sub-waves, especially later in the pandemic, as travel restrictions are relaxed and mutations occur, leading to a messier model. This is taken into account by the Reverse Chaos method, below in section 3.4.

3.4 REVERSE CHAOS: THEORY

Some variables are deeply time-based, not simply travel behavior, but also seasonal changes in virus behavior. For example, during Winter, the air is very dry, the virites tend to not experience de-aerosolization.

Normally, in the Summer, virite particles are encased in water droplets spread by coughing or sneezing, and stay in the air much like motes in a sunbeam. Such water droplets encounter and absorb water in the atmosphere, gradually becoming heavier, and no longer able to maintain their status as aerosol particles. This causes the droplets to fall to earth in a process known as de-aerosolization.

In the dry air of Winter, this process is much less effective at naturally removing contaminants from the air, so the temperature and internal humidity has a great effect on the spread of disease completely independently of travel or accumulation of persons together. Even such places as grocery stores can more readily spread such droplets, which can “hang” in the air for a protracted time, infecting other individuals. Thereby an additional paradigm variable for seasonality should be applied. Such a paradigm variable can be unmasked by comparing similar states close together that experience the same temperature swings, and by the termination shock during spring, when the effects begin to vanish.

During the Similar Waves method, an occurrence may arise where one or more sub-waves are spotted inside of another wave, creating a wavelet with a much smaller amplitude and wavelength than the parent wave under observation. This implies that some occurrence has caused a deviation in the trajectory of the wave. The wavelets cannot be simply subtracted from the parent wave, as the wavelet and parent experience synergies; by having an increase in contagion, for example, a smooth line or curve can experience a jump to a suddenly higher value, reaching the peak much earlier than would otherwise be the case, affecting not only the amplitude, which could easily be subtracted out as $F_{\text{TOTAL}} = F_1(t) – F_2(t)$, but also affects the wavelength, so the resulting double-wave equation would appear more like as follows, with both X_1 and X_2 being variables, or even functions, that must be solved for:

$$F_{TOTAL} = e^{it(x1-x2)} \tag{11}$$

A variation on Rainbow Tables can use pre-solving for a handful of known problems, and then recognize when these problems occur, and make adjustments in real-time.

For example, the change in amplitude and wavelength after a known mutation can be recorded, as well as changes after the onset of cold weather, holiday travel, as an offset from the amplitude and wavelength of a baseline immediately before the mutation or travel at each location. Then, for example during holiday travel, that offset can be applied at known times in the future, as an additional increase. For events such as mutations or cold snaps that do not occur at preset times, a running predication window of multiple potential events can be run concurrently. For example, a prediction of the future n weeks with no events, and a secondary or tertiary predication of the future n weeks if a mutation were to occur right now, or a cold snap were to occur right now. As the trajectory continues for the next several weeks, if it suddenly deviates from a baseline trajectory and starts to very closely match an alternate trajectory, the system can conclude that a mutation might have occurred at the crucial point and then emphasize the remainer of the trajectory of that wave cycle with that in mind, giving feedback to the operator of future events before the mutation has even been isolated in a medical laboratory.

In this way, one or more side-windows can run concurrently with the Similar Wave method and the algorithm can modify predicted outcomes if certain patterns are realized. The modified prediction is then fed forward to the next iteration of Similar Flow and Similar Wave methods until concurrence is reached, same as the iterations of Similar Waves in isolation.

3.5 REVERSE CHAOS: IN USE

The procedure will now be described in some detail, repeating equation (10), with the additional modification that the outside amount O (for outside) has already been subtracted, and therefore is always equal to 0, along with its derivatives O':

$$x''(t) - C_1\, x'(t) + C_2\, x(t) = 0 \qquad 11$$

This is a very classical homogenous differential equation, and therefore should exhibit very classical wave-like behavior. However, frequently, something will occur mid-wave that changes the complete trajectory of the remainder of the wave, leading to a one or more sub-waves or wavelets. However, these wavelets must be created because either C_1 or C_2 has changed mid-wave. Either the virus has become more contagious, either through excess travel or replication mutation, or the ceiling has changed up or down, because of vaccination or evasion mutation, or possibly both, such as an evasion mutation followed by a vaccination for that mutation, or by vaccination followed by travel or mutation.

Because of the fact that classical differential equations with different values of C_1 or C_2 are well understood, waves with many different values of C_1 and C_2 can be pre-computed. After the very first wave is computed in a more simple mode, waves with different values of C_1 and C_2 can be extrapolated out and stored as virtual expected states alongside the "real life" states, with rising and falling values, alongside the actual measured outcomes of the actual "real life" state data of the various states. For example, C_1 (AKA R_0) and a C_2 of 1 through 10 can be pre-computed and the results populating 100 (10X10) virtual states' data.

If these values change mid-wave, the values will be similar to one of these 100 waves for a period of time and then suddenly deviate at some time t, mid-wave, to a different prerecorded value. Additionally, in an alternative embodiment, the R-squared values of the equation solver for the Similar Waves method will be much higher than it ordinarily is, as it does not precisely match any single-wave situation. Additionally, when a change occurs mid-wave, a wavelet will be observed in the graph of the wave near the point of change. All three of these features, in tandem, can be used for validation of the Reverse Chaos method, as they should always be found together.

The Reverse Chaos method, as explained previously, can keep a running tally of, for example, all 100 possibilities to compare a given state to. If the C_1 and C_2 values do not change mid-wave, a state should, very roughly, follow a standard wave trajectory. The reverse Chaos subroutine matches the values of the state in question versus the expected values of the virtual state table, which should roughly measure the final outcome of C_1 and C_2 measured by the solver in the Similar Waves method. When this does not happen, the wave is halved and measured again. If convergence is still not reached, the halving may be repeated until it is. This results in at least one section that does match a wave, and has a valid single solution to the differential equation under the similar waves method, successfully completing that portion of the similar waves method. However, this leaves another section of ¾, ½, etc. of the wave that does not match the current solution and requires one or more additional solutions. The remaining portion of the wave, after a pause of one week, can then be compared against the 99 remaining virtual waves to see if one follows the reminder of the wave more closely. If not, there are two sub-waves and the halving process is repeated to create a third section of the wave. After the sectioning is completed, the second section only (followed by the third section only, if any) is then fed into the Similar Waves equation solver and new C_1 and C_2 are derived for the remainder of the wave.

The reverse Chaos method is expected to be relatively rare, compared to the Similar Waves method and only handles outliers, where a major event happens mid-wave. Numerous waves may come and go before a large change in the C_1 and C_2 values become necessary. However, even though they are rare, it is expected to be the best source of data when measuring a pandemic in real time, as a change mid-wave gives an indication that the underlying event has occurred, and if it is a mutation, provides insight that a mutation has occurred before it is generally known, and additionally, what the nature of that mutation might be.

While the real strength in the Reverse Chaos method is in real-time data, it can also be used to lookback to previous training data and find a nexus where a variable has changed suddenly, implying a mutation. If one is found, the second "real-time" algorithm can be run on each half of the state so found, one pre-mutation, and one post-mutation.

3.6 PUTTING IT TOGETHER

Once concurrence is reached, in that none of the three methodologies add meaningful changes to the predicted behavior in a cycled iteration, the resulting prediction can be compared against the CDC’s predictions and the actual observed data of what historically occurred after that point in time. A graph is displayed of the CDC prediction, actual results, and newly predicted results ,and the similarities therebetween can be described.

As a possible embodiment, if the results are changed significantly, for example by more than 1% over the previous results, the process repeats instead of terminates. Thereby the function as a whole must necessarily run at least twice, but further runs of this routine are data-dependent, and depend on the time to convergence for a given data set.

The sum of the results from the similar waves method that is subtracted from the total in all 50 states is compared to the total infections. If the subtraction amount in all states is less than 1% of the total infection in all states, convergence is presumed to have been achieved, and further looping is not necessary.

3.7 DATA CLEANING

Certain states are removed from the variable calculation phase as potentially falsified data. States with such data can still have some of the calculations performed, but not all. Besides the effect of "healing" around missing data, as elaborated on in section 4.1, states with no wave behavior may also contain falsified data, and so no wave processing is performed under the Reverse Chaos method. Because any falsified data would not follow any particular pattern, any attempt to incorporate such patternless data would be more likely to introduce random noise than actionable insight.

3.8 EXPERIMENTAL PROCEDURE

This section describes what was done to generate the results that are presented in Chapter 4 and presents the pseudocode for those algorithms. Appendix A contains the entire Python notebook.

The first step generated the linear model for the Similar Flow approach, which was done using Algorithm 1. The matrix *X* is created by using the first 53 weeks of data to predict the values for weeks 2 to 54. This was done for each state, and the results were combined into the matrix *X*. The matrix *Z* contains the predictions for each state for the weeks 54 to 173.

The following pseudocode describes the process:

Let s_{it} be the number of actual COVID-19 cases for state *i* in week t.

Let z_{it} be the predicted number of COVID-19 cases for state *i* in week t.

```
Algorithm 1. SIMILAR FLOW
Given: for i = 1, …, 60, t = 1, …, 173.
1. Set $s_{it} = 1000$ for i = 61, t = 1, …, 173. // "Outside" Row
2. For i = 1 to 60 // loop over states
3.      Set $y = [s_{i2}, \dots, s_{i54}]$
4.      Using , for i = 1, …, 61, t = 1, …, 53, and y, use linear regression to determine a vector of
        coefficients x used for predicting the values for state i.
5. Form the matrix X from the i = 1, …, 60 of each individual x vector.
6. Form the column vector $[z_{1,53}, \dots, z_{61,53}] = [s_{1,53}, \dots, s_{61,53}]$ // the last week of the training data
7. Set $[z_{1\dots 53},] = [s_{1\dots 53}]$ // "Predict" the current training data
8. For t = 53 to 173 // loop over remaining weeks in time horizon
9.      Calculate $\left[z_{1,t}, \dots, z_{60,t}\right] = X\left[z_{1,t-1}, \dots, z_{61,t-1}\right]$
10.     Set $z_{61,t} = 1000$
11. Form the matrix Z = $\left[z_{1,54}, \dots, z_{60,173}\right]$ and the matrix S = $\left[s_{1,54}, \dots, s_{60,173}\right]$
12. Calculate $[deltas] = Z - S$ // Calculate prediction error
```

The next step determined the one-dimension discrete Fourier transforms for each state, which generated amplitudes and wavelengths for each state. This was done using Algorithm 2.

```
Algorithm 2. SIMILAR WAVES
Given: $deltas_{it}$, for $i$ = 1, …, 60, $t$ = 1, …, 53  // prediction error from algorithm 1
$X$, the matrix of regression coefficients
$S$, the matrix $[s_{1,54}, \dots, s_{61,173}]$

// s_it = internal + external + sinusoidal, so we
// increment each state's remaining error by the purely internal contagion so that
// the final prediction = all minus external contagion from any other state, as in equation [10].

1. Set $D$ = $[X_{1,1}, \dots, X_{61,61}]$ // the diagonal of $X$
2. Calculate $T = S - DS$ // prediction errors and purely internal contagion, from the X matrix, which is the internal contagion to itself in the next t increment.
3. For $i$ = 1 to 60 // loop over states
        Determine the one-dimension discrete Fourier transforms of $[s_{i,54}, \dots, s_{i,173}]$,
        which returns a set of frequencies $F_i$ and amplitudes $A_i$ for state $i$.
        Calculate adjusted amplitudes $A_i^* = \frac{2}{54}|A_i|$ and wavelengths $\lambda_i$
        (the reciprocals of the frequencies).
```

For New York City, we used the first three amplitudes (1100, 800, and 1400) and wavelengths (54, 27, and 18) to create a sinusoidal model that was added to the predictions in the matrix *Z*, as shown in Algorithm 3, to create new predictions *W*.

```
Algorithm 3. [for state $i$]

Given:

$z_{it}$ for $t$ = 54, …, 173  // Predicted cases for state $i$.

$A_{i,1}^*$, $A_{i,2}^*$, and $A_{i,3}^*$ // adjusted amplitudes for state $i$.

$\lambda_{i,1}$, $\lambda_{i,2}$, and $\lambda_{i,3}$ // wavelengths for state $i$.

1. For $t$ = 66 to 173 // loop over weeks

2.      Calculate $w_{it} = z_{it} + 2A_{i,1}^*\sin\left(\frac{2\pi t}{\lambda_{i,1}}\right) - 2A_{i,2}^*\sin\left(\frac{2\pi t}{\lambda_{i,2}}\right) - 2A_{i,3}^*\sin\left(\frac{2\pi t}{\lambda_{i,3}}\right)$
```

For the entire county and selected jurisdictions (the District of Columbia (DC), Virginia, and New York City), we used the CDC's exponential model to create a baseline prediction *B*, as shown in Algorithm 4. We used the following basic reproduction numbers: DC: 1.005; Virginia: 1.0036; New York City: 1.008; all of the USA: 1.00451355. The value for the USA was found by fitting an exponential curve to the nationwide totals from weeks 40 to 100.

```
Algorithm 4. [for state $i$]
Given:
$s_{i,53}$ // Actual cases for state $i$.
$R_{0,i}$ // Basic reproduction number for state $i$.
1. Set $B_{i,54} = R_{0,i} s_{i,53}$
2. For $t$ = 55 to 173 // loop over weeks
3.     Calculate $B_{it} = R_{0,i} B_{i,t-1}$
```

For the entire country, we included a sinusoidal model (using the sums of the states' amplitudes and the wavelengths 54, 27, and 18) and a factor for the impact of the vaccinations starting at week 112, as shown in Algorithm 5, which yielded the national prediction *U*. The amplitudes were rounded to the nearest thousands, so the values used were 101,000, 150,000, and 184,000.

```
Algorithm 5.

Given:

$z_{it}$ for $i$ = 1, …, 60; $t$ = 54, …, 173  // Predicted cases for state $i$ and week $t$.

$A^*_{i,1}$, $A^*_{i,2}$, and $A^*_{i,3}$ for $i$ = 1, …, 60 // adjusted amplitudes for state $i$.

$\lambda_1$, $\lambda_2$, and $\lambda_3$ // Wavelengths

$v$ = -10000 // constant for impact of vaccination

1. For $a$ = 1 to 3
2.         Calculate $A^U_a = A^*_{1,a} + \cdots + A^*_{60,a}$ // sum of the amplitudes (rounded)
3. For $t$ = 54 to 173 // loop over weeks
4.         If $t$ < 112
5.                 $u_t = \sum_{i=1}^{60} z_{it} - 2A^U_1 \sin\left(\frac{2\pi t}{\lambda_1}\right) - 2A^U_2 \sin\left(\frac{2\pi t}{\lambda_2}\right) - 2A^U_3 \sin\left(\frac{2\pi t}{\lambda_3}\right)$
6.         Else
7.                 $v = 61v$
8.                 $u_t = \sum_{i=1}^{60} z_{it} - 2A^U_1 \sin\left(\frac{2\pi t}{\lambda_1}\right) - 2A^U_2 \sin\left(\frac{2\pi t}{\lambda_2}\right) - 2A^U_3 \sin\left(\frac{2\pi t}{\lambda_3}\right) + v$
```

```
Algorithm 6. REVERSE CHAOS (embodiment 1: lookback to previous predictions)

Given: $deltas_{it}$, $s_{it}$ , $z_{it}$   for $i$ = 1, …, 60, t = 1, …, 173,  // deltas, z, s states from SIMILAR WAVES

1. For $i$ = 1 to 60 // loop over states
2.         Fit $z_{it}$ to a single sinusoid y(t) = A Sin(b t) + c
           errorBaseline = 2*Average($error_i$) from above fit.
3.         if ($error_i$) ) > errorbaseline) )
4.                 perform two one-state half-time REVERSE CHAOS:
5.                         REVERSE CHAOS (s[i, 0…(53 /2)] )
6.                         REVERSE CHAOS (s[i, (53 / 2) … 53] )
7.                         Print results for manual look
```

```
Algorithm 7. REVERSE CHAOS (embodiment 2: Real-time analysis)
Given: $deltas_{it}$, $s_{it}$ , $z_{it}$   for $i$ = 1, …, 60, t = 1, …, 173,  // deltas, z, s states from SIMILAR WAVES
1. For t = {current week} down to 1, step by -1 // count backwards from now to beginning of pandemic
// populate [A], a 10x10x{current t} table of errors
2.      For C_1 = .5 to 1.5, step by 0.1
3.          For C_2 = .5 to 1.5, step by 0.1
4.              $A_{C1,C2}$ = Solve x" (t)  – C_1  x' (t) + C_2  x(t) = 0, x(t)=sum(s(t))
5. For t = {current week} down to 1, step by -1 // count backwards to beginning of pandemic
6.      Let k_j = lowest errored column, the minimum value.
7.      if (k_j != previous_kj) AND (previous_kj != 0)
            // if the lowest errored C1 and C2 is different, mutation has occurred
            // if previous_kj = 0, then the variable hasn’t been set yet. (first iteration)
8.          Print("mutation at time:" t "of strength:" k, j)
9.          Return t
10.     previous_kj = k_j


11. Print "no mutation found"
```

4. Results

4.1 SIMILAR FLOW METHOD

Table 2: The x vector coefficients from the state of Colorado:

1	0	2	0	3	0	4	0	5	0	6	0
7	0.3357675	8	0	9	0	10	0	11	0	12	0
13	0	14	0	15	0	16	0.13530653	17	0	18	0
19	0	20	0	21	0	22	0	23	0	24	0
25	0	26	0	27	0	28	0	29	0	30	0
31	0	32	0	33	0.08497	34	0	35	0	36	0
37	0	38	0	39	0	40	0.01574976	41	0	42	0
43	0	44	0	45	0	46	0	47	0	48	0
49	0	50	1.08451019	51	0	52	0	53	0	54	0
55	0	56	0	57	0	58	0	59	0	60	1.97079
61	0										

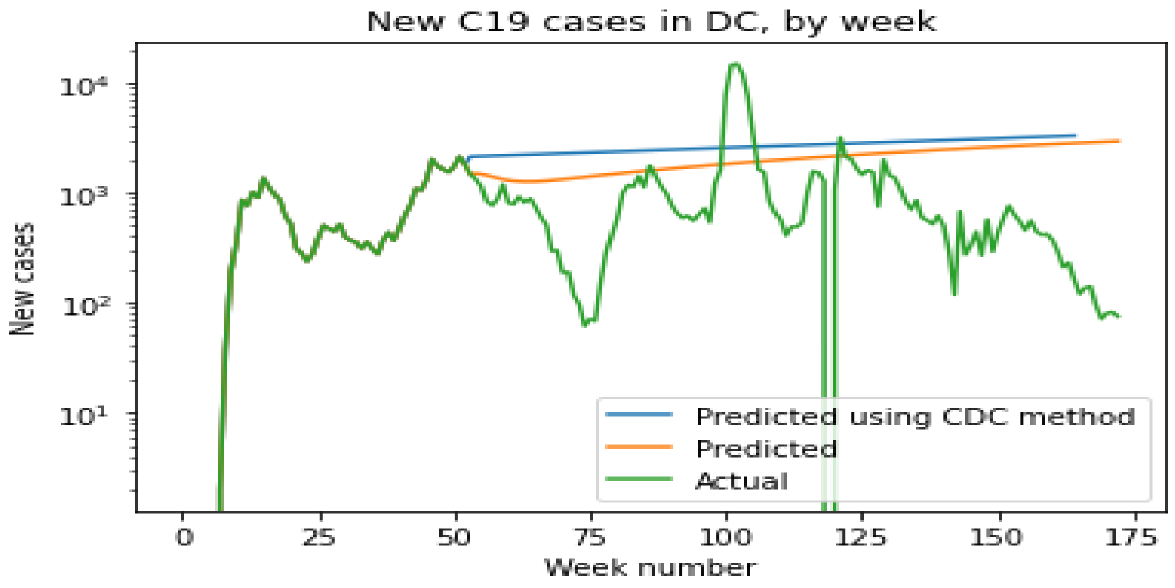
The Similar Flow method reduces the calculation, as expected, to a small number of either neighboring states or other high-value sources. Much of the infectivity of a given week can be predicted using the infectivity of these "friend" states to each given state, that have a nonzero coefficient. Because of this, the baseload of infection can be estimated out to a very high margin for many states, as a linear combination of their previous week's infectivity and the Friend States' previous week infectivity. Colorado's results are posted above as exemplary. In this table, Colorado, for example, has no contribution from states 1- 6, a contribution from state 7 (itself), state 16 (Iowa), and a small contribution from states 33 (North Dakota) and 40 (NY City), with a larger share coming from state 50, (South Dakota), and state 60 (Wyoming). So the result at t+1 is predicted to be roughly this linear coefficient times the values of these states at time t, for each such state.

Maryland and Virginia, unsurprisingly, both have most of their external contribution only from DC. DC has almost all of its infectivity from within DC and from "Outside", one of the few states to receive a contribution from Outside. This, again, makes logical sense, in that DC has a large amount of international travel and so would acquire most of its infectivity from Outside, especially early in the pandemic, which is the training data currently under consideration. Both NY and DC, additionally, had externally-acquired outbreaks very early in the pandemic, so that in this particular case, most of the contagion was imported into international cities, and then spread to the rest of the US. Contrast to states that mostly generated their own internal contagion, and then became a net exporter of contagion to the world, such as Florida, which is a common source of contagion, but not a common sink of contagion.

It is noted that Puerto Rico's contagion did not seriously affect, or be affected by, any other area. All of its contagion seemed to be internal. As a result, its waves may not provide meaningful insight into the waves of any other states.

This model forced a y-intercept of zero for three reasons. First, an attempt using a very small y-intercept led to drastically, obviously spurious results, secondly because the infectivity was indeed 0 at the beginning of the pandemic so it makes sense that the logical outcome would follow from the mathematical, and finally, because any infectivity that would normally be carried independent of other states is handled by the "outside" factor, which, strangely enough, wasn't used in all of the states; mostly New York, DC, Nevada (Las Vegas), and to a lesser extent, South Carolina, had significant contribution from "Outside", likely because these are good tourist areas, and therefore with a larger contribution unaccounted for from the other states' general travels within the US.

An additional requirement made to the model was that all infectivity coefficients must be non-negative, one because this makes logical sense that there are no anti-infected spreading anti-contagion, and when it was allowed, the system allowed states to have billions infected to compensate for "negative" billions infected elsewhere, which while being technically tighter to the training data, represents an overfit, and is useless at predicting realistic outcomes, since a state cannot realistically have negative infected.



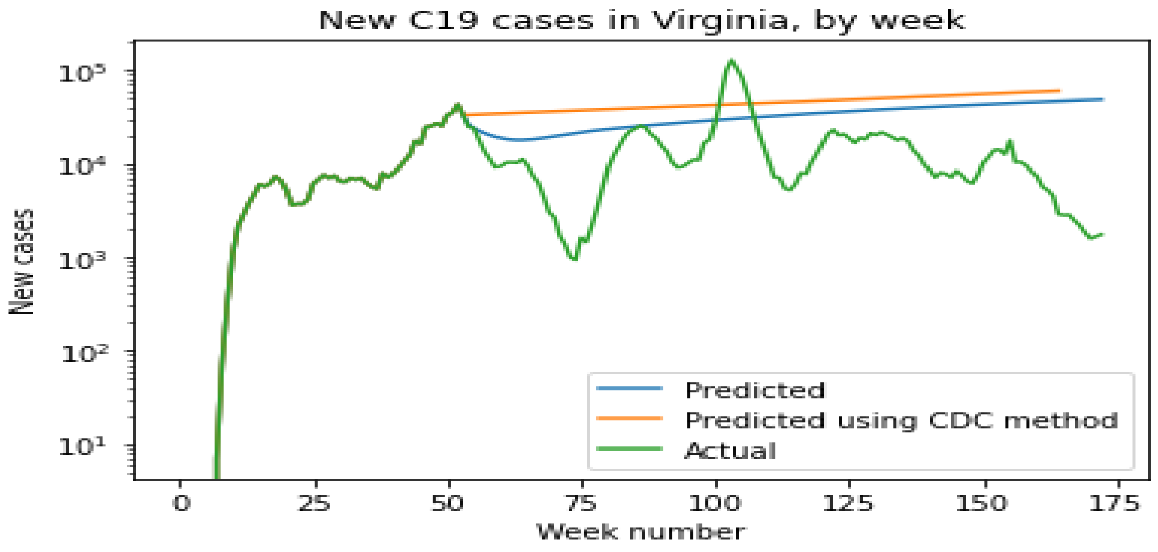


Figure 4: New cases in DC and Virginia on a logarithmic scale.

Colorado is chosen as exemplary in this section for ability to "heal" around any potential false data, for example by Wyoming. If Wyoming's reported data does not form a good linear combination of any other states because it contains false data, the genuine contagion would still grow and decline corresponding to reality, regardless of any fictitious reporting. The actual, genuine contagion will show correlation with other features, for example North and South Dakota. Fictitious Wyoming data will start to see less and less linear correlation with the other states, such as North and South Dakota, as well as Colorado here, who is importing contagion from Wyoming. However, since the genuine contagion of Wyoming shows a strong linear correlation with North and South Dakota and also with Colorado, the genuine measurement of Colorado will also start to see a large correlation with North and South Dakota, that is not explained by any apparent correlation with Wyoming. Therefore, the randomized, potentially falsified data will see less and less importance, and be less strongly linearly correlated with the genuine data. Instead, the genuine data will show a stronger correlation with other genuine data, one step removed, in this case. This shows an impressive ability for the system to "heal" around sparse or false data by gathering much of the information from the genuine data it does have. In this way, the extrapolation has healed around any falsification by Wyoming by also including some factors from some of Wyoming's own factors directly: North and South Dakotas. In this way, the prediction for Colorado stayed accurate, even though some of the input data may have been falsified.

The Similar Flow method thus used more states' data points if sourcing from one of these states that were later disqualified for possibly having bad data, essentially attempting to correlate instead with the states that the disqualified states **should** have logically been correlating with, but instead the disqualified states were later to be found to be correlating with nothing in particular/randomness. As a result, the states that were trying to import contagion correlation from the states that would later be disqualified, instead imported contagion correlation from the states that the disqualified states should have properly been correlated with.

