MODELING DNA REPLICATION USING THE SINE-GORDON EQUATION

BY

SUSAN ROGOWSKI

A Thesis Submitted to the Graduate Faculty of

WAKE FOREST UNIVERSITY GRADUATE SCHOOL OF ARTS AND SCIENCES

in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF ARTS

Mathematics and Statistics

May 2019

Winston-Salem, North Carolina

Approved By:

Sarah Raynor, Ph.D, Advisor

Frank Moore, Ph.D., Chair

John Gemmer, Ph.D.

Acknowledgments

Being able to complete my Master's in Math at Wake Forest University has truly been the opportunity of a lifetime. When I look at the "numbers", it still does not add up that I not only was given this opportunity but have also had such incredible success. I'm beyond thankful to the people in this department that took a chance on me.

Thank you to my mentors, Dr. Raynor and Dr. Gemmer, for supporting and encouraging me through this process. Many people, especially women and especially in STEM, have an extreme fear of failing. Sometimes, we fear failing so much that we don't even want to try. Dr. Gemmer, you have created an environment where students feel comfortable enough to fail. Thank you for creating a space where I'm not afraid to learn from my mistakes. Dr. Raynor, your work to support women and provide opportunities for them is inspiring! Thank you for "Raynorating" all my tough problems and being a great female role model.

Finally, I'd like to thank my family for their unyielding love and support. I would especially like to thank my grandmother, Ellen Morris, for initially introducing me to this incredible department. It is amazing how influential your network can be. Thank you for your not-so-gentle nudging to get me here and giving me the opportunity to recognize my full potential.

Table of Contents

List of Figures

Abstract

DNA replication begins when locally unzipped regions of several broken hydrogen bonds form which cause a partial unwinding of the double helix. These regions are often referred to as bubbles and their formation can be modeled using a chain of coupled pendula. The angular oscillations that occur are usually modeled using the sine-Gordon equation. In this model, the discrete analog of the sine-Gordon equation is derived. Instead of taking the continuous limit, the space step between pendula remains nonzero and the forces on each pendulum are considered. Additionally, the randomization of nitrogen bases, Adenine, Thymine, Guanine, and Cytosine, within the chain of pendula, is modeled by randomly selecting the corresponding coefficients for each base. The system of differential equations is solved and plotted using Verlet integration in MATLAB.

Chapter 1: Introduction

In this thesis we study a chain of coupled pendula using the sine-Gordon equation. This problem is motivated by the dynamics of DNA replication.

The DNA molecule is made of two polynucleotide chains which attach together to form a double helix. Each of these chains consist of nucleotides which contain one of the four nitrogen bases, adenine (A), thymine (T), guanine (G), or cytosine (C). The two polynucleotide chains attach together by the hydrogen bonds between complementary base pairs. The process of DNA replication begins when locally unzipped regions with several broken hydrogen bonds unwind the linked helices [2]. These regions are often referred to as DNA bubbles (**Fig. 1.1**).

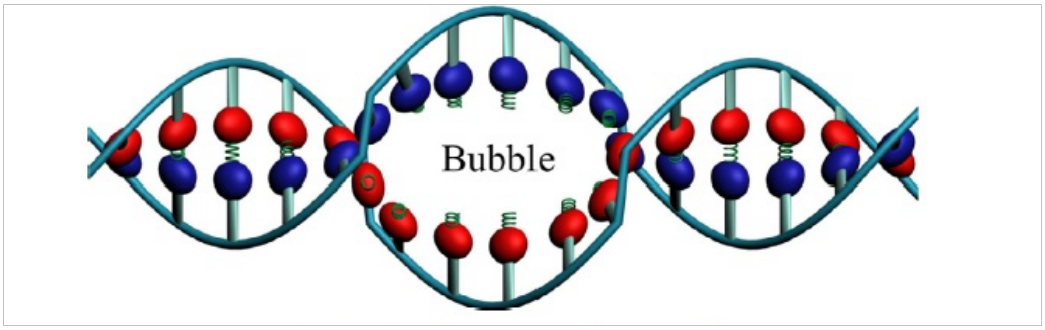


Figure 1.1: DNA double chain containing one bubble [1]

The formation of these bubbles allow us to consider the DNA molecule as a complex dynamical system where nonlinear waves activate and move along the polynucleotide chains [1]. There are several internal motions which occur in DNA molecules. These motions include transverse, longitudinal, and angular displacements of nitrous bases. With transverse motion, the polynucleotide chain only shifts up and down causing the nitrogen bases to move perpendicular to the direction of the wave as the wave moves along the chain. This motion does not allow the nitrogen bases to fully

rotate up to form the DNA bubble. Longitudinal motion results in a similar problem; the chain would shift right to left only allowing the nitrogen bases to move parallel to the direction of the wave. This is why in our model we will focus on consider the angular oscillations of nitrous bases. These oscillations are usually modeled using the Sine-Gordon equation:

$$I\frac{@^2y}{@t^2} \quad K^{0}a^2\frac{@^2y}{@x^2} + V\sin(y) = 0 \tag{1.1}$$

In this equation, $y(x;t)$ represents the angular displacement of the bases, I is the moment of inertia at each base, and K^{0} is the torsional stiffness of the chain. The coefficient V describes the interaction energy between bases and a is distance on the chain between nearest base pairs. The torsional stiffness along the chain occurs because of the covalent bonds that are present between neighboring bases. These bonds cause a restoring force to occur when the chain twists open. Also, the energy between base pairs is caused by the hydrogen bonds that exist between complementary base pairs.

In a polynucleotide chain where only one nitrogen base appears, $K^{0}; I; V$, and a remain constant. These types of chains have already been well studied. However, the DNA molecule is more heterogeneous in nature. So these studies have limited application to the actual behavior of this molecule.

In this thesis, we study the behavior of an inhomogenous polynucleotide chain. We specifically focus on expanding the model used in the work of Yakushevich et al. [1, 3, 4, 5]. Their model studies the DNA molecule as a complex dynamical system in which nonlinear waves can form and move along the polynucleotide chain.

Next, we will review the derivation of the model used in [1] and discuss what changes we will make to expand this model for this thesis. To obtain this model, Yakushevich drew an analogy between the angular oscillations of bases and the os-

cillations of pendula. In this setting, each pendulum represents a nitrogen base and the polynucleotide chain can be considered a chain of coupled pendula (Fig. 1.2). To start, the oscillations of a single pendulum is considered. To find the equation for this, Newton's second law of motion, $F = ma$ is used. First, because $y(x;t)$ represents angular displacement linear acceleration must be converted to angular acceleration by scaling the acceleration term by the length of the pendulum. So, one obtains $a = R^2 \frac{@^2 y}{@t^2}$. Then, F is the total sum of the forces acting on the pendulum. Because a single pendulum is being considered, there is only one force acting on it which is caused by the hydrogen bond present between complementary base pairs. Experimental evidence [6] has shown that this is well modeled by $F = \quad V \sin(y)$ where V represents the interaction energy between complementary base pairs. Now, combining these terms together, one has;

$$F = ma$$

$$V \sin(y) = mR^2 \frac{@^2 y}{@t^2}$$

$$I \frac{@^2 y}{@t^2} + V \sin(y) = 0$$

Here, note that $I = mR^2$ and $V = mgR$ where m is the mass of the pendulum, R is the length, and g is the gravitational constant.

To obtain the model for a chain of N coupled pendula representative of N base pairs, the covalent bond present between neighboring base pairs within a polynucleotide chain must be accounted for. So, to calculate the behavior of the n-th base pair in the chain, there are two additional forces to consider. These forces are represented by f_{right} and f_{left}. Then, note that these forces will act in opposite directions which is represented by alternating signs. So, adding these terms in, one obtains;

$$m_n R_n^2 \frac{@^2 y_n}{@t^2} + m_n g R_n \sin(y_n) \quad f_{right} + f_{left} = 0 \tag{1.2}$$

Now, due to the torsional stiffness of the polynucleotide chain, the covalent bonds are modeled by the extending and compressing of a spring between pendula. So, one expects the forces caused by the covalent bond to follow Hooke's Law. Thus, $f_{right} = K_{right}R_n(R_{n+1}y_{n+1} \quad R_ny_n)$ where K_{right} is the torsional stiffness of the chain. Similarly, $f_{left} = K_{left}R_n(R_ny_n \quad R_{n\ 1}y_{n\ 1})$. Then, f_{right} and f_{left} represent the forces acting on the n-th pendulum by the right and left pendulum respectively. Moving forward, this particular model assumes that $K_{right} = K_{left} = K$.

