NUCLEATION IN A TWO COMPONENT METAL ALLOY

Kalea Sebesta

Department of Applied Mathematics
University of Nevada, Reno

George Mason University Project Mentors

Dr. Evelyn Sander
Dr. Tom Wanner
Tom Stephens

Abstract

The research discussed in this report explores a particular phase separation phenomenon, known as nucleation, in a two component metal alloy. The motivation behind this study is to use a phenomenological model to understand some general features of binary alloys. This will allow for examination, explanation, and later prediction, based on behaviors that are occurring in these alloys.

This final report is submitted in partial fulfillment of the
National Science Foundation Research Experience for Undergraduates (REU) Program
Department of Mathematical Sciences
George Mason University
June 03, 2013 - August 02, 2013

Contents

1 Introduction

The phase separation phenomenon known as nucleation occurs in multiple component metal alloys. These alloys are seen in material sciences; therefore, understanding their properties is an important topic for discussion. For the purpose of this research, the two component metal alloy was examined in hopes to understand from a phenomenological stand point, how these composition of metals behave. Learning the characteristics of the binary case will allow for intuition and insight into more complex models.

In a 2011 volume of the SIAM Applied Dynamical Systems Journal, this idea of nucleation in a stochastic system for a three component metal alloy was examined using topological approaches and much discovery was made [2]. The approach taken in this research on the binary case, follows the same methods used in the three component case. The binary case focuses on the formation and behavior of the droplets, which can be described as the separation of one metal from the other within an alloy. After analyzing the behaviors of these droplets, the aim is to find an equation or method that will allow for accurately predicting droplet behavior based on the characteristics of the alloy.

The purpose of this research is to be able to provide explanation through analysis, as well as a prediction method for the two component alloy with Neumann Boundary Conditions. After this is accomplished, the same path will be followed, with regards to approach, to make discoveries in the case of periodic boundary conditions.

2 Background and Research Methods

2.1 Methodologies

In order to understand the concept of nucleation, which is a technical term for droplet formation, and how the calculation of this phase separation is done, one must understand the geometric interpretations and numerical methods behind it. This research included and relied on topological aspects, theorems and theories and their connection to numerical methods. The numerical methods behind calculating these topological spaces is crucial to understand. One important topological aspect is the Euler Characteristic; which directly corresponds to the betti number calculations. The betti number calculations is how the number of droplets are counted. Performing numerical calculations on the gathered data also brings up an important feature involved, which is the occurrence of large deviation and the use of the stochastic process.The stochastic process implies that the equation being used is including additive noise.

The topological and geometric aspects one needs to be familiar with, revolves around this idea of betti numbers. β_0 can be interpreted as the number of connected components in a geometric figure, β_1 represents the number of "holes" in the connected component, and β_2 is the number of "voids" in the connected component. The β_2 can be thought of as the holes in the holes, or the enclosed cavities [3]. In this research the β_2 was equal to zero. As seen in the figures 1 and 2, there are corresponding β_0 and β_1 for each of the components in the alloy.

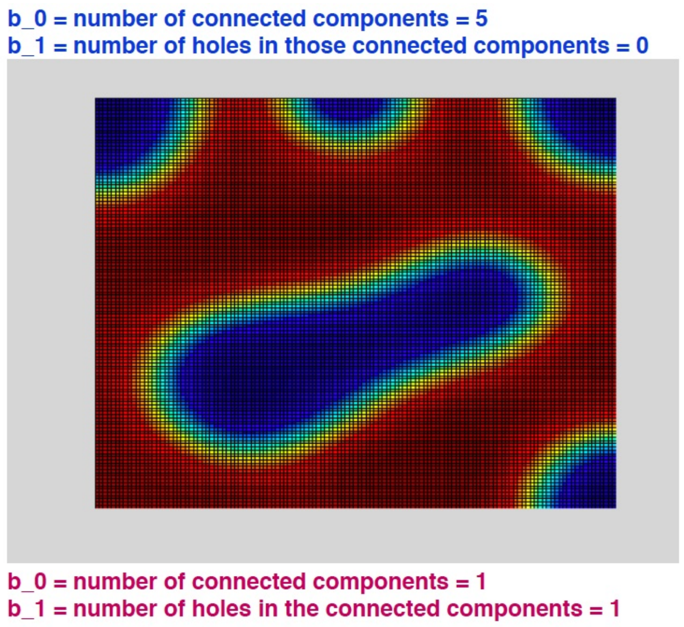

Figure 1: *β_0 and β_1 are shown for each component in the mixture. For the purpose of basic understanding, disregard the light blue and yellow colors in this figure. Assume the image is either red or blue (yellow will be considered red and light blue will be considered blue).*