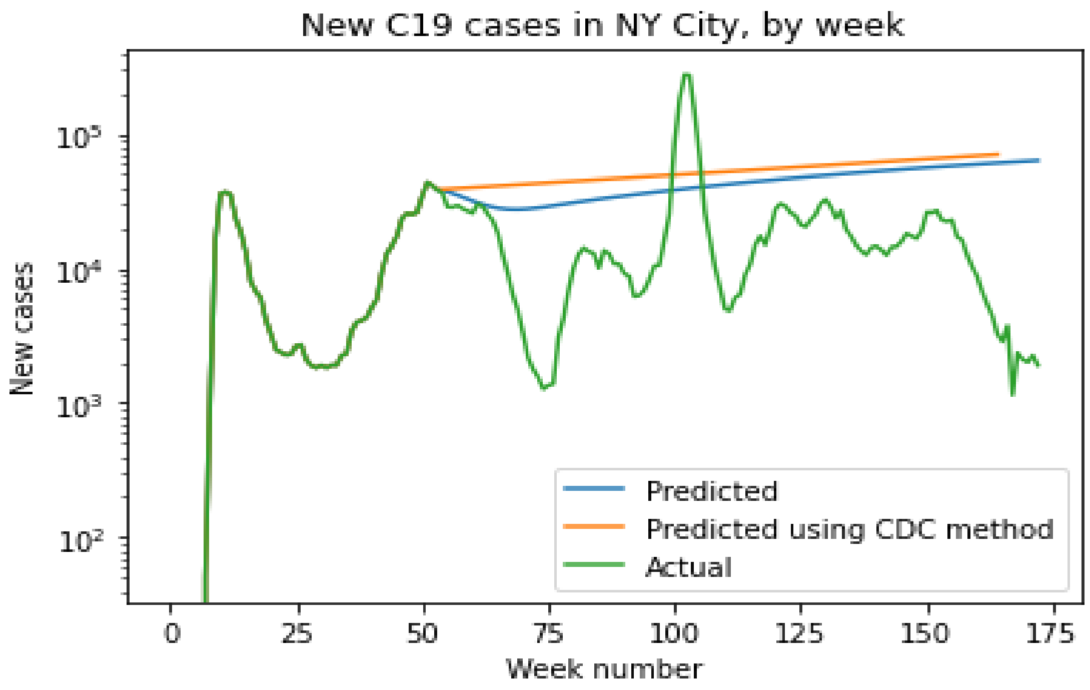


Figure 5: New cases in New York City on a logarithmic scale, by week

4.2 SIMILAR WAVES METHOD

Table 3: Output of the computed wavelengths and amplitudes for arbitrary states: (note the wavelengths are the same)

California (CA) Wavelengths:				
54	27	18	13.5	10.8
California (CA) corresponding Amplitudes:				
37005.9392	40317.6563	29671.2868	18941.71288	13437.72134
Colorado(CO) Wavelengths:				
54	27	18	13.5	10.8
Colorado(CO) corresponding Amplitudes:				
6505.13957	5252.29222	3137.15396	1430.489556	1139.572272

A vast majority of the states exhibited very similar wave behavior. The error and the internal contagion were fit to sinusoids to find the wavelength and amplitude of the waves using the Fast

Fourier Transform function to find the frequencies. This model found that nearly all the states had a wavelength of approximately 1 year, which is logically expected because of the effects of deaerosolization. This predicts that the pandemic will become more extreme in winter months and less extreme in summer months. Further waves are found at the half-year mark, and these are likely due to contagion vs. immunity itself. Temporary sub-waves by state are also created, as vaccinations occur and the virus mutates, changing wave behavior somewhat, but this is discussed in the reverse chaos method, below.

Interestingly enough, all of these states still under consideration exhibited very similar wavelength and amplitude variations (shape), differing only in the time to onset (phase). This makes sense in the case of COVID-19, because of the early-contagion lockdowns resulted in less cross-national travel. Because of this, most contagion sources were from within the state itself once its bubble had been sufficiently "popped". The state would then go on to follow the "standard" wave once this had occurred, with only minor variation from the input from other states, accounted for in the Similar Flow method. The primary state exceptions were Florida and DC, which were tourist areas which saw significant traffic from other states or regions, which could, theoretically have varied their wave mid-wave with an onslaught of new cases that was not grown internally. However, contrariwise, these are also some of the very first states to be infected, so they were responsible for infecting other states, rather than being the states infected themselves, and none of the other states had their bubble affected significantly after it had already been popped. As a result, while these states had tourists that could theoretically have suddenly raised their contagion rates in violation of the model, these areas instead saw well tourists that then became sick on vacation, and then went home to infect their home states. As a result, these states did not see sudden upticks mid-wave, but instead caused other places to increase their infectivity mid-wave. However, since each state had a relatively small proportional population from that other state on vacation at any time, this did not suddenly uptick the other states either, but was instead folded into, as a continuous value in the "regular" wave, which is much easier to model. As a result, the only changes in wave shape come from mutations and vaccinations, which are accounted for in the reverse chaos method, below.

A few of the states did not exhibit good wave behavior and had to be removed from consideration in the Reverse Chaos method for different reasons. Some states had effectively zero amplitude in their frequency of occurrence at any wavelength whatsoever, and therefore could not be solved nontrivially: Florida and Arkansas. Further, Wyoming had a suspiciously low amplitude, but was still solvable. After subtracting the linear coefficients from the Similar Flow method, the rate of occurrence of these states did not further go up and down significantly, but were always effectively anchored to some arbitrary value, with a very small random deviation, completely independent of any initial conditions whatsoever. The values reported from these states followed a trajectory that was essentially anchored to a linear value with a minor random variation on top, instead of forming a wave of any possible wavelength, giving a high indication for randomness. These two states' lack of any solvable wavelength makes it likely that some of their data might be falsified, so their predictions were not considered in the third phase (Reverse Chaos Method) as they are therefore likely to contain false data input and, therefore, output.

Additionally, Alaska and Rhode Island suffered similar problems, likely because of the low number of infected. The very low amplitude of the signal made it difficult to discern a pattern.

The number of iterations required to form convergence to a single value for these states was increased to 1000 and then 2000 iterations, but Python was still unable to converge on a value. Upon manual inspection of the raw data for these states, the noise apparent exceeded signal for these values. It was therefore considered unlikely that these numbers would ever converge on a single, true value for these states, so Reverse Chaos was not performed on the states of Florida, Arkansas, Alaska, and Rhode Island.

The states can then have their predictions moderated by the largest waves discovered in that states' numbers. Again, this effect is attenuated the further from the training data the line gets.

This portion of the algorithm as a whole is especially strong if there is a lot of raw data, allowing the system to discount uncorrelated data and magnify data that fits a particular pattern, and is therefore best-suited for large data sets of potentially noisy data, even with very sparse real data. It doesn’t work as well in very small data sets, or with very small populations.

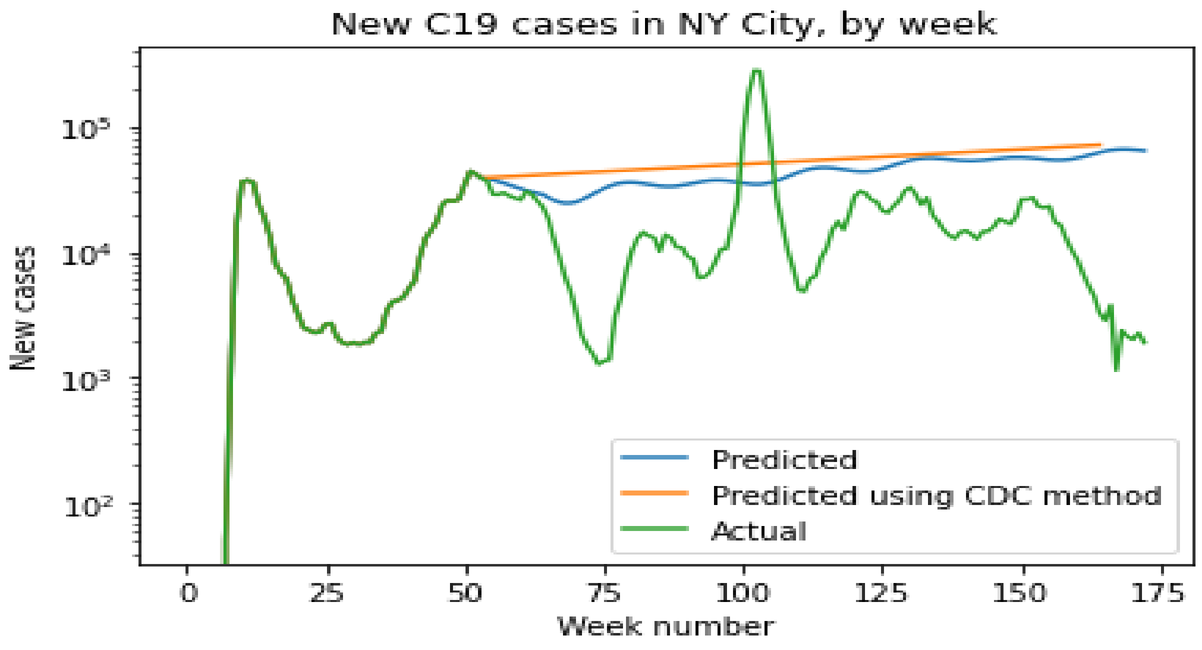


Figure 6: New cases in New York City, including waves, on a log scale, by week

4.3 REVERSE CHAOS METHOD

There were two major wavelength swings. The first swing is around week 50 when, first, the vaccine was distributed, severely increasing the ceiling. However, at about the same time, the Delta variant mutation occurred, drastically raising propagation, as well as evasion in removing some of the newfound ceiling. This might also have been cause of the rapid displacement of one strain over another in that the old strain now had a drastically higher ceiling than the new strain.

The second major swing was just before week 100 when the Omicron variant suddenly spiked cases by some orders of magnitude. Even though a follow-up vaccine was introduced to combat this strain as well, it and future vaccines were not taken by the public as readily as the first, and so other vaccines had far more diminutive effects. The predictions are moderated to account for pre-planned vaccines' effects at week 112 by adding a one-time vaccine impulse downwards of strength 10,000.

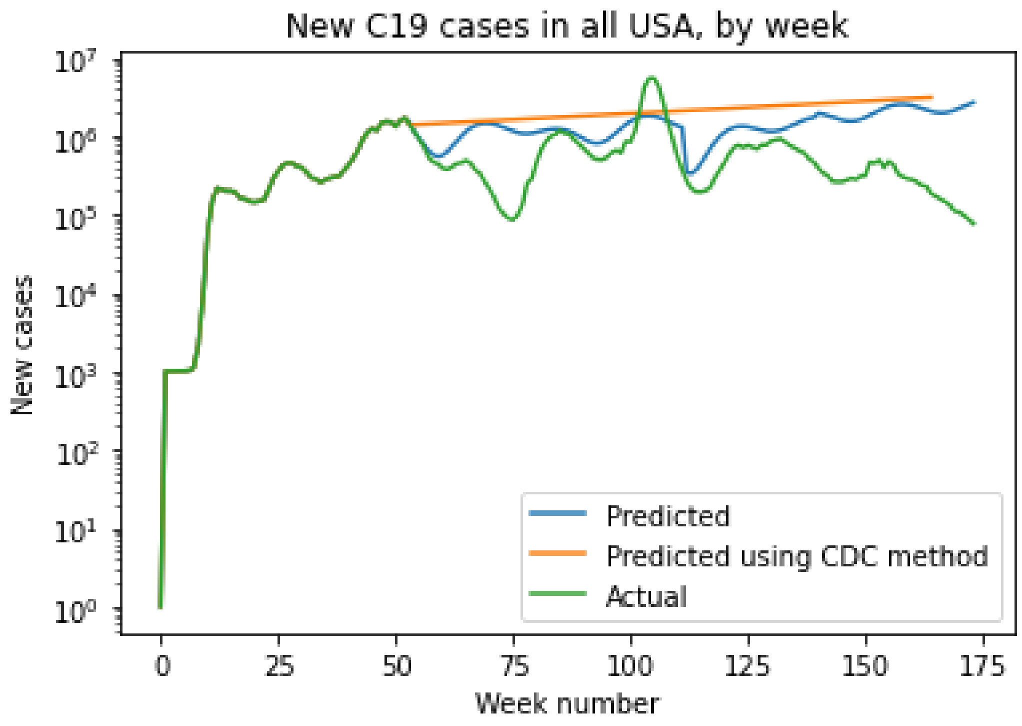


Figure 7: New cases in the entire USA on a logarithmic scale. Spikes occur of different amplitudes around weeks 50 and 100. Reverse spikes occur around weeks 50, 100, and 112.

The reverse chaos method is most useful when a mutation takes place that changes a value mid-wave. There are two main reasons for this to happen: Vaccination, which drastically cuts the wave downward and mutation, which drastically cuts upward. Vaccination doesn’t really need to be predicted, since it is known ahead of time, so the only values that really need to be tracked are when the upward force of the virus that increases suddenly mid-wave. When this happens, a second mini-wave occurs within the bounds of the "regular" wave, and can be tracked with multiple methodologies.

In the current data set, the first instance is vaccination halting the upward flow of the wave, wherein the wave begins to curve downwards, immediately followed by the Delta mutation which halts the downward flow as well. This leads to a flat-top wave plateau, where the upward and downward forces on the wave are roughly equal until the downward force begins to "win" at the end of the wave.

The second instance is the Omicron mutation, where the downward trend suggested by the wavelength, is instead replaced with a drastic upward trend of several thousand percent, completely restarting this wave. However, the same wavelength resumes after this uptick, suggesting the same factors are at play, but simply with a larger number of units of amplitude. This implies that such changes can be modeled very well as a one-time bump to the process.

Multiple embodiments and methodologies can be used when a sub-wave exists in the regular wave. The easiest to program is a sudden uptick in error in the one-wave model; the one-wave model will suddenly see a much worse fit than the model did previously, however this requires setting a baseline of a level of acceptable error, which might not be useful in regions where the data is less complete as it was for COVID-19 in the United States, as some error would be generated by the nature of the incomplete data, leading to false positives.

If viewing the data for only a few regions visually, the easiest way is to simply observe a visual mini-wave, or alternatively, sudden jump in wave intensity (depending on if the mutation occurred at the downhill or uphill portion of the wave, respectively). This works very well if the mutation happens on a single graph of time-series data, however is not practical when measuring input interactions from multiple regions over multiple timelines, (ex. The 60 X 53 matrix creates over 3180 visual-mathematical calculations) however could be done with a large enough workforce or a well-trained machine-learning model.

If such a beat note is found in real-time in "this week's" data, this can provide for a very good opportunity for staying ahead of the mutation and identifying the states where the mutation was present in significant quantities. Because the mutation can be modeled as having occurred identically in all 51 states at different times, the algorithm is agnostic to where the mutation actually occurred for COVID-19, since mutations for this particular virus tended to very quickly become the dominant strain after mutation, displacing other strains of COVID-19; different behavior might be observed for other diseases. Because of this, the disease progression can be modeled such that, at a specific point in time, all copies of the virus in a given state suddenly become copies of a different virus with a better bandwidth, independent of weather the mutation actually happened in that state or whether the mutation was imported from elsewhere.

In the current non-real time data, when a mutation is identified in a state, the state is divided into a pre-mutation and a post-mutation time zone. Since the transposition from a pre-mutation to a post-mutation state takes several weeks while one strain displaces another, there is no focus on attempting to discover "which" week in particular was responsible after the fact. Instead, the wave is cleanly divided into two halves. One of the halves will be "standard", without mutation, and one of the halves will feature the mutation. It was not found necessary in the current data set, but theoretically this process would be repeated to subdivide into quarters, or technically even further. Further subdivisions might not actually create more knowledge, however; it would probably create more accurate predictions of COVID-19 spread to treat two mutations happening back-to-back as a single mutation, mathematically, as any mathematical operation about only one would quickly be drowned out by the propagation of the second since, again, COVID-19 quickly displaces previous iterations of itself in a population. More granular subdivision might produce greater insight for other viruses.

4.4 WHAT WAS LEARNED

Because the current CDC model treats contagion as a logit-like function with exponential growth the new method leads to more accurate leading predications as a quasi-sinusoid, since it predicts what the future R_0 might be, instead of having it need to be updated after-the-fact to fit previous data. As a result, the deviations in the graph can be predicted ahead of time, instead of retrofit to existing data. Here, number of infected = $R_0{}^t$ with R_0 generally around 1.005, as derived at the moment of R(1 year) for units of t = 1 week. This implies for each newly infected, the number of infected in one week will be 1.005 times that. This is the long-form predictive model used by the CDC, as mutations change the amount of contagion frequently throughout the outbreak, so numerical predictions are not published by the CDC that far out. [13]

The figures below describe covid data for the entire US using both the CDC's methodology and the new methodology. The CDC methodology has the additional problem that the values were calculated after the fact, instead of before. Because it does not have that advantage, the new methodology begins to drift further away from actual results as it interpolates more and more beyond its training data. In the first year of interpolation especially, the new method sees much less error than the CDC's method, but decays from this the further it must interpolate.

The cumulative absolute error for the CDC method for 100 weeks of interpolation was 147 million. The cumulative absolute value error for 100 weeks of interpolation for the new method was 89 million, a difference of about 58 million less error. Neither method predicted how severe the Omicron variant spike would be, but the new method was much more accurate for most other times.

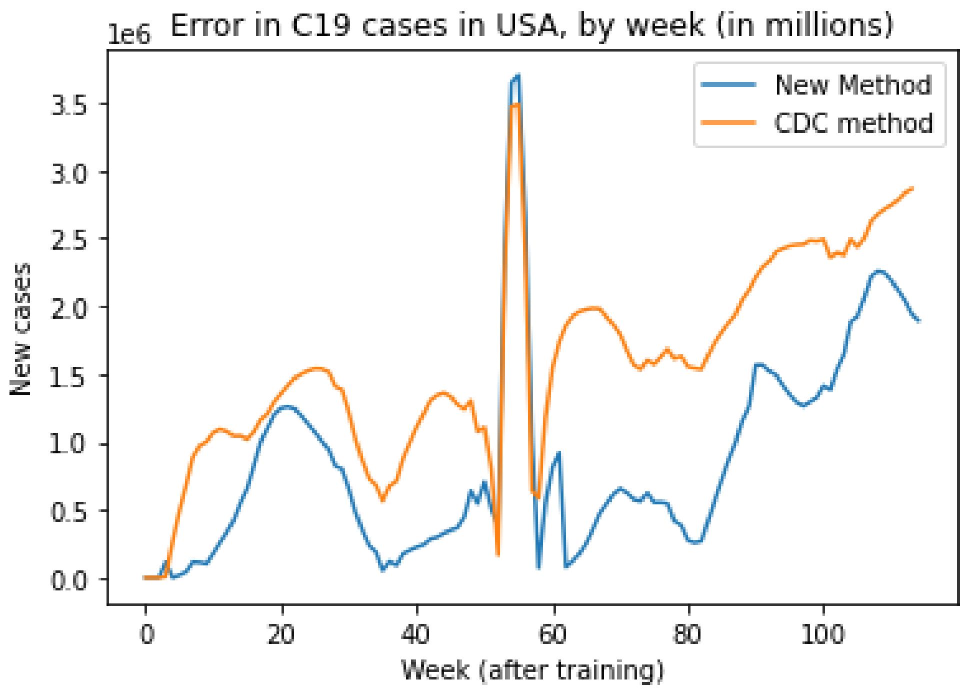


Figure 8: Absolute value of error in new cases in the entire USA using all methodologies, in weeks of interpolating new data. The sum of the absolute value of this error is 147 Million for the CDC method, and 89 million for the new method.

5. Summary and Conclusions, Similar work

5.1 CONCLUSIONS

Because the current CDC model treats contagion as a logit-like function with exponential growth, the new method leads to more accurate leading predications as a quasi-sinusoid, since it predicts what the future R_0 might be, instead of having it need to be updated after-the-fact to fit previous data. As a result, the curves in the graph can be predicted ahead of time, instead of retrofit to existing data. The figures describe covid data for the entire US using both the CDC's methodology and the new methodology. The new methodology begins to drift further away from actual results as it interpolates more and more beyond its training data. In the first year of interpolation especially, the new method sees much less error than the CDC's method, but decays from this, the further it must interpolate.

A continuous look-ahead model might have even greater predictability. If all past data is used as training data and the algorithm is asked to look no more than 25 weeks ahead, a more accurate model could result. However, part of the purpose currently is to form a better long-term prediction method, which is a different methodology than those used currently.

There are a few other limitations of the new methodology. As mentioned earlier, the methodology takes a little bit of information from many, many data points. This works very well even if the real data is relatively sparse in a large amount of noise, but does not work if there are few data points to begin with. This means that a very, very low granularity is not optimal, and works better with bins that are relatively large in comparison to the whole.

This work was done with US data and not global data, but there is no reason a larger data set cannot be used. There may be a problem in places where a significant portion of the population is located in a single capital city or a small number of hub cities, as the remainder of the area might take up most contagion from those areas and not from within or nearby neighbors.

There are many benefits of the new methodology beyond simply a more precise estimate. The algorithm's ability to "heal" around potentially falsified data can be used in data sets where much of the data is similarly incomplete or falsified. The data that is correct will show linear correlations with other data that is also correct. Data that does not linearly correlate with anything will remain an "island" of data and will be shut off from contaminating future results as much as in some other models. This is also a type of drawback with the methodology in that unrecoverable data is instead dropped and "healed around", rather than attempting to brute-force it. Regions where almost all of the data is fabricated will not be able to make very accurate predictions,. Even though the data will not contaminate the data for other, more trustworthy, regions.

An unexpected benefit is the similarity of the waves. Because the wave shapes are similar differential equations among the many states, and both before and after key events, this greatly simplifies the calculations that need to be performed. The wave shape can be isolated and the effects of events can be simple permutations or acceleration factors that can be applied to an individual wave. This benefit may not exist in all possible data sets, however, leading to states with different wavelengths and drastically different phases in some implementations.

5.2 SUMMARY

Pandemics and other natural forces can be modeled with a distinct wave behavior. This wave can be modeled as a second-order differential equation, creating a sinusoid with a repeating pattern. Early sections can allow for a good mapping of the sinusoid, as they are more free of sub-waves that interfere with computation. Once the easy parameters are solved for, the remaining parameters can be solved with this extra information in mind. This sinusoid mapping can then be applied to nonsmooth, chaotic sections, which can be difficult to find a straightforward solution for without the foresight of the easy, smoother section first. Once the chaotic features are determined, the controlling variables can be solved for to a desired level of precision.

Falsified information can be treated as noise, and can be accounted for by determining which features of the data are commonly found in tandem and which features are not in tandem with any other features. Features that show no correlation to anything else in the data might be either random noise or fabricated data. Data points that "play well with others" can be separated from data points that follow a completely different function or rule. In this way, genuine data can be solved for independently of noisy or false data. This allows for an empirical methodology of finding outliers that may not represent actionable information and excluding them early in the calculation, so the remainder of the calculation can be routed around such outliers when a pattern is found. This enables the algorithm to "heal" around missing or false data.

A method of finding such noise is to separate each dependency of the data into different bins. Genuine correlations should be common to many of the bins. Noise should not be common to other noise or genuine data, except extremely rarely. As a result, the genuine data patters becomes increasingly emergent as correlations between data builds up, and noise can be dropped from the pattern.

This is especially useful in "real-time" data where a number of measurements are seen as signal, and are then suddenly dropped all at once in favor of a new set of measurements that is seen as the new second signal, and the old signal is dropped as noise. This implies that two signals are present in the data and the second signal has recently become more powerful than the original signal, and can be the result of an invisible factor in the data that has recently changed, a change that could not be seen directly, but can be seen in the drastic change in behavior of its dependencies. In analyzing the results of the dependencies, situations that cause such behavior change can be predicted, even if the nature of the behavior change itself is not known.

The number of new cases of COVID-19, by week, by state, from the CDC were input and the linear correlations were found therebetween, by finding a linear solution between the number of new cases in a given state versus the number of new cases in all of the states in the previous week, including itself. This provided a good baseline of the "ripple in a pond" effect to map out the traffic between the various (potentially nearby) states determining how the effect of one state affects the other states over time.

The immune resistance to the pandemic caused by immunity and vaccination can then be modeled as a second order differential equation, creating a sinusoidal wave pattern. In this way, each state's immunity over time is a function of the historical population of infected in that state. For COVID-19 in the US, the solutions for all the states were very similar, providing confidence in the model and cross-validation for the methodology.

A number of states not only didn’t fit the numbers, but couldn’t fit any numbers, having no wave behavior at all, causing the algorithm to route around such data. The algorithm under consideration has certain assumptions which were made based on the idea of contagion, which imposes certain conditions on the data. Other types of data may have similar rules that cannot logically be violated. Data that implicitly violates those conditions can be dropped as noise. This could be caused by the population being too low for any signal to overcome the random noise of the process, or by an adversary inputting arbitrary data into the input parameters for that state. However, the algorithm was able to route around both forms of noise similarly, as it detects when these conditions of the data are no longer being met, and so the data is no longer relied upon as much as other data that does meet these conditions.

Sudden, drastic changes in the amount of noise actually provides insight of a potential hidden signal, such as a mutation. The algorithm can then introduce this newly found pattern to fine-tune its assumptions and parameters. The final estimation for the contagion was therefore much more accurate and precise than the generally accepted methodology, which has a great deal of inertia and is slow to change, versus an algorithm that notices drastic changes in data right away. As a result, the new algorithm focused more closely on valid data and away from noise, allowing for a better prediction.

This algorithm of data mining real data in potentially false data, is useful in other types of problems as well. Amng those are difficult physics simulations and types of economic modeling.

5.3 DIFFICULT PHYSICS SIMULATIONS

There are many physics simulations that cannot be feasibly performed because the conditions change at different speeds at different places in the simulation. For example, near black holes, the ‘speed of time’ increases, slowing the relative progression of events. A possible model to this is a time counterforce to the progression of events, as in the above differential equations. As a result, a physics simulation can be remapped onto the surface of a wave, and a calculation in proper time can be mapped to the similar place in the wave. This can be aided by a rainbow table of flagpoints that can be placed along the wave to act as an aid to navigation on the wave, so it can be easier for a computer to determine where, and when, objects are located on the progression of time. This creates a "reverse" differential equation, where time is the dependent variable rather than the independent variable. After this, everything can be dependent upon the new flow of time. For example, by mapping a function onto the surface of an arctangent curve.