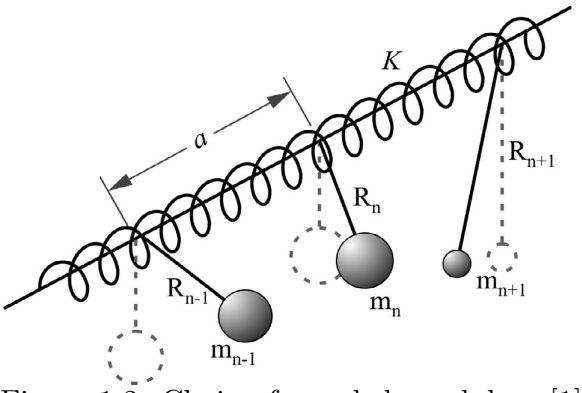


Figure 1.2: Chain of coupled pendulum [1]

Now, plugging in the f_{right} and f_{left} terms into Eq. (1.2) and letting $I_n = m_nR_n^2$, one obtains;

$$\begin{aligned} I_n\frac{@^2y_n}{@t^2} &= KR_n(R_{n+1}y_{n+1} \quad R_ny_n) \quad KR_n(R_ny_n \quad R_{n\ 1}y_{n\ 1}) \quad m_ngR_n\sin(y_n) \\ &= KR_n(R_{n+1}y_{n+1} \quad 2R_ny_n + R_{n\ 1}y_{n\ 1}) \quad m_ngR_n\sin(y_n) \end{aligned} \tag{1.3}$$

Then, notice that the coefficents are spatially dependent. So, letting x_n represent the coordinate of the n-th base pair in the chain, one has;

$$I_n = I(x_n) = m_nR_n^2 \qquad m_n = m(x_n) \qquad R_n = R(x_n) \tag{1.4}$$

Now, consider the function $R(x_n)y(x_n;t_n)$. Then, the second derivative of this function can be approximated using a centered difference approximation. This approximation is explained in detail in the Appendix. Using this approximation, $\frac{@^2}{@x^2}(R(x_n)y(x_n;t_n))$

becomes;

$$\frac{@^2}{@x^2}(R(x_n)y(x_n;t_n)) \quad \frac{R(x_{n+1}y(x_{n+1};t_n) \quad 2R(x_n)y(x_n;t_n)+R(x_{n\ 1})y(x_{n\ 1};t_n)}{\Delta x^2}$$

$$\Delta x^2\frac{@^2(R_ny_n)}{@x^2} \quad R_{n+1}y_{n+1} \quad 2R_ny_n+R_{n\ 1}y_{n\ 1} \tag{1.5}$$

Then, using (1.5) and (1.4), Eq. (1.3) can be written as;

$$I(x_n)\frac{@^2y_n}{@t^2}=KR(x_n)\Delta x^2\frac{@^2(R(x_n)y_n)}{@x^2} \quad m(x_n)gR(x_n)\sin(y_n) \tag{1.6}$$

Now, as shown in Figure 1.2, a will represent the spacing between neighboring base pairs. So, $\Delta x^2=a^2$. Next, to derive the complete PDE model, the continuum limit is taken. It is important to note that taking the continuum limit rigorously is beyond the scope of this thesis. However, it is assumed that there is a nice relationship between K and the spacing a so that as the spacing decreases, the spring constant will increase proportionally. Thus, the continuum limit yields;

$$\lim_{a\ !\ 0}\ I(x_n)\frac{@^2y_n}{@t^2} \quad KR(x_n)a^2\frac{@^2(R(x_n)y_n)}{@x^2}+m(x_n)gR(x_n)\sin(y_n) \tag{1.7}$$

$$=\lim_{a\ !\ 0}I(x)\frac{@^2y}{@t^2} \quad KR(x)a^2\frac{@^2(R(x)y)}{@x^2}+m(x)gR(x)\sin(y)=0$$

Next, to derive the final equation of the model used in [1] $\frac{@^2(R(x)y(x;t))}{@x^2}$ is considered. This derivative is taken by first taking the first partial derivitive using the product rule. So, one has;

$$\frac{@(R(x)y(x;t))}{@x}=\frac{@R(x)}{@x}y(x;t)+R(x)\frac{@y(x;t)}{@x}$$

$$\frac{@^2(R(x)y(x;t))}{@x^2}=\frac{@}{@x}\ \frac{@R(x)}{@x}y(x;t)+R(x)\frac{@y(x;t)}{@x}$$

$$=\frac{@^2R(x)}{@x^2}y(x;t)+\frac{@R(x)}{@x}\frac{@y(x;t)}{@x}+\frac{@R(x)}{@x}\frac{@y(x;t)}{@x}+R(x)\frac{@^2y(x;t)}{@x^2}$$

$$=y\frac{@^2R(x)}{@x^2}+2\frac{@R(x)}{@x}\frac{@y}{@x}+R(x)\frac{@^2y}{@x^2} \tag{1.8}$$

Now, using (1.8), Eq. (1.7) can be written as

$$I(x)\frac{@^2y}{@t^2} = KR(x)a^2 \ \ y\frac{@^2R(x)}{@x^2} + 2\frac{@R(x)}{@x}\frac{@y}{@x} + R(x)\frac{@^2y}{@x^2} \qquad m(x)gR(x)\sin(y)$$

$$= KR^2(x)a^2\frac{@y^2}{@x^2} + KR(x)a^2 \ \ 2\frac{@y}{@x}\frac{@R(x)}{@x} + y\frac{@^2R(x)}{@x^2} \qquad m(x)gR(x)\sin(y) \tag{1.9}$$

It is important to note here that in the homogeneous case when one only considers a chain of one type of base pair, $R(x)$ is constant. Thus, $\frac{@R(x)}{@x} = \frac{@^2R(x)}{@x^2} = 0$. So, the middle term in Eq. (1.9) goes to zero and we are left with the classical sine-Gordon equation (Eq. (1.1)).

After this derivation, Yakushevich et al. [1] considers a chain of n homogeneous regions. To understand the movement of DNA bubbles throughout the chain, the approximate energy profile of the DNA sequence was calculated.

In this study, we make several adjustments to this model. First, we will not assume that $K_{right} = K_{left}$. This will dictate how we account for the boundaries between regions of homogeneous base pairs. Thus, we will use different methods to account for the jumps in coefficients at these boundaries. Additionally, instead of quantitatively calculating the energy profile, we would like to make a qualitative comparison to other models.

In order to accurately take into account the heterogeneous nature of DNA, we would like to randomly generate sequences of base pairs. So, for each of our n-homogeneous regions, we will have a random number of A, T, C, and G bases present. Then, we will use the method of concentrations [7] to create the homogeneous structure within each region. Additionally, a completely inhomogeneous chain is simulated where each base pair may differ.

This thesis will be organized as follows. Chapter 2 will discuss the derivation and set up of the equation for our model. In Chapter 3 we will discuss the methods used

to solve the system of differential equations. We conclude with a discussion chapter on relevance and future direction.

Chapter 2: Model

2.1 Derivation

In this model, the discrete analog of the sine-Gordon equation is derived. Instead of taking the continuum limit, the space step between pendula remains nonzero. We begin by considering Newton's second law of motion, $F = ma$. In this model we consider a chain of coupled pendula, and we would like to understand the behavior of motion within the chain during the replication process. So, we will denote $y(x_i; t_j) = y_i^j$ to be the angular displacement of the i^{th} pendulum in the system at the j^{th} time step. Because we are not yet taking time steps in this section, we will simplify this notation to $y_i^j = y_i$. Then, we will use $F = ma$ to solve for y_i. So, our equation becomes $F_i = m_i a_i$, where m_i is the mass of the i^{th} pendulum in the chain, F_i is the sum of the forces on the i^{th} pendulum, and a_i is the acceleration. Now, because y_i represents the angular displacement, as in the introduction we have that $a_i = R_i^2 \frac{@^2 y_i}{@t^2}$, where R_i represents the length of the pendulum.

Since we would like to study a system where nitrogen bases are randomly selected for each pendulum in the chain, we compute F_i by considering the i^{th} pendulum and let the two nearest neighboring pendula represent possibly distinct nitrogen bases. Then, similar to the derivation in [1] there are three forces we need to consider on the center pendulum, the two forces acting on the i^{th} pendulum by the left and right neighbor and the force that occurs between chains due to the hydrogen bonds between complementary base pairs.

The left and right forces occur due to the covalent bonds between neighboring nitrogen bases so that a restoring force occurs when there is a relative displacement: $R_i y_i \quad R_{i+1} y_{i+1}(t)$. First, let us consider the force acting on the i^{th} pendulum by

the right neighbor. This can be modeled by a linear scaling of the distance between these pendula, so we will have $f_{\text{right}} = K_{i+1;i}R_i(R_{i+1}y_{i+1} \quad R_iy_i)$ [1]. The force from the left neighbor will have a similar form, $f_{\text{left}} = \quad K_{i\ \ 1;i}R_i(R_iy_i \quad R_{i\ \ 1}y_{i\ \ 1})$, where $K_{i\ \ 1;i}$ represents the torsional stiffness of the covalent bonds between the left or right pundulum with the i^{th} pendulum.

The third force acting on the pendulum is caused by the hydrogen bond present between complementary base pairs. There has been experimental evidence [6] that has shown that this force is well modeled by $f_{\text{center}} = k_{1\ \ 2}R_i^2\sin(y_i)$. Here we have that $k_{1\ \ 2}$ represents the coefficient of the interaction energy between complementary base pairs.

Now, we can add these forces together to obtain the left side of our equation. So, we have;

$$\begin{aligned} F_i &= f_{\text{right}} + f_{\text{left}} + f_{\text{center}} \\ &= K_{i+1;i}R_i(R_{i+1}y_{i+1} \quad R_iy_i) \quad K_{i\ \ 1;i}R_i(R_iy_i \quad R_{i\ \ 1}y_{i\ \ 1}) \quad k_{1\ \ 2}R_i^2\sin(y_i) \qquad (2.1) \\ &= K_{i+1;i}R_iR_{i+1}y_{i+1} \quad (K_{i+1;i} + K_{i\ \ 1;i})R_i^2y_i + K_{i\ \ 1;i}R_iR_{i\ \ 1}y_{i\ \ 1} \quad k_{1\ \ 2}R_i^2\sin(y_i) \end{aligned}$$

Then, setting this equal to our right side of the equation, we have

$$\begin{aligned} m_iR_i^2\frac{@^2y_i}{@t^2} = K_{i+1;i}R_iR_{i+1}y_{i+1} \quad (K_{i+1;i} + K_{i\ \ 1;i})R_i^2y_i \\ + K_{i\ \ 1;i}R_iR_{i\ \ 1}y_{i\ \ 1} \quad k_{1\ \ 2}R_i^2\sin(y_i) \qquad (2.2) \end{aligned}$$