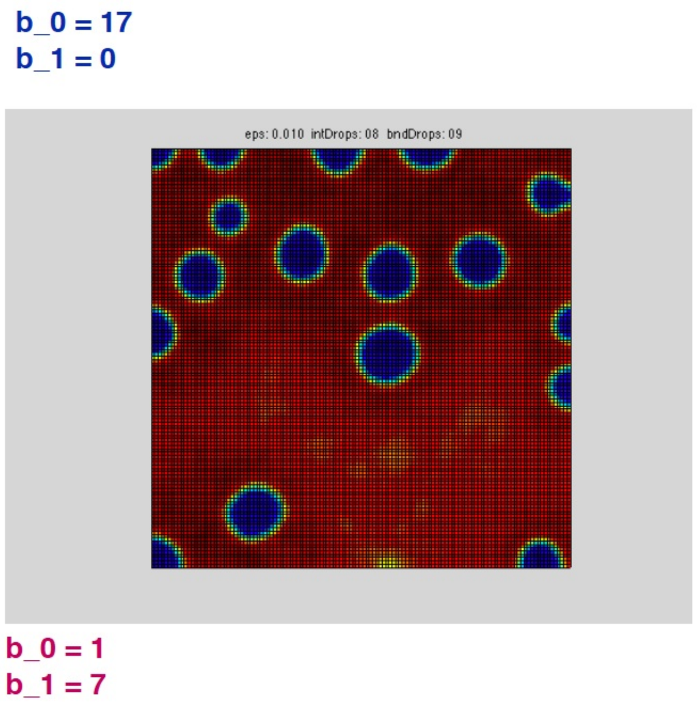

Figure 2: *β_0 and β_1 are shown for each component in the mixture. Disregard the light blue and yellow colors in this figure, assume the image is either red or blue.*

These geometric figures can also be transformed into a numerical matrix to be made machine readable. For the sake of explanation let the red area in the figures be referred to as component one and the blue area, as component two (also referred to as droplets). A grid is made on top of the figures and a dot is placed in the middle of each square. If the dot lies within a droplet (component two) then a one is assigned to that location on the grid. If the dot lies in component one then a zero is placed at that location. The logic behind this process is to make a matrix that is filled with zeros everywhere there is component one and ones placed everywhere there is component two. Through this process the computer is able to count the groups of ones which directly reflects the number of droplets or what is technically known as the β_0. Figure 4 is a visual representation where the ones are denoted as X's and the blank boxes as zeros.

Taking these concepts and combining them together along with using machine technology, specifically the software Matlab, data was able to be generated. This occurred through the combination of integration and supporting functions that were created. Matlab also allowed for data to be stored, and ultimately was used to preform numerical analysis. Matlab's graphical capabilities allow for visual analysis to be performed along side the numerical computations.

2.2 Equations

The primary equation used in this research was the Cahn-HIlliard, fourth ordered partial differential equation with additive noise:

$$u_t = -\Delta(\epsilon^2 \Delta u - f(u)) + \sigma_{noise}\xi, \tag{1}$$

with Neumann Boundary Conditions:

$$\frac{\partial u}{\partial v} = \frac{\partial \Delta u}{\partial v} = 0 \quad on \quad \partial\Omega.$$

Where $v =$ the outward unit normal vector. Also $f(u)$ takes the nonlinear form as follows,

$$f(u) = u - u^3.$$

When discussing the phenomenon that is phase separation, it is key to understand that the components in the alloy have energy states and ideally want to minimize that energy. The following equation represents the energy solution to the Cahn-Hilliard equation at a specific time; the energy is a general statement and is not unique. The energy equation is also not physical, it is merely a formula that is decreasing over time. Figure 3 puts a visual to this concept.

$$E_\epsilon[u] = \int_\Omega \left(\frac{\epsilon^2}{2} \cdot |\bigtriangledown u|^2 + F(u) \right) dx. \tag{2}$$

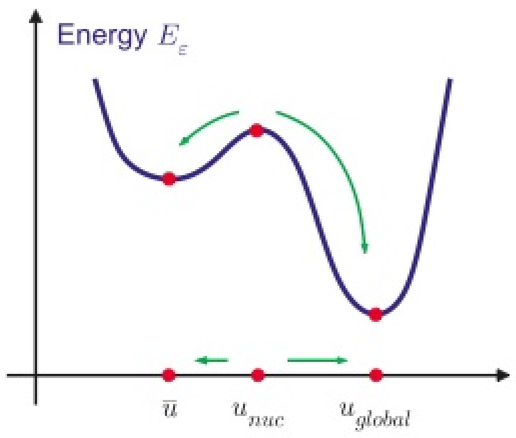

Figure 3: *Visual representation on the generic energy equation. Each point in this graph represents a function in infinite-dimensional space. This means depending on the specific function at a specific point in space may have a different level of energy, where these energies can be local or global minimums.*

A secondary equation used and also very important to the research, was the calculation of the Euler Characteristic:

$$\chi = V - E + F = \beta_0 - \beta_1 + \beta_2 \tag{3}$$

The visual representation of this equation was seen in earlier Figures 1 and 2. This equation points out the relationship between the vertices, edges, and faces of a polyhedra, and its connection to the betti numbers. This equation allows for a calculation to be made on a smooth surface object through identifying topological characteristics. Similar to how a geometric figure is transformed into a matrix for numerical calculations, this equation is used to take the location of numbers from matrix and identify if a droplet lives at that location. As mentioned before β_0 is the number of connected components, β_1 is the number of one dimensional holes, and β_2 is the number of two dimensional holes (voids).