The reverse chaos method especially adds an incredible amount of insights to such problems that are numerically unstable, in that a small change on the border results in drastic changes elsewhere, for example supersonic fluids. For example, in the equation $y=1/x$, if 0 is approached from the negative direction, the result decreases without bound, but if it increases in the positive direction, the function increases without bound. Values very near to zero have difficulty being mapped to a single solution, since the result may be a very large, or very negative number. One can take an empirical value of a number of nearby values to determine an approximate location along the function. For example, if my current values is one billion and one, I can recognize that my input value must be between 1/one billion and 1/two billion, and that it could not possibly have been negative 1/two billion. In this way, I can navigate between various flagpoints to determine the location on the wave along both the t and the f(t) axis.

5.4 ECONOMIC MODELING

Other uses for this procedure are also envisioned. The success of the model on very highly numerical data, such as virus propagation, can also provide insights into the use of the model in less empirical circumstances, such as customer service.

There is an inertial effect of someone who regularly shops at one store being familiar with the layout, and would tend not to shop at a nearby store, even if the price were very slightly cheaper, or the customer service is somewhat better. In this case, the customer only switches to the other store if the expected difference in customer experience and prices at the new store are greater than that individual customer's inertial effects that are anchoring the customer to the old store. As a result, multiple nearby stores can drop customer service or raise prices in tandem; as long as the increase is slow and gradual enough, a condition state is never reached such that a given store's deviation from the surrounding stores' value is greater than the customer's inertial hesitancy to switch stores. The minimum customer service a store must provide is thus heavily dependent on the nearby stores that sell similar items. One store can raise prices or cut service by (amount less than inertia constant), causing other vendors to also cut service and raise prices by (amount less than inertia constant), bringing them back into par. This has important connotations in economic modeling, creating a natural monopolistic force even in absence of a single monopoly.

The inertial effect or monopolistic pressure can be calculated as the amount of price increase that is finally necessary to pull away a given amount of custom from nearby establishments. This is the customer's switching inertia. Measuring this switch inertia can be useful in antitrust policies to determine the amount of monopolistic pressure a given entity emits, empirically. This measurement of the amount of price or value difference required to get a customer to switch stores is proportional to the ability of the store's magnetism to keep customers at that store. In a True Monopoly, this value is infinite. In a perfectly level system, this inertia value is 0.

For example, there is a pressure force on gas prices that is applied to a station from the adjacent stations. People may go next door to save ten cents on gas who would be hesitant to drive all the way across town to save the same amount. As a result, the rate of change of gas prices is heavily "contagious" from adjacent gas stations; gas stations have a tendency to somewhat match nearby stations. If some gas stations are far away from any others, this pressure does not exist, causing "islands" of high gas prices to develop, independent of the wholesale price of fuel or locality taxes, as island gas stations can get away with charging more, without customers fleeing to an adjacent station. The gas station that is far distant from any other can raise prices quite significantly before it finally becomes worthwhile to drive all that distance. In addition, if there is a peninsula of gas stations, the price may change, but very slowly, as there is a large propagation delay across the peninsula, leading to a large price gradient across the peninsula, as a wave of prices gradually travels down the peninsula.

Alternatively, this differential equation can be used in price management routines to see how much a given price would influence not only the demand curve in a single step, but how much a price increase or decrease would affect other vendors of the product, and what the final state of such a price change would be, both in the number of customers and the profit so derived.

For example, a gas station raising prices by x cents per gallon may not reduce the demand curve, as other gas stations can respond similarly, allowing all such stations to increase profits.

Similarly, reducing prices can lead to a "race to the bottom", as other competitors match the lower price. Similarly, cutting costs by reducing product quality may trigger a similar race to the bottom, as other vendors cut quality by the same amount to match prices. As a result the quality/price curve is not as sharp as it may first appear, as not only customers, but other providers will change to match.

The price of gas can be modeled as a system of linear equations, where each gas station is one node and has a listed price for regular unleaded gas. The price of gas at that station the next week is then proportional to the price of gas at that station the previous week, as well as the price of adjacent gas stations in the previous week; if the price of other gas stations is higher, the station may raise their prices to compensate. If the price is lower, the price may similarly lower to compensate, to avoid driving off business. This causes a sudden change in gas prices to “spread” across the map over the course of several weeks. The addition of an “Outside” value of the price of gas to account for a global increase in raw materials can also be added, similar to the "Outside" state in this thesis. The price of gas in several gas stations can then be tracked and a vector of linear coefficients can be determined between the various gas stations’ prices to determine the ability of each one to influence the other. If one station finds itself in an island or peninsula, that can be evidence that that station would be better off raising prices to take advantage of a captive audience (such as at an airport). Similarly, if a station realizes that a number of stations are dependent on its prices, that may provide an opportunity to either slowly raise prices in tandem, or undercut competitors with a deep, sudden price cut, depending on circumstances.

A similar methodology can be used to monitor monopolistic practices of a small number of vendors of a product. For example, both Coke and Pepsi have seen high profits as of late by both increasing prices. The ability to switch to the adjacent vendor on the price “island” is limited by the only other common vendor on the “island”. If both vendors raise prices in tandem, there would be little switching from one to the other due to price.

Additionally, if changing prices by one gas station does not effect a second gas station in any way, it can be inferred that the stations are on separate islands, and are therefore not true competitors, as there is nothing a gas station on one island can do to draw customers away from the second island. This can be a sign by anti-trust regulators to determine local monopolies that have drastically different service than nearby alleged competitors, that no one is actually capable of switching to.

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        import scipy as sp

        from datetime import datetime as dt
        from numpy import nan as NaN
        from sklearn.linear_model import LinearRegression

        import os
        import math

        print("done")
```

```
done
```

```
In [2]: %pwd
        #Where am I?
```

```
Out[2]: 'C:\\Users\\admin\\anaconda3\\aThesis'
```

```
In [3]: #Reading data and patitioning it to the relivant variables.
        rawDataSet = pd.read_excel('c19.xls', sheet_name = '19')

        # 50 states + DC = 51 "states"
        state = rawDataSet["state"]
        valuesOfDataSet = rawDataSet["new_cases"]
        deaths = rawDataSet["new_deaths"]
        endDate = rawDataSet["end_date"]

        print("Reading data")
```

```
Reading data
```

```
In [4]: fig = plt.figure()
        plt.show()
        #test
```

```
<Figure size 432x288 with 0 Axes>
```

```
In [5]: maxTime = 173
        # There are 173 weeks of data in the CDC data set of 2020-01-23 thru 2023-05-10

        maxStates = 60
        # 50 "real" states, 1 district (DC), 1 NY City, and 8 of territories, etc.
```

In [6]:
```python
#initialization for for loop vars
j = 0
t = 0
yValues = np.ones(99)
finalXvalues = np.ones(61)
thousand = np.ones(maxTime) * 1000
thousandRow = pd.DataFrame(thousand).transpose()

# reshape as a 60 X 173 array
states = valuesOfDataSet.values.reshape(maxStates, maxTime)
statesNames = state.values.reshape(maxStates, maxTime)
statesNames = statesNames[:, 0]
states = pd.DataFrame(states)

#add the names of the states as the column headers
states = states.transpose()
states.columns = statesNames
states = states.transpose()

# An extra 61 node for Outside the US
states = pd.concat([states, thousandRow])
# Note: the Outside row is the state of " 0 ", which looks like "O", which is a good


states
# 60 State object Rows, 173 weeks in columns.
# Week 0 is the first week of the pandemic; week 172 is the final week.
# This comes out to 0- 172 instead of 1- 173.
```

Out[6]:

	0	1	2	3	4	5	6	7	8	9	...	163	164
AK	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	11.0	52.0	...	582.0	448.0
AL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	58.0	411.0	...	2471.0	1890.0
AR	0.0	0.0	1.0	3.0	1.0	3.0	0.0	8.0	70.0	279.0	...	1448.0	1240.0
AS	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
AZ	0.0	1.0	0.0	0.0	0.0	0.0	0.0	8.0	18.0	374.0	...	3220.0	4892.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
WA	1.0	0.0	0.0	0.0	0.0	0.0	44.0	329.0	848.0	1644.0	...	4042.0	4020.0
WI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	108.0	510.0	...	4178.0	3856.0
WV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	38.0	...	1414.0	1412.0
WY	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	18.0	34.0	...	281.0	257.0
0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	...	1000.0	1000.0

61 rows × 173 columns

```
In [7]: oldSize = 53
        interpolateDistance = 119

        currentStates = states.loc[:, :oldSize]
        yValues = (states.loc[:, 1:oldSize+1])
        finalXvals = ()

        #includes internal spread (state itself), however, Outside, the 61st outside “state”,
        for j in range(maxStates): # loop in states, j= 1 to 60: each in states
            #starting at t=2, for each state, y= that state’s value at t,
            #and is a linear combination of the values at all 60 nodes at t-1:
            #Y =  X1*s1 + X2*s2, … X52 * s60 for some X values 1- 60.
            #This is a basic linear line of best fit, and is actually only 1 line of code whe
            regress = LinearRegression(fit_intercept = False, positive = True)
            tempHold = currentStates.transpose()
            tempHold.columns = tempHold.columns.astype(str)
            re = regress.fit(tempHold, yValues.iloc[j, :] )
            XvaluesIteration = re.coef_

            # Not doing it this way any more
            #finalXvalues = math.linearBestFit(yValues, states)
            #finalXvalues = states.index[j], re.coef_, re.intercept_

            #The names of each state, values
            print(states.index[j], XvaluesIteration)
            finalXvals = np.append(finalXvals, XvaluesIteration)

            #tempHold = states.iloc[: , oldSize:oldSize + interpolateDistance]
            #tempHold = tempHold.transpose()
            #tempHold = re.predict(tempHold)
            #print(tempHold)
            #futureStates = pd.concat([tempHold, futureStates])
```

```
AK [4.33840852e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 3.72566559e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 4.16275469e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.79590545e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 5.81967986e-02 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 4.41529380e-05 0.00000000e+00 1.52180652e-02
 0.00000000e+00]
AL [3.09769083e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 7.91040352e-02 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 1.37814333e-02 2.86551257e-02 0.00000000e+00
 0.00000000e+00 9.34387804e-01 0.00000000e+00 0.00000000e+00
 1.74841917e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 1.36689531e-02 0.00000000e+00
 1.66340977e+02 2.61684336e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.09071165e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 2.12927772e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 2.13658345e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 6.47464817e-01]
AR [0.00000000e+00 5.74572992e-02 5.55928964e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 9.65968076e-03 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 3.38371502e+01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.96025058e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 4.29747034e-02 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 7.70983793e-02 4.89013421e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 1.77988320e-01 0.00000000e+00
 2.52210742e-01]
AS [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
AZ [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 6.76681294e-01 2.57057104e-02 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 6.85015368e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.72266641e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 7.59569859e-01 8.50017033e+02
 0.00000000e+00 0.00000000e+00 1.84653016e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 4.19946659e-01]
CA [0.         0.         0.         0.         0.33796258 0.43902781
 0.         0.         0.         0.09782357 0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.97688814 0.         0.         0.         0.
 0.         0.         1.4591806  0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.        ]
CO [0.         0.         0.         0.         0.         0.
 0.33576748 0.         0.         0.         0.         0.
 0.         0.         0.         0.13530653 0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.08496784 0.         0.         0.
 0.         0.         0.         0.01574976 0.         0.
 0.         0.         0.         0.         0.         0.
 0.         1.08451019 0.         0.         0.         0.
 0.         0.         0.         0.         0.         1.97079352
 0.        ]
```

```
CT [0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.09932846 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.13419056 0.         0.         0.
 0.         0.         0.         0.         0.72265285 0.
 0.         0.15030951 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.        ]
DC [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 5.16381211e-01 0.00000000e+00 5.33220397e-05 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 4.27326283e-03 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 8.24144783e-03
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 5.66567898e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 9.28862477e-02]
DE [0.         0.         0.         0.         0.         0.00325545
 0.         0.         0.         0.04492638 0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.00593024 0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.0033142  0.         0.         0.
 0.0295854  0.         0.         0.         0.48047137 0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.07589609]
FL [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 4.37308089e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 6.66413033e-01 0.00000000e+00
 0.00000000e+00 1.64303356e-01 0.00000000e+00 0.00000000e+00
 5.25694292e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 2.03352053e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 3.11467389e+03
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.86861224e+00]
FSM [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 8.96519695e-08 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 4.85899265e-06 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 1.26132665e-07 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00]
GA [0.00000000e+00 8.20484758e-01 0.00000000e+00 0.00000000e+00
 1.11773142e-01 2.03572851e-02 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 1.15174545e-01 0.00000000e+00
 8.63021666e-03 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 6.29878395e-02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 5.28781882e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.71690294e+03
 3.24294683e-02 0.00000000e+00 5.44822530e-02 7.07531138e-03
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 9.03481647e-01]
GU [0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.73686895 0.04153216 0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         4.26294809 0.
 0.         0.         0.00476813 0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.        ]
HI [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 6.71377579e-04 0.00000000e+00
 0.00000000e+00 1.62908279e-02 6.75553788e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.00485116e+01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 4.76945214e-03 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 3.87262119e+01
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 2.34005096e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 4.86093515e-03]
IA [0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.07437522 4.89885033 0.         0.11564757 0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         1.06835334 0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.66032224 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.14746971]
```

```
ID [8.67135329e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 1.62457553e-02 0.00000000e+00
 0.00000000e+00 1.04841094e+00 0.00000000e+00 0.00000000e+00
 1.73981144e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 2.45767295e+01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.19129758e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 2.23380371e-01 1.50644685e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.35325268e-01]
IL [0.         0.         0.         0.         0.         0.
 0.         0.         2.73045171 0.         0.         0.
 0.1054787  0.         0.         0.         0.         0.25329707
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         1.45878765 0.         0.         0.
 0.         0.         0.         0.05105017 0.         0.
 0.         0.         0.         0.         0.32846927 0.
 0.         4.05689125 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.31914274]
IN [1.83866889e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 3.37762888e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 1.74515217e+00 0.00000000e+00 8.28176239e-02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 9.67826177e-03 0.00000000e+00 0.00000000e+00
 1.41327593e+01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 1.17471579e+00 0.00000000e+00
 0.00000000e+00 9.77808457e-01 1.33350624e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 2.38681797e-01 0.00000000e+00
 0.00000000e+00]
KS [0.08858036 0.         0.         0.         0.         0.
 0.1222471  0.         0.         0.         0.         0.
 0.05027515 0.82264582 0.         0.09496475 0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.79290342 0.06818557 0.         0.         0.
 0.         0.         0.         0.         0.14478736 0.
 0.        ]
KY [0.00000000e+00 0.00000000e+00 3.85801830e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 5.53916098e-03 0.00000000e+00
 0.00000000e+00 1.85842928e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 4.58066528e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 9.77430232e-02 0.00000000e+00 5.45444423e-01 2.21662183e+02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 1.14243235e+00 0.00000000e+00
 0.00000000e+00]
LA [0.         0.         0.         0.         0.07996661 0.
 0.         0.         0.         0.         0.02963014 0.
 0.         0.         0.         0.15898998 0.         0.
 0.         0.         0.         0.44496871 0.         0.
 0.         0.00534616 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.01175926 0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.03121161 0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.79208377]
MA [ 0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.         27.35892033  0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.28041119  0.03141124  0.          0.
  0.          0.          0.          0.          2.25645764  0.
  0.          0.          0.06276643  0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.        ]
MD [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 2.63721064e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.33143817e-01
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 1.30838661e-01 0.00000000e+00 7.79377714e-02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 2.01509568e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 9.06680415e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 3.06500844e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 2.25673332e-03 0.00000000e+00
 3.93755853e-01]
ME [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 1.22720279e-03 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.24311507e+02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 5.15789417e-02 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 2.49201844e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 4.47794185e-03 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 1.37275669e-01 0.00000000e+00
 0.00000000e+00]
```

```
MI [0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.10410175 0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.43825002 0.         0.         0.         0.
 0.         0.         1.45065234 0.         0.         0.
 0.         0.         0.         0.063103   0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.28466062 0.         0.         0.         0.
 0.         0.         0.         0.         0.         1.9347288
 0.20833302]
MN [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 2.63041863e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 5.32172326e-01
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 8.46773963e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.65043501e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 6.86083314e+02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00]
MO [1.12197893e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.14447163e-01 5.29819220e+00 0.00000000e+00 1.74392916e-02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 5.86448186e+01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.63269462e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 9.69110113e-02 2.00279782e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00]
MP [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 4.04304932e-06 0.00000000e+00
 0.00000000e+00 3.61475829e-03 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 1.14112943e-05 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 8.69850322e-06
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 1.79191666e-05 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.42423894e-03]
MS [0.00000000e+00 4.28720909e-03 2.28530908e-02 0.00000000e+00
 9.47862400e-04 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 2.70598155e-01 4.19116600e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.17863439e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 4.81095940e-03 0.00000000e+00
 4.95576530e+01 1.31617635e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 4.66920728e-02 1.46597096e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 8.04725926e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 4.24200585e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 5.14181930e-01]
MT [0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         1.69034837 0.         0.         0.         0.
 0.         0.         0.01118974 0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.13338696 0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.28174607 0.         0.         0.         0.
 0.         0.         0.         0.06572043 0.         0.
 0.        ]
NC [0.00000000e+00 0.00000000e+00 1.16086143e+00 0.00000000e+00
 7.24221721e-02 7.51834633e-03 0.00000000e+00 0.00000000e+00
 0.00000000e+00 7.69653993e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 4.31417773e+01 7.30080745e-02 0.00000000e+00 2.02093700e-01
 0.00000000e+00 0.00000000e+00 5.82685500e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 5.24700367e-02 0.00000000e+00 4.93167048e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 6.17889854e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 6.10261893e-02 0.00000000e+00
 4.13484180e-01]
ND [0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         4.69025155 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         2.74284778 0.
 0.         0.         0.25927085 0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.36085167 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.        ]
NE [0.         0.         0.         0.         0.         0.
 0.         0.         0.16591866 0.         0.         0.
 0.02127342 0.         0.         0.01932181 0.         0.02065921
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.06764903 0.04593345 0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         1.0833431  0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.0893231 ]
```

```
NH [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 3.34727588e-03 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.78573083e+02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 4.21115103e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 2.17190874e-03
 0.00000000e+00 0.00000000e+00 2.19457127e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 1.35389757e-01 0.00000000e+00
 0.00000000e+00]
NJ [0.         0.         0.         0.         0.         0.
 0.16226932 0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.15488539 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.08038639 0.52737166 0.         0.
 0.         0.         0.         0.         0.90812213 0.
 0.         0.72866887 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.        ]
NM [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 7.12750102e-03 8.97631318e-06 1.62091469e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 3.54898847e-02 1.42866991e-01 0.00000000e+00 0.00000000e+00
 7.94713690e-02 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 3.23509406e-02 1.27150207e+03
 0.00000000e+00 1.37614774e-01 4.46217725e-03 0.00000000e+00
 8.20758196e-02 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 7.41908080e-02 0.00000000e+00
 0.00000000e+00]
NV [1.19689027 0.         0.         0.         0.02659907 0.
 0.         0.         0.         0.         0.0222601  0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.1182556  0.         0.
 0.         0.         0.1201737  0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.08246552 0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.0564439 ]
NY [0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.06236821 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.45061854 0.3029183  0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.00299471 0.         0.         0.
 0.         0.         0.         0.         2.13383046 0.
 0.02458514]
NYC [0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.87341923 0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.62621572 0.
 0.68045929]
OH [2.70795275 0.         0.         0.         0.03295101 0.
 0.67700651 0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.15107002 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.7834657  0.
 0.         0.         0.19170376 0.         0.         0.
 0.         0.         0.         0.         1.46761989 0.
 0.        ]
OK [0.00000000e+00 0.00000000e+00 3.97232615e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 1.81170587e-05 0.00000000e+00
 3.98872085e-03 8.71514018e-01 0.00000000e+00 6.31417131e-02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 8.44550128e-02 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 5.18122673e-02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 3.25675452e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 3.52073896e+02
 2.93674651e-02 0.00000000e+00 9.31112729e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 7.11487908e-01 0.00000000e+00
 0.00000000e+00]
OR [0.48206416 0.         0.11273074 0.         0.00791134 0.
 0.05597792 0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.0279397  0.         0.
 0.         0.         0.02522256 0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.10193906 0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.01861357 0.00182314 0.         0.
 0.         0.         0.         0.         0.         0.
 0.02264588]
PA [1.22382397 0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.3455752  0.         0.         0.         0.
 0.         0.         0.         0.         2.51702382 0.
 0.         0.         0.00284436 0.         0.         0.
 0.05874878 0.12090951 0.         0.         2.39963416 0.
 0.         0.         0.09720822 0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.        ]
PR [1.44450921e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 3.24898491e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 3.63790698e-02
 1.11990999e+01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 2.90797957e-03 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 5.61696096e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 3.85408429e+00 0.00000000e+00
 0.00000000e+00 2.85458222e-02 0.00000000e+00 0.00000000e+00
 0.00000000e+00]
PW [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
RI [ 0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.05688679  0.          0.         11.28600904  0.
  0.          0.          0.          0.          0.          0.
  0.01973239  0.          0.02334211  0.          0.          0.
  0.          0.          0.02135015  0.          0.3672282   0.
  0.          0.12428071  0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.        ]
RMI [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 9.08331850e-04 0.00000000e+00 2.75767036e-06
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00]
SC [0.00000000e+00 0.00000000e+00 2.73071315e-01 0.00000000e+00
 2.05616382e-01 8.56979307e-03 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 3.68283912e-03 7.16783215e+02
 0.00000000e+00 1.05823811e+00 1.16194157e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 2.63985481e+01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 3.36820279e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 5.00396850e-01]
SD [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 3.55679007e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 4.77398618e+01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 6.94786826e-02 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 6.17611066e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 5.51658417e-03 0.00000000e+00 0.00000000e+00
 0.00000000e+00]
TN [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 7.18809198e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.59361617e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.77262367e+02 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 4.33340299e-01 0.00000000e+00 0.00000000e+00
 5.05012555e-02 0.00000000e+00 0.00000000e+00 6.78637906e-02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 2.41031785e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 3.27562710e-02]
TX [0.00000000e+00 0.00000000e+00 2.38400317e+00 0.00000000e+00
 4.88781858e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 3.08269121e-01 0.00000000e+00
 5.50421215e-02 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 3.63256804e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 9.46913629e-02 0.00000000e+00
 3.08370923e+01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 2.10774875e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 2.60402572e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 4.17513894e+03
 0.00000000e+00 0.00000000e+00 4.76272474e-01 4.59288089e-02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.04822641e+00]
UT [0.         0.         0.28094675 0.         0.0048118  0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.14623845 0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.0428161  0.
 0.         1.09080658 0.02579313 0.         0.22771938 0.
 0.         0.         0.         0.         0.23994446 0.
 0.09385827]
VA [0.00000000e+00 0.00000000e+00 1.83746504e-01 0.00000000e+00
 0.00000000e+00 1.20029272e-02 0.00000000e+00 0.00000000e+00
 1.43108667e+00 6.31300343e-01 8.03597337e-03 7.42739259e+03
 2.68443364e-02 8.22713568e-01 1.06963929e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 7.05170079e-02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 6.35009419e-02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 2.46452289e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 1.47921529e-01 4.62293811e+02
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 2.50369293e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 2.42022320e-01 0.00000000e+00
 1.46613370e-01]
VI [2.93690556e-03 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 3.36126037e-04 5.63842192e+01
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 3.48592554e-05 0.00000000e+00
 1.17845812e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 6.66384329e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.49572637e-03]
VT [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.72725640e-04 0.00000000e+00 2.51078796e-03 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 3.26700447e+01
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 7.89052826e-03 0.00000000e+00 0.00000000e+00 0.00000000e+00
 7.86304991e-02 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 5.23747256e-03 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 6.52980858e+01
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.20243637e-01
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00]
WA [0.00000000e+00 0.00000000e+00 1.85553506e-01 0.00000000e+00
 3.49100841e-02 0.00000000e+00 3.21678621e-01 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 1.05724678e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.86334857e+02
 0.00000000e+00 0.00000000e+00 5.27148593e-02 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 1.53895910e-02 0.00000000e+00
 4.93828161e-01]
```

```
WI [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 6.82420380e-02 1.09370789e+01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.13806272e+02 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.17453479e-02 3.38599002e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 1.18906901e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00]
WV [0.191932   0.         0.         0.         0.         0.
 0.01273428 0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.00373553 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.04076951 0.
 0.         0.04965205 0.0269017  0.         0.         0.
 0.         0.         0.         0.         0.67623847 0.
 0.        ]
WY [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.05712190e-03 6.62034908e-02 0.00000000e+00 2.12079707e-03
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 1.42883030e-02 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.05569605e+02
 0.00000000e+00 3.10306663e-01 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 3.19591640e-03 0.00000000e+00 3.01662937e-01
 0.00000000e+00]
```

In [8]:
```
finalXvals = finalXvals.reshape(60, 61)# 60 STATES HAVE 61 ENTRIES EACH (61's 'Outsic
# to each of themselves, all other states, and the Outside state, for each of the 60
# this is correct, x[1] has 61 entries. They are the same as state 1.
finalXvals = np.array(finalXvals)

futureStates = pd.DataFrame(currentStates).copy()
newLine = range(61)
thousand = [1000]
thousand = pd.DataFrame(thousand)

# Prediction loop, for future distance of (interpolatDistance)
for t in range(oldSize, interpolateDistance + oldSize):
    tempVector = futureStates.iloc[:, t] # All of the values of all 61 states at the

    # dot product vector with matrix finalXvals to make the prediction.
    newLine = np.dot( np.array(tempVector), finalXvals.transpose() )
    newLine = pd.DataFrame(newLine)
    newLine = pd.concat([newLine, thousand]) # add the thousand outside vector elemen

    #print(futureStates.shape)

    futureStates = np.concatenate((futureStates, newLine), axis = 1)
    futureStates = pd.DataFrame(futureStates)

futureStates.head
```

Out[8]:
```
<bound method NDFrame.head of         0       1       2       3       4       5
6       7       8    \
0      0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0    11.0
1      0.0     0.0     0.0     0.0     0.0     0.0     0.0     3.0    58.0
2      0.0     0.0     1.0     3.0     1.0     3.0     0.0     8.0    70.0
3      0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
4      0.0     1.0     0.0     0.0     0.0     0.0     0.0     8.0    18.0
..     ...     ...     ...     ...     ...     ...     ...     ...     ...
56     1.0     0.0     0.0     0.0     0.0     0.0    44.0   329.0   848.0
57     0.0     0.0     0.0     0.0     0.0     0.0     0.0     3.0   108.0
58     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     1.0
59     0.0     0.0     0.0     0.0     0.0     0.0     0.0     2.0    18.0
60  1000.0  1000.0  1000.0  1000.0  1000.0  1000.0  1000.0  1000.0  1000.0