Now, dividing each side by $m_iR_i^2$ we solve for $\frac{@^2y_i}{@t^2}$ and Eq. (2.2) becomes;

$$\frac{@^2y_i}{@t^2} = \frac{K_{i+1;i}R_{i+1}}{m_iR_i}y_{i+1} \quad \frac{(K_{i+1;i} + K_{i\ \ 1;i})}{m_i}y_i + \frac{K_{i\ \ 1;i}R_{i\ \ 1}}{m_iR_i}y_{i\ \ 1} \quad \frac{k_{1\ \ 2}}{m_i}\sin(y_i) \qquad (2.3)$$

It is most important to note here that our coefficients are spatially dependent. For a chain of coupled pendulum, we can generalize this using the equation

$\ddot{y}_i = \ _{i;+}y_{i+1} \quad \ _{i;0}y_i + \ _{i;} \ y_{i \ 1}$. These coefficients can be represented in a tri-diagonal matrix given below:

$$\Lambda = \begin{matrix} \ _{1;0} & \ _{1;+} & 0 & & 0 \\ \ _{2;} & \ _{2;0} & \ _{2;+} & & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & & \ _{N;} & \ _{N;0} \end{matrix} \tag{2.4}$$

Then, the generalized equation above can be converted to vector notation and becomes $\ddot{\boldsymbol{y}} = \Lambda \boldsymbol{y}$. Now, we can also use this to convert our system to vector notation. Thus, Eq. (2.3) will become:

$$\frac{@^2 \boldsymbol{y}}{@t^2} = \Lambda \boldsymbol{y} \quad D \sin(\boldsymbol{y}) \text{ where } D = \begin{matrix} \frac{k_{1 \ 1}}{m_1} & & 0 \\ \vdots & \ddots & \vdots \\ 0 & & \frac{k_{1 \ 2}}{m_N} \end{matrix} \tag{2.5}$$

From Eq. (2.3), we have that;

$$\begin{aligned} \ _{i;0} &= \frac{(K_{i+1;i} + K_{i \ 1;i})}{m_i}; \\ \ _{i;+} &= \frac{K_{i+1;i} R_{i+1}}{m_i R_i}; \text{ and} \\ \ _{i;} &= \frac{K_{i \ 1;i} R_{i \ 1}}{m_i R_i} \end{aligned} \tag{2.6}$$

From this derivation, we can find the kinetic and potential energy of the system. This will allow us to derive the Hamiltonian for the system which we will use to check the stability of our results.

2.2 Hamiltonian

In order to show that our system is working properly, we would like to check the stability. To do that, we will derive the Hamiltonian and show that it is conserved. In a later chapter, we will test the Hamiltonian with our results in MatLab.

In general the Hamiltonian is the total energy of a system. So, we have $H = V + T$, where V is the total potential energy and T is the total kinetic energy of the system. Kinetic energy is given by $T = \frac{1}{2}mv^2$. Here, v is the velocity function and because y_i represents angular displacement, we will have $v_i = R_i \frac{@y_i}{@t}$. To approximate these terms, we will use a central difference method for the first time derivative, as described in Appendix B. To notate this, we must now introduce superscripts to represent our time steps. So, y_i^j will represent the angular displacement of the i^{th} pendulum at time j. Thus, letting Δt represent our time step, the kinetic energy for the i^{th} pendulum will be:

$$T_i = \frac{1}{2}m_i \left(R_i \frac{@y_i}{@t} \right)^2$$
$$\frac{1}{2}m_i R_i^2 \left(\frac{y_i^{j+1} - y_i^{j-1}}{2\Delta t} \right)^2 \tag{2.7}$$

Before deriving the potential energy for the system, we will first prove the following theorem about the conservation of the Hamiltonian;

Theorem *Given* $\vec{F} = -\nabla V$ *and* $H = \sum_{i=1}^{N} \frac{1}{2}m_i \left(\frac{@y_i}{@t} \right)^2 + V(\vec{y})$, *then* $\frac{@H}{@t} = 0$

Proof:

Given $\vec{F} = -\nabla V$, we have that $F_i = m_i \frac{@^2 y_i}{@t^2} = -\frac{@V}{@y_i}$. Now, taking $\frac{@H}{@t}$, we have;

$$\begin{aligned}
\frac{@H}{@t} &= \frac{@}{@t}\sum_{i=1}^{N}\frac{1}{2}m_i\left(\frac{@y_i}{@t}\right)^2 + V(\mathbf{y}) \\
&= \sum_{i=1}^{N}\frac{1}{2}m_i 2\frac{@y_i}{@t}\frac{@^2y_i}{@t^2} + \frac{@}{@t}V(\mathbf{y}) \\
&= \sum_{i=1}^{N}\frac{@y_i}{@t}m_i\frac{@^2y_i}{@t^2} + \sum_{i=1}^{N}\frac{@V}{@y_i}\frac{@y_i}{@t} \\
&= \sum_{i=1}^{N}\frac{@y_i}{@t}\left(m_i\frac{@^2y_i}{@t^2} + \frac{@V}{@y_i}\right) \\
&= \sum_{i=1}^{N}\frac{@y_i}{@t}\left(\frac{@V}{@y_i} + \frac{@V}{@y_i}\right) \\
&= 0
\end{aligned}$$

■

Now, we would like to use this theorem to guarantee stability of our system. So, given the forces F_i derived in the previous section, we must check that the vector of these forces is the gradient of V. By Clairaut's theorem, the second mixed partials V will be equal. Thus, if $\mathbf{F}$ is a gradient of V, we will have;

$$\frac{@F_i}{@y_j} = \frac{@^2V}{@y_j@y_i} = \frac{@^2V}{@y_i@y_j} = \frac{@F_j}{@y_i} \quad 8\ i \neq j$$

Then, because the domain of our model is simply connected, the above condition is necessary and sufficient for $\mathbf{F} = \quad r\,V$. So, given Eq. (2.1) for F_i notice that when $j \neq i \quad 1$, $\frac{@F_i}{@y_j} = \frac{@F_j}{@y_i} = 0$. Then, for the $j = i+1$ case we have;

$$\frac{@F_i}{@y_{i+1}} = \frac{@}{@y_{i+1}}\left(K_{i+1;i}R_iR_{i+1}y_{i+1} \quad (K_{i+1;i} + K_{i\ \ 1;i})R_i^2y_i + K_{i\ \ 1;i}R_iR_{i\ \ 1}y_{i\ \ 1} \quad k_{1\ \ 2}R_i^2\sin(y_i)\right)$$

$$= K_{i+1;i}R_iR_{i+1}$$

$$\frac{@F_{i+1}}{@y_i} = \frac{@}{@y_{i+1}} \; K_{i+2;i+1}R_{i+1}R_{i+2}y_{i+2} \quad (K_{i+2;i+1} + K_{i+1;i})R_{i+1}^2 y_{i+1}$$
$$+ K_{i+1;i}R_{i+1}R_i y_i \quad k_{1\ 2}R_{i+1}^2 \sin(y_{i+1})$$
$$= K_{i+1;i}R_iR_{i+1}$$

Lastly, we must check the $j = i \quad 1$ case. So, we have;

$$\frac{@F_i}{@y_{i\ 1}} = \frac{@}{@y_{i\ 1}} \; K_{i+1;i}R_iR_{i+1}y_{i+1} \quad (K_{i+1;i} + K_{i\ 1;i})R_i^2 y_i + K_{i\ 1;i}R_iR_{i\ 1}y_{i\ 1} \quad k_{1\ 2}R_i^2 \sin(y_i)$$
$$= K_{i\ 1;i}R_iR_{i\ 1}$$

$$\frac{@F_{i\ 1}}{@y_i} = \frac{@}{@y_{i\ 1}} \; K_{i\ 1;i}R_{i\ 1}R_i y_i \quad (K_{i\ 1;i} + K_{i\ 1;i\ 2})R_{i\ 1}^2 y_{i\ 1}$$
$$+ K_{i\ 1;i\ 2}R_{i\ 1}R_{i\ 2}y_{i\ 2} \quad k_{1\ 2}R_{i\ 1}^2 \sin(y_{i\ 1})$$
$$= K_{i\ 1;i}R_iR_{i\ 1}$$

Thus, we have shown that $\frac{@F_i}{@y_j} = \frac{@F_j}{@y_i}; \ 8 \ i \ 6= j$ and we can conclude that $F = \quad rV$. We will use this to derive the total potential energy V.

Now, recall that $F_i = (m_iR_i^2)\frac{@^2y_i}{@t^2}$. In the previous section, after dividing by $m_iR_i^2$ to solve for $\frac{@^2y_i}{@t^2}$, we were able to define a matrix Λ to simplify our expression for $\frac{@^2y_i}{@t^2}$. We will now define a similar matrix Γ to simplify our notation for F_i.

$$\Gamma = \begin{matrix} {}_{1;1} & {}_{1;2} & 0 & & 0 \\ {}_{2;1} & {}_{2;2} & {}_{2;3} & & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & & {}_{N;N\ 1} & {}_{N;N} \end{matrix} \tag{2.8}$$

Then, from Eq. (2.1), we have;

$$\begin{aligned} {}_{i;i} &= (K_{i+1;i} + K_{i\ 1;i})R_i^2 \\ {}_{i;i\ 1} &= K_{i\ 1;i}R_iR_{i\ 1} \\ {}_{i;i+1} &= K_{i+1;i}R_iR_{i+1} \end{aligned} \tag{2.9}$$

Using this, we can write $\vec{F}$ in matrix-vector notation. So, we have;

$$\vec{F} = \Gamma\vec{y} \quad A\sin(\vec{y}); \text{where } A = \begin{matrix} k_1 \ {}_2R_1^2 & & 0 \\ \vdots & \ddots & \vdots \\ 0 & & k_1 \ {}_2R_N^2 \end{matrix} \tag{2.10}$$