In order to take topological surfaces, such as the droplets seen in the previous figures, and numerically calculate their Euler Characteristic, a systematic approach is taken. First, the figures are discretized on grid and then translated into a matrix in the form of ones and zeros. Ones occur if part of a droplet is in the grid and a zero if it is not. From there the vertices are calculated by tracing through the matrix, picking a location, checking its value, and compares it to the neighboring values. If a one exists in the neighbor above and to the left, a vertex exists. If the neighbor above and to the right holds a one a vertex exists, and similar is done for the bottom, left, and bottom, right neighbors. Then this process moves two places over and repeats. To calculate the edges a similar process is performed. A location is picked, it is checked if it holds a one or zero value, then checks if there is a neighbor above, below, left, or right that is occupied by a one. If a neighbor holds a one then an edge exists. As before this process continues throughout the matrix. Finally, to calculate the faces, a location is picked and if the space is occupied by a one then a face exists at that location. Figure 4 is a visual representation of this process.

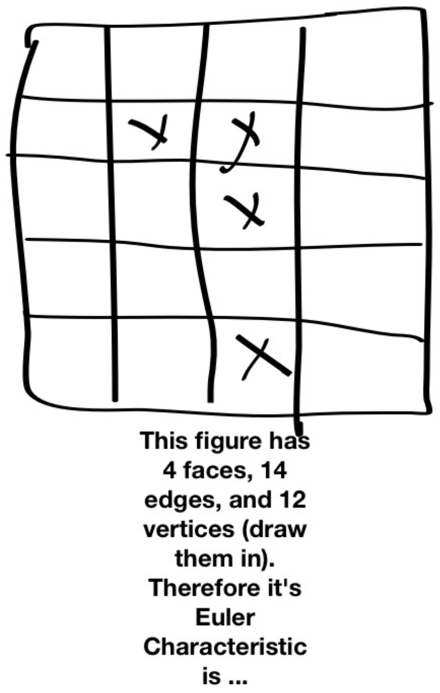
This figure has 4 faces, 14 edges, and 12 vertices (draw them in). Therefore it's Euler Characteristic is ...

Figure 4: *The X's in this grid represent the discretized droplets and the resulting Euler Characteristic is found to be 2. This means that the matrix represents 2 droplets.*

Built within the code used for this research were functions, one designed specifically to trace through the matrix given to it and calculate the Euler Characteristic (findEuler.m), which essentially does the process described earlier with calculating the vertices, edges, and faces. There is another important function that then takes what is found in findEuler.m and computes the betti numbers for both components one and two (computeBetti.m).

It is also important to understand the derivation of the partial differential equation that was used in performing this research. The variational form being used is that of, $E_\epsilon[u]$. For simplicity, the Cahn-Hilliard equation is first being considered without noise, for $0 < x < 1$ and $0 < y < 1$,

$$\frac{\partial u}{\partial t} = -\bigtriangleup (\epsilon_n^2 \bigtriangleup u + f(u)). \tag{4}$$

$$\bigtriangleup = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 y}{\partial y^2}.$$

$$Let = \begin{cases} \tilde{t} = \gamma^2 t \\ \tilde{x} = \gamma x \\ \tilde{y} = \gamma y \end{cases}$$

where γ is some constant. Notice that if $0 < x < 1$ and $0 < y < 1$ then $0 < \tilde{x} < \gamma$ and

$0 < \tilde{y} < \gamma$. Now using the notation $\tilde{\triangle}u = \frac{\partial^2 u}{\partial \tilde{x}^2} + \frac{\partial^2 y}{\partial \tilde{y}^2}$, the left hand side of the original Cahn-Hilliard equation becomes $\frac{\partial u}{\partial t} = \frac{\partial u}{\partial \tilde{t}}\frac{\partial \tilde{t}}{\partial t}$ and $\frac{\partial \tilde{t}}{\partial t} = \frac{\partial}{\partial t}[\gamma^2 t] = \gamma^2$. Thus resulting in,

$$\frac{\partial u}{\partial t} = \gamma^2 \frac{\partial u}{\partial t}. \tag{5}$$

Doing similar manipulation to the right hand side of the Cahn-Hilliard equation, the following equations are obtained,

$$\begin{gathered}\frac{\partial u}{\partial x} = \gamma \frac{\partial u}{\partial \tilde{x}}, \\ \triangle u = \gamma^2 \tilde{\triangle} u.\end{gathered} \tag{6}$$

Combining the left and right hand sides of the Cahn-Hilliard manipulation equations (5) and (6), the follow equation is obtained on $[0, \gamma]^2$,

$$\frac{\partial u}{\partial t} = -\tilde{\triangle}(\epsilon_n^2 \gamma^2 \tilde{\triangle} u + f(u)) \tag{7}$$

let $\gamma^2 = \frac{\epsilon_0}{\epsilon_n}$ then we have the equation on $\left[0, \frac{\epsilon_0}{\epsilon_n}\right]^2$ is as follows,

$$\frac{\partial u}{\partial \tilde{t}} = -\tilde{\triangle}(\epsilon_0^2 \tilde{\triangle} u + f(u)). \tag{8}$$

This implies that changing the value of ϵ can also be seen when fixing $\epsilon = \epsilon_0$ and changing the domain size.