         9  ...           163           164           165           166  \
0     52.0  ...   4390.482430   4414.549648   4438.565733   4462.530793
1    411.0  ...  36217.297658  36411.162160  36604.614797  36797.656443
2    279.0  ...  23860.313714  23988.857075  24117.127346  24245.125105
3      0.0  ...      0.000000      0.000000      0.000000      0.000000
4    374.0  ...  70574.955499  70967.668398  71359.546990  71750.593040
..     ...  ...           ...           ...           ...           ...
56  1644.0  ...  24493.358588  24624.646651  24755.655788  24886.386592
57   510.0  ...  44801.079935  45042.158516  45282.724905  45522.780189
58    38.0  ...  11699.195045  11764.633607  11829.933145  11895.093952
59    34.0  ...   4207.021603   4229.788727   4252.507480   4275.177965
60  1000.0  ...   1000.000000   1000.000000   1000.000000   1000.000000

             167           168           169           170           171  \
0    4486.444937   4510.308274   4534.120911   4557.882957   4581.594518
1   36990.287967  37182.510240  37374.324130  37565.730504  37756.730226
2   24372.850930  24500.305398  24627.489083  24754.402561  24881.046406
3       0.000000      0.000000      0.000000      0.000000      0.000000
4   72140.808310  72530.194561  72918.753551  73306.487033  73693.396760
..           ...           ...           ...           ...           ...
```

```
56  25016.839652  25147.015557  25276.914897  25406.538258  25535.886228
57  45762.325457  46001.361795  46239.890283  46477.912004  46715.428034
58  11960.116322  12025.000550  12089.746927  12154.355747  12218.827302
59   4297.800284   4320.374540   4342.900836   4365.379272   4387.809952
60   1000.000000   1000.000000   1000.000000   1000.000000   1000.000000

             172
0    4605.255703
1   37947.324161
2   25007.421189
3       0.000000
4   74079.484480
..           ...
56  25664.959391
57  46952.439450
58  12283.161883
59   4410.192976
60   1000.000000

[61 rows x 173 columns]>
```

In [9]:
```
predict = futureStates.copy()
predict = predict.loc[:, oldSize: (oldSize + interpolateDistance) ]
actual = states.loc[:, oldSize: (oldSize + interpolateDistance) ]
difference = predict.values - actual
print(difference)
```

```
     53            54            55            56            57            58   \
AK  0.0     86.699611     -4.457053    171.075491    110.042695    159.313441
AL  0.0    697.249744   5198.265478   8665.068476   8364.005567   7897.941692
AR  0.0     -5.046033   1231.996000   5593.755156   6314.955709   1851.176476
AS  0.0      0.000000      0.000000      0.000000      0.000000      0.000000
AZ  0.0  13449.257309  16067.166989  22855.806708  21135.607426  22144.834657
..  ...           ...           ...           ...           ...           ...
WA  0.0   -186.983570   2032.259982   3727.152758   3081.250508   3117.223618
WI  0.0   1166.127387   3110.693431   4787.567652   6259.296421   7285.480320
WV  0.0    656.276888   1357.647467   1995.384076   2162.359572   1976.191352
WY  0.0     39.621452    122.784276    209.736505    170.724305    478.858457
0   0.0      0.000000      0.000000      0.000000      0.000000      0.000000

             59            60            61            62   ...           163  \
AK   186.515580    259.138255    314.927161    162.471057  ...   3808.482430
AL 10218.852879   6114.728876  10798.920200  11256.401316  ...  33746.297658
AR  6758.128106   6837.932641   7406.997641   7665.310531  ...  22412.313714
AS     0.000000      0.000000      0.000000      0.000000  ...      0.000000
AZ 19752.887515  21842.476925  22464.615306  21982.372677  ...  67354.955499
..          ...           ...           ...           ...  ...           ...
WA  3833.921592   3179.349151   3213.804612   1672.773288  ...  20451.358588
WI  8901.895362   9925.526886  10452.124487  10784.870184  ...  40623.079935
WV  2019.696816   1576.767069   1238.316784    839.742130  ...  10285.195045
WY   604.627897    730.616860    783.807104    898.449425  ...   3926.021603
0      0.000000      0.000000      0.000000      0.000000  ...      0.000000

            164           165           166           167           168  \
AK  3966.549648   3942.565733   3948.530793   4085.444937   4152.308274
AL 34521.162160  34642.614797  34733.656443  34970.287967  35550.510240
AR 22748.857075  23127.127346  23464.125105  23454.850930  23786.305398
AS     0.000000     -1.000000     -3.000000     -2.000000      0.000000
AZ 66075.668398  68708.546990  68487.593040  69209.808310  68855.194561
..          ...           ...           ...           ...           ...
WA 20604.646651  21181.655788  21662.386592  21746.839652  22718.015557
WI 41186.158516  41539.724905  41391.780189  42279.325457  43307.361795
```

```
WV  10352.633607  10801.933145  10741.093952  10943.116322  11057.000550
WY   3972.788727   4039.507480   4073.177965   4166.800284   4162.374540
0       0.000000      0.000000      0.000000      0.000000      0.000000

             169           170           171           172
AK   4173.120911   4277.882957   4393.594518   4406.255703
AL  35814.324130  36051.730504  36292.730226  36650.324161
AR  23576.489083  24041.402561  23983.046406  24350.421189
AS     -3.000000      0.000000      0.000000     -2.000000
AZ  69510.753551  69300.487033  70487.396760  71471.484480
..           ...           ...           ...           ...
WA  22885.914897  23141.538258  23726.886228  24095.959391
WI  43905.890283  44945.912004  45169.428034  45428.439450
WV  11410.746927  11583.355747  11955.827302  12089.161883
WY   4227.900836   4225.379272   4258.809952   4044.192976
0       0.000000      0.000000      0.000000      0.000000

[61 rows x 120 columns]
```

In [ ]:

In [10]:
```
#Similar Waves phase
```

In [11]:
```
# find internal contageon = each state from end of training Xn * X

autoCorrelate = list()

for j in range(maxStates): # for each state, grab its correlation with itself in the
    autoCorrelate.append(finalXvals[j, j] )

autoCorrelate.append(0) # for the outside state

autoCorrelate
```

Out[11]:
```
[0.4338408517134921,
 0.0,
 0.5559289639052066,
 0.0,
 0.6766812941736081,
 0.439027806633437,
 0.3357674789919865,
 0.0,
 0.516381210512584,
 0.04492637588811812,
 0.6664130333739763,
 0.0,
 0.008630216656186242,
 0.7368689523066959,
 0.6755537884749124,
 0.11564756933637281,
 0.17398114397538755,
 0.2532970719053044,
 0.0,
 0.0,
 0.0,
 0.44496870703847313,
 0.0,
 0.07793777143841489,
 0.0515789417451891,
 0.43825001991080337,
 0.08467739634533024,
 0.0,
 0.0,
 0.13161763460178547,
 0.13338695548649157,
 0.20209369971255745,
 0.2592708539806615,
 0.04593345149281187,
 0.4211151034566458,
 0.0,
 0.07947136904516236,
 0.0,
 0.4506185447127101,
 0.8734192290484437,
 0.0,
 0.0,
 0.0,
 0.12090951203043761,
 0.0,
 0.0,
 0.3672281951985964,
 0.0,
 0.33682027853552216,
 0.6176110659653083,
 0.2410317854078414 8,
 0.04592880885971314,
 0.2277193782746182,
 0.25036929298173327,
 0.6663843288595278,
 0.12024363729798618,
 0.0,
 0.11890690127342535,
 0.676238471657536,
 0.30166293731430227,
 0]
```

In [12]:
```
print(oldSize)
```

```
53
```

In [13]:
```
tot = list()
for j in range(oldSize + 1):
    newLine = np.multiply(autoCorrelate, states.iloc[:, j] )
    tot.append(newLine)

tot = pd.DataFrame(tot)#reshape(118,61)
tot.head()

# not being used:
# tempVector = futureStates.iloc[:, t] # All of the values of all 61 states at the cu
# dot product vector with matrix finalXvals to make the prediction.
```

Out[13]:

	AK	AL	AR	AS	AZ	CA	CO	CT	DC	DE	...	TX	UT	VA	VI	VT	WA
0	0.0	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.0	0.224632	...	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.000000	0.0	0.676681	0.878056	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.555929	0.0	0.000000	2.634167	0.0	0.0	0.0	0.179706	...	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	1.667787	0.0	0.000000	3.512222	0.0	0.0	0.0	0.044926	...	0.0	0.0	0.0	0.0	0.0	0.0

	AK	AL	AR	AS	AZ	CA	CO	CT	DC	DE	...	TX	UT	VA	VI	VT	WA
4	0.0	0.0	0.555929	0.0	0.000000	6.146389	0.0	0.0	0.0	0.044926	...	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 61 columns

In [14]:

```python
s_it = currentStates.copy()
s_it = s_it.transpose()
# Python's for loops use per-column, not per-row, so states are on top and time is or
# instead of visa-versa, as it is in the other sections.
s = 0

print(s_it)
```

```
        AK       AL       AR   AS       AZ        CA       CO       CT  \
0      0.0      0.0      0.0  0.0      0.0       0.0      0.0      0.0   
1      0.0      0.0      0.0  0.0      1.0       2.0      0.0      0.0   
2      0.0      0.0      1.0  0.0      0.0       6.0      0.0      0.0   
3      0.0      0.0      3.0  0.0      0.0       8.0      0.0      0.0   
4      0.0      0.0      1.0  0.0      0.0      14.0      0.0      0.0   
5      0.0      0.0      3.0  0.0      0.0      22.0      0.0      0.0   
6      0.0      0.0      0.0  0.0      0.0      28.0      1.0      0.0   
7      0.0      3.0      8.0  0.0      8.0     140.0     41.0      5.0   
8     11.0     58.0     70.0  0.0     18.0     468.0    238.0    119.0   
9     52.0    411.0    279.0  0.0    374.0    2312.0   1137.0   1081.0   
10    86.0    703.0    374.0  0.0   1012.0    5169.0   2327.0   3217.0   
11    86.0   1439.0    490.0  0.0   1313.0    8972.0   2542.0   5316.0   
12    65.0   1693.0    668.0  0.0   1243.0    7368.0   2671.0   6806.0   
13    37.0   1385.0    901.0  0.0   1504.0   10980.0   2531.0   5981.0   
14    18.0   1287.0    699.0  0.0   1736.0   11112.0   4394.0   5282.0   
15    19.0   1790.0    649.0  0.0   2498.0   12343.0   2805.0   4431.0   
16    15.0   2053.0    723.0  0.0   2509.0   12265.0   2616.0   3840.0   
17    15.0   2364.0    925.0  0.0   2690.0   13039.0   2352.0   3360.0   
18    23.0   3077.0   1042.0  0.0   2413.0   14969.0   2060.0   2759.0   
19    91.0   2691.0   1876.0  0.0   5035.0   18736.0   2057.0   1657.0   
20    98.0   3705.0   2724.0  0.0   7498.0   18533.0   1251.0    843.0   
21    97.0   5282.0   2682.0  0.0  11085.0   20884.0   1371.0    662.0   
22   108.0   4937.0   4294.0  0.0  19253.0   33293.0   1587.0    554.0   
23   199.0   6769.0   3365.0  0.0  23925.0   42569.0   1978.0    660.0   
24   253.0   8502.0   3449.0  0.0  24499.0   57083.0   2238.0    536.0   
25   421.0  12045.0   4108.0  0.0  22740.0   58522.0   3267.0    527.0   
26   508.0  12400.0   4649.0  0.0  19264.0   66518.0   3704.0    575.0   
27   684.0  12058.0   5594.0  0.0  17657.0   62193.0   4152.0   1329.0   
28   601.0  10671.0   5415.0  0.0  13935.0   49959.0   3319.0    685.0   
29   491.0   8896.0   4779.0  0.0   7233.0   61840.0   2499.0    482.0   
30   558.0   7477.0   4230.0  0.0   6117.0   53386.0   2176.0    607.0   
31   465.0   8208.0   3913.0  0.0   3909.0   41022.0   2102.0    906.0   
32   474.0   8336.0   4444.0  0.0   3392.0   33604.0   2071.0    888.0   
33   547.0   5821.0   4874.0  0.0   3680.0   28054.0   2013.0    763.0   
34   529.0   6362.0   4935.0  0.0   3366.0   23947.0   3075.0   1295.0   
35   642.0   6474.0   5589.0  0.0   5381.0   25106.0   4262.0   1149.0   
36   833.0   7875.0   5539.0  0.0   3220.0   23721.0   4229.0   1235.0   
37  1020.0   6313.0   5183.0  0.0   4033.0   21364.0   4890.0   1814.0   
38  1341.0   7535.0   6206.0  0.0   5096.0   24635.0   6715.0   2497.0   
39  1538.0   7985.0   6427.0  0.0   6275.0   23575.0   8616.0   3010.0   
40  2614.0  12170.0   7040.0  0.0   7253.0   32061.0  12285.0   4256.0   
41  2727.0  10152.0   7168.0  0.0   9468.0   33103.0  20509.0   6246.0   
42  3598.0   9710.0  10652.0  0.0  14530.0   46951.0  29132.0   9368.0   
43  4316.0  14851.0  11340.0  0.0  17939.0   66610.0  37337.0  12287.0   
44  4112.0  15830.0  12594.0  0.0  27748.0  101588.0  35899.0  12124.0   
```

```
45  4524.0  17510.0  11911.0  0.0  30129.0  108911.0  34116.0  12274.0
46  4622.0  23359.0  15149.0  0.0  41622.0  184852.0  31348.0  19122.0
47  3417.0  25453.0  15451.0  0.0  46684.0  263456.0  22623.0  17233.0
48  2223.0  28929.0  14873.0  0.0  43988.0  305346.0  17252.0  12924.0
49  1724.0  22251.0  15229.0  0.0  39216.0  271394.0  13697.0  12958.0
50  2026.0  27365.0  19429.0  0.0  62191.0  280190.0  16083.0  15791.0
51  1973.0  26811.0  19328.0  0.0  67049.0  311715.0  16469.0  21122.0
52  1377.0  18659.0  14198.0  0.0  48815.0  247717.0  11987.0  13558.0
53  1181.0  19432.0  12587.0  0.0  48017.0  158085.0  10395.0  13205.0

        DC      DE  ...        TX       UT       VA     VI     VT       WA  \
0      0.0     5.0  ...       0.0      0.0      0.0    0.0    0.0      1.0
1      0.0     0.0  ...       0.0      0.0      0.0    0.0    0.0      0.0
2      0.0     4.0  ...       0.0      0.0      0.0    0.0    0.0      0.0
3      0.0     1.0  ...       0.0      0.0      0.0    0.0    0.0      0.0
4      0.0     1.0  ...       0.0      0.0      0.0    0.0    0.0      0.0
5      0.0     2.0  ...       0.0      0.0      0.0    0.0    0.0      0.0
6      0.0     0.0  ...       0.0      0.0      0.0    0.0    0.0     44.0
7      2.0     4.0  ...      21.0      2.0      7.0    0.0    1.0    329.0
8     37.0    20.0  ...     156.0     50.0     71.0    2.0    7.0    848.0
9    192.0   130.0  ...     925.0    272.0    317.0   15.0   44.0   1644.0
10   355.0   300.0  ...    2421.0    660.0   1088.0   13.0  182.0   3259.0
11   854.0   764.0  ...    4003.0    830.0   2164.0   15.0  278.0   2689.0
12   757.0   900.0  ...    4321.0    723.0   2848.0    6.0  221.0   2061.0
13  1009.0  1308.0  ...    4237.0    909.0   3769.0    3.0   80.0   1804.0
14   900.0  1418.0  ...    6041.0   1032.0   4697.0    3.0   39.0   1670.0
15  1355.0  1338.0  ...    6611.0   1092.0   6026.0    9.0   44.0   1960.0
16  1123.0  1180.0  ...    7609.0   1030.0   5757.0    3.0   30.0   1482.0
17   967.0  1142.0  ...    7590.0   1102.0   6166.0    0.0   14.0   1351.0
18   855.0   752.0  ...    5366.0    988.0   7336.0    0.0   17.0   1609.0
19   610.0   608.0  ...    7845.0   1809.0   6655.0    1.0   24.0   1906.0
20   521.0   426.0  ...   21117.0   2375.0   5267.0    2.0   80.0   2115.0
21   310.0   411.0  ...   18944.0   2493.0   3598.0    1.0   64.0   2308.0
22   281.0   423.0  ...   29179.0   3461.0   3738.0    3.0   34.0   3185.0
23   237.0   784.0  ...   46674.0   3970.0   3708.0   14.0   46.0   3732.0
24   277.0   791.0  ...   50227.0   4154.0   4162.0   32.0   47.0   4448.0
25   384.0   837.0  ...   63779.0   4302.0   6153.0  121.0   44.0   5636.0
26   503.0   752.0  ...   59348.0   4415.0   6862.0   77.0   54.0   5755.0
27   470.0   701.0  ...   55386.0   3605.0   7601.0   65.0   52.0   5769.0
28   444.0   593.0  ...   58536.0   3099.0   7061.0   96.0   24.0   5111.0
29   516.0   484.0  ...   78709.0   2741.0   7471.0  158.0   33.0   4447.0
30   395.0   464.0  ...   59198.0   2453.0   6503.0  189.0   57.0   4201.0
31   368.0   504.0  ...   32903.0   2678.0   6434.0  202.0   41.0   3660.0
32   355.0   720.0  ...   31176.0   2682.0   7058.0  114.0   56.0   3326.0
33   310.0   653.0  ...   23266.0   2847.0   6777.0   53.0   34.0   2744.0
34   356.0   710.0  ...   36350.0   4198.0   7070.0   35.0   38.0   2919.0
35   307.0   711.0  ...   35316.0   6224.0   6229.0   58.0   34.0   3238.0
36   276.0   821.0  ...   27697.0   7254.0   5677.0   28.0   28.0   3998.0
37   371.0   912.0  ...   24971.0   7446.0   5423.0    4.0   42.0   4021.0
38   435.0   893.0  ...   24682.0   8574.0   7914.0    6.0   84.0   4448.0
39   366.0   963.0  ...   34157.0   9034.0   7167.0    9.0   72.0   4713.0
40   475.0  1014.0  ...   44225.0  11036.0   7983.0   16.0  144.0   5157.0
41   628.0  1182.0  ...   41803.0  12884.0   9079.0   35.0  125.0   7363.0
42   778.0  1769.0  ...   57949.0  18358.0  10674.0   25.0  224.0  10833.0
43  1086.0  2610.0  ...   62126.0  22409.0  12330.0   50.0  534.0  14514.0
44  1051.0  3383.0  ...   97579.0  22011.0  17447.0   58.0  757.0  19245.0
45  1326.0  3941.0  ...   92464.0  18517.0  16175.0   68.0  498.0  17005.0
46  2012.0  5562.0  ...   96907.0  20363.0  24651.0  109.0  797.0  20885.0
47  1748.0  5939.0  ...  121612.0  18370.0  25128.0  130.0  740.0  17673.0
48  1624.0  4344.0  ...  133398.0  17084.0  26893.0  139.0  706.0  16528.0
49  1532.0  4508.0  ...  114742.0  14372.0  25210.0   64.0  604.0  13170.0
50  1724.0  5635.0  ...  150087.0  21015.0  32965.0   75.0  782.0  17350.0
```

```
51  2118.0  5247.0  ...  153453.0  19967.0  35266.0   60.0  1162.0  19270.0
52  1803.0  4538.0  ...  136650.0  13453.0  43025.0  139.0  1142.0  13123.0
53  1462.0  3958.0  ...  112996.0  12287.0  32937.0   68.0   962.0  11156.0

         WI       WV      WY       0
0       0.0      0.0     0.0  1000.0
1       0.0      0.0     0.0  1000.0
2       0.0      0.0     0.0  1000.0
3       0.0      0.0     0.0  1000.0
4       0.0      0.0     0.0  1000.0
5       0.0      0.0     0.0  1000.0
6       0.0      0.0     0.0  1000.0
7       3.0      0.0     2.0  1000.0
8     108.0      1.0    18.0  1000.0
9     510.0     38.0    34.0  1000.0
10    935.0    152.0   120.0  1000.0
11   1154.0    292.0   129.0  1000.0
12   1011.0    220.0    86.0  1000.0
13   1124.0    239.0    50.0  1000.0
14   1675.0    168.0   101.0  1000.0
15   2380.0    130.0    76.0  1000.0
16   2003.0    157.0    66.0  1000.0
17   2508.0    168.0    93.0  1000.0
18   3053.0    323.0    73.0  1000.0
19   2935.0    188.0    41.0  1000.0
20   2194.0    119.0    80.0  1000.0
21   1861.0    179.0   134.0  1000.0
22   2309.0    257.0   161.0  1000.0
23   3438.0    347.0   215.0  1000.0
24   3952.0    729.0   233.0  1000.0
25   5574.0    851.0   232.0  1000.0
26   6122.0    653.0   306.0  1000.0
27   6202.0   1067.0   360.0  1000.0
28   5890.0    877.0   259.0  1000.0
29   5329.0    862.0   186.0  1000.0
30   5224.0    783.0   304.0  1000.0
31   4768.0    738.0   293.0  1000.0
32   4871.0   1119.0   227.0  1000.0
33   6208.0   1157.0   268.0  1000.0
34   9374.0   1388.0   437.0  1000.0
35  13217.0   1313.0   624.0  1000.0
36  16343.0   1335.0   782.0  1000.0
37  16425.0   1300.0   967.0  1000.0
38  19880.0   1673.0  1247.0  1000.0
39  33637.0   1911.0  1727.0  1000.0
40  29343.0   2333.0  2339.0  1000.0
41  35727.0   2921.0  3366.0  1000.0
42  44062.0   4215.0  4500.0  1000.0
43  49316.0   6064.0  5367.0  1000.0
44  41777.0   6785.0  5400.0  1000.0
45  28488.0   6853.0  3604.0  1000.0
46  31250.0   8557.0  3222.0  1000.0
47  25982.0   8388.0  2517.0  1000.0
48  22282.0   9086.0  2170.0  1000.0
49  15782.0   8289.0  1473.0  1000.0
50  19773.0  10456.0  1807.0  1000.0
51  22021.0   9569.0  2296.0  1000.0
52  15101.0   7427.0  1635.0  1000.0
53  12482.0   6094.0  1254.0  1000.0

[54 rows x 61 columns]
```

In [15]:
```
s_it = s_it - tot
s_it
```

Out[15]:
```
            AK       AL           AR   AS           AZ            CA           CO      CT
 0    0.000000      0.0     0.000000  0.0     0.000000      0.000000     0.000000     0.0    0
 1    0.000000      0.0     0.000000  0.0     0.323319      1.121944     0.000000     0.0    0
 2    0.000000      0.0     0.444071  0.0     0.000000      3.365833     0.000000     0.0    0
 3    0.000000      0.0     1.332213  0.0     0.000000      4.487778     0.000000     0.0    0
 4    0.000000      0.0     0.444071  0.0     0.000000      7.853611     0.000000     0.0    0
 5    0.000000      0.0     1.332213  0.0     0.000000     12.341388     0.000000     0.0    0
 6    0.000000      0.0     0.000000  0.0     0.000000     15.707221     0.664233     0.0    0
 7    0.000000      3.0     3.552568  0.0     2.586550     78.536107    27.233533     5.0    0
 8    6.227751     58.0    31.084973  0.0     5.819737    262.534986   158.087340   119.0   17
 9   29.440276    411.0   123.895819  0.0   120.921196   1296.967711   755.232376  1081.0   92
10   48.689687    703.0   166.082567  0.0   327.198530   2899.665268  1545.669076  3217.0  171
11   48.689687   1439.0   217.594808  0.0   424.517461   5033.042519  1688.479068  5316.0  413
12   36.800345   1693.0   296.639452  0.0   401.885151   4133.243121  1774.165064  6806.0  366
13   20.947888   1385.0   400.108004  0.0   486.271334   6159.474683  1681.172511  5981.0  487
14   10.190865   1287.0   310.405654  0.0   561.281273   6233.523013  2918.637697  5282.0  435
15   10.757024   1790.0   288.202102  0.0   807.650127   6924.079783  1863.172221  4431.0  655
16    8.492387   2053.0   321.063359  0.0   811.206633   6880.323952  1737.632275  3840.0  543
17    8.492387   2364.0   410.765708  0.0   869.727319   7314.516429  1562.274889  3360.0  467
18   13.021660   3077.0   462.722020  0.0   780.168037   8397.192763  1368.318993  2759.0  413
19   51.520482   2691.0   833.077264  0.0  1627.909684  10510.375015  1366.326296  1657.0  295
20   55.483597   3705.0  1209.649502  0.0  2424.243656  10396.497660   830.954884   843.0  251
21   54.917437   5282.0  1190.998519  0.0  3583.987854  11715.343286   910.662786   662.0  149
22   61.145188   4937.0  1906.841029  0.0  6224.855043  18676.447234  1054.137011   554.0  135
23  112.665671   6769.0  1494.299036  0.0  7735.400037  23880.025299  1313.851927   660.0  114
24  143.238265   8502.0  1531.601003  0.0  7920.984974  32021.975714  1486.552382   536.0  133
25  238.353001  12045.0  1824.243816  0.0  7352.267370  32829.214700  2170.047646   527.0  185
26  287.608847  12400.0  2064.486247  0.0  6228.411549  37314.748358  2460.317258   575.0  243
27  387.252857  12058.0  2484.133376  0.0  5708.838389  34888.543622  2757.893427  1329.0  227
28  340.261648  10671.0  2404.644660  0.0  4505.446166  28025.609808  2204.587737   685.0  214
29  277.984142   8896.0  2122.215481  0.0  2338.564199  34690.520438  1659.917070   482.0  249
30  315.916805   7477.0  1878.420483  0.0  1977.740524  29948.061515  1445.369966   607.0  191
```

	AK	AL	AR	AS	AZ	CA	CO	CT	
31	263.264004	8208.0	1737.649964	0.0	1263.852821	23012.201316	1396.216759	906.0	177
32	268.359436	8336.0	1973.451684	0.0	1096.697050	18850.909586	1375.625551	888.0	171
33	309.689054	5821.0	2164.402230	0.0	1189.812837	15737.513913	1337.100065	763.0	149
34	299.498189	6362.0	2191.490563	0.0	1088.290764	13433.601115	2042.515002	1295.0	172
35	363.474173	6474.0	2481.913021	0.0	1739.777956	14083.767887	2830.959005	1149.0	148
36	471.610571	7875.0	2459.709469	0.0	1041.086233	13306.821399	2809.039331	1235.0	133
37	577.482331	6313.0	2301.620180	0.0	1303.944341	11984.609939	3248.097028	1814.0	179
38	759.219418	7535.0	2755.904850	0.0	1647.632125	13819.549984	4460.321379	2497.0	210
39	870.752770	7985.0	2854.044549	0.0	2028.824879	13224.919459	5723.027401	3010.0	177
40	1479.940014	12170.0	3126.260094	0.0	2345.030573	17985.329492	8160.096521	4256.0	229
41	1543.915997	10152.0	3183.101187	0.0	3061.181507	18569.862517	13622.744773	6246.0	303
42	2037.040616	9710.0	4730.244676	0.0	4697.820796	26338.205451	19350.421802	9368.0	376
43	2443.542884	14851.0	5035.765549	0.0	5800.014264	37366.357800	24800.449637	12287.0	525
44	2328.046418	15830.0	5592.630629	0.0	8971.447449	56988.043180	23845.283272	12124.0	508
45	2561.303987	17510.0	5289.330111	0.0	9741.269288	61096.042552	22660.956687	12274.0	641
46	2616.787583	23359.0	6727.232126	0.0	13457.171174	103696.831888	20822.361069	19122.0	973
47	1934.565810	25453.0	6861.341579	0.0	15093.810463	147791.490176	15026.932323	17233.0	845
48	1258.571787	28929.0	6604.668520	0.0	14222.143232	171290.615356	11459.339452	12924.0	785
49	976.058372	22251.0	6762.757809	0.0	12679.266368	152244.487447	9097.992840	12958.0	740
50	1147.038434	27365.0	8627.856160	0.0	20107.513634	157178.798859	10682.851635	15791.0	833
51	1117.032000	26811.0	8583.004986	0.0	21678.195907	174863.447255	10939.245388	21122.0	1024
52	779.601147	18659.0	6304.920570	0.0	15782.802625	138962.348824	7962.155229	13558.0	871
53	668.633954	19432.0	5589.522131	0.0	15524.794298	88681.289188	6904.697056	13205.0	707

54 rows × 61 columns

In [16]:
```
for s in s_it: # loop in states. Note s is the postal abbreviation here.
    print(s)
    s_i = s_it.loc[:, s]
    sizeOf = s_i.size #in case we change interpolate distance later
    s_i = s_i.to_numpy()

    frequencies = sp.fft.fftfreq(sizeOf, d = 1.0)
    fourierAmps = sp.fft.fft(s_i)
    amplitudes = (2/(sizeOf)) * np.abs(fourierAmps)