From this, we can re-write Eq. (2.1) as;

$$F_i = \ {}_{i;i+1}y_{i+1} \quad {}_{i;i}y_i + \ {}_{i;i \ 1}y_{i \ 1} \quad A_{i;i}\sin(y_i) \tag{2.11}$$

Now, using $F_i = \frac{@V}{@y_i}$, we can integrate to solve for V. So, starting with $i = 1$, we have;

$$F_1 = \frac{@V}{@y_1} = \ {}_{1;2}y_2 \quad {}_{1;1}y_1 \quad A_{1;1}\sin(y_1)$$

$$V = \int \ {}_{1;2}y_2 \quad {}_{1;1}y_1 \quad A_{1;1}\sin(y_1)\, dy_1$$

$$= \ {}_{1;2}y_2y_1 \quad {}_{1;1}\frac{y_1^2}{2} + A_{1;1}\cos(y_1) + f_1(y_2;\dots;y_N) \tag{2.12}$$

Then, note that;

$$F_2 = \frac{@V}{@y_2} = \ {}_{2;1}y_1 \quad {}_{2;2}y_2 + \ {}_{2;3}y_3 \quad A_{2;2}\sin(y_2) \tag{2.13}$$

So, differentiating Eq. (2.12) with respect to y_2 should yield Eq. (2.13). We can use this to solve for f_1. Thus, we have;

$$\frac{@V}{@y_2} = \ {}_{1;2}y_1 + \frac{@f_1}{@y_2} = \ {}_{2;1}y_1 \quad {}_{2;2}y_2 + \ {}_{2;3}y_3 \quad A_{2;2}\sin(y_2) \tag{2.14}$$

Now, it is important to note here that $K_{i+1;i} = K_{i;i+1}$. Thus, given Eq. (2.9) we have;

$${}_{i;i+1} = K_{i+1;i}R_iR_{i+1} = K_{i;i+1}R_{i+1}R_i = \ {}_{i+1;i}$$

Therefore, Γ is a symmetric matrix. Now, because ${}_{1;2} = \ {}_{2;1}$, Eq. (2.14) gives us;

$$\frac{@f_1}{@y_2} = \quad {}_{2;2}y_2 + \ {}_{2;3}y_3 \quad A_{2;2}\sin(y_2)$$

$$\Rightarrow f_1 = \int \quad _{2;2}y_2 + \ _{2;3}y_3 \quad A_{2;2}\sin(y_2)\, dy_2$$

$$= \quad _{2;2}\frac{y_2^2}{2} + \ _{2;3}y_3y_2 + A_{2;2}\cos(y_2) + g(y_1; y_3; \dots; y_N)$$

Plugging this into Eq. (2.12), we have;

$$V = \ _{1;2}y_2y_1 \quad _{1;1}\frac{y_1^2}{2} + A_{1;1}\cos(y_1)$$

$$_{2;2}\frac{y_2^2}{2} + \ _{2;3}y_3y_2 + A_{2;2}\cos(y_2) + g(y_1; y_3; \dots; y_N) \quad (2.15)$$

In order to completely solve for V, we would need to integrate each F_i term and continue to solve for the function of integration. Instead, we will use Eq. (2.15) to make a reasonable guess. Then, we can check our guess with our condition $F_i = \ \frac{@V}{@y_i}$. So, we have;

$$V = \sum_{i=1}^{N} \quad \frac{_{i;i}y_i^2}{2} + A_{i;i}\cos(y_i) \ + \sum_{i=1}^{N-1} \ _{i;i+1}y_iy_{i+1} \quad (2.16)$$

Now, differentiating this with respect to y_i, we have;

$$\frac{@V}{@y_i} = \quad _{i;i}y_i + \ _{i;i\ 1}y_{i\ 1} + \ _{i;i+1}y_{i+1} \quad A_{i;i}\cos(y_i) = F_i$$

Thus, we have that Eq. (2.16) is the correct equation for V. So, given $H = V + T$ and Eq. (2.7) for kinetic energy, we have;

$$H = V + \sum_{i=1}^{N} T_i$$

$$= \sum_{i=1}^{N} \quad \frac{_{i;i}y_i^2}{2} \quad A_{i;i}\cos(y_i) \ + \sum_{i=1}^{N-1} \quad _{i;i+1}y_iy_{i+1} + \sum_{i=1}^{N} \frac{1}{2}m_iR_i^2 \ \frac{@y_i}{@t}^{\ 2} \quad (2.17)$$

Now, given that we derived V using the condition $\vec{F} = \ r\,V$, we have from our theorem that H will be conserved. Therefore, we will expect for our Hamiltonian to

remain constant for each time step. Using this result, we will be able to test that our model is consistent with expected behavior.

2.3 Change of Variables

The parameters for a homogeneous DNA chain are known and given in Table 1 [4]. Notice that these values are of extremely small magnitude. Given the original sine-Gordon equation, Eq. (1.1), one would need to divide through by $I = mR^2$ to solve for y. This would mean dividing by a magnitude of 10^{-44}, which would cause numerical resolution issues. Thus, we would like to derive new parameters which have unit magnitude in order to create a more computationally robust model. To do this, we will rescale time.

To determine our new scaling, we will first consider the homogeneous equation and use the parameters given in Table 1.

Table 1 - Coefficients from [4]

Type of Base	I 10^{-44} (kg m^2)	K' 10^{-18} (N m)	R (Å)	$k_{1,2}$ (N/m)
A	7.61	2.27	5.8	0.062
T	4.86	1.56	4.8	0.062
G	8.22	2.20	5.7	0.096
C	4.11	1.50	4.7	0.096

Here we have that $I = m_i R_i^2$, $K' = K_i R_i^2$. This is considering the homogeneous case so that the coefficients stay constant. So, we will have $R_{i+1} = R_{i-1} = R_i = R$ and $K_{i+1;i} = K_{i-1;i} = K$. Thus, Eq. (2.3) becomes:

$$I\frac{\partial^2 y_i}{\partial t^2} = K'(y_{i+1} - 2y_i + y_{i-1}) - k_{1,2}R^2 sin(y_i)$$

$$\frac{\partial^2 y_i}{\partial t^2} = \frac{K'}{I}(y_{i+1} - 2y_i + y_{i-1}) - \frac{k_{1,2}R^2}{I}sin(y_i(t))$$

$$= \frac{K}{m}(y_{i+1} - 2y_i + y_{i-1}) - \frac{k_{1,2}}{m}\sin(y_i) \tag{2.18}$$

Now, let $= \frac{K}{m}$ and $= \frac{k_1 \ _2}{m}$. Then, Eq. (2.18) becomes:

$$\frac{@^2 y_i}{@t^2} = \quad (y_{i+1} \quad 2y_i + y_{i \ 1}) \qquad sin(y_i) \tag{2.19}$$

Now, consider the units for and . So, we have;

$$= \left[\frac{K}{m} \right] = \frac{N=m}{kg} = \frac{\frac{kg\ m}{s^2} \frac{1}{m}}{kg} = 1=s^2$$

$$= \left[\frac{k_1 \ _2}{m} \right] = \frac{(N=m)}{kg} = \frac{\frac{kg\ m}{s^2} \frac{1}{m}}{kg} = 1=s^2$$

Then, using Table 1, we can derive our new parameters given in Table 2.

Table 2 - Coefficients for Eq. (2.5)		
Type of Base	10^{25} $(1=s^2)$	10^{23} $(1=s^2)$
A	2.98	2.74
T	3.21	2.94
G	2.68	3.79
C	3.65	5.16

Notice that has a slightly larger magnitude (10^{25}) than (10^{23}). This is because the covalent bonds between base pairs are stronger than the hydrogen bonds between complementary base pairs. The weaker nature of the hydrogen bond allows the double helix to unzip, generating DNA replication.

This magnitude also dictates how we will re-scale our equation. We would like to use the larger coefficient to change the variable for time in the equation and create a dimensionless equation. This will allow more convenient visualization and faster computations. We would like our new time coeffient to have unit 1. So, define $= p\_t$. Then, checking the units we have

$$= \quad p\_t \quad = \quad \sqrt{\frac{1}{s^2}} s = 1$$

Now, with $= \sqrt{}\, t$, Eq. (2.19) becomes:

$$\frac{@^2 y_i}{@\ ^2} = \quad (y_{i+1} \quad 2y_i + y_{i\ 1}) \qquad \sin(y_i) \tag{2.20}$$

Then, we divide all terms in Eq. (2.20) by .

$$\frac{@^2 y_i}{@\ ^2} = (y_{i+1} \quad 2y_i + y_{i\ 1}) \quad - \sin(y_i) \tag{2.21}$$

Now, it is simple to extend this change of variables to the inhomogeneous equation. To do this, consider Eq. (2.5) from our Derivation section. Here, the inhomogeneous case is given in matrix notation. The spatially dependent coefficients that are present in the first and second terms are given in matrices Λ and D. Notice that the entries for Λ; $_{i;0}$; $_{i;+1}$; $_{i;}$ will have the same order of magnitude as . Similarly, the entries for D will have the same order of magnitude as . So, to re-scale Eq. (2.5) we will use the maximum entry of matrix Λ which we will denote as $_M$. Then, we again define $= \sqrt{}_M\, t$ and Eq. (2.21) becomes;

$$\frac{@^2 \mathbf{y}}{@\ ^2} = \frac{1}{M}\Lambda \mathbf{y} \quad \frac{1}{M} D \sin(\mathbf{y}) \tag{2.22}$$

Chapter 3: Method

3.1 Method of Concentrations

In order to initially take into account the inhomogeneity of the DNA structure, we first implement the method of concentrations. This idea was first presented in the work of Dominguez-Adame et al [7] and was further implemented in the work of Yakushevich et al [4]. Taking into account the concentrations of each nucleotide, A, T, C, and G, in the chain, we replace the coefficients in Eq. (2.4) with the following:

$$I = C_A I_A + C_T I_T + C_C I_C + C_G I_G \qquad K = C_A K_A + C_T K_T + C_C K_C + C_G K_G$$

$$R = C_A R_A + C_T R_T + C_C R_C + C_G R_G \qquad k_{1\ 2} = k_{A\ T}(C_A + C_T) + k_{G\ C}(C_G + C_C)$$

$$m = C_A m_A + C_T m_T + C_C m_C + C_G m_G$$

Note that $C_j = N_j/N$ is the concentration of the jth nitrogen base where j represents the corresponding nitrogen base ($j \in \{A, T, C, G\}$). N_j is the number of jth nitrogen bases and N is the total number of bases in the chain. Also, note that this is only an approximation of the heterogeneous nature of the DNA molecule, however it is a good step towards modeling it accurately.