2.3 Parameters

Within this research, parameters were established which allowed the use of equations (1) and (2) as well as the machine software. These parameters allowed for the equations to be tailored for the specific binary alloy case. Changing the parameters that were used, resulted in an energy change of the components within the alloy. Thus, effecting the equilibrium solution equation and the ultimate phase separation characteristics. The parameters specified in this research are μ, which represents the mass of the components. The mass of the components can also be thought of as the difference between the two concentrations of the components in the alloy ($\int u dx$ where u = concentration of component 1 $-$ concentration of component 2). Component is another way to say a type of metal. Since this was the binary case if $\mu = 0$ that meant there was equal mass in the alloy between component one and component two. ϵ represents the interaction lengths of the molecules. The ϵ value is the major factor that effects the number, size, and the rate at which droplets form. To make a more realistic model, σ was established to represent the noise intensity level. There are also parameters established which allow for numerous iterations of the code and provide a larger supply of data to ultimately run analysis on. There is also a variable d, which is the thickness of the boundary droplets. For the purpose of this preliminary research the parameters were set to $\mu = 0.6$, $\sigma = .09$, and the ϵ value was being varied.

2.4 Models

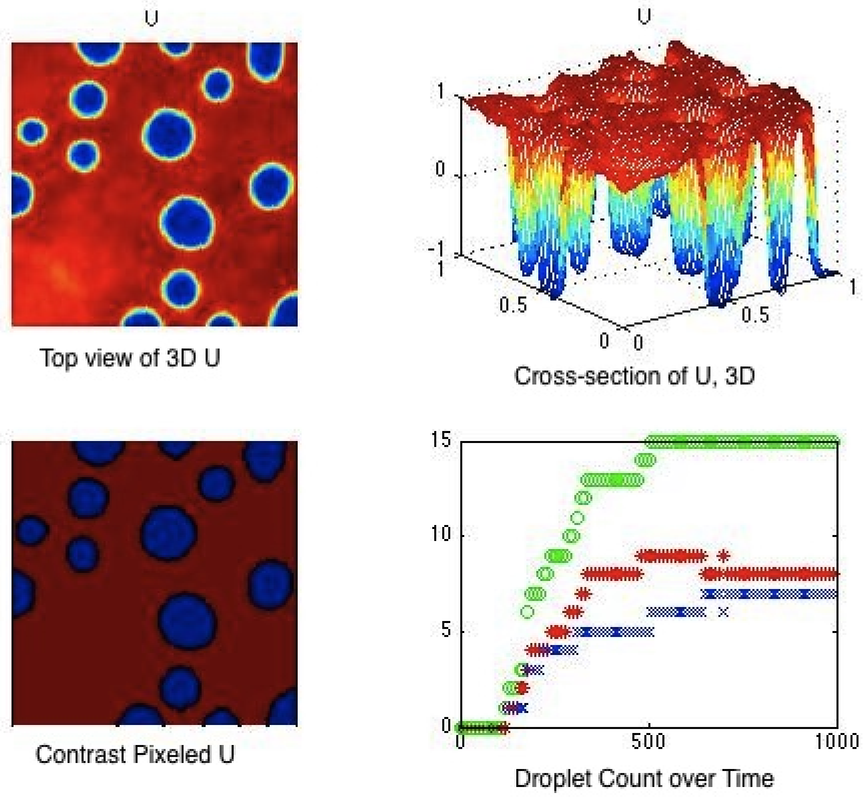

Figure 5: *The multiple subplot model for the two component metal alloy. The 3 dimensional view of the cross-section of U shows how the concentration of the metals changes at each location for a specific time. The yellowish and light blueish colors seen in these figures were among the concentrations that were thresholded to equal either red or blue in order to accurately calculate the droplets.*

The above image, Figure 5 is the multiple subplots model used in this research. The top left hand subplot labeled 'U' is a top view of the the three dimensional cross-section seen in the subplot on the top right hand side. The three dimensional cross-section 'U' is a visual of the two components separating within the alloy. The bottom left subplot displays the image associated with the matrix comprised of ones and negative ones. The bottom left hand subplot shows the betti curves, the green is β_0 (total droplets), the red is boundary droplets ($\beta_0 n - \beta_1 p$) and blue is the interior droplets ($\beta_1 p$); where n represents the negative space (the blue area/component two) and p represents positive space(red area/component one). The betti curve subplot also shows the time step associated with the formation of the first droplet as well as the last droplet. Since nucleation is the phase separation that is occurring in the alloy, the betti curve subplot allows for association between a time step and nucleation. This is a behavior of the binary case that will be highly focused on in this research.