    # getting all frequencies, not only top ones, because why not?
    #top_indices = np.argpartition(np.abs(fourierAmps), -2)[-2:]

    # Exclude zero-freqencires
    amplitudes = amplitudes[frequencies != 0]
    frequencies = frequencies[frequencies != 0]

    #dominantFreqs = [frequencies[j] for j in top_indices]
    #dominantAmps = [np.abs(fourierAm[j] ) for j in top_indices]

    # Wavelength = 1 / frequency
    print("Wavelengths:")
    print(1 / frequencies)
    # print("Amps1:", dominantAmps)
    print("Amps:")
    print(amplitudes)

    #tempHold = sp.integrate.solve_ivp(func(t, s_i), (53,91), [0])
    #errors_it = fourier.error
```

```
AK
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[819.07507582 515.74491509 342.78975059 132.20330949  70.34771811
  88.24584004  79.13183276  70.80088881  68.35317785  55.41500457
  49.69329091  33.13659861  17.69861029  22.35345957  24.79815266
  38.77939323  37.67978607  19.16559297   5.36827941  22.52441404
  40.15059787  29.01681982   7.16215508  11.25544674  25.22919031
  17.22038727   3.62761232  17.22038727  25.22919031  11.25544674
   7.16215508  29.01681982  40.15059787  22.52441404   5.36827941
  19.16559297  37.67978607  38.77939323  24.79815266  22.35345957
  17.69861029  33.13659861  49.69329091  55.41500457  68.35317785
  70.80088881  79.13183276  88.24584004  70.34771811 132.20330949
 342.78975059 515.74491509 819.07507582]
AL
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[7326.23777686 6433.54790844 4253.96731913 2332.96022085 1779.05026766
  826.43277615  631.20969835  926.79228264  714.80303579  770.21480064
  555.20294516  537.03392333  553.70173488  412.14484887  330.34864618
  321.30696832  595.98791378  436.83497526  276.82613923  661.1605675
  754.96299802  849.83104285  716.46606717  399.91450681  550.16293232
  396.87645625  137.07407407  396.87645625  550.16293232  399.91450681
  716.46606717  849.83104285  754.96299802  661.1605675   276.82613923
  436.83497526  595.98791378  321.30696832  330.34864618  412.14484887
  553.70173488  537.03392333  555.20294516  770.21480064  714.80303579
  926.79228264  631.20969835  826.43277615 1779.05026766 2332.96022085
 4253.96731913 6433.54790844 7326.23777686]
AR
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[2475.20140005 1717.72592397 1034.74407938  684.84251533  431.58558755
  392.7768081   490.18257195  232.68742481  297.98491918  152.54410626
  214.26978878  140.26848661  103.9599597    43.86569781  103.3942578
  154.84885157  103.88080654  131.97079094  155.42833826   93.94401202
  156.8090525    83.89176727  139.99095376  160.53922409  162.74915011
  114.52614743   20.78910332  114.52614743  162.74915011  160.53922409
  139.99095376   83.89176727  156.8090525    93.94401202  155.42833826
  131.97079094  103.88080654  154.84885157  103.3942578    43.86569781
  103.9599597   140.26848661  214.26978878  152.54410626  297.98491918
  232.68742481  490.18257195  392.7768081   431.58558755  684.84251533
 1034.74407938 1717.72592397 2475.20140005]
AS
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0.]
```

```
AZ
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[3729.43419515 5466.88415282 2307.37770094 2415.87453985  693.2168274
 1281.74752634  852.88314425  952.09297073  918.96745968  651.18547513
  595.5847208   370.02868481  301.9562907   121.30696113  134.33862043
  286.39235827  400.04249158  458.24032065  520.5358166   375.94035269
  471.73637485  347.69999293  424.76174063  485.88990494  353.92933936
  269.13163818  205.06788286  269.13163818  353.92933936  485.88990494
  424.76174063  347.69999293  471.73637485  375.94035269  520.5358166
  458.24032065  400.04249158  286.39235827  134.33862043  121.30696113
  301.9562907   370.02868481  595.5847208   651.18547513  918.96745968
  952.09297073  852.88314425 1281.74752634  693.2168274  2415.87453985
 2307.37770094 5466.88415282 3729.43419515]
CA
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[37005.93920734 40317.65626101 29671.28677629 18941.71287956
 13437.72134301  7410.40012042  4020.09151589  3828.00073514
  3449.08097242  5143.28286105  4625.40413925  5159.31320224
  4470.83081721  2865.55956978  2076.88225282   176.47039198
   185.12390821  1338.07921601  1210.65532574   950.44057516
  1226.94716976  1531.70044698  1825.50863201  1970.63188048
  1907.44002746  1218.9891465    626.91759121  1218.9891465
  1907.44002746  1970.63188048  1825.50863201  1531.70044698
  1226.94716976   950.44057516  1210.65532574  1338.07921601
   185.12390821   176.47039198  2076.88225282  2865.55956978
  4470.83081721  5159.31320224  4625.40413925  5143.28286105
  3449.08097242  3828.00073514  4020.09151589  7410.40012042
 13437.72134301 18941.71287956 29671.28677629 40317.65626101
 37005.93920734]
CO
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[6505.13956845 5252.29222303 3137.15396436 1430.48955554 1139.57227178
 1430.718206   1238.46383823  709.86995848  340.41672489  250.18743644
  276.89016939  246.10181028  256.08348261  154.85618599  234.2877643
  140.97064259  149.81595307  151.25751435   59.75874862   95.06574089
  262.5006444   289.90184475  197.66141557  192.1635595   142.31261959
  112.33839319  125.53994647  112.33839319  142.31261959  192.1635595
  197.66141557  289.90184475  262.5006444    95.06574089   59.75874862
  151.25751435  149.81595307  140.97064259  234.2877643   154.85618599
  256.08348261  246.10181028  276.89016939  250.18743644  340.41672489
  709.86995848 1238.46383823 1430.718206   1139.57227178 1430.48955554
 3137.15396436 5252.29222303 6505.13956845]
CT
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[5007.78588554 4562.36465941 3341.17194659  791.55938424  268.63736964
  786.50537911 1100.08554365 1051.86863216  755.24323701  639.94520864
  888.42155282  619.48324868  408.06361029  316.47214508  283.78564136
  327.73070085  411.36420705  220.456656     96.46763467  184.16995136
  337.65291396  432.04293363  490.39071951  453.60500554  505.15445185
  371.96859874  354.7037037   371.96859874  505.15445185  453.60500554
  490.39071951  432.04293363  337.65291396  184.16995136   96.46763467
  220.456656    411.36420705  327.73070085  283.78564136  316.47214508
  408.06361029  619.48324868  888.42155282  639.94520864  755.24323701
 1051.86863216 1100.08554365  786.50537911  268.63736964  791.55938424
 3341.17194659 4562.36465941 5007.78588554]
DC
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[145.27346047 256.65185446 215.71571338  27.74199852  48.87262423
  41.1766427   44.14918274  52.29628494  50.82102078  37.97830251
  34.6484084   39.9421977   10.75021695  10.8295779    3.2454682
  20.34263142  18.02669806  14.67924768   8.4274244   13.01826837
  12.9622446   17.32587825  32.01814428  17.36111237   6.85216493
  19.04739595  21.33296216  19.04739595   6.85216493  17.36111237
  32.01814428  17.32587825  12.9622446   13.01826837   8.4274244
  14.67924768  18.02669806  20.34263142   3.2454682   10.8295779
  10.75021695  39.9421977   34.6484084   37.97830251  50.82102078
  52.29628494  44.14918274  41.1766427   48.87262423  27.74199852
 215.71571338 256.65185446 145.27346047]
DE
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[1344.51777054 1326.69290559  972.83529044  365.76119496  218.91345821
  207.94630538  216.61366692  267.57131474  207.76050996  187.0207299
  172.28486267  109.34556415   47.18919294   27.43931779   95.91079347
  125.31843813  138.69590777  152.79249948  102.18693197   84.10549564
   67.40742711   53.13445097   68.85470986   64.90708323   59.00929516
   62.51580284   78.66976815   62.51580284   59.00929516   64.90708323
   68.85470986   53.13445097   67.40742711   84.10549564  102.18693197
  152.79249948  138.69590777  125.31843813   95.91079347   27.43931779
   47.18919294  109.34556415  172.28486267  187.0207299   207.76050996
  267.57131474  216.61366692  207.94630538  218.91345821  365.76119496
  972.83529044 1326.69290559 1344.51777054]
FL
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[6666.69173163 8800.87087822 5797.99046613 3836.96038624 2397.9429135
 2423.02292248 1810.00866943 1808.03280923 1146.84134175 1150.30598703
  896.75657154  903.51766943  891.29944265  644.78761719  633.3092023
  206.50139003  459.15840599  348.71705357  509.45112485  369.79082471
  329.71907266  572.62970711  315.13968479  756.40213767  284.17652079
  630.19762716  167.90543987  630.19762716  284.17652079  756.40213767
  315.13968479  572.62970711  329.71907266  369.79082471  509.45112485
  348.71705357  459.15840599  206.50139003  633.3092023   644.78761719
  891.29944265  903.51766943  896.75657154 1150.30598703 1146.84134175
 1808.03280923 1810.00866943 2423.02292248 2397.9429135  3836.96038624
 5797.99046613 8800.87087822 6666.69173163]
FSM
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.        -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[0.03703704 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704
 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704
 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704
 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704
 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704
 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704
 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704
 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704
 0.03703704 0.03703704 0.03703704 0.03703704 0.03703704]
GA
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.        -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[12692.48173685 13435.54292665  9505.61757824  5955.29753213
  4923.95171731  4538.04866663  2844.15466165  1650.50294262
  2669.21317066  3035.94788645  2142.28858675  1065.26569332
   722.50647037  1365.58007346  1033.58178992   487.04816016
  1372.37646225  2056.86136141  1576.53295817   318.77771156
   855.59851005  1867.63391032  2281.81806898  1078.22056614
   354.28661597  1052.56319948  2035.3556      1052.56319948
   354.28661597  1078.22056614  2281.81806898  1867.63391032
   855.59851005   318.77771156  1576.53295817  2056.86136141
  1372.37646225   487.04816016  1033.58178992  1365.58007346
   722.50647037  1065.26569332  2142.28858675  3035.94788645
  2669.21317066  1650.50294262  2844.15466165  4538.04866663
  4923.95171731  5955.29753213  9505.61757824 13435.54292665
 12692.48173685]
GU
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.        -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[56.16992553 31.84735513 12.02542543 11.27464894 17.85732084 12.65836898
  4.44897877  6.69398888  7.29233594  9.25391469  6.32340294  2.73807839
  7.01802278  7.91860132  6.9659055   4.33297552  1.01926233  3.55525907
  5.97753908  6.79569899  4.92167487  1.43297726  2.05314308  3.48592315
  3.3919765   1.65307676  0.1754207   1.65307676  3.3919765   3.48592315
  2.05314308  1.43297726  4.92167487  6.79569899  5.97753908  3.55525907
  1.01926233  4.33297552  6.9659055   7.91860132  7.01802278  2.73807839
  6.32340294  9.25391469  7.29233594  6.69398888  4.44897877 12.65836898
 17.85732084 11.27464894 12.02542543 31.84735513 56.16992553]
HI
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[165.61718143  49.12255265 112.86870997  23.27966338  62.4397216
  40.52744698  17.72591107  35.51422867   2.72346027  16.1378401
  14.73441501  11.59121088   5.55738173  11.88883996   9.85795315
   3.53132093   5.76300569   8.37165614  11.17646746   7.16652708
   7.64643936   4.64308446  11.93292024   9.77659089   7.91303077
  11.53881213   4.66241222  11.53881213   7.91303077   9.77659089
  11.93292024   4.64308446   7.64643936   7.16652708  11.17646746
   8.37165614   5.76300569   3.53132093   9.85795315  11.88883996
   5.55738173  11.59121088  14.73441501  16.1378401    2.72346027
  35.51422867  17.72591107  40.52744698  62.4397216   23.27966338
 112.86870997  49.12255265 165.61718143]
IA
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[6022.21380013 4069.47089319 1907.64622492 1610.83309703 1565.85658363
 2143.41023874 1336.0116614   916.02661772  699.33292841  891.58146742
  785.35846879  569.01865799  388.22518346  341.77424468  441.67312254
  337.15818825   86.12732016  204.67800362  143.90844779  291.13822943
  258.34096796   88.96263079  291.00020337  292.25305949  428.12429549
  403.00095805  404.50935254  403.00095805  428.12429549  292.25305949
  291.00020337   88.96263079  258.34096796  291.13822943  143.90844779
  204.67800362   86.12732016  337.15818825  441.67312254  341.77424468
  388.22518346  569.01865799  785.35846879  891.58146742  699.33292841
  916.02661772 1336.0116614  2143.41023874 1565.85658363 1610.83309703
 1907.64622492 4069.47089319 6022.21380013]
ID
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.        -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[2983.66958758 1678.741041   1273.75713689  109.40526467  183.67899191
  194.42133213  158.97376376  247.38949994  150.71547183  245.5144589
  158.27159672  228.24077336  167.23949079   81.60580412  109.74971052
  142.01401824   87.42167185   78.31005868   20.45401503   44.10944447
   23.32682565   94.5623639   170.76737235  180.25130533  177.1667854
  153.93172729   86.82375976  153.93172729  177.1667854   180.25130533
  170.76737235   94.5623639    23.32682565   44.10944447   20.45401503
   78.31005868   87.42167185  142.01401824  109.74971052   81.60580412
  167.23949079  228.24077336  158.27159672  245.5144589   150.71547183
  247.38949994  158.97376376  194.42133213  183.67899191  109.40526467
 1273.75713689 1678.741041   2983.66958758]
IL
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.        -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[16427.89082726 13646.8322085   6298.61605222  3803.66389622
  2339.52685591  3906.18858519  2511.24092887  1622.72078104
   371.12995987  1103.08350674  1583.03570347  1601.21427737
  1058.14010541   483.56458219   634.02755574   550.00504205
   427.67610651   477.94205519   717.53184998   687.80421945
   381.1218245    367.50737812   507.90316761   879.76334873
   824.81870588   428.19993129    24.77947495   428.19993129
   824.81870588   879.76334873   507.90316761   367.50737812
   381.1218245    687.80421945   717.53184998   477.94205519
   427.67610651   550.00504205   634.02755574   483.56458219
  1058.14010541  1601.21427737  1583.03570347  1103.08350674
   371.12995987  1622.72078104  2511.24092887  3906.18858519
  2339.52685591  3803.66389622  6298.61605222 13646.8322085
 16427.89082726]
IN
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.        -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[14609.89674777 10788.91813661  6470.21261316  1580.08248881
   543.73460385  1687.84997528  1913.88029594  1485.93147199
   830.9088344    803.07540462  1052.15046133  1240.15815021
   986.32244165   546.89072988   647.69501143   571.24243379
   668.57096734   611.38840959   137.26022574   239.64306929
   370.33093194   717.69050786   701.75479893   600.52272256
   508.72577986   315.36627995   126.07407407   315.36627995
   508.72577986   600.52272256   701.75479893   717.69050786
   370.33093194   239.64306929   137.26022574   611.38840959
   668.57096734   571.24243379   647.69501143   546.89072988
   986.32244165  1240.15815021  1052.15046133   803.07540462
   830.9088344   1485.93147199  1913.88029594  1687.84997528
   543.73460385  1580.08248881  6470.21261316 10788.91813661
 14609.89674777]
KS
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[6409.13482864 4063.21942997 2383.92774426  368.65730869  444.48514123
 1385.8860294   980.73842952  704.48351142   71.05554349  537.5800089
  440.64557012  502.45664449  480.3770715   265.56872059  254.95759597
  199.40784837  146.61034876  196.94595152  132.5589957   117.53743111
  249.83572928  265.2036481   285.17025186  447.68698574  408.26222491
  196.60209157   31.96296296  196.60209157  408.26222491  447.68698574
  285.17025186  265.2036481   249.83572928  117.53743111  132.5589957
  196.94595152  146.61034876  199.40784837  254.95759597  265.56872059
  480.3770715   502.45664449  440.64557012  537.5800089    71.05554349
  704.48351142  980.73842952 1385.8860294   444.48514123  368.65730869
 2383.92774426 4063.21942997 6409.13482864]
KY
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[8681.43720934 5488.76360051 3387.38038598 1332.16054471 1345.12501876
 1523.49444333 1708.24608851 1445.77279113 1144.99936624  831.27969601
  675.79238836  352.8643535   150.53129087  121.87262974  319.75086794
  662.11724903  793.00937469  792.96898038  678.30027851  887.12457091
  838.61962877  524.85214812  631.93006838  614.22807216  458.54299425
  318.11908258  206.18518519  318.11908258  458.54299425  614.22807216
  631.93006838  524.85214812  838.61962877  887.12457091  678.30027851
  792.96898038  793.00937469  662.11724903  319.75086794  121.87262974
  150.53129087  352.8643535   675.79238836  831.27969601 1144.99936624
 1445.77279113 1708.24608851 1523.49444333 1345.12501876 1332.16054471
 3387.38038598 5488.76360051 8681.43720934]
LA
```

```
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[2270.15749946 3054.9789841  2502.18926227 1341.95140885  336.82722317
  661.97243033  853.18792474 1010.75255427  460.52272309  130.94861347
  118.29412317  398.52399156  437.47968845  102.05487355  248.35557949
  112.21393781  418.95193226  350.90134777  120.27500167  173.36566382
  195.43376366  496.33957543  327.17257693  217.2598769    87.82663982
  130.30239905  387.16516191  130.30239905   87.82663982  217.2598769
  327.17257693  496.33957543  195.43376366  173.36566382  120.27500167
  350.90134777  418.95193226  112.21393781  248.35557949  102.05487355
  437.47968845  398.52399156  118.29412317  130.94861347  460.52272309
 1010.75255427  853.18792474  661.97243033  336.82722317 1341.95140885
 2502.18926227 3054.9789841  2270.15749946]
MA
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[9817.70392174 8877.71656874 8054.81115987 3038.9403646  1270.25888169
 1852.63562373 2066.93959402 1629.78895871 1348.97382327 1603.67793055
 1608.51794324 1088.74391919  714.90945935  255.64173877  462.3750498
  686.08819289  703.9056465   560.05715954  148.77542294  367.31121207
  588.13493177  690.56187684  937.47356046  934.27248508  810.3638727
  683.2288845   510.48148148  683.2288845   810.3638727   934.27248508
  937.47356046  690.56187684  588.13493177  367.31121207  148.77542294
  560.05715954  703.9056465   686.08819289  462.3750498   255.64173877
  714.90945935 1088.74391919 1608.51794324 1603.67793055 1348.97382327
 1629.78895871 2066.93959402 1852.63562373 1270.25888169 3038.9403646
 8054.81115987 8877.71656874 9817.70392174]
MD
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
```

```
Amps:
[4101.69603686 4975.42342486 3393.1569845   398.44232314  877.18186838
 1009.23273244 1208.53788751  974.27971923  784.81924848  467.05340567
  576.34449804  603.95624457  331.31232502   64.96752107   75.66755843
  213.29851365  401.21162359  327.02409692  198.05247365  169.15575652
  208.82957554  250.97589868  257.01853652  422.77480439  302.25499357
  243.12105274  261.69492065  243.12105274  302.25499357  422.77480439
  257.01853652  250.97589868  208.82957554  169.15575652  198.05247365
  327.02409692  401.21162359  213.29851365   75.66755843   64.96752107
  331.31232502  603.95624457  576.34449804  467.05340567  784.81924848
  974.27971923 1208.53788751 1009.23273244  877.18186838  398.44232314
 3393.1569845  4975.42342486 4101.69603686]
ME
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[946.15867303 822.53948854 570.42758983 357.55565519 240.05272958
 201.54829261 164.47230678 134.93519362 129.32815926 142.22503094
 157.95673595 127.74611621 100.73478601  44.60960672  35.98448928
  12.72234662  26.13030058  37.41617498  34.93044779  44.16324876
  43.60636323  49.08933358  57.90852144  59.71660296  56.2460779
  47.72978123  35.54822633  47.72978123  56.2460779   59.71660296
  57.90852144  49.08933358  43.60636323  44.16324876  34.93044779
  37.41617498  26.13030058  12.72234662  35.98448928  44.60960672
 100.73478601 127.74611621 157.95673595 142.22503094 129.32815926
 134.93519362 164.47230678 201.54829261 240.05272958 357.55565519
 570.42758983 822.53948854 946.15867303]
MI
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[7444.97863716 6184.16943414 4071.22986552 1607.74190751 1606.24912838
 1258.59939611 1257.42667091 1135.0746161   358.91135027   82.19021928
  253.18189033  398.47273876  249.68304237  243.88010504  341.07454374
  188.21346187  334.43218752  129.53690143  229.31276017  138.78519076
  319.53016193  293.11317488  229.12409376  117.04554058   80.80246578
  139.02444584  219.64424221  139.02444584   80.80246578  117.04554058
  229.12409376  293.11317488  319.53016193  138.78519076  229.31276017
  129.53690143  334.43218752  188.21346187  341.07454374  243.88010504
  249.68304237  398.47273876  253.18189033   82.19021928  358.91135027
 1135.0746161  1257.42667091 1258.59939611 1606.24912838 1607.74190751
 4071.22986552 6184.16943414 7444.97863716]
```