Now, using this method on the entire chain of pendula creates a homogeneous system which takes into account the inhomogenous nature of DNA. However, in order to work towards a more realistic model, we split the chain into regions where each region has a randomly generated concentration of nitrogen bases.

So, we first look at a chain of a hundred pendula. Then, for every 25 pendulum, a random number is generated for N_j so that $N_A + N_T + N_G + N_C = 25$. From this we derive C_j and the coefficients listed above. Then, we can combine these to generate the and values for Eq. (2.19). Now, because every region will have a different

set of concentrations, we will have four distinct values for and . Thus, in order to change variables for the entire chain of pendula, we must use the maximum . We can denote this as $_M$. Next, letting $k = 1; 2; 3;$ or 4 denote which region of the chain we are in, Eq. (2.21) becomes

$$\frac{@^2 y_i}{@ \ ^2} = \frac{k}{M}(y_{i+1} \quad 2y_i + y_{i \ 1}) \quad \frac{k}{M}\sin(y_i) \tag{3.1}$$

Eq. (3.1) will work within each homogeneous region of our chain. However, between each region it is important to note that our $_k$ will change abruptly. To handle this behavior, recall Eq. (2.3). From this equation, we have three distinct coefficients in front of $y_{i+1}; y_i$, and $y_{i \ 1}$. So, at each jump between regions, we will have a new equation for $\frac{@^2 y_i}{@t^2}$ for the last and first pendulum of each region. Letting J_k represent the last pendulum of the k^{th} region and $J_k + 1$ represent the first pendulum in the neighboring region, we have;

$$\frac{@^2 y_{J_k}}{@t^2} = \frac{K_{J_k+1;J_k} R_{J_k+1}}{m_{J_k} R_{J_k}} y_{J_k+1} \quad \frac{K_{J_k+1;J_k} + K_{J_k \ 1;J_k}}{m_{J_k}} y_{J_k} + \frac{K_{J_k \ 1;J_k} R_{J_k \ 1}}{m_{J_k} R_{J_k}} y_{J_k \ 1} \quad \frac{k_{1 \ 2;J_k}}{m_{J_k}} \sin(y_{J_k}) \tag{3.2}$$

$$\frac{@^2 y_{J_k+1}}{@t^2} = \frac{K_{J_k+2 J_k+1} R_{J_k+2}}{m_{J_k+1} R_{J_k+1}} y_{J_k+2} \quad \frac{K_{J_k+2;J_k+1} + K_{J_k;J_k+1}}{m_{J_k+1}} y_{J_k+1} + \frac{K_{J_k;J_k+1} R_{J_k}}{m_{J_k+1} R_{J_k+1}} y_{J_k} \quad \frac{k_{1 \ 2;J_k+1}}{m_{J_k+1}} \sin(y_{J_k+1}) \tag{3.3}$$

Note, we have used the method of concentrations to create four homogeneous regions. So, we have four different values for $K; R; m$, and $k_{1 \ 2}$. We can denote this as we did above with $_k$. So, because y_{J_k} corresponds to the end of the k^{th} region, we will have $K_k = K_{J_k \ 1;J_k}; R_k = R_{J_k} = R_{J_k \ 1}; m_k = m_{J_k}$, and $k_{1 \ 2;J_k} = k_{1 \ 2;k}$ where k

corresponds to the particular homogeneous region. Thus, the equations above become:

$$\frac{@^2 y_{J_k}}{@t^2} = \frac{K_{J_k+1;J_k} R_{k+1}}{m_k R_k} y_{J_k+1} \quad \frac{K_{J_k+1;J_k} + K_k}{m_k} y_{J_k} + \frac{K_k R_k}{m_K R_k} y_{J_k \ 1} + \frac{k_{1 \ 2;k}}{m_k} \sin(y_{J_k})$$

$$= \frac{K_{J_k+1;J_k} R_{k+1}}{m_k R_k} y_{J_k+1} \quad \frac{K_{J_k+1;J_k} + K_k}{m_k} y_{J_k} + \ _k y_{J_k \ 1} \quad _k \sin(y_{J_k}) \tag{3.4}$$

$$\frac{@^2 y_{J_k+1}}{@t^2} = \frac{K_{k+1} R_{k+1}}{m_{k+1} R_{k+1}} y_{J_k+1} \quad \frac{K_{k+1} + K_{J_k;J_k+1}}{m_{k+1}} y_{J_k+1} + \frac{K_{J_k;J_k+1} R_k}{m_{k+1} R_{k+1}} y_{J_k} \quad \frac{k_{1 \ 2;k+1}}{m_{k+1}} \sin(y_{J_k+1})$$

$$= \ _{k+1} y_{J_k+1} \quad \frac{K_{k+1} + K_{J_k;J_k+1}}{m_{k+1}} y_{J_k+1} + \frac{K_{J_k;J_k+1} R_k}{m_{k+1} R_{k+1}} y_N \quad _{k+1} \sin(y_{J_k+1}) \tag{3.5}$$

Now, note that the coefficients in front of $y_{J_k \ 1}$ and y_{J_k} in each of these equations are of the same order of magnitude as in Table 2 above. So, once again in order to reflect the change of variables in our system, we must divide each term by $_M$. Thus, our final equations for jumps between homogeneous regions become:

$$\frac{@^2 y_{J_k}}{@ \ ^2} = \frac{1}{_M} \quad \frac{K_{J_k+1;J_k} R_{k+1}}{m_k R_k} y_{J_k+1} \quad \frac{K_{J_k+1;J_k} + K_k}{m_k} y_{J_k} + \ _k y_{J_k \ 1} \quad _k \sin(y_{J_k}) \tag{3.6}$$

$$\frac{@^2 y_{J_k+1}}{@ \ ^2} = \frac{1}{_M} \quad _{k+1} y_{J_{k+1}} \quad \frac{K_{k+1} + K_{J_k;J_k+1}}{m_{k+1}} y_{J_k+1} + \frac{K_{J_k;J_k+1} R_k}{m_{k+1} R_{k+1}} y_N \quad _k \sin(y_{J_k+1}) \tag{3.7}$$

As we saw in the derivation section, these coefficients are spatially dependent and can be represented in a tri-diagonal matrix as seen in Eq. (2.4). Then, given Eqs. (3.6-7), we have derived new rules for the entries to this matrix. Above, we let J_k correspond to the k^{th} jump between homogeneous regions. So, from Eq. (3.1), we have at $J_{k \ 1} + 1 \quad i \quad J_k \quad 1$;

$$_{i;0} = \ _{i;+} = \ _{i;} \ = \frac{k}{M} \tag{3.8}$$

Then, from Eq. (3.6), we have at $i = J_k$;

$$\begin{aligned} {}_{J_K;+} &= \frac{1}{M} \frac{K_{J_k+1;J_k} R_{k+1}}{m_k R_k} \\ {}_{J_k;0} &= \frac{1}{M} \frac{K_{J_k+1;J_k} + K_k}{m_k} \\ {}_{J_k;} &= \frac{k}{M} \end{aligned} \tag{3.9}$$

Lastly, from Eq. (3.7), we have at $i = J_k + 1$;

$$\begin{aligned} {}_{J_{k+1};+} &= \frac{k+1}{M} \\ {}_{J_k+1;0} &= \frac{1}{M} \frac{K_{k+1} + K_{J_k;J_k+1}}{m_{k+1}} \\ {}_{J_k+1;} &= \frac{1}{M} \frac{K_{J_k;J_k+1} R_k}{m_{k+1} R_{k+1}} \end{aligned} \tag{3.10}$$

Finally, note that each of these parameters have been derived in this section with the exception of $K_{J_k;J_k+1}$ and $K_{J_k+1;J_k}$. First, we have that $K_{J_k+1;J_k} = K_{J_k;J_k+1}$ because they both represent the same spring constant between the $J_k + 1$ and J_k^{th} pendula. Now, within the homogeneous regions, we assume that the spring constant between pendula remain the same. However, at the jump, we must account for the change. In our next section, we will explain how we handle the differing spring constants.

3.2 Taking the Average for K

Each nitrogen base has a unique structure and reacts differently to one another. This is illustrated by the differing coefficients for the hydrogen bonds. This would imply that each covalent bond differs between base pairs so that there is a unique K to represent the torsional stiffness. To organize and select the correct values, consider

the following symmetric matrix.

$$\begin{bmatrix} K_{A;A} & K_{A;T} & K_{C;A} & K_{G;A} \\ K_{A;T} & K_{T;T} & K_{C;T} & K_{G;T} \\ K_{C;A} & K_{C;T} & K_{C;C} & K_{C;G} \\ K_{G;A} & K_{G;T} & K_{C;G} & K_{G;G} \end{bmatrix}$$

Note that the literature only gives values for the homogenous case [4]. However, we can consider a similar approach to the Method of Concentrations and use the average to approximate these values. So, we have;

$$K_{A;T} = \frac{K_{A;A} + K_{T;T}}{2} \qquad K_{A;C} = \frac{K_{A;A} + K_{C;C}}{2} \qquad K_{A;G} = \frac{K_{A;A} + K_{G;G}}{2}$$

$$K_{C;T} = \frac{K_{T;T} + K_{C;C}}{2} \qquad K_{G;T} = \frac{K_{T;T} + K_{G;G}}{2} \qquad K_{C;G} = \frac{K_{C;C} + K_{G;G}}{2}$$

The values for $K_i^{\emptyset} = K_i R_i^2$ are given in [1]. From this, we can solve for K_i which is shown in Table 3.