3 Results and Discussion

3.1 Time Horizon for Decomposition

In order to learn the behaviors associated with the two component metal alloy, this idea of finding the alloy's nucleation point was the first area to be explored. Due to the fact that nucleation occurs in a somewhat stochastic (random) way, it was realized that the nucleation point would be better looked at as a time horizon of decomposition. For the purpose of this research the μ was kept at 0.6 and σ at 0.09. Since the amount of mass for each component in the alloy was not equal, using conventional methods to identify a droplet was causing inaccurate readings. Specifically this was an issue in the early time steps. Every little fluctuation away from the starting mass (the concentration of metal one $-$ concentration of metal two) was being considered a droplet which is not true and as a result, was effecting the betti number calculations. The importance of correct betti number calculation is because β_0 represents the total number of droplets and when a true droplet forms, then nucleation has begun. Therefore, to combat the issue of premature droplet counting, the data that was being use became thresholded before betti calculations were performed. This is explained by Figure 5. Remembering the energy equation mentioned earlier in equation (2), the droplets only truly form when an equilibrium state is reached for a component. The thresholding allowed for the initial insignificant fluctuations (where a equilibrium state was not reached) to be ignored and for the actual time horizon of decomposition to be calculated using the new, accurate betti numbers.

Once the thresholding was done, looking at the betti curves allowed for intuition to be made on the approximate time horizon. At first, both the start time of decomposition (the first time step where $\beta_0 = 1$) and the end time of decomposition (the first time step where $\beta_0 =$ the max β_0) was calculated. Remembering that the motivation behind this project is to find when nucleation occurs in a two component alloy, the end time of decomposition became the main focus. After the max β_0 is reached nucleation has ended and a process called coarsening occurs, thus stressing the importance of calculating the ending time of decomposition. Once the end time decomposition of the time horizon was calculated for one run, then numerous runs where done with the same parameters and an average was taken. This average time step was used as a parameter referred to as tend within the Matlab code; this governed the run time of the data, making it less computational expensive. Averaging the time decomposition for the many different ϵ values over numerous runs allowed for a smoother curve of the betti numbers to be made and better approximation of prediction to occur. The code written to calculate the average decomposition step is as follows.

```
function [avgStep] = getTimeDecompStep(mu,epsilon,sigma,startRun,endRun)

%initialize varaiables
finalStep = 498;

t_decompArray = zeros(1, endRun-startRun+1);
bettiArray = zeros(endRun-startRun, finalStep+1);
```

```

%create bettiArray from saved data
    %for loop startRun:endRun
        %load saved data
        for run=startRun:endRun
            for step = 0:finalStep

                S= load(sprintf('../../../data/neumannBC/U_mu%0.5f_eps%
                                    ...0.5f_sg%0.5f_step%05d_r%04d.mat',
                                        ...mu,epsilon,sigma,step,run));
                step = step+1;
                bettiArray(run-startRun+1, step)= S.betti0n;
            end
            %finds the time step that holds the largest element
            max(bettiArray(run-startRun+1));

            %find(bettiArray == max_b0n);
            newArray = find(bettiArray(run-startRun+1,:) == ...
                max(bettiArray(run-startRun+1, :)));
            run_t_decompElement = min(newArray);

            %vector of the t_decomp values of each run
            t_decompArray(run-startRun+1)= run_t_decompElement;

        end

        %t_decompArray
        avgStep= mean(t_decompArray)

end
```

Once the getTimeDecompStep code was made then the code to calculate the average time decomposition step (getAvgTimeDecompStep.m) was made on the same premise, to take in the μ, ϵ, σ, startRun, and endRun and calculate the average time decomposition step for these parameters. This code was then integrated into the integrator function designed for this research (ch2d_noise_neumannIntegrator.m) and allowed for the max iterations through the data to be three times the amount of the average decomposition time that was found for the specified parameters. This was another way to save time while still obtaining the critical information that was being sought. The theory behind incorporating these codes was to be able to interpolate a function that describes the behavior of the data which ultimately allows the data to become normalized, to a certain extent.