```
MN
Wavelengths:
[ 54.         27.         18.         13.5        10.8
   9.          7.71428571  6.75        6.          5.4
   4.90909091  4.5         4.15384615  3.85714286  3.6
   3.375       3.17647059  3.          2.84210526  2.7
   2.57142857  2.45454545  2.34782609  2.25        2.16
   2.07692308 -2.         -2.07692308 -2.16       -2.25
  -2.34782609 -2.45454545 -2.57142857 -2.7        -2.84210526
  -3.         -3.17647059 -3.375      -3.6        -3.85714286
  -4.15384615 -4.5        -4.90909091 -5.4        -6.
  -6.75       -7.71428571 -9.        -10.8       -13.5
 -18.        -27.        -54.       ]
Amps:
[10007.18195855  8030.59596298  4810.577719    3711.48686786
  2215.89838224  2775.05617132  2004.53564326  1657.13526238
   734.07022365   268.71107977   413.15043678   421.93431125
   396.93639116   433.24646088   524.29977664   438.19145899
   330.07750976   281.39061677   217.97705685    89.35382402
   379.86552168   338.94438463   505.10640808   390.72229793
   282.54093675   177.90011827    70.9544522    177.90011827
   282.54093675   390.72229793   505.10640808   338.94438463
   379.86552168    89.35382402   217.97705685   281.39061677
   330.07750976   438.19145899   524.29977664   433.24646088
   396.93639116   421.93431125   413.15043678   268.71107977
   734.07022365  1657.13526238  2004.53564326  2775.05617132
  2215.89838224  3711.48686786  4810.577719    8030.59596298
 10007.18195855]
MO
Wavelengths:
[ 54.         27.         18.         13.5        10.8
   9.          7.71428571  6.75        6.          5.4
   4.90909091  4.5         4.15384615  3.85714286  3.6
   3.375       3.17647059  3.          2.84210526  2.7
   2.57142857  2.45454545  2.34782609  2.25        2.16
   2.07692308 -2.         -2.07692308 -2.16       -2.25
  -2.34782609 -2.45454545 -2.57142857 -2.7        -2.84210526
  -3.         -3.17647059 -3.375      -3.6        -3.85714286
  -4.15384615 -4.5        -4.90909091 -5.4        -6.
  -6.75       -7.71428571 -9.        -10.8       -13.5
 -18.        -27.        -54.       ]
Amps:
[11822.2230374   5937.29755001  4008.03490862   347.23959751
   797.88684812  1848.09123377  2063.9909597    639.93585557
   346.26809352   587.82069994   997.73690006   699.82245328
   839.88221641   539.28652098   265.57308036   567.08909451
   626.15483654   343.51588698   523.76256547   139.63054539
   324.62454921   363.64927633   691.53654272   269.53437299
   605.52242634   508.56797081   209.           508.56797081
   605.52242634   269.53437299   691.53654272   363.64927633
   324.62454921   139.63054539   523.76256547   343.51588698
   626.15483654   567.08909451   265.57308036   539.28652098
   839.88221641   699.82245328   997.73690006   587.82069994
   346.26809352   639.93585557  2063.9909597   1848.09123377
   797.88684812   347.23959751  4008.03490862  5937.29755001
 11822.2230374 ]
MP
Wavelengths:
[ 54.         27.         18.         13.5        10.8
   9.          7.71428571  6.75        6.          5.4
   4.90909091  4.5         4.15384615  3.85714286  3.6
   3.375       3.17647059  3.          2.84210526  2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[1.48040859 0.83577182 0.49020488 0.71918681 0.59437648 0.71137726
 0.59609842 0.45924716 0.51851852 0.22271021 0.77586116 0.8961527
 0.5783696  0.62975736 0.64853247 0.60945008 0.51369629 0.12830006
 0.51187554 0.21947807 0.58232706 0.27417205 1.21845064 0.85797448
 0.8117587  0.60064832 0.07407407 0.60064832 0.8117587  0.85797448
 1.21845064 0.27417205 0.58232706 0.21947807 0.51187554 0.12830006
 0.51369629 0.60945008 0.64853247 0.62975736 0.5783696  0.8961527
 0.77586116 0.22271021 0.51851852 0.45924716 0.59609842 0.71137726
 0.59437648 0.71918681 0.49020488 0.83577182 1.48040859]
MS
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[3451.27889915 3266.57369884 2336.23663017  973.74222372  993.4893166
  358.94055049  465.27257808  610.80071144  520.60128311  417.87235474
  506.58390632  194.34306974  341.99234745  118.22654208  197.27194852
  224.42725157  183.73245602  151.99851481  167.16520138  387.13894699
  254.01095353  286.47495943  162.82599863  237.0271702   189.9641238
  188.03381641  138.81252922  188.03381641  189.9641238   237.0271702
  162.82599863  286.47495943  254.01095353  387.13894699  167.16520138
  151.99851481  183.73245602  224.42725157  197.27194852  118.22654208
  341.99234745  194.34306974  506.58390632  417.87235474  520.60128311
  610.80071144  465.27257808  358.94055049  993.4893166   973.74222372
 2336.23663017 3266.57369884 3451.27889915]
MT
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[2288.05748586 1301.90385747  682.38253504  284.94995281  209.64322858
  224.55012243  242.11900873  226.62075259  101.26049648   51.13763765
  110.2222135   140.67494287  132.88208579   96.22238526   69.0414555
   66.48115535   87.1870163    97.00626484   56.92218691   65.7114693
  104.8454033   114.7383714   107.72394653   51.19015887   49.59948252
   22.73282714   47.47113677   22.73282714   49.59948252   51.19015887
  107.72394653  114.7383714   104.8454033    65.7114693    56.92218691
   97.00626484   87.1870163    66.48115535   69.0414555    96.22238526
  132.88208579  140.67494287  110.2222135    51.13763765  101.26049648
  226.62075259  242.11900873  224.55012243  209.64322858  284.94995281
  682.38253504 1301.90385747 2288.05748586]
NC
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[10684.96487257  9540.01495977  5596.9770672   4196.92360799
  2728.13136296  2006.24880818  1719.61804051  1971.97032156
  1512.61504898  1425.60624566  1225.01112714   844.26518285
   622.25393153    61.56251069   425.04414003   590.35166729
   813.3928694    818.98639667   559.71543052   674.88086185
   832.18751038   857.83481577   934.46625447   732.11356371
   597.25507484   346.82250117   255.83240154   346.82250117
   597.25507484   732.11356371   934.46625447   857.83481577
   832.18751038   674.88086185   559.71543052   818.98639667
   813.3928694    590.35166729   425.04414003    61.56251069
   622.25393153   844.26518285  1225.01112714  1425.60624566
  1512.61504898  1971.97032156  1719.61804051  2006.24880818
  2728.13136296  4196.92360799  5596.9770672   9540.01495977
 10684.96487257]
ND
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[1994.48948612 1280.41500892  727.28959944  553.34648729  386.01210109
  344.33687722  217.78869134   83.27170265   51.99251185   79.3615671
   82.08566771   55.24020286   73.46769497   68.14419332  102.50724907
   71.90552731   61.40803328   89.9929409    99.27290602  105.69661929
  125.61765423  138.07061289  134.64977827  100.33320252   74.09356055
   49.95713191   33.71689335   49.95713191   74.09356055  100.33320252
  134.64977827  138.07061289  125.61765423  105.69661929   99.27290602
   89.9929409    61.40803328   71.90552731  102.50724907   68.14419332
   73.46769497   55.24020286   82.08566771   79.3615671    51.99251185
   83.27170265  217.78869134  344.33687722  386.01210109  553.34648729
  727.28959944 1280.41500892 1994.48948612]
NE
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[3912.60062742 3019.58059808 1176.50347517  797.62429993  405.4637297
  836.53683406  594.7759879   355.09707776  126.10692873  237.74023156
  284.79851062  313.29201432  201.33287534  148.54443542  152.80111789
  148.02467086   94.39117379   70.27110505   58.35279852   56.22143394
  106.75208266  116.09108754  158.40616499  197.80060006   79.70018068
  145.63116868   78.26879278  145.63116868   79.70018068  197.80060006
  158.40616499  116.09108754  106.75208266   56.22143394   58.35279852
   70.27110505   94.39117379  148.02467086  152.80111789  148.54443542
  201.33287534  313.29201432  284.79851062  237.74023156  126.10692873
  355.09707776  594.7759879   836.53683406  405.4637297   797.62429993
 1176.50347517 3019.58059808 3912.60062742]
NH
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[965.88117934 836.4898093  563.40224721 277.80676276 146.94827034
  92.86117054 129.43415022 139.58644872 129.96982231 116.66581881
 114.44821477 108.83291799  58.56522991   5.34770347  30.79442482
  62.78863786  79.34019732  78.80701113  56.78431267  45.76202503
  41.8108793   38.43100411  42.57682907  39.47151914  24.50251341
  24.56377547  21.33298045  24.56377547  24.50251341  39.47151914
  42.57682907  38.43100411  41.8108793   45.76202503  56.78431267
  78.80701113  79.34019732  62.78863786  30.79442482   5.34770347
  58.56522991 108.83291799 114.44821477 116.66581881 129.96982231
 139.58644872 129.43415022  92.86117054 146.94827034 277.80676276
 563.40224721 836.4898093  965.88117934]
NJ
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[10926.72383008 10904.52690105  9239.16651427  3135.13325007
  1302.48245607  1364.08408767  3313.52479808  2862.10944213
  1746.92085136  1116.65949432  1287.83630977  1300.44839123
  1209.67730658   600.61102159   672.85245898   493.00620194
   806.04272509   944.51302874   748.90272888   473.06551964
   615.02284976   656.84841191   954.60330337   887.12717965
   687.34922556   581.9676491    719.66666667   581.9676491
   687.34922556   887.12717965   954.60330337   656.84841191
   615.02284976   473.06551964   748.90272888   944.51302874
   806.04272509   493.00620194   672.85245898   600.61102159
  1209.67730658  1300.44839123  1287.83630977  1116.65949432
  1746.92085136  2862.10944213  3313.52479808  1364.08408767
  1302.48245607  3135.13325007  9239.16651427 10904.52690105
 10926.72383008]
NM
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[3791.61971949 3101.49539114 1699.05247703  403.12977468  195.83636988
  626.40939504  651.92483514  612.0605313   438.82418363  127.17397148
   71.04371525  218.53802583  301.17306982  205.05802978  138.22960904
  176.00920791  201.68825677  272.69057873  258.95395619  183.91611101
  104.11941879  153.18640374  220.84343586  217.06970122  237.01641581
  110.24130605   33.17312437  110.24130605  237.01641581  217.06970122
  220.84343586  153.18640374  104.11941879  183.91611101  258.95395619
  272.69057873  201.68825677  176.00920791  138.22960904  205.05802978
  301.17306982  218.53802583   71.04371525  127.17397148  438.82418363
  612.0605313   651.92483514  626.40939504  195.83636988  403.12977468
 1699.05247703 3101.49539114 3791.61971949]
NV
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[5142.63806554 4398.99995457 3283.10145568  628.54109102  592.35781142
  550.84364889  698.93141971  805.43618002  563.91640631  416.5014994
  454.45644974  322.81861229  400.49615164  200.17285774  148.59399532
  241.20612596  121.28772934  294.61220239  135.30639242  141.9490577
  183.09515932  258.77399542  362.04262017  479.45072132  269.24517987
  243.7991069   117.37037037  243.7991069   269.24517987  479.45072132
  362.04262017  258.77399542  183.09515932  141.9490577   135.30639242
  294.61220239  121.28772934  241.20612596  148.59399532  200.17285774
  400.49615164  322.81861229  454.45644974  416.5014994   563.91640631
  805.43618002  698.93141971  550.84364889  592.35781142  628.54109102
 3283.10145568 4398.99995457 5142.63806554]
```

```
NY
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[8565.41086551 6813.39512225 7581.02222694 4378.53158778 1287.97071068
 1125.11264513 2238.30257814 2424.78917768 1571.64003605  883.94615432
  825.6657491   941.74231126  606.16564422  466.21268423  470.58582099
  367.60179641  627.20284929  602.64093552  440.88967708  461.51499613
  756.01569747  725.83406047  540.95403311  527.06107252  613.12315353
  462.72932475  352.96741129  462.72932475  613.12315353  527.06107252
  540.95403311  725.83406047  756.01569747  461.51499613  440.88967708
  602.64093552  627.20284929  367.60179641  470.58582099  466.21268423
  606.16564422  941.74231126  825.6657491   883.94615432 1571.64003605
 2424.78917768 2238.30257814 1125.11264513 1287.97071068 4378.53158778
 7581.02222694 6813.39512225 8565.41086551]
NYC
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[1103.62559657  817.95626452 1296.9768627   915.66086997  362.61069031
   58.10298687  412.16818486  513.32752878  355.04214735  178.20958173
  106.21667235  180.02142391  175.98386137  127.12577066   89.76809392
   82.59213111   79.87135934   66.03144971   87.253199    131.89846229
  123.85766053   77.11909229   74.05378403  115.21836011  126.71053154
   94.3643609    56.24405589   94.3643609   126.71053154  115.21836011
   74.05378403   77.11909229  123.85766053  131.89846229   87.253199
   66.03144971   79.87135934   82.59213111   89.76809392  127.12577066
  175.98386137  180.02142391  106.21667235  178.20958173  355.04214735
  513.32752878  412.16818486   58.10298687  362.61069031  915.66086997
 1296.9768627   817.95626452 1103.62559657]
OH
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[21430.00845029 16861.14724897 10775.39975877  3194.92911098
   568.03205212  2424.37091894  2633.46788434  2854.49301882
  2017.5784515   1742.79032985  1971.59324845  1592.06280592
  1090.43113429   492.96448241   597.26462994  1063.39143645
   894.4873444    964.71018342   508.07755391   355.78174557
   837.82518316  1375.32981707  2095.21469837  1744.58265868
  1377.54653093  1085.7731998    375.03703704  1085.7731998
  1377.54653093  1744.58265868  2095.21469837  1375.32981707
   837.82518316   355.78174557   508.07755391   964.71018342
   894.4873444   1063.39143645   597.26462994   492.96448241
  1090.43113429  1592.06280592  1971.59324845  1742.79032985
  2017.5784515   2854.49301882  2633.46788434  2424.37091894
   568.03205212  3194.92911098 10775.39975877 16861.14724897
 21430.00845029]
OK
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[8642.65618425 5279.69612744 3601.99568786 1719.66822724  902.21507017
 1556.60520217 1698.08203133 1161.13922937  816.78661838  346.5804335
  818.51201871  828.84697434  833.54142694  477.65756285  216.21952931
  261.31628906  501.67156079  453.71271495  613.03372594  449.97646796
  698.28500578  503.56678697  745.05942497  695.5756235   731.48301185
  329.08709471  252.48148148  329.08709471  731.48301185  695.5756235
  745.05942497  503.56678697  698.28500578  449.97646796  613.03372594
  453.71271495  501.67156079  261.31628906  216.21952931  477.65756285
  833.54142694  828.84697434  818.51201871  346.5804335   816.78661838
 1161.13922937 1698.08203133 1556.60520217  902.21507017 1719.66822724
 3601.99568786 5279.69612744 8642.65618425]
OR
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[3025.32552462 2281.90628943 1532.43098191  432.3716243    93.67627795
  339.93290208  496.79929576  449.76368983  339.54371667  250.05500433
  244.70527372  201.24626112   89.34023471   56.86385379    9.46002901
   23.40828587   82.0735042   113.05474717  141.67637086  136.82297334
  138.73638778  141.22269754  123.25576778  146.27293061  148.83497874
   88.54883759   35.51851852   88.54883759  148.83497874  146.27293061
  123.25576778  141.22269754  138.73638778  136.82297334  141.67637086
  113.05474717   82.0735042    23.40828587    9.46002901   56.86385379
   89.34023471  201.24626112  244.70527372  250.05500433  339.54371667
  449.76368983  496.79929576  339.93290208   93.67627795  432.3716243
 1532.43098191 2281.90628943 3025.32552462]
PA
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[16501.82232772 14186.59660525 10221.62767293  3591.3478182
  1695.58822686   531.93229751  1608.25031421  2093.56781283
  1789.54686428  1588.58304522  1846.43593574  1816.72152982
  1171.70991701   570.54746198   674.22066705   726.22101675
   955.0026778    815.78384957   451.24911946   254.06861784
   549.03804545   800.07712992   929.59503191   849.80177403
   897.86332395   730.83481915   539.6964418    730.83481915
   897.86332395   849.80177403   929.59503191   800.07712992
   549.03804545   254.06861784   451.24911946   815.78384957
   955.0026778    726.22101675   674.22066705   570.54746198
  1171.70991701  1816.72152982  1846.43593574  1588.58304522
  1789.54686428  2093.56781283  1608.25031421   531.93229751
  1695.58822686  3591.3478182  10221.62767293 14186.59660525
 16501.82232772]
PR
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[2175.63393207  990.39953637  934.03641484  227.55414979  266.24365607
   70.58075488  276.65931213  113.43277941  202.63045215  217.77525675
   65.68100117  235.56985612  293.60576958  224.72976315   27.00904962
  155.78803328  116.92391798   39.9013186    41.06855834  106.9338783
  174.87475047  219.08387183  197.55702929   71.32172464  129.64603589
  163.8950502    22.48148148  163.8950502   129.64603589   71.32172464
  197.55702929  219.08387183  174.87475047  106.9338783    41.06855834
   39.9013186   116.92391798  155.78803328   27.00904962  224.72976315
  293.60576958  235.56985612   65.68100117  217.77525675  202.63045215
  113.43277941  276.65931213   70.58075488  266.24365607  227.55414979
  934.03641484  990.39953637 2175.63393207]
PW
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0.]
RI
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[1434.94224072 1299.0426196   795.63203807   31.03967883   21.21744726
  187.09401626  286.28234211  195.79895325  146.14829405  148.68240226
  128.64818538   42.30041733   38.1358657    74.23445885  117.19585534
  126.55800087  118.00716291   79.92220585   26.27543667   30.67430159
   41.36848473   62.64485962   70.36249004   81.06114055   73.88141312
   49.37265919   51.98103196   49.37265919   73.88141312   81.06114055
   70.36249004   62.64485962   41.36848473   30.67430159   26.27543667
   79.92220585  118.00716291  126.55800087  117.19585534   74.23445885
   38.1358657    42.30041733  128.64818538  148.68240226  146.14829405
  195.79895325  286.28234211  187.09401626   21.21744726   31.03967883
  795.63203807 1299.0426196  1434.94224072]
RMI
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[0.14646336 0.14150022 0.13353894 0.12307162 0.11085224 0.09799079
 0.08609604 0.07735196 0.07407407 0.07735196 0.08609604 0.09799079
 0.11085224 0.12307162 0.13353894 0.14150022 0.14646336 0.14814815
 0.14646336 0.14150022 0.13353894 0.12307162 0.11085224 0.09799079
 0.08609604 0.07735196 0.07407407 0.07735196 0.08609604 0.09799079
 0.11085224 0.12307162 0.13353894 0.14150022 0.14646336 0.14814815
 0.14646336 0.14150022 0.13353894 0.12307162 0.11085224 0.09799079
 0.08609604 0.07735196 0.07407407 0.07735196 0.08609604 0.09799079
 0.11085224 0.12307162 0.13353894 0.14150022 0.14646336]
SC
Wavelengths:
```

```
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[4.22234909e+03 4.75237744e+03 2.55554356e+03 2.83158204e+03
 1.14648544e+03 1.61584235e+03 9.88694168e+02 1.00288583e+03
 1.01997038e+03 8.92938164e+02 7.94789386e+02 7.40526308e+02
 5.33685955e+02 2.36795096e+02 2.48368516e+02 3.87551329e+02
 4.24139679e+02 5.90278949e+02 2.92390926e+02 3.56479683e+02
 2.75533303e+02 3.29891719e+02 4.34611121e+02 3.69555428e+02
 2.95799936e+02 1.89149222e+02 1.89129032e+00 1.89149222e+02
 2.95799936e+02 3.69555428e+02 4.34611121e+02 3.29891719e+02
 2.75533303e+02 3.56479683e+02 2.92390926e+02 5.90278949e+02
 4.24139679e+02 3.87551329e+02 2.48368516e+02 2.36795096e+02
 5.33685955e+02 7.40526308e+02 7.94789386e+02 8.92938164e+02
 1.01997038e+03 1.00288583e+03 9.88694168e+02 1.61584235e+03
 1.14648544e+03 2.83158204e+03 2.55554356e+03 4.75237744e+03
 4.22234909e+03]
SD
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[1069.25183606  721.59157567  296.94723659  199.37187021  158.78272531
  152.07834644   88.04435818   28.29462335   19.93790224   44.31442786
   76.09325798   33.47753414    3.91232383   48.06851056   39.27955498
   55.82517062   25.88683443   40.88427376   36.49895137   51.81510914
   46.02061406   27.95989076   17.21023983   23.93651444   18.48587783
   20.17822893   17.32080246   20.17822893   18.48587783   23.93651444
   17.21023983   27.95989076   46.02061406   51.81510914   36.49895137
   40.88427376   25.88683443   55.82517062   39.27955498   48.06851056
    3.91232383   33.47753414   76.09325798   44.31442786   19.93790224
   28.29462335   88.04435818  152.07834644  158.78272531  199.37187021
  296.94723659  721.59157567 1069.25183606]
TN
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[10830.65641603  8283.91099249  6231.48169566  2920.4024771
  2635.16936094   929.92577435   641.37525526   477.80130298
   673.48068     1153.24980196  1186.62237689  1242.44630643
  1090.39322539   665.3214453    545.22318045   563.44120998
   514.15809653   436.207397     642.01910589  1075.97604664
  1216.3189405    927.77986828   581.56115733   373.44546953
   520.7488093    558.09212729   543.84289066   558.09212729
   520.7488093    373.44546953   581.56115733   927.77986828
  1216.3189405   1075.97604664   642.01910589   436.207397
   514.15809653   563.44120998   545.22318045   665.3214453
  1090.39322539  1242.44630643  1186.62237689  1153.24980196
   673.48068      477.80130298   641.37525526   929.92577435
  2635.16936094  2920.4024771   6231.48169566  8283.91099249
 10830.65641603]
TX
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[35072.20962815 34831.99730416 22746.13381804 11777.15420811
  8754.46816208  7161.98351333  7725.99826277  5279.37922447
  5454.32902596  5634.93851115  1797.86181494  5574.91746521
  2529.24775094  3102.89340813  1078.32899398  1411.16703923
   809.08109895  4097.55545941  2098.84947667  3808.73051597
  3198.25612777   979.82785511  2945.57033403  1748.04459542
  2806.29202472  1505.45654314   152.43937476  1505.45654314
  2806.29202472  1748.04459542  2945.57033403   979.82785511
  3198.25612777  3808.73051597  2098.84947667  4097.55545941
   809.08109895  1411.16703923  1078.32899398  3102.89340813
  2529.24775094  5574.91746521  1797.86181494  5634.93851115
  5454.32902596  5279.37922447  7725.99826277  7161.98351333
  8754.46816208 11777.15420811 22746.13381804 34831.99730416
 35072.20962815]
UT
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[6008.44620261 4027.46767899 1646.33178649  672.51330742  374.10039851
 1042.6652906  1092.0058392   751.36948706  296.8283095   312.96677214
  310.65446881  333.05759477  307.87272542  238.6789876   257.41447114
  313.83540647  253.21527143  306.8775577   215.50512328  252.36565971
  319.96261411  319.89241936  294.21767166  295.0213713   258.43168276
  177.96774042    8.75251371  177.96774042  258.43168276  295.0213713
  294.21767166  319.89241936  319.96261411  252.36565971  215.50512328
  306.8775577   253.21527143  313.83540647  257.41447114  238.6789876
  307.87272542  333.05759477  310.65446881  312.96677214  296.8283095
  751.36948706 1092.0058392  1042.6652906   374.10039851  672.51330742
 1646.33178649 4027.46767899 6008.44620261]
VA
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[5982.25244066 5740.79515326 4384.03506739 2037.3523543  2066.62215338
 1657.6725084  1486.44936902 1636.51636043 1377.77643415 1218.84857767
 1125.10651517  977.6380868   793.39362296  586.80998368  466.11166141
  582.65034811  548.02401788  692.27613721  621.73315158  437.33064335
  344.15533377  366.25672589  544.7367658   403.74659294  468.81503691
  330.51330412  137.62664499  330.51330412  468.81503691  403.74659294
  544.7367658   366.25672589  344.15533377  437.33064335  621.73315158
  692.27613721  548.02401788  582.65034811  466.11166141  586.80998368
  793.39362296  977.6380868  1125.10651517 1218.84857767 1377.77643415
 1636.51636043 1486.44936902 1657.6725084  2066.62215338 2037.3523543
 4384.03506739 5740.79515326 5982.25244066]
VI
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[12.44596941 10.55538969 14.94574324  3.18528422  3.24774078  1.68353365
  3.13227781  3.27242798  3.01904679  6.3524965   1.10327975  3.84776365
  2.22334583  1.58138049  0.29710484  1.9595758   0.95055312  1.76895683
  0.26165937  1.44829013  0.55896064  1.40260179  1.06858042  1.48049937
  1.22156002  2.14973675  0.40775249  2.14973675  1.22156002  1.48049937
  1.06858042  1.40260179  0.55896064  1.44829013  0.26165937  1.76895683
  0.95055312  1.9595758   0.29710484  1.58138049  2.22334583  3.84776365
  1.10327975  6.3524965   3.01904679  3.27242798  3.13227781  1.68353365
  3.24774078  3.18528422 14.94574324 10.55538969 12.44596941]
VT
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[246.662728   191.73050915 142.86970213  76.52363125  53.83458057
  47.56426266  73.81698887  79.47338169  50.95410767  30.45788978
  31.38451807  51.69699437  39.63121985  16.38478375   5.71918647
  10.03702328  23.93032043  23.31515452  14.56952475   2.92048179
  12.77356861  17.7458808   25.23195661  24.88450701  23.36002621
  14.29294278   0.71683852  14.29294278  23.36002621  24.88450701
  25.23195661  17.7458808   12.77356861   2.92048179  14.56952475
  23.31515452  23.93032043  10.03702328   5.71918647  16.38478375
  39.63121985  51.69699437  31.38451807  30.45788978  50.95410767
  79.47338169  73.81698887  47.56426266  53.83458057  76.52363125
 142.86970213 191.73050915 246.662728  ]
WA
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[5547.80345695 4665.27068875 3332.68254612  907.66288361  130.31403317
  857.21146668 1062.45791907 1103.24773293  713.1817721   486.14709495
  376.85745861  399.77196648  371.48731659   99.90256126  152.67210804
  187.45825252  237.19096169  252.42792232  279.76666843  167.8066631
  272.01783429  326.56060279  514.48459234  482.96720028  386.61097959
  226.65053759   55.51851852  226.65053759  386.61097959  482.96720028
  514.48459234  326.56060279  272.01783429  167.8066631   279.76666843
  252.42792232  237.19096169  187.45825252  152.67210804   99.90256126
  371.48731659  399.77196648  376.85745861  486.14709495  713.1817721
 1103.24773293 1062.45791907  857.21146668  130.31403317  907.66288361
 3332.68254612 4665.27068875 5547.80345695]
WI
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[13052.64553938  7819.7277068   3380.43384039  1700.781966
  1443.1114676   1806.43364894  1536.03308357   730.81400907
   162.49411273   432.4690934    634.20533698  1013.6525241
  1184.87070471   863.28731715   751.90526722   602.8861846
```