Table 3 - Coefficients from [1]

Type of Base	$K^{\emptyset}$ 10 ^{18} (N m)	R (Å)	K (N/m)
A	2.27	5.8	6.75
T	1.56	4.8	6.77
G	2.20	5.7	6.77
C	1.50	4.7	6.79

Then, we can use our equations above to solve for our remaining values and we can fill in our matrix above. So, we will have

$$\begin{bmatrix} 6.75 & 6.76 & 6.77 & 6.76 \\ 6.76 & 6.75 & 6.78 & 6.77 \\ 6.77 & 6.78 & 6.79 & 6.78 \\ 6.76 & 6.77 & 6.78 & 6.77 \end{bmatrix}$$

In the previous section, we derived homogeneous regions within our chain using the method of concentrations. Within these regions, only $K_{A;A}; K_{T;T}; K_{C;C}$ and $K_{G;G}$

were used to derive our homogeneous spring constant K. So, within these regions, we will not need to consider the spring constant between differing base pairs. However, at the jumps between homogeneous regions, we will have two pendula representing different regions. In this way, they must be treated as two different base pairs. Thus, we must account for this using the average for the spring constant. Using the same notation from the previous section, we will have $K_{N+1;N} = \frac{K_k + K_{k+1}}{2}$.

3.3 Verlet Integration

The numerical scheme used to solve our system of differential equations is known in the literature under various names; including the Verlet-Störmer Method and the leapfrog method [8]. To derive this scheme, we consider the classical sine-Gordon equation.

$$\frac{@^2y}{@t^2} = \frac{@^2y}{@x^2} \quad \sin(y) \tag{3.11}$$

Now, we have that y represents our angular displacement in terms of space and time. To represent this, let $y(t_j; x_i) = y_i^j$. Now, we must introduce a discrete temporal and spatial mesh. The temporal mesh is given by $t_j \ 2 \ ft_1; t_2; \cdots; t_{M+1}g$ with time step $\Delta t = t_{j+1} \quad t_j$. The spatial mesh will be $x_i \ 2 \ fx_1; x_2; \cdots; x_N g$. Then, let M be the total time steps taken and N be the number of pendula in our chain. Now, letting $x_N = L$ where L is the total length of the chain, we will have $\Delta x = \frac{L}{N}$. Also, we have $x_1 = t_1 = 0$. So, using this mesh and the notation above, we can replace $\frac{@^2y}{@t^2}$ and $\frac{@^2y}{x^2}$ with a centered difference approximation. This approximation is derived in Appendix B. So, we have

$$\frac{@^2y}{@t^2} = \frac{y_i^{j+1} \quad 2y_i^j + y_i^{j\ 1}}{\Delta t^2} + O(\Delta t^2); \text{ and} \tag{3.12}$$

$$\frac{@^2y}{@x^2} = \frac{y_{i+1}^j \quad 2y_i^j + y_i^{j\ 1}}{\Delta x^2} + O(\Delta x^2): \tag{3.13}$$

Now, plugging these equations into Eq. (3.11), we obtain the discrete equation;

$$\frac{y_i^{j+1} \quad 2y_i^j + y_i^{j\ 1}}{\Delta t^2} = \frac{y_{i+1}^j \quad 2y_i^j + y_i^{j\ 1}}{\Delta x^2} \quad \sin(y_i^j): \tag{3.14}$$

Then, solving for the $j+1$ time step, we obtain;

$$y_i^{j+1} = \Delta t^2 \ \frac{y_{i+1}^j \quad 2y_i^j + y_i^{j\ 1}}{\Delta x^2} \quad \Delta t^2 \sin(y_i^j) + 2y_i^j \quad y_i^{j\ 1}: \tag{3.15}$$

Then, notice that because this scheme uses the information from the $j \quad 1^{st}$ time step, this will not work when $j+1=2$. So, we need a separate equation for this time step. Now, note that in Eq. (3.12-13), we had error terms $O(\Delta t^2)$ and $O(\Delta x^2)$. This means that Eq. (3.15) is a second order numerical scheme. Thus, our equation for y_i^2 must also be second order. So, to find this equation we Taylor expand for y_i^2 about 0 up to $O(\Delta t^3)$.

Now, note that $y_i^2 = y(t_2; x_i)$. Then, with time step Δt, we have that $t_2 = t_1 + \Delta t = 0 + \Delta t$. Thus, $y_i^2 = y(\Delta t; x_i)$. So, Taylor expanding $y(\Delta t; x_i)$ about 0 gives;

$$y_i^2 = y(\Delta t; x_i) = y(0; x_i) + \frac{@y(0; x_i)}{@t}\Delta t + \frac{@^2y(0; x_i)}{@t^2}\frac{\Delta t^2}{2} + O(\Delta t^3) \tag{3.16}$$

Now, the first two terms in this equation will be our initial position and initial velocity. Then, for the third term, we have $\frac{@^2y(0;x_i)}{@t^2} = \frac{@^2y}{@t^2}_{\ t=0}$. For this term, we can use our original equation (3.11). So, denoting our initial conditions as

$y(0;x_i) = f(x_i)$; $\frac{@y}{@t}_{t=0} = g(x_i)$, Eq. (3.16) becomes:

$$y_i^2 = f(x_i) + g(x_i)\Delta t + \frac{\Delta t^2}{2} \frac{@^2 y}{@x^2}_{t=0} \sin(y)_{t=0}$$

$$= f(x_i) + \Delta t g(x_i) + \frac{\Delta t^2}{2} \frac{y_{i+1} \quad 2y_i + y_{i\ 1}}{\Delta x^2} \sin(y_i) \tag{3.17}$$

Next, we can convert these equations to vector-matrix notation. So, because N is the total number of space steps or pendula in the system and M is the total number of time steps, we will let y be an $N \quad M$ matrix where each column is a time step. Now, in MatLab notation, we have that $y(:;j)$ is the j^{th} column of y. Then, for our initial conditions, we will have $y(:;1) = f(x)$ and $g(x) = 0$.

Now, notice that calculating the second spatial derivative at the j^{th} time step, $\frac{@^2 y^j}{@x^2}$, can be written as an $N \quad N$ matrix acting on the $y(:;j)$ vector. So, we have;

$$\frac{@^2 y^j}{@x^2} = \frac{1}{\Delta x^2}\Lambda y(:;j) = \frac{1}{\Delta x^2} \begin{matrix} 2 & 1 & 0 & & 0 \\ 1 & 2 & 1 & & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & & 1 & 2 \end{matrix} \begin{matrix} y_1^j \\ y_2^j \\ \vdots \\ y_N^j \end{matrix}$$

$$= \frac{1}{\Delta x^2} \begin{matrix} 2y_1^j + y_2^j \\ y_1^j \quad 2y_2^j + y_3^j \\ \vdots \\ y_{N\ 1}^j \quad 2y_N^j \end{matrix} \tag{3.18}$$

Then, using the above and converting to Matlab notation, we can re-write Eq. (3.17) and Eq. (3.15) as;

$$y(:;2) = y(:;1) + \frac{\Delta t^2}{2\Delta x^2} \Lambda y(:;1) \quad \sin(y(:;1)) \tag{3.19}$$

$$y(:;j+1) = \frac{\Delta t^2}{\Delta x^2}\Lambda y(:;j) \quad \Delta t^2 \sin(y(:;j)) + 2y(:;j) \quad y(:;j \quad 1) \tag{3.20}$$

Now, note that in our inhomogeneous model which is derived in the first section includes spatially dependent coefficients in front of the $\frac{@^2 y}{@x^2}$ and $\sin(y)$ terms. As

explained in the Derivation and Method of Concentrations section, the coefficients in front of $\frac{@^2y}{@x^2}$ can be built into the matrix Λ. Also, shown in Eq. (2.5), the coefficients in front of the $\sin(y)$ term can be built into a diagonal matrix D. Thus, for our inhomogeneous model, Eq. (2.19-20) become;

$$y(:;2) = y(:;1) + \frac{\Delta t^2}{2\Delta x^2} \ \Lambda y(:;1) \quad D\sin(y(:;1)) \tag{3.21}$$

$$y(:;j+1) = \frac{\Delta t^2}{\Delta x^2}\Lambda y(:;j) \quad \Delta t^2 D\sin(y(:;j) + 2y(:;j) \quad y(:;j \quad 1) \tag{3.22}$$

3.4 Stability of the Verlet Scheme

In order to ensure that Verlet Integration gives accurate approximations of the solution, we must ensure that it is numerically stable. So, we will derive the appropriate Courant - Friedrichs - Lewy (CFL) condition [9].