3.2 Average Betti Curves/ Droplet Curves

As mentioned earlier, once the data was able to be thresholded and correct betti numbers calculated, betti curves were able to be plotted on the real time run model. This was seen in Figure 5, in the bottom left hand corner. This allowed for a clear representation numerically, of how the droplets were forming and a relative time step for when nucleation was beginning and ending. From here, the getAvgTimeDecompStep was used as a tool to create and plot the average betti curves associated with parameters that were specified. The code that was created to implement this was plotBettiZero_Averages.m. The plot has three different curves that are being plotted; the blue curve represents the $\beta_0 n$ (total droplets), the red curve is the $\beta_1 p$ (interior droplets), and the green is $\beta_0 n - \beta_1 p$ (boundary droplets). Since the data used in this research was run twenty times per parameter the average betti curves still appear somewhat jagged. Running 60 runs the curves begin to smooth out. The theory is with more runs, such as one hundred, the average betti curves would become smooth. This means the average number of droplets that are forming would become smoother as well since the betti curves are the droplet counts. The code that was created to implement this was AvgerageDrops_withPlot. There is control within this code to run it only to as far as the average decomposition time step, which was found while examining the time horizon. This allows for a visual check of the numerical output of interior and boundary droplets that the code calculates. The below figures show the plots associated with the average betti curves for $\epsilon = 0.006$ Figure 6, and the corresponding average droplets image Figure 7.

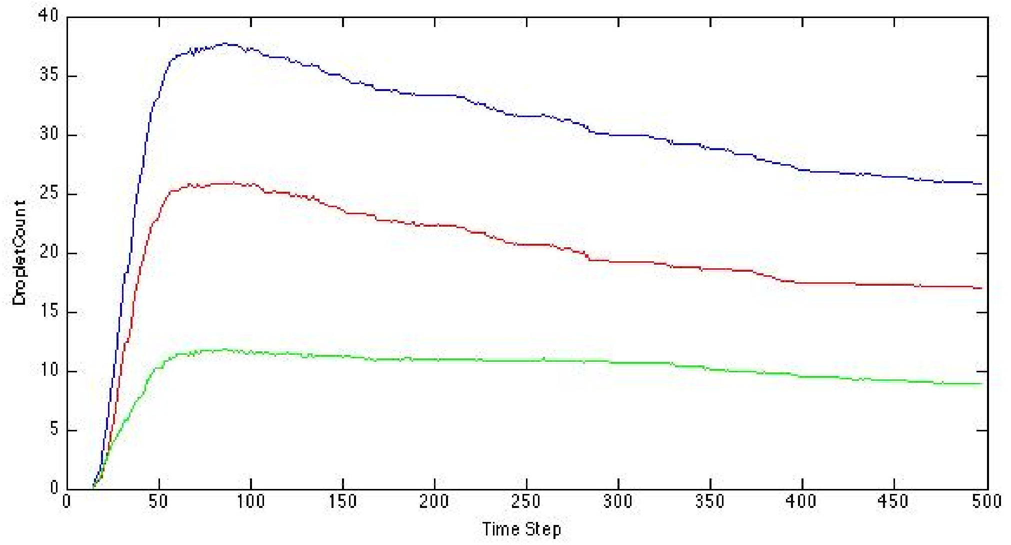

Figure 6: *Average Bettie Curve before smoothing data with* $\epsilon = 0.006$.

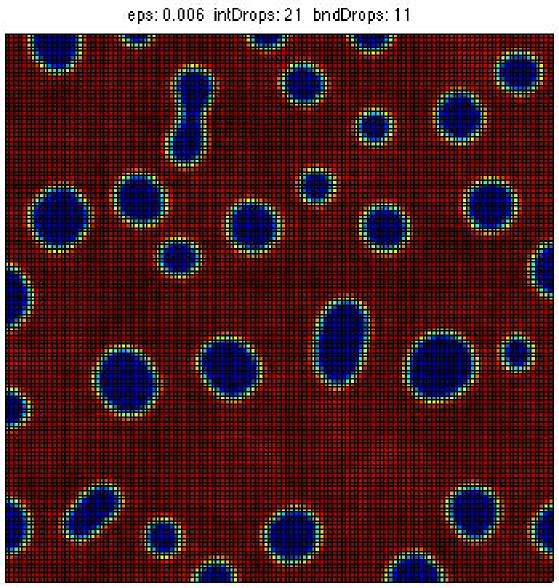

Figure 7: *Average drops with* $\epsilon = .006$, *interior drops* $= 21$, *boundary drops* $= 11$.

3.3 Calculating d

After being able to calculate the average interior and boundary droplets from AverageDrops_withPlot.m for the different ϵ values, this idea of calculating the boundary droplet thickness d, was examined. Identifying d which is related to the domain mentioned in equation (8), is important when attempting to predict the average droplets. Using the manipulation done on the Cahn-Hilliard equation seen in section 2.2, the number of interior and boundary droplets for ϵ_n on $[0,1]^2$ is the same as at ϵ_0 on $[0, \frac{\epsilon_0}{\epsilon_n}]^2$. Using this fact for ϵ_0, it is assumed that there are i_0 droplets in the interior region $[d, 1-2d]^2$ and b_0 droplets in the boundary, doing some substitutions the following equations are obtained, where the i_0, b_0, ϵ_0 are fixed and the ϵ_n changes:

$$i_n = i_0 \frac{\left(\dfrac{\epsilon_0}{\epsilon_n} - 2d\right)^2}{(1-2d)^2} \tag{9}$$

$$b_n = b_0 \frac{\epsilon_0}{\epsilon_n} \tag{10}$$

After manipulation and substitutions of these equations (9) and (10), the resulting equation to calculate d is as follows:

$$d = \frac{\sqrt{\dfrac{e_0}{e_n}} - \sqrt{\dfrac{i_n}{i_0}}}{2 - 2\sqrt{\dfrac{i_n}{i_0}}} \tag{11}$$