```
   261.6595915    152.84589175   365.54942479   284.74643374
   189.5625737    554.18042102   808.98602508   662.04505385
   253.67901997    63.79604934   278.65385074    63.79604934
   253.67901997   662.04505385   808.98602508   554.18042102
   189.5625737    284.74643374   365.54942479   152.84589175
   261.6595915    602.8861846    751.90526722   863.28731715
  1184.87070471  1013.6525241    634.20533698   432.4690934
   162.49411273   730.81400907  1536.03308357  1806.43364894
  1443.1114676   1700.781966    3380.43384039  7819.7277068
 13052.64553938]
WV
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[1030.59507396  723.62946687  488.34845378  243.12446839  128.89085059
  146.97847756  145.7760981   117.41100124   80.69715908   53.92715132
   59.6030044    58.14686883   35.44557108    7.15985733   30.74169014
   36.54376582   53.12454258   39.8809942    44.28551825   36.82131843
   55.78506888   55.3119531    68.67301053   44.48411931   41.48046655
   25.37091993    1.73871932   25.37091993   41.48046655   44.48411931
   68.67301053   55.3119531    55.78506888   36.82131843   44.28551825
   39.8809942    53.12454258   36.54376582   30.74169014    7.15985733
   35.44557108   58.14686883   59.6030044    53.92715132   80.69715908
  117.41100124  145.7760981   146.97847756  128.89085059  243.12446839
  488.34845378  723.62946687 1030.59507396]
WY
Wavelengths:
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[1018.12698049  692.93940512  394.90900941  200.96237806  204.24050892
  222.57067153  187.7419337   116.82951046   27.93335079   32.303842
   53.19795448   65.56281877   72.45900352   38.36869381   23.45187006
   29.11253272   25.26202165   23.65599987   17.05165156    1.34614303
   18.76395163   33.49864222   43.94563419   45.2848553    33.48691269
   17.51430266    6.85404895   17.51430266   33.48691269   45.2848553
   43.94563419   33.49864222   18.76395163    1.34614303   17.05165156
   23.65599987   25.26202165   29.11253272   23.45187006   38.36869381
   72.45900352   65.56281877   53.19795448   32.303842     27.93335079
  116.82951046  187.7419337   222.57067153  204.24050892  200.96237806
  394.90900941  692.93940512 1018.12698049]
0
Wavelengths:
```

```
[ 54.          27.          18.          13.5         10.8
   9.           7.71428571   6.75         6.           5.4
   4.90909091   4.5          4.15384615   3.85714286   3.6
   3.375        3.17647059   3.           2.84210526   2.7
   2.57142857   2.45454545   2.34782609   2.25         2.16
   2.07692308  -2.          -2.07692308  -2.16        -2.25
  -2.34782609  -2.45454545  -2.57142857  -2.7         -2.84210526
  -3.          -3.17647059  -3.375       -3.6         -3.85714286
  -4.15384615  -4.5         -4.90909091  -5.4         -6.
  -6.75        -7.71428571  -9.         -10.8        -13.5
 -18.         -27.         -54.        ]
Amps:
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0.]
```

In [17]:
```
#Algorithm 2. SIMILAR WAVES
#Given: deltas_it, for i = 1, …, 51, t = 1, …, 100.  // error from step 1
#1. Set s_it= deltas_it
#2. For i = 1 to 51 // loop over states
#3.     Determine y_i, for s = 1, …, 52 = SolveIDP(ODEint(s_i)).y
#        Determine errors_it = y.error
# Set [z_1,38,…,z_51,38]=[s_1,38,…,s_51,38]
#5. For t = 1 to 53 // loop over weeks in 2021
#6.       Calculate [z_(1,38+t),…,z_(51,38+t)]=interpolate(y[0],y[1],[z_(1,37+t),…,z_
#7. Set Δ=z-s // Calculate prediction error
```

In [18]:
```
# End Fourier Xform secton
```

In [19]:
```
# Reverse Chaos
```

In [20]:
```
curveFitX = s_it.copy()
stdDeviationsList = []


def interpolatingFunctionSolver(x, a, b, c):
    return a * np.sin(b * x) + c
    #print(amplitude, frequency, phase)

print("defined")
```

```
defined
```

```
In [21]: for s in curveFitX: # Loop in states. Note s is the postal abbreviation here.
             if s not in ("AR", "RI", "AK", "FL", 0):
                 # Python does not like Rhode Island, Alaska, Arkansas, or Florida. Does not 
                 # Also getting rid of the Outside state with postal abbreviation " 0 ".
                 s_i = curveFitX[s]
                 sizeOf = s_i.size #in case we change interpolate distance later
                 xVars = range(sizeOf)
                 yVars = np.array(s_i)

                 tuples, covarienceMatrix = sp.optimize.curve_fit(interpolatingFunctionSolver
                 # maxfev is the number of iterations until convergence is reached before hal
                 # that there is no convergence in the data.
                 stdDeviation = np.sqrt(np.diag(covarienceMatrix))

                 #extract tuples
                 amplitude, frequency, phase = tuples
                 amplitude = abs(amplitude)

                 print(s, "Amplitude:", amplitude, "Frequency:", frequency, "Phase:", phase)
                 print(stdDeviation)
                 stdDeviationsList.append(sum(stdDeviation) )
```

```
AL Amplitude: 593.3759046556825 Frequency: 1.549307871571894 Phase: 8319.24131964569
[1.60447574e+03 8.65124849e-02 1.12619087e+03]
AS Amplitude: 3.0184440937519395e-20 Frequency: 1.0000000000005007 Phase: 1.592417287
947139e-19
[3.22576864e-20 7.50761717e-13 2.26768726e-20]
AZ Amplitude: 872.7318200034628 Frequency: 5.459878925951252 Phase: 4420.402846111828
[1.10609758e+03 4.08859697e-02 7.77680135e+02]
CA Amplitude: 469.53647759918385 Frequency: 3.2097543158251676 Phase: 34276.976066814
845
[1.02358640e+04 5.69625811e-01 6.74558351e+03]
CO Amplitude: 591.9924666665868 Frequency: 9.422578453362302 Phase: 4949.345385297252
[1.01532080e+07 4.11233755e+01 9.29127435e+02]
CT Amplitude: 381.08408702191895 Frequency: 4.5605449599084515 Phase: 4585.6751206343
95
[1.13577507e+03 9.49630707e-02 7.96381415e+02]
DC Amplitude: 51.615688207640964 Frequency: 1.0621453356738495 Phase: 321.10924586234
46
[5.48485023e+01 3.40186919e-02 3.84930557e+01]
DE Amplitude: 27.091971055245754 Frequency: 2.8775651419227883 Phase: 1362.6390917366
434
[318.21153192   0.38217255 223.4467797 ]
FSM Amplitude: 0.03679282578513693 Frequency: 1.0166323757462619 Phase: 0.01751876591
4710455
[0.02648425 0.02284368 0.01855956]
GA Amplitude: 149.09219232356324 Frequency: -2.3090640296313443 Phase: 16257.13884803
7637
[3.38260415e+03 7.18240922e-01 2.36763540e+03]
GU Amplitude: 5.680602337062402 Frequency: 1.3007768756165907 Phase: 36.8692723512063
4
[10.44767143  0.05862229  7.32486727]
HI Amplitude: 18.312179673014974 Frequency: 1.7008214838719968 Phase: 154.84546595512
41
[31.94523173  0.05668957 22.50413292]
IA Amplitude: 406.1198923988358 Frequency: 9.241051631488979 Phase: 5175.826294312551
[1.22011134e+03 9.01555493e-02 8.41520982e+02]
ID Amplitude: 234.19266538858412 Frequency: 1.4373817924252172 Phase: 2468.2517772632
95
[5.23668886e+02 7.22396302e-02 3.68490824e+02]
```

```
IL Amplitude: 1489.1298343318747 Frequency: 4.704808720470669 Phase: 15410.8465976793
76
[3.28970578e+03 7.24332069e-02 2.32283725e+03]
IN Amplitude: 1144.2342786274737 Frequency: 7.592037756894067 Phase: 11432.2923832268
92
[2.79393746e+03 7.76368580e-02 1.95798151e+03]
KS Amplitude: 589.1091475285235 Frequency: 7.853790568550572 Phase: 5081.577167848025
[1.15928316e+03 6.49198833e-02 8.19947075e+02]
KY Amplitude: 2702.562630833312 Frequency: 12.271360762595185 Phase: 6375.1497140394
[1.62764219e+03 1.89578170e-02 1.14166328e+03]
LA Amplitude: 274.3704748181654 Frequency: 10.645459504800227 Phase: 4029.68523914753
5
[7.18595571e+02 8.64708670e-02 5.08321545e+02]
MA Amplitude: 374.5043851356571 Frequency: 2.321262995828477 Phase: 9436.391134930249
[2.33201288e+03 2.02685478e-01 1.64361535e+03]
MD Amplitude: 1110.310360427106 Frequency: 7.109649345223755 Phase: 5916.113408630476
[1.08207040e+03 3.11913679e-02 7.59336065e+02]
ME Amplitude: 108.53548812432823 Frequency: 1.3647756244465956 Phase: 672.98818681408
45
[2.13143850e+02 6.24414576e-02 1.49347134e+02]
MI Amplitude: 139.14895212422752 Frequency: 8.941352988735623 Phase: 6284.43764616016
1
[1.56320718e+03 3.49723930e-01 1.08933180e+03]
MN Amplitude: 161.76844563537364 Frequency: 14.778465816740608 Phase: 7739.9052383641
93
[2.09348479e+03 4.27820229e-01 1.48128714e+03]
MO Amplitude: 596.3156578770296 Frequency: 5.128855967968418 Phase: 9496.390214431965
[2.00660719e+03 1.10917898e-01 1.41924943e+03]
MP Amplitude: 0.5147131762346421 Frequency: 1.0499673692672558 Phase: 2.4438261367337
55
[0.49170359 0.03138997 0.34735833]
MS Amplitude: 152.3959375162408 Frequency: 1.748975591943349 Phase: 4320.302778758395
[7.94313799e+02 1.71982825e-01 5.61750817e+02]
MT Amplitude: 12.342267643826561 Frequency: 7.50163899570433 Phase: 1484.800127866175
2
[391.92811881   1.04875573 277.3243581 ]
NC Amplitude: 1081.2204592818807 Frequency: 5.630268630617744 Phase: 10946.0891893054
94
[2.39163003e+03 6.99402571e-02 1.67488507e+03]
ND Amplitude: 109.82896981615387 Frequency: 2.7010298848114562 Phase: 1331.7710758363
892
[3.70067648e+02 1.09306738e-01 2.60323170e+02]
NE Amplitude: 1005.7804298041634 Frequency: 5.963767080303897 Phase: 3225.86035735340
37
[7.44324780e+02 2.36467144e-02 5.26218285e+02]
NH Amplitude: 137.22072270469334 Frequency: 0.939207755963404 Phase: 681.097396461086
[2.06981349e+02 4.88642988e-02 1.45679663e+02]
NJ Amplitude: 657.1149657548731 Frequency: 2.9382722894175446 Phase: 12523.0093959504
14
[2.73596585e+03 1.32811573e-01 1.90953282e+03]
NM Amplitude: 74.47447918686092 Frequency: 15.667044919981326 Phase: 2916.38471504020
6
[6.91674182e+02 5.94494338e-01 5.34565513e+02]
NV Amplitude: 913.8162443953474 Frequency: 6.770707411499629 Phase: 5085.342147593719
[1.09839475e+03 3.76024844e-02 7.67742873e+02]
NY Amplitude: 192.17852140184976 Frequency: 7.774934532403679 Phase: 7980.80537528076
5
[2.09024916e+03 3.48550538e-01 1.46788785e+03]
NYC Amplitude: 92.72412193055976 Frequency: 2.3470409130004346 Phase: 1369.2085185901
756
[3.24567557e+02 1.13940222e-01 2.28671152e+02]
OH Amplitude: 442.1572045280467 Frequency: 1.9262861611415758 Phase: 16257.6018885057
```

```
82
[4.25445532e+03 3.17557839e-01 3.00876723e+03]
OK Amplitude: 447.341076659758 Frequency: 8.856480409829723 Phase: 7030.680550537367
[1.60670385e+03 1.20086517e-01 1.13963354e+03]
OR Amplitude: 50.7105571088771 Frequency: 7.916874664534869 Phase: 2593.863315133426
[5.93058507e+02 3.85234140e-01 4.19219348e+02]
PA Amplitude: 11741581.660781732 Frequency: 7.269754435343402e-05 Phase: -9160.022801
48443
[7.16322636e+11 4.46046475e+00 4.20408683e+03]
PR Amplitude: 183.9475273454338 Frequency: -2.6405977450101346 Phase: 1711.5373063757
98
[3.78982710e+02 6.72789408e-02 2.67105488e+02]
PW Amplitude: 3.0184440937519395e-20 Frequency: 1.0000000000005007 Phase: 1.592417287
947139e-19
[3.22576864e-20 7.50761717e-13 2.26768726e-20]
RMI Amplitude: 0.08015318395195978 Frequency: 0.9887987062484072 Phase: 0.07681868322
630851
[0.08329162 0.034259   0.05891122]
SC Amplitude: 6297.297822413428 Frequency: 0.08619876682517952 Phase: 6735.0328621526
1
[1.04750662e+03 6.64162839e-03 9.46408230e+02]
SD Amplitude: 21.381083213680714 Frequency: 2.0661704214497663 Phase: 761.71293760192
68
[193.65674619   0.2882305  135.64855841]
TN Amplitude: 8420.451531620933 Frequency: -3.1405242497530597 Phase: 9974.9913909766
26
[1.56092403e+08 2.08987991e+01 1.57913369e+03]
TX Amplitude: 1012.281261004448 Frequency: 1.962031049262049 Phase: 40376.24766329864
[8.34453164e+03 2.64142463e-01 5.85899140e+03]
UT Amplitude: 402.72949880656137 Frequency: 10.849877177258838 Phase: 4878.4239075426
73
[1.09303402e+03 8.62466174e-02 7.65846455e+02]
VA Amplitude: 6209116.472001418 Frequency: 6.360695082183843e-05 Phase: -3684.1638988
961504
[4.02106496e+11 4.14215449e+00 1.57847396e+03]
VI Amplitude: 2.6321304588600336 Frequency: 0.9399069933524776 Phase: 14.654731073651
098
[3.50941489 0.04311611 2.46894454]
VT Amplitude: 44.5893606671943 Frequency: 1.421283208287651 Phase: 181.1700008268364
[5.50654524e+01 3.92791769e-02 3.85881884e+01]
WA Amplitude: 141.71065799010412 Frequency: 9.472406661079695 Phase: 5661.93196626577
6
[1.11343795e+03 3.58739311e-01 8.28122626e+02]
WI Amplitude: 6711.595545347998 Frequency: 6.101217488679808 Phase: 8222.70976069831
[2.21726569e+03 9.68315693e-03 1.54472713e+03]
WV Amplitude: 46.45182482663391 Frequency: 3.8214287802880595 Phase: 705.815193638226
2
[1.98194542e+02 1.40173274e-01 1.39943863e+02]
WY Amplitude: 20.419263186101457 Frequency: 4.191692341793655 Phase: 658.032485958739
9
[192.12120168   0.30922427 135.74330061]
```

In [22]:

```
avgError = sum(stdDeviationsList) / (len(stdDeviationsList) )
print(avgError)
```

```
19974919782.733772
```

In [23]:
```
y = 0

for x in stdDeviationsList:
    y = y+1
    if x > (2 * avgError):
        print(y, " : ", x)
```

```
41  :  716322640344.3839
50  :  402106497468.12103
```

In [24]:
```
# 41 here = PA
# 50 here = VA
# Potential mutation. redo on these, halved.
```

In [25]:
```
for s2 in curveFitX: # loop in states. Note s is the postal abbreviation here.
    if s2 in ("VA", "PA"):
        s_i2 = curveFitX[s2]
        sizeOf2 = s_i.size / 2.0 #div 2
        xVars2 = range(int(sizeOf2) )
        yVars2 = np.array(s_i2[:int(sizeOf2)])

        tuples, covarienceMatrix = sp.optimize.curve_fit(interpolatingFunctionSolver,
        stdDeviation = np.sqrt(np.diag(covarienceMatrix))

        #extract tuples
        amplitude, frequency, phase = tuples
        amplitude = abs(amplitude)

        print(s2, "Amplitude:", amplitude, "Frequency:", frequency, "Phase:", phase)
        print(stdDeviation)
```

```
PA Amplitude: 1076.8156059971875 Frequency: 19.788424907358817 Phase: 3498.1142180757
706
[8.83553897e+02 5.48489560e-02 6.22250116e+02]
VA Amplitude: 117.15661725974043 Frequency: 28.57954650939507 Phase: 2230.08914940454
22
[5.69661222e+02 3.40581992e-01 4.02758238e+02]
```

In [26]:
```
# pre-half only has an amplitude of 117 and 1k. Interesting. Let's look at post-half
```

In [27]:
```
for s3 in curveFitX: # loop in states. Note s is the postal abbreviation here.
    if s3 in ("VA", "PA"):
        s_i3 = curveFitX[s3]
        sizeOf3 = s_i.size / 2.0 #div 2
        xVars3 = range(int(sizeOf3), int(sizeOf3 * 2.0) )
        yVars3 = np.array(s_i3[:int(sizeOf2)])

        tuples, covarienceMatrix = sp.optimize.curve_fit(interpolatingFunctionSolver,
        stdDeviation = np.sqrt(np.diag(covarienceMatrix))

        #extract tuples
        amplitude, frequency, phase = tuples
        amplitude = abs(amplitude)

        print(s2, "Amplitude:", amplitude, "Frequency:", frequency, "Phase:", phase)
        print(stdDeviation)
```

```
0 Amplitude: 108.90959173826906 Frequency: 8.95446934608311 Phase: 3501.6365029644026
[9.11417582e+02 2.03740152e-01 6.41706079e+02]
0 Amplitude: 504.9217207232555 Frequency: -0.5515007548316523 Phase: 2227.59025798185
5
[5.76067746e+02 2.70128424e-02 4.02548062e+02]
```

In [28]:
```
# These two states have problems only because the amplitude is trivial; 108 cases or
# Similar to Alaska, but for different reasons. A mutation of impulse 108 can safely
```

In [29]:
```
# Following code is no longer used:
```

for s3 in curveFitX: # loop in states. Note s is the postal abbreviation here. if s3 in ("VA", "PA"):

s_i3 = curveFitX[s] sizeOf3 = s_i.size / 4.0 #div 2 xVars3 = range(int(sizeOf *3.0), int(sizeOf* 4.0))

yVars3 = np.array(s_i)

```
        tuples, covarienceMatrix =
    sp.optimize.curve_fit(interpolatingFunctionSolver, xVars3, yVars3,
    maxfev=2000)
        stdDeviation = np.sqrt(np.diag(covarienceMatrix))

        #extract tuples
        amplitude, frequency, phase = tuples
        amplitude = abs(amplitude)

        print(s2, "Amplitude:", amplitude, "Frequency:", frequency,
    "Phase:", phase)
        print(stdDeviation)
```

for s3 in curveFitX: # loop in states. Note s is the postal abbreviation here. if s3 in ("VA", "PA"):

s_i3 = curveFitX[s] sizeOf3 = s_i.size / 4.0 #div 2 xVars3 = range(int(sizeOf *2.0), int(sizeOf* 3.0))

yVars3 = np.array(s_i)

```
        tuples, covarienceMatrix =
    sp.optimize.curve_fit(interpolatingFunctionSolver, xVars3, yVars3,
    maxfev=2000)
        stdDeviation = np.sqrt(np.diag(covarienceMatrix))

        #extract tuples
        amplitude, frequency, phase = tuples
        amplitude = abs(amplitude)

        print(s2, "Amplitude:", amplitude, "Frequency:", frequency,
    "Phase:", phase)
        print(stdDeviation)
```

In [30]:
```
#Algorithm 3. REVERSE CHAOS
#Given: s_it  for i = 1, …, 51, t = 1, …, 53 s // s states from SIMILAR WAVES
#1. errorBaseline = 2*Average(〖error〗_it)
#2. For i = 1 to 51 // loop over states
#3.     if (〖error〗_i) ) > errorbaseline) )
#4.             perform a single SIMILAR_WAVES(s[i], t/2)
#5. Set [z_1,38,…,z_51,38]=[s_1,38,…,s_51,38]
# cut in half.
```

```
In [ ]:
```

```
In [31]: # The following 2 sections are no longer being used, but variables are being initiali
```

```
In [32]: restartParadigm = 0.01
         # This is no longer used. Cleaning is now done after each step instead of a second rou
         # other way could concevebly work too, but this way is logistically simpler due to th
         # of Jupyter Notebook.

         X = actual.copy()
         X = np.array(X)
         delta = np.sum(X)

         #Python is case sensitive
         #transpose uses T

         denominator = np.sum(np.array(states.copy() ) )

         deltaNormed = delta / denominator

         if deltaNormed >= restartParadigm:
             print("restart")
```

```
restart
```

```
In [33]: curr = 100
         first = 12
         second = 77
         third = 165
         CDC = np.array(range(1, 3) )

         for j in range(3, first):
             curr = curr**1.05
             CDC = np.append(CDC, curr)

         for j in range(first, second):
             curr = curr**1.0035
             CDC = np.append(CDC, curr)

         for j in range(second, third):
             curr = curr**0.999
             CDC = np.append(CDC, curr)

         print(CDC)
```

```
[1.00000000e+00 2.00000000e+00 1.25892541e+02 1.60324539e+02
 2.06656943e+02 2.69781708e+02 3.56913623e+02 4.78840947e+02
 6.51929818e+02 9.01386164e+02 1.26664943e+03 1.29872060e+03
 1.33172034e+03 1.36567852e+03 1.40062605e+03 1.43659492e+03
 1.47361827e+03 1.51173040e+03 1.55096681e+03 1.59136431e+03
 1.63296098e+03 1.67579628e+03 1.71991109e+03 1.76534775e+03
 1.81215013e+03 1.86036369e+03 1.91003553e+03 1.96121448e+03
 2.01395114e+03 2.06829795e+03 2.12430928e+03 2.18204151e+03
 2.24155309e+03 2.30290462e+03 2.36615895e+03 2.43138129e+03
 2.49863924e+03 2.56800295e+03 2.63954520e+03 2.71334148e+03
 2.78947016e+03 2.86801254e+03 2.94905301e+03 3.03267917e+03
 3.11898195e+03 3.20805575e+03 3.29999858e+03 3.39491223e+03
 3.49290241e+03 3.59407888e+03 3.69855567e+03 3.80645126e+03
 3.91788870e+03 4.03299586e+03 4.15190563e+03 4.27475609e+03
 4.40169077e+03 4.53285887e+03 4.66841548e+03 4.80852185e+03
 4.95334565e+03 5.10306123e+03 5.25784994e+03 5.41790039e+03
 5.58340877e+03 5.75457921e+03 5.93162408e+03 6.11476439e+03
 6.30423011e+03 6.50026061e+03 6.70310506e+03 6.91302285e+03
 7.13028404e+03 7.35516982e+03 7.58797304e+03 7.82899867e+03
 7.75912079e+03 7.68993556e+03 7.62143549e+03 7.55361318e+03
 7.48646134e+03 7.41997274e+03 7.35414024e+03 7.28895679e+03
 7.22441540e+03 7.16050920e+03 7.09723137e+03 7.03457516e+03
 6.97253393e+03 6.91110110e+03 6.85027015e+03 6.79003466e+03
 6.73038827e+03 6.67132470e+03 6.61283774e+03 6.55492126e+03
 6.49756917e+03 6.44077548e+03 6.38453427e+03 6.32883966e+03
 6.27368587e+03 6.21906715e+03 6.16497786e+03 6.11141238e+03
 6.05836518e+03 6.00583080e+03 5.95380381e+03 5.90227887e+03
 5.85125069e+03 5.80071404e+03 5.75066376e+03 5.70109473e+03
 5.65200189e+03 5.60338027e+03 5.55522491e+03 5.50753093e+03
 5.46029350e+03 5.41350786e+03 5.36716928e+03 5.32127309e+03
 5.27581468e+03 5.23078949e+03 5.18619301e+03 5.14202077e+03
 5.09826837e+03 5.05493145e+03 5.01200568e+03 4.96948682e+03
 4.92737064e+03 4.88565298e+03 4.84432971e+03 4.80339675e+03
 4.76285008e+03 4.72268571e+03 4.68289970e+03 4.64348814e+03
 4.60444719e+03 4.56577304e+03 4.52746191e+03 4.48951007e+03
 4.45191384e+03 4.41466958e+03 4.37777368e+03 4.34122258e+03
 4.30501274e+03 4.26914068e+03 4.23360296e+03 4.19839616e+03
 4.16351690e+03 4.12896187e+03 4.09472774e+03 4.06081127e+03
 4.02720923e+03 3.99391841e+03 3.96093568e+03 3.92825790e+03
 3.89588198e+03 3.86380488e+03 3.83202357e+03 3.80053506e+03
 3.76933640e+03 3.73842466e+03 3.70779697e+03 3.67745044e+03]
```

In [34]:

```
curveFitX2 = states.copy()
curveFitX2 = curveFitX2.transpose()


def solver2(t, b):
    return constT * (b ** t)

print("init")
```

```
init
```

In [35]:

```
startWeek = 40
# starting at 40 after the initial pulse for all states; otherwise we have a lot of e

for s in curveFitX2:
    yVars = pd.DataFrame(curveFitX2[s] )
    yVars = np.array(yVars).flatten()
    constT = yVars[startWeek]
    xVars = np.array(range(startWeek, len(yVars) + startWeek) )


    #print(xVars, yVars)
    tuples, covarience = sp.optimize.curve_fit(solver2, xVars, yVars, maxfev=2000)

    print(s, tuples, "covarience: ", covarience)
```

```
AK [0.99742655] covarience:  [[6.82770461e-07]]
AL [0.99851998] covarience:  [[5.61199041e-07]]
AR [0.99864187] covarience:  [[6.54734789e-07]]
AS [1.] covarience:  [[inf]]
AZ [1.00399865] covarience:  [[6.75153245e-07]]
CA [1.00506895] covarience:  [[7.71136656e-07]]
```