In general, it is harder to guarantee stability for a nonlinear PDE like the sine-Gordon equation. However if we remove the sine term from the sine-Gordon equation, we are left with the linear wave equation;

$$\frac{@^2y}{@t^2} = c^2\frac{@^2y}{@x^2} \tag{3.23}$$

Stability of the linear piece of the sine-Gordon equation is a necessary condition for stability of the nonlinear equation itself. So, we can use Eq. (3.23) to derive the necessary CFL condition for our numerical scheme. Now, the general solution to the wave equation is known [9]. Given the initial condition $y(0;x) = f(x)$, we have;

$$y(t;x) = \frac{f(x \quad ct)}{2} + \frac{f(x+ct)}{2} \tag{3.24}$$

Thus, our actual solution for the point $y(t_{j+1};x_i)$ must depend on the values $f(x_i \quad c\Delta t)$ and $f(x_i + c\Delta t)$. However, our numerical approximation uses the points

$y(t_j; x_{i+1}); y(t_j; x_{i-1})$, and $y(t_j; x_i)$ to approximate the solution $y(t_{j+1}; x_i)$. So, notice that if $x_i \quad c\Delta t$ lie outside of $[x_{i+1}; x_{i-1}]$ then the actual solution (Eq. (3.24)) will change without changing our numerical approximation [9]. Geometrically, we have;

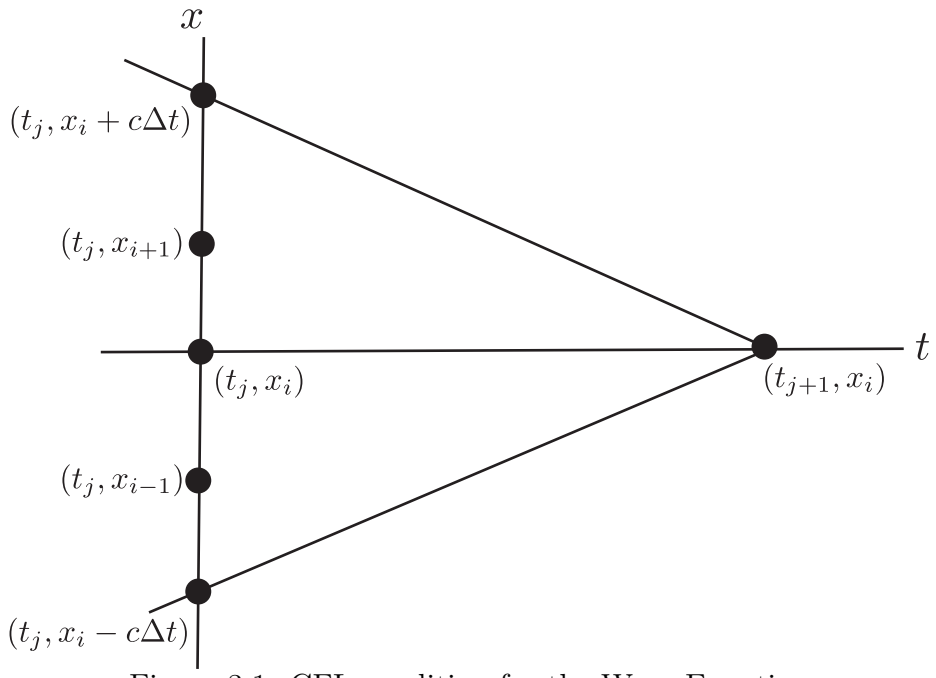


Figure 3.1: CFL condition for the Wave Equation

Therefore, we must require that:

$$\begin{aligned} & x_{i-1} \quad x_i \quad c\Delta t \quad x_{i+1} \\ & x_{i-1} \quad x_i \quad c\Delta t \quad x_{i+1} \quad x_i \\ & \Delta x \quad c\Delta t \quad \Delta x \\ &) \quad \frac{c\Delta t}{\Delta x} \quad 1 \end{aligned} \tag{3.25}$$

Then, Eq. (3.25) is the known CFL condition for the linear wave equation [9].

Now, it is important to note that in our inhomogeneous model, instead of the constant c in front of $\frac{\partial^2 y}{\partial x^2}$ we have spatially dependent coefficients. Recall that these coefficients are built into matrix Λ and the entries for this matrix are given in Eqs. (3.8-10). So, in order to ensure stability in our model, we must take c to be the

maximum value of the entries of Λ. Let us denote this value as $_M$. Then, Eq. (3.25) will become;

$$\frac{_M \Delta t}{\Delta x} \quad 1 \tag{3.26}$$

Then, the value of Δx will be the approximate spacing between each nitrogen base given in [5]. So, this condition will be used to determine our appropriate time step. Thus, we will have;

$$\Delta t \quad \frac{\Delta x}{_M} \tag{3.27}$$

Then, to make sure this inequaility is satisfied, we will set Δt to be;

$$\Delta t = \frac{1}{10}\frac{\Delta x}{_M} \tag{3.28}$$

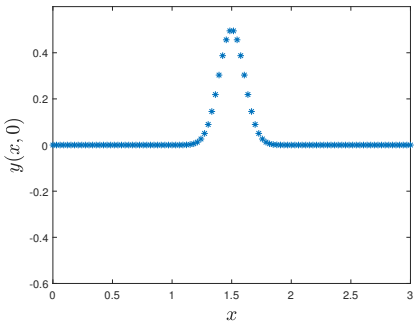
Finally, we note that this approach can be justified rigorously using von Nuemann stability analysis [9]. However, this is beyond the scope of this thesis.

Chapter 4: Results

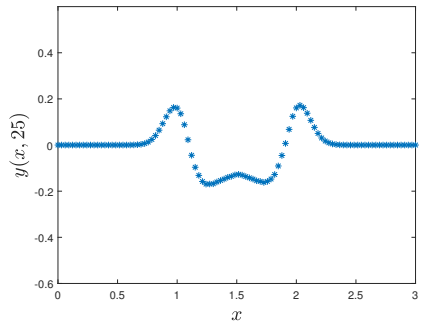
In this chapter, we use the model and methods described in the previous chapters to create simulations of different types of DNA chains. Specifically, we use the Method of Concentrations to generate a chain of pendula with four homogeneous regions. We also simulated a completely inhomogeneous model where each pendulum represented differing nitrogen bases. In both simulations, we tested a chain of varying lengths of pendula and ran the simulation for varying amounts of time.

4.1 Method of Concentrations Simulation

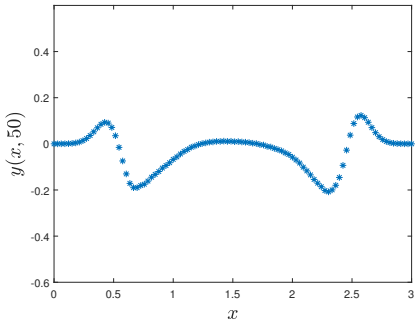
In our work, we first generated a code which modeled chains of varying numbers of pendula using the Method of Concentration. Using this method, we generated four homogeneous regions within the chain with three inhomogeneous jumps between regions. We ran this model with varying time steps M and varying N number of pendula. The code for this simulation is given in Appendix A.1.



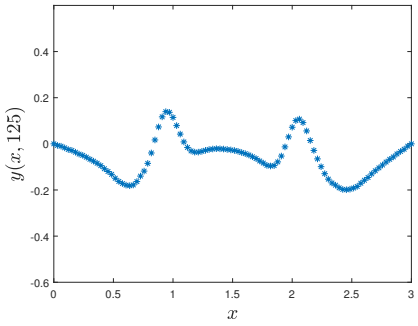
(a) Initial pro le of $y(x;t)$



(b) Solution $y(x;t)$ at $t = 25$



(c) Solution $y(x;t)$ at $t = 50$



(d) Solution $y(x;t)$ at $t = 125$

Figure 4.1: Chain of $N = 100$ pendula with $M = 500$ timesteps

As Figure 4.1 illustrates, the solution demonstrated wave-like behavior. The wave divides and moves towards each boundary smoothly. As the wave hits an inhomogeneous jump between regions, partial transmission and partial reflection occurs causing some back-propagation of the wave. Over time, as the wave passes through more inhomgeneous jumps the solution becomes less smooth due to the back-propagation that occurs.

4.2 Inhomogeneous Model Simulation

The next model we simulated using MatLab was a completely inhomogeneous model. In this model, we were able to generate a chain of N pendula where each pendulum represented a randomly selected base pair. We ran this model with varying N values and varying M time steps. The code for this simulation is given in Appendix A.2

(a) Initial pro le of $y(x;t)$

(b) Solution $y(x;t)$ at $t = 100$

(c) Solution $y(x;t)$ at $t = 300$

(d) Solution $y(x;t)$ at $t = 800$

(e) Solution $y(x;t)$ at $t = 2000$

Figure 4.2: Chain of$N = 200$ pendula with $M = 10000$ timesteps

Similarly to the Method of Concentrations, we see wave-like behavior in our solutions to the inhomogeneous model. However, as Figure 4.1(b) illustrates, because each pendula represents a possibly di erent nitrogen base, back-propagation occurs immediately.

4.3 Combining Simulations

Now, as shown in Appendix A.1, the Method of Concentrations consisted of four homogeneous regions. Within these regions, we randomly selected the number of A, T, C, and G nitrogen bases for each region and used those numbers to create the coe - cients. For the last simulation, we wanted to compare the Method of Concentrations model with the completely inhomogenous model. To do this, we used the random chain generated in our inhomogeneous model to create the homogeneous regions for the Method of Concentrations model. So, we divided the chain into four regions and then counted the number of A, T, C, and G nitrogen bases for each region. This count was then used to create the coe cients. We then graphed both solutions to illustrate and compare their di erences. For this comparison, the code does not change except for generating the coe cients used in the Method of Concentrations simulation. The new coe cient code is given in Appendix A.3.

(a) Initial pro le of $y(x;t)$

(b) Solution $y(x;t)$ at $t = 1000$

(c) Solution $y(x;t)$ at $t = 2000$

(d) Solution $y(x;t)$ at $t = 3000$

(e) Solution $y(x;t)$ at $t = 4000$

(f) Solution $y(x;t)$ at $t = 5000$

Figure 4.3: Chain of $N = 4000$ pendula with $M = 10000$ timesteps using both Method of Concentrations and completely Inhomgeneous models

Figure 4.3(a-b) illustrates that the pulses from both the Method of Concentrations chain and the Inhomogeneous chain line up nicely at rst. However, as the Method of Concentrations chain goes through the rst jump between regions, back propagation begins and the two chains begin to alter dramatically from one another.