3.4 Predicting Interior and Boundary Droplets

Obtaining the d calculation describe in the previous section as well as the initial values that are found, paves the way to predict boundary and interior droplets. This was done with varying ϵ so that analysis of the binary alloy case could be done and compare its results to what was found in the three component case. The idea behind this was using the following two equations, where equation (12) predicts the average boundary droplets and equation (13) predicts the average interior droplets.:

$$b_n = b_0 \frac{e_0}{e_n} \tag{12}$$

$$i_n = i_0 \frac{\left(\dfrac{\epsilon_0}{\epsilon_n} - 2d\right)^2}{(1-2d)^2}. \tag{13}$$

Manipulating equation (13), d was calculated with a specific e_0 and e_n and the i_0 corresponding to the e_0. Then d is used to predict the i_n for any given ϵ_n. Once the prediction droplets were calculated, the accuracy of the calculation needed to be found. For the sake of readability, let the actual average droplets that were calculated in AverageDrops_withPlot.m be represented by A_a and let the predicted average droplets that were calculated by equations (12) and (13) be represented by A_p. Calculating the relative error, which is proof of accuracy in the prediction calculation,

$$\frac{|A_a - A_p|}{A_p}$$

is the next step. The relative error is critical to have when analyzing if the equations (12) and (13) are accurate to predict these droplets. Numerous ϵ ranges were set and the interior and boundary droplet averages with actual and predicted amounts were made. This was done for eight runs, then twenty runs, then sixty runs to see if the increased data would reduce the relative error since the predictions were done on averages. Thus with more data, a better read and prediction should occur.

Initial Parameters for Tables 1 and 2:
Average Fixed Step = 90
Total Runs = 20
$\epsilon_0 = 0.0110$ and $\epsilon_n = 0.0105$
$d = 0.2301$

Table 1: Interior Average Droplet Counts

ϵ	Actual Average	Predicted Average	Relative Error
.0110	3.8	3.8	0.0000%
.0105	4.5	4.5	0.0000%
.0010	5.15	5.3383	3.5273%
.0095	6.5	6.3482	2.3912%
.0090	7.8	7.5727	3.0016%
.0085	8.35	9.0691	7.9291%

Table 2: Boundary Average Droplet Counts

ϵ	Actual Average	Predicted Average	Relative Error
.0110	5.05	5.05	0.0000%
.0105	5.5	5.05	0.0000%
.0010	6.05	5.555	8.9108%
.0095	5.75	5.8474	1.6657%
.0090	5.9	6.1722	4.4100%
.0085	5.95	6.5353	8.9559%

Tables 1 and 2 are the actual and predicted droplets that form in the interior and on the boundary. For these tables a finer mesh of ϵ was calculated over. Also the average time step of decomposition that was found over these specific ϵ values was calculated to be 90. Running the AverageDrops_withPlot.m code for each ϵ the actual average droplets were found. From there, using $\epsilon_0 = 0.011$ and $\epsilon_n = 0.0105$ and the corresponding i_0 and i_n, d was calculated to be 0.2301. As seen in the tables the relative error isn't as low as one would hope; the aim is to get the relative error to closer to zero, 10^{-4} would be ideal. The boundary droplet calculations are even more unsatisfying than the interior predictions. One thought for the inaccuracy in prediction values is the low number of runs on the data.

Initial Parameters for Tables 3 and 4:

Average Fixed Step = 150

Total Runs = 60

$\epsilon_0 = 0.0110$

$d = 0.1653$

Table 3: Interior Average Droplet Counts

ϵ	Actual Average	Predicted Average	Relative Error
.011	6.3333	6.3333	0.0000%
.010	8.3667	8.3667	0.0000%
.0090	11.05	11.2362	1.657%
.0080	15.3	15.4167	0.7570%

Table 4: Boundary Average Droplet Counts

ϵ	Actual Average	Predicted Average	Relative Error
.011	6.650	6.650	0.0000%
.010	8.0833	7.3150	10.503%
.0090	8.4833	8.1278	4.374%
.0080	9.15	9.1438	0.067%

Focusing on four ϵ values but for triple the number of runs, Tables 3 and 4 show that the interior droplet prediction got better since the relative error decreased. However, in the boundary droplet prediction something peculiar is occurring. The substantial increase in the relative error at $\epsilon = 0.01$ is disturbing and leads to discussion of the cause for this drastic jump.