```
CO [0.99903979] covarience:  [[7.19894827e-07]]
CT [1.0020155] covarience:  [[6.20235319e-07]]
DC [1.00492157] covarience:  [[1.03729129e-06]]
DE [1.0040935] covarience:  [[6.44089471e-07]]
FL [1.00402015] covarience:  [[5.73503729e-07]]
FSM [1.] covarience:  [[inf]]
GA [1.00271241] covarience:  [[5.94348335e-07]]
GU [0.99837739] covarience:  [[1.0755265e-06]]
HI [1.00834903] covarience:  [[9.00539325e-07]]
IA [0.9948715] covarience:  [[5.97873325e-07]]
ID [0.99479509] covarience:  [[5.06462971e-07]]
IL [0.99754509] covarience:  [[5.51695198e-07]]
IN [0.99776516] covarience:  [[5.652213e-07]]
KS [0.99818189] covarience:  [[7.77298839e-07]]
KY [1.00012177] covarience:  [[5.32185684e-07]]
LA [1.0041005] covarience:  [[7.22878272e-07]]
MA [1.00322123] covarience:  [[6.80795594e-07]]
MD [1.00293611] covarience:  [[6.27299069e-07]]
ME [1.01001436] covarience:  [[5.24493589e-07]]
MI [1.00025822] covarience:  [[4.36343718e-07]]
MN [0.99872801] covarience:  [[5.16070732e-07]]
MO [0.99704378] covarience:  [[5.10007817e-07]]
MP [1.01709947] covarience:  [[2.29218325e-06]]
MS [1.0004246] covarience:  [[5.80322742e-07]]
MT [0.99200351] covarience:  [[7.88962833e-07]]
NC [1.00193899] covarience:  [[5.75741029e-07]]
ND [0.98927864] covarience:  [[1.29172075e-06]]
NE [0.9949347] covarience:  [[7.35088691e-07]]
NH [1.00688786] covarience:  [[6.31735632e-07]]
NJ [1.00274553] covarience:  [[5.96350762e-07]]
NM [0.998517] covarience:  [[5.51667553e-07]]
NV [0.99973574] covarience:  [[6.30784457e-07]]
NY [1.00610397] covarience:  [[5.95693371e-07]]
NYC [1.00804744] covarience:  [[9.43046916e-07]]
OH [1.00111568] covarience:  [[4.92666278e-07]]
OK [0.99950115] covarience:  [[6.3174937e-07]]
OR [1.00489266] covarience:  [[5.27983928e-07]]
PA [1.00239623] covarience:  [[4.88906358e-07]]
PR [1.00664855] covarience:  [[4.31129013e-07]]
PW [1.] covarience:  [[inf]]
RI [0.99995525] covarience:  [[8.25567281e-07]]
RMI [1.02720012] covarience:  [[9.09718373e-06]]
SC [1.00269407] covarience:  [[6.57884944e-07]]
SD [0.98631936] covarience:  [[1.5841277e-06]]
TN [0.99833908] covarience:  [[5.48592126e-07]]
TX [1.0007573] covarience:  [[5.42582855e-07]]
UT [0.99601722] covarience:  [[8.51804399e-07]]
VA [1.00361311] covarience:  [[4.32461744e-07]]
VI [1.01295321] covarience:  [[1.11862496e-06]]
VT [1.01087571] covarience:  [[9.38959423e-07]]
WA [1.00523572] covarience:  [[5.7118272e-07]]
WI [0.99286093] covarience:  [[9.8984376e-07]]
WV [1.00342283] covarience:  [[4.51261712e-07]]
WY [0.99427495] covarience:  [[7.79427197e-07]]
0 [1.] covarience:  [[0.]]
```

```
C:\Users\admin\anaconda3\lib\site-packages\scipy\optimize\minpack.py:828: OptimizeWar
ning: Covariance of the parameters could not be estimated
  warnings.warn('Covariance of the parameters could not be estimated',
```

```
In [ ]:
```

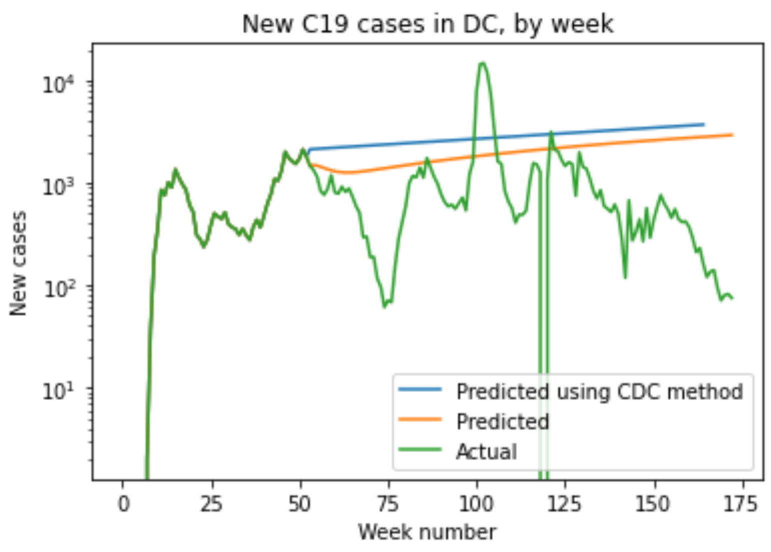
```
In [ ]:
```

```
In [36]: # Graphing sections:
```

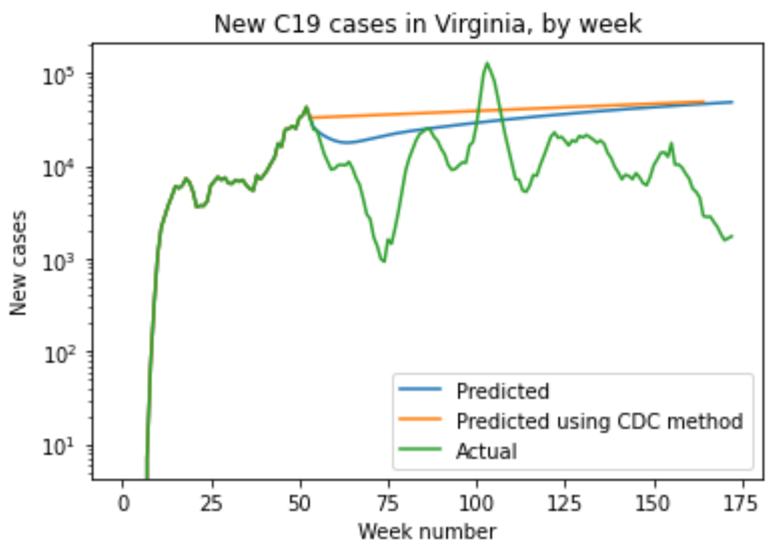
```
In [37]: tempHold = np.array(states.loc["DC"] )
         CDC = tempHold[:53]
         print(CDC)

         tempHold = states.loc["DC"]
         curr = tempHold.loc[51]

         for j in range(53, third):
             curr = curr * 1.005
             CDC = np.append(CDC, curr)

         print(CDC)
```

```
[0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
 2.000e+00 3.700e+01 1.920e+02 3.550e+02 8.540e+02 7.570e+02 1.009e+03
 9.000e+02 1.355e+03 1.123e+03 9.670e+02 8.550e+02 6.100e+02 5.210e+02
 3.100e+02 2.810e+02 2.370e+02 2.770e+02 3.840e+02 5.030e+02 4.700e+02
 4.440e+02 5.160e+02 3.950e+02 3.680e+02 3.550e+02 3.100e+02 3.560e+02
 3.070e+02 2.760e+02 3.710e+02 4.350e+02 3.660e+02 4.750e+02 6.280e+02
 7.780e+02 1.086e+03 1.051e+03 1.326e+03 2.012e+03 1.748e+03 1.624e+03
 1.532e+03 1.724e+03 2.118e+03 1.803e+03]
[0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 2.00000000e+00
 3.70000000e+01 1.92000000e+02 3.55000000e+02 8.54000000e+02
 7.57000000e+02 1.00900000e+03 9.00000000e+02 1.35500000e+03
 1.12300000e+03 9.67000000e+02 8.55000000e+02 6.10000000e+02
 5.21000000e+02 3.10000000e+02 2.81000000e+02 2.37000000e+02
 2.77000000e+02 3.84000000e+02 5.03000000e+02 4.70000000e+02
 4.44000000e+02 5.16000000e+02 3.95000000e+02 3.68000000e+02
 3.55000000e+02 3.10000000e+02 3.56000000e+02 3.07000000e+02
 2.76000000e+02 3.71000000e+02 4.35000000e+02 3.66000000e+02
 4.75000000e+02 6.28000000e+02 7.78000000e+02 1.08600000e+03
 1.05100000e+03 1.32600000e+03 2.01200000e+03 1.74800000e+03
 1.62400000e+03 1.53200000e+03 1.72400000e+03 2.11800000e+03
 1.80300000e+03 2.12859000e+03 2.13923295e+03 2.14992911e+03
 2.16067876e+03 2.17148215e+03 2.18233956e+03 2.19325126e+03
 2.20421752e+03 2.21523861e+03 2.22631480e+03 2.23744637e+03
 2.24863361e+03 2.25987677e+03 2.27117616e+03 2.28253204e+03
 2.29394470e+03 2.30541442e+03 2.31694149e+03 2.32852620e+03
 2.34016883e+03 2.35186968e+03 2.36362903e+03 2.37544717e+03
 2.38732441e+03 2.39926103e+03 2.41125733e+03 2.42331362e+03
 2.43543019e+03 2.44760734e+03 2.45984538e+03 2.47214460e+03
 2.48450533e+03 2.49692785e+03 2.50941249e+03 2.52195955e+03
 2.53456935e+03 2.54724220e+03 2.55997841e+03 2.57277830e+03
 2.58564219e+03 2.59857040e+03 2.61156326e+03 2.62462107e+03
 2.63774418e+03 2.65093290e+03 2.66418756e+03 2.67750850e+03
 2.69089604e+03 2.70435052e+03 2.71787228e+03 2.73146164e+03
 2.74511895e+03 2.75884454e+03 2.77263876e+03 2.78650196e+03
 2.80043447e+03 2.81443664e+03 2.82850882e+03 2.84265137e+03
 2.85686462e+03 2.87114895e+03 2.88550469e+03 2.89993221e+03
 2.91443188e+03 2.92900403e+03 2.94364906e+03 2.95836730e+03
 2.97315914e+03 2.98802493e+03 3.00296506e+03 3.01797988e+03
 3.03306978e+03 3.04823513e+03 3.06347631e+03 3.07879369e+03
 3.09418766e+03 3.10965859e+03 3.12520689e+03 3.14083292e+03
 3.15653709e+03 3.17231977e+03 3.18818137e+03 3.20412228e+03
 3.22014289e+03 3.23624360e+03 3.25242482e+03 3.26868695e+03
 3.28503038e+03 3.30145553e+03 3.31796281e+03 3.33455262e+03
 3.35122539e+03 3.36798151e+03 3.38482142e+03 3.40174553e+03
 3.41875426e+03 3.43584803e+03 3.45302727e+03 3.47029240e+03
 3.48764387e+03 3.50508209e+03 3.52260750e+03 3.54022053e+03
 3.55792164e+03 3.57571124e+03 3.59358980e+03 3.61155775e+03
 3.62961554e+03 3.64776362e+03 3.66600243e+03 3.68433245e+03
 3.70275411e+03]
```

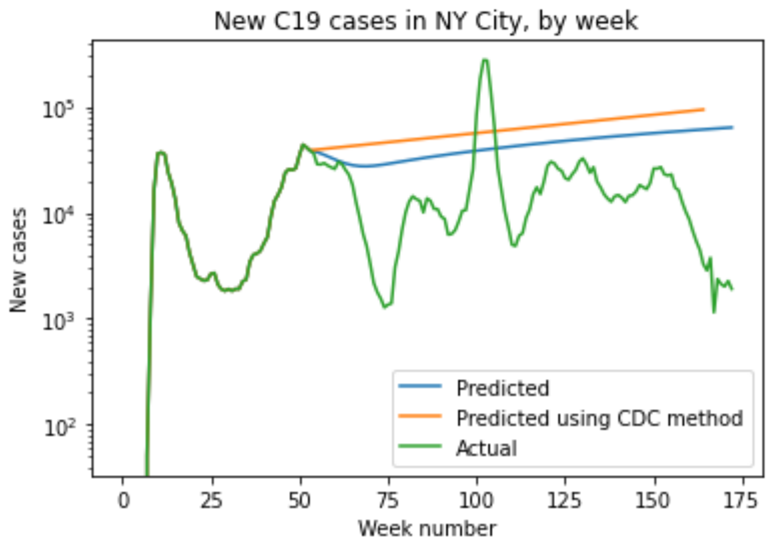
In [38]:
```
fig = plt.figure()

plt.title("New C19 cases in DC, by week")
plt.plot(CDC, label = "Predicted using CDC method")
plt.plot(futureStates.loc[8], label = "Predicted")
plt.plot(states.loc["DC"], label = "Actual")
plt.yscale("log")
plt.ylabel("New cases")
plt.xlabel("Week number")
plt.legend()
plt.show()
```



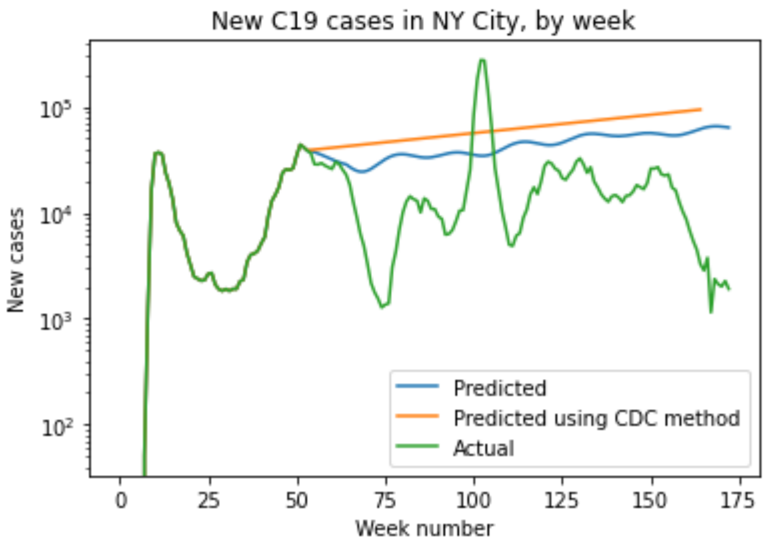
In [39]:
```
fig = plt.figure()

tempHold = states.loc["VA"]
curr = tempHold.loc[53]
CDC = np.array(tempHold[:53] )

for j in range(53, third):
    curr = curr*1.0036
    CDC = np.append(CDC, curr)

plt.title("New C19 cases in Virginia, by week")
plt.plot(futureStates.loc[53], label = "Predicted")
plt.plot(CDC, label = "Predicted using CDC method")
plt.plot(states.loc["VA"], label = "Actual")
plt.yscale("log")
plt.ylabel("New cases")
plt.xlabel("Week number")
plt.legend()
plt.show()
```



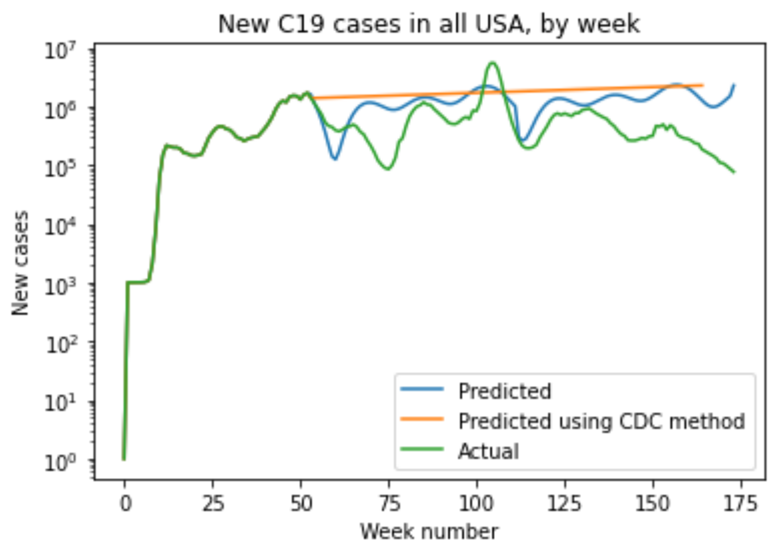
In [40]:
```
fig = plt.figure()
prediction = futureStates.loc[39]
prediction = np.array(prediction)
t = 0

tempHold = states.loc["NYC"]
curr = tempHold.loc[53]
CDC = np.array(tempHold[:53] )

for j in range(53, third):
    curr = curr*1.008
    CDC = np.append(CDC, curr)


plt.title("New C19 cases in NY City, by week")
plt.plot(prediction, label = "Predicted")
plt.plot(CDC, label = "Predicted using CDC method")
plt.plot(states.loc["NYC"], label = "Actual")
plt.yscale("log")
plt.ylabel("New cases")
plt.xlabel("Week number")
plt.legend()
plt.show()
```



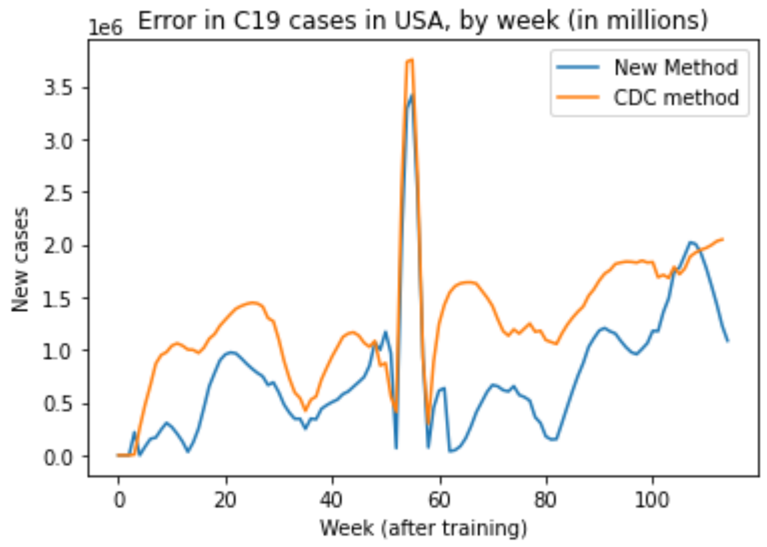
In [41]:
```
fig = plt.figure()
prediction = futureStates.loc[39]
prediction = np.array(prediction)
t = 0

for j in prediction:
    t = t + 1
    if (t < 109): #interpolate forward an additional 109 weeks beyond 54.
        # amplitudes taken from table above, rounded to the nearest hundred
        addTo = 1100 * math.sin(2 * t * math.pi / 54.0)
        addTo = addTo - (800 * math.sin(2 * t * math.pi / 27.0) )
        addTo = addTo - (1400 * math.sin(2 * t * math.pi / 18.0) )
        addTo = addTo * 2
        # times two because a sine is both up and down, but here basline is the troug
        prediction[64+t] = prediction[64+t] + addTo


plt.title("New C19 cases in NY City, by week")
plt.plot(prediction, label = "Predicted")
plt.plot(CDC, label = "Predicted using CDC method")
plt.plot(states.loc["NYC"], label = "Actual")
plt.yscale("log")
plt.ylabel("New cases")
plt.xlabel("Week number")
plt.legend()
plt.show()
```


New C19 cases in NY City, by week
New cases
Week number
Predicted
Predicted using CDC method
Actual
10^5
10^4
10^3
10^2
0
25
50
75
100
125
150
175

```
In [53]: fig = plt.figure()
         sumStates = np.array(1)
         sumPredStates = np.array(1)

         for t in futureStates:
             tempHold = np.sum(futureStates.iloc[:, t] )
             sumPredStates = np.append(sumPredStates, tempHold)

         for t in states:
             tempHold = np.sum(states.iloc[:, t] )
             sumStates = np.append(sumStates, tempHold)

         tempHold = sumStates
         curr = sumStates[53]
         CDC = np.array(tempHold[:53] )

         for j in range(53, third):
             curr = curr * 1.00451355
             # this number is derived in the final cell of this workbook.
             CDC = np.append(CDC, curr)


         chaosNumb = sumPredStates.copy()

         # accounting for waves
         for t in range(53, 174): # sum of states' amplitudes, rounded to nearest thousand.
             addTo = -101000 * math.sin(2 * t * math.pi / 54.0)
             addTo = addTo - (150000 * math.sin(2 * t * math.pi / 27.0) )
             addTo = addTo - (184000 * math.sin(2 * t * math.pi / 18.0) )
             addTo = addTo * 2
             # times two because a sine is both up and down, but here basline is the trough.
             chaosNumb[t] = chaosNumb[t] + addTo

         # accouning for vaccination at t = 109
         for t in range(112, maxTime):
             addTo = -10000
             durationStart = 112
             #vaccination of Omicron/Delta began at 112th week
             durationEnd = maxTime
             durationOfWithoutPulse = durationEnd - durationStart
             addTo = addTo * durationOfWithoutPulse
             # all at once as a pulse, instead of a cumulative duration effect because it's ea
             # still need the pulse active for the entire rest of t, anyway.
             chaosNumb[t] = chaosNumb[t] + addTo

         plt.title("New C19 cases in all USA, by week")
         plt.plot(chaosNumb, label = "Predicted")
         plt.plot(CDC, label = "Predicted using CDC method")
         plt.plot(sumStates, label = "Actual")
         plt.yscale("log")
         plt.ylabel("New cases")
         plt.xlabel("Week number")
         plt.legend()
         plt.show()
```



```
In [51]:  fig = plt.figure()
          tempHold = CDC[0:165] - sumStates[0:165]
          CDCoff = tempHold[50:-1]
          meOff = (chaosNumb - sumStates)[50:165]

          plt.title("Error in C19 cases in USA, by week (in millions)")
          plt.plot( abs(meOff), label = "New Method")
          plt.plot( abs(CDCoff), label = "CDC method")
          plt.ylabel("New cases")
          plt.xlabel("Week (after training)")
          plt.legend()
          plt.show()
```



```
In [54]:  million = 1000000

          print("CDC error (in millions): ", sum( abs(CDCoff)[0:160])/million)
          print("new method error (in millions): ", sum( abs(meOff)[0:160])/million)
```

```
CDC error (in millions):  146.93080645181885
new method error (in millions):  88.51721664252544
```

```
In [45]:  print("End.")
```

End.

```
In [46]: # Bonus round
```

```
In [55]: startWeek = 40
         stopWeek = 100
         # giving it a stopweek of 100 so it still has 54 weeks of training

         yVars = pd.DataFrame(sumStates[startWeek:stopWeek])
         yVars = np.array(yVars).flatten()
         constT = yVars[startWeek]
         xVars = np.array(range(startWeek, stopWeek) )


         tuples, covarience = sp.optimize.curve_fit(solver2, xVars, yVars, maxfev=2000)

         print(s, tuples, "covarience: ", covarience)
```

```
0 [1.00451355] covarience:  [[1.50433893e-06]]
```

```
In [ ]:
```

References:

1. Dettmer, P. (2021). Immune: A journey into the mysterious system that keeps you alive. Penguin Random House. New York. ISBN: 978-0-593-24131-8

2. Nohara, Y., & Manabe, T. (2022). Impact of human mobility and networking on spread of covid-19 at the time of the 1st and 2nd epidemic waves in Japan: An effective distance approach. PLOS ONE, 17(8). https://doi.org/10.1371/journal.pone.0272996

3. Brockmann, D., & Helbing, D. (2013). The hidden geometry of complex, network-driven contagion phenomena. Science, 342(6164), 1337–1342. https://doi.org/10.1126/science.1245200

4. Jacobs, R., Teunis, P., & van de Kassteele, J. (2020). Tracing the origin of food-borne disease outbreaks. Epidemiology, 31(3), 327–333. https://doi.org/10.1097/ede.0000000000001169

5. Pastore Y Piontti, A, Gomes Da Costa, M, Samay N, Perra N, Vespignani A, Ndro, A. (2014) the infection tree of global epidemics. Network Science, 2(1), 132–137. https://doi.org/10.1017/nws.2014.5 http://journals.cambridge.org/article_S2050124214000058

6. Pinto, P. C., Thiran, P., & Vetterli, M. (2012). Locating the source of diffusion in large-scale networks. Physical Review Letters, 109(6). https://doi.org/10.1103/physrevlett.109.068702

7. Schlundt J, Toyofuku H, Jansen J, & Herbst S. A. (2004). Emerging food-borne zoonoses. Revue Scientifique Et Technique De L'OIE, 23(2), 513–533. https://doi.org/10.20506/rst.23.2.1506

8. May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature,* 261(5560), 459–467. https://doi.org/10.1038/261459a0

9. Volterra, V. (1927). Fluctuations in the abundance of a species considered mathematically. *Nature*, *119*(2983), 12–13. https://doi.org/10.1038/119012b0

10. Katie Gostic et al.; (2024). Technical Blog: Improving CDC's Tools for Assessing Epidemic Growth. https://www.cdc.gov/forecast-outbreak-analytics/about/technical-blog-rt.html

11. Goncalves, B., Balcan, D., & Vespignani, A. (2013). Human mobility and the worldwide impact of intentional localized highly pathogenic virus release. Sci Rep 3, 810 (2013). https://doi.org/10.1038/srep00810

12. Miller, M. C., & Miller, J. M. (2015). The masses and spins of neutron stars and stellar-mass black holes. Physics Reports, 548, 1–34. https://doi.org/10.1016/j.physrep.2014.09.003

13. CDC, (2023) Ensemble model, https://covid19forecasthub.org/doc/ensemble/

14. Gleick, J. (1987). Chaos: Making a new science. Viking. ISBN: 0-670-81178-5

15. Ray, E. L., et al. (2023). Comparing trained and untrained probabilistic ensemble forecasts of covid-19 cases and deaths in the United States. International Journal of Forecasting, 39(3), 1366–1383. https://doi.org/10.1016/j.ijforecast.2022.06.005

16. Cramer, E. Y., Ray, et al. (2022). Evaluation of individual and ensemble probabilistic forecasts of covid-19 mortality in the United States. Proceedings of the National Academy of Sciences, 119(15). https://doi.org/10.1073/pnas.2113561119

17. Afzal, A., Saleel, C.A., Bhattacharyya, S. et al. (2022) Merits and Limitations of Mathematical Modeling and Computational Simulations in Mitigation of COVID-19 Pandemic: A Comprehensive Review. Arch Computat Methods Eng 29, 1311–1337 (2022). https://doi.org/10.1007/s11831-021-09634-2

18. Rahmandad, H., Xu, R., & Ghaffarzadegan, N. (2022). Enhancing long-term forecasting: Learning from covid-19 models. PLOS Computational Biology, 18(5). https://doi.org/10.1371/journal.pcbi.1010100

19. Zhang, L., Tan, C., & Yu, F. (2017a). An improved rainbow table attack for long passwords. Procedia Computer Science, 107, 47–52. https://doi.org/10.1016/j.procs.2017.03.054