Initial Parameters for Tables 5 and 6:
Average Fixed Step = 81
Calculated by averaging the time decomposition step over the runs and different ϵ
Total Runs = 60
$\epsilon_0 = 0.0110$
$d = 0.3090$

Table 5: Interior Average Droplet Counts

ϵ	Actual Average	Predicted Average	Relative Error
.011	2.9	2.9	0.0000%
.010	4.6167	4.6167	0.0000%
.0090	6.3833	7.2555	12%
.0080	7.3833	11.3884	35%

Table 6: Boundary Average Droplet Counts

ϵ	Actual Average	Predicted Average	Relative Error
.011	3.5667	3.5667	0.0000%
.010	4.9667	3.9234	26%
.0090	5.8	4.3593	33%
.0080	5.5167	4.9042	12%

For Tables 5 and 6 the same ϵ values, number of runs, and d value were used but the fixed time step that the averages were taken at were changed from 150 to 81. The fixed time step was set to 81 because 81 was the average time step calculated between the four ϵ values. The interior and boundary droplet predictions increased even more. This provoked the idea that the fixed time step was causing the increase in relative error. After further investigation into the code it was realized that the average time steps for each ϵ referred to a different actual time; thus the notation to go off of the actual average time and not the time step may prove to be a better strategy.

Initial Parameters for Tables 7 and 8:
Average Fixed Step = final time step
Calculated by averaging the time decomposition over the runs and different ϵ
Average Decomp Time = 0.020255 Total Runs = 60
$\epsilon_0 = 0.011$
$d = 0.1189$

Table 7: Interior Average Droplet Counts

ϵ	Actual Average	Predicted Average	Relative Error
.011	5.4833	5.4833	0.0000%
.010	7.0167	7.0167	0.0000%
.0090	8.7667	9.1468	4.1555%
.0080	12.2833	12.2061	0.6324%

Table 8: Boundary Average Droplet Counts

ϵ	Actual Average	Predicted Average	Relative Error
.011	6.5500	6.5500	0.0000%
.010	7.3667	7.2050	2.2443%
.0090	7.7333	8.0056	3.4013%
.0080	8.7000	9.0062	3.3999%

Following the new strategy of calculating the average actual time between the ϵ values, the time was calculated to be 0.020255. This number was then set within the ch2d_noise_neumannIntegrator.m code as tend, where tend = $3 * 0.020255$. Then the average droplet count was made by calculating the average at the final time step of 500 since all the time steps now go out to the same actual time. Tables 7 and 8 show the results gained through this process and it is seen that the relative error for the droplets are more constant and have decreased slightly. From here, there is intuition that smoothing of the data prior to calculating the droplets could reduce the relative error being seen in these tables.

4 Conclusion and Future Work

4.1 Analysis of Neumann Boundary Conditions

The purpose of this research was to examine the phase separation phenomenon, also known as nucleation in a binary alloy. Along with examining the nucleation time, the hope was to identify behaviors of this alloy. Using the three component metal alloy case as a template [2] along with topological theory [3], the Cahn-Hilliard equation with additive noise was chosen to be used as this research's model. Incorporating Euler Characteristic into the model, the betti number calculations were performed. Identifying that small fluctuations at the early time steps were incorrectly calculating betti numbers, thresholding the data was implemented to correct this problem. Using topology, it can been seen that these betti curves represent the droplet formation numbers. From there, the calculation of the time horizon of decomposition was pin pointed for μ, σ, and the varying ϵ values and a narrower scope on when nucleation was occurring, was seen. Using the data gathered, averages of the time decomposition were calculated and then used to calculate the thickness of the boundary drops, d. Once d was calculated, equations to calculate the average interior (12) and boundary droplets (13) were used to construct the average prediction tables.

There remains future work to be done for the Neumann Boundary Conditions. Since there remains a relatively high relative error for the boundary drops, it would be worth wild to research it in further detail. Running analysis using the average time decomposition versus the average time decomposition step may decrease the relative error as seen in Table 7 and 8. Applying a smoothing function which can be done in Matlab, to the data prior to calculating the number of droplets is being looked into; this too may reduce the error being seen in the actual/prediction tables. Also examining the rate of nucleation for a given set of parameters seems to be an area of interest.

4.2 Periodic Boundary Conditions

Similar to the research done in the Neumann Boundary Condition case, the same procedure will be done for the Periodic Boundary Conditions. This means manipulation of the matrix padding in the initial coding step will need to be done. The manipulation will govern what will be considered what, and essential where a droplet will be counted. Analyzing how the parameters changing correspond to nucleation in the Periodic Conditions is an area to be research as well.

References

[1] Evelyn Sander and Thomas Wanner, et al., *Nonlinear Analysis and Computation for Partial Differential Equations*, Textbook

[2] Jonathan P. Desi, Hanein H. Edrees, Joseph J. Price, Evelyn Sander and Thomas Wanner, *The Dynamics of Nucleation in Stochastic Cahn-Morral Systems*, SIAM J. Applied Dynamical Systems, Vol. 10, No. 2, pp. 707-743

[3] Henle's Book, *A Combinatorial Introduction to Topology*, Ch. 5 *Homology of Complexes*