Chapter 5: Conclusion and Discussion

In this thesis we studied the discrete analog of the sine-Gordon equation and used this to model a chain of pendula which represented a DNA chain of nitrogen bases. We were able to model two types of chains. The first type consisted of four homogeneous regions which were created using the Method of Concentrations. The second type was a completely inhomogeneous chain where each pendula could represent a different nitrogen base.

As the results illustrated, our solutions in both cases demonstrate behavior similar to solutions to the wave equation. In Section 3.4, it was discussed that the sine-Gordon equation is the wave equation with the added nonlinear sine term. Additionally, it was shown in Section 2.3 that the parameters of a DNA molecule yield a coefficient of smaller magnitude in front of the sine term. Thus, the sine term contributes less to the solution than the $\frac{\partial^2 y}{\partial t^2}$ term does. So, it makes sense that our results illustrated solutions similar to the wave-equation.

However, it is important to remind ourselves what the sine term represents in our model. The DNA molecule consists of two polynucleotide chains that are attached together by the hydrogen bonds present between complementary base pairs. This hydrogen bond is modeled by the sine term and the parameters in front of this term. So, to model a complete DNA molecule, it is necessary to have the sine term present.

In the future, we would like to continue to analyze the completely inhomogeneous chain that was modeled. We would also like to take the continuum limit to derive the PDE for the completely inhomogeneous chain. From this, we could compare solutions to the PDE with solutions to the discrete analog of the sine-Gordon equation.

Additionally, because the DNA molecule consists of two polynucleotide chains, we

would like to incorporate a second chain into our model. In this way, we would be able to model the complete DNA molecule and the replication process.

Bibliography

[1] AA Grinevich, AA Ryasik, and LV Yakushevich. Trajectories of dna bubbles. Chaos, Solitons & Fractals, 75:62{75, 2015.

[2] Bruce Alberts, Dennis Bray, Karen Hopkin, Alexander D Johnson, Julian Lewis, Martin Ra , Keith Roberts, and Peter Walter. Essential cell biology. Garland Science, 2013.

[3] Ludmila Yakushevich. On the mechanical analogue of dna. Journal of biological physics, 43(1):113{125, 2017.

[4] Larisa A Krasnobaeva and Ludmila V Yakushevich. Rotational dynamics of bases in the gene coding interferon alpha 17 (ifna17). Journal of bioinformatics and computational biology, 13(01):1540002, 2015.

[5] LV Yakushevich and LA Krasnobaeva. A new approach to studies of non-linear dynamics of kinks activated in inhomogeneous polynucleotide chains. International Journal of Non-Linear Mechanics, 43(10):1074{1081, 2008.

[6] SW Englander, NR Kallenbach, AJ Heeger, JA Krumhansl, and S Litwin. Nature of the open state in long polynucleotide double helices: possibility of soliton excitations. Proceedings of the National Academy of Sciences, 77(12):7222{7226, 1980.

[7] Francisco Dom nguez-Adame, Angel Sanchez, and Yuri S Kivshar. Soliton pinning by long-range order in aperiodic systems. Physical Review E, 52(3):R2183, 1995.

[8] Ernst Hairer, Christian Lubich, and Gerhard Wanner. Geometric numerical integration illustrated by the strmer/verlet method. Acta Numerica, 12:399{450, 2003.

[9] Peter J Olver. Introduction to partial di erential equations. Springer, 2014.

Appendix A: MatLab Code

A.1 Model Using Method of Concentrations

The rst model we coded consisted of a chain of 500 pendula which we divided into four homogeneous sections using the Method of Concentrations. To begin, we built a code to generate the coe cients for the homogeneous regions and for the jumps in between regions.

Listing A.1: Coe cient Code for Method of Concentration

```
%Coefficients Setup%
%%%%%%%%%%%Realistic Coefficents%%%%%%%%%%%%%%%%%%
IA = 7.61*10^-44;
IT = 5.86*10^-44;
IG = 8.22*10^-44;
IC = 4.11*10^-44;
KAp = 2.27*10^-18; %KA*RA^2
KTp = 1.56*10^-18;
KGp = 2.20*10^-18;
KCp = 1.50*10^-18;
RA = 5.8*10^-10;
RT = 4.8*10^-10;
RG = 5.7*10^-10;
RC = 4.7*10^-10;
kAT = 0.062;
kGC = 0.096;
KA = round(KAp/(RA^2),9);
KT = round(KTp/(RT^2),9);
KC = round(KCp/(RC^2),9);
KG = round(KGp/(RG^2),9);
mA = 2.26*(10^-25);
mT = 2.10938*(10^-25);
mG = 2.53001*(10^-25);
mC = 1.86057*(10^-25);

%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%Set-up for Concentrations
   %%%%%%%%%%%%%
```

```
NA = zeros(1,4);
NT = zeros(1,4);
NC = zeros(1,4);
NG = zeros(1,4);
CA = zeros(1,4);
CT = zeros(1,4);
CG = zeros(1,4);
CC = zeros(1,4);
I = zeros(1,4);
K = zeros(1,4);
R = zeros(1,4);
k = zeros(1,4);
Kp = zeros(1,4);
mass = zeros(1,4);
temp = zeros(1,4);
beta = zeros(1,4);
zeta = zeros(1,4);
kappa = zeros(1,4);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:4
n=25;
m=1:n;
a=m(sort(randperm(25,4)));
b=diff(a);
b(end+1)=25-sum(b);
NA(i) = b(1);
NT(i) = b(2);
NC(i) = b(3);
NG(i) = b(4);
CA(i) = NA(i)/25;
CT(i) = NT(i)/25;
CG(i) = NG(i)/25;
CC(i) = NC(i)/25;
I(i) = IA*CA(i) + IT*CT(i) + IG*CG(i) + IC*CC(i);
Kp(i) = KAp*CA(i) + KTp*CT(i) + KGp*CG(i) + KCp*CC(i);
K(i) = KA*CA(i) + KT*CT(i) + KG*CG(i) + KC*CC(i);
R(i) = RA*CA(i) + RT*CT(i) + RG*CG(i) + RC*CC(i);
k(i) = kAT*(CA(i) + CT(i)) + kGC*(CG(i) + CC(i));
mass(i) = mA*CA(i) + mT*CT(i) + mG*CG(i) + mC*CC(i);
temp(i) = (Kp(i))/I(i);
beta(i) = (k(i)*(R(i))^2)/I(i);
end
```

```
%%%%average K for between regions%%%%
KJ1 = (K(1)+K(2))/2;
KJ2 = (K(2) + K(3))/2;
KJ3 = (K(3) + K(4))/2;
%%%%%alphas at jumps%%%%
alphaJ1p = (KJ1*R(2))/(mass(1)*R(1)); %alpha_jump1_+
alphaJ10 = (KJ1 + K(1))/mass(1);%alpha_jump1_0
alphaJ110 = (K(2) + KJ1)/(mass(2));%alpha_jump1+1_0
alphaJ11m = (KJ1*R(1))/(mass(2)*R(2)); %alpha_jump1+1_-
alphaJ2p = (KJ2*R(3))/(mass(2)*R(2)); %alpha_jump2_+
alphaJ20 = (KJ2 + K(2))/(mass(2));%alpha_jump2_0
alphaJ210 = (K(3) + KJ2)/mass(3);%alpha_jump2+1_0
alphaJ21m = (KJ2*R(2))/(mass(3)*R(3));
alphaJ3p = (KJ3*R(3))/(mass(3)*R(3));
alphaJ30 = (KJ3 + K(3))/(mass(3));
alphaJ310 = (K(4) + KJ3)/mass(4);
alphaJ31m = (KJ3*R(3))/(mass(4)*R(4));
%%%%%%%%%%%%%%
alpha = [temp(1), temp(2), temp(3), temp(4), alphaJ1p,
   alphaJ10, alphaJ110, alphaJ11m, alphaJ2p, alphaJ20,
   alphaJ210, alphaJ21m, alphaJ3p, alphaJ30, alphaJ310,
   alphaJ31m];
%%%%%%%%%%%KAPPAS AT JUMPS%%%%%%%%%%%%%
kappaJ1p = round(alphaJ1p/max(alpha),9);
kappaJ10 = round(alphaJ10/max(alpha),9);
kappaJ110 = round(alphaJ110/max(alpha),9);
kappaJ11m = round(alphaJ11m/max(alpha),9);
kappaJ2p = round(alphaJ2p/max(alpha),9);
kappaJ20 = round(alphaJ20/max(alpha), 9);
kappaJ210 = round(alphaJ210/max(alpha),9);
kappaJ21m = round(alphaJ21m/max(alpha),9);
kappaJ3p = round(alphaJ3p/max(alpha),9);
kappaJ30 = round(alphaJ30/max(alpha),9);
kappaJ310 = round(alphaJ310/max(alpha),9);
kappaJ31m = round(alphaJ31m/max(alpha),9);
kappaJ1 = [kappaJ1p, kappaJ10, kappaJ110, kappaJ11m];
kappaJ2 = [kappaJ2p, kappaJ20, kappaJ210, kappaJ21m];
kappaJ3 = [kappaJ3p, kappaJ30, kappaJ310, kappaJ31m];

%%%%%%%%%%%%%%%
%%%%%%%%%%CHANGE OF VARIABLES%%%%%%%%%%%%%%%
zeta(1) = round(beta(1)/max(alpha),9);
```

```
zeta(2) = round(beta(2)/max(alpha),9);
zeta(3) = round(beta(3)/max(alpha),9);
zeta(4) = round(beta(4)/max(alpha),9);
kappa(1) = round(alpha(1)/max(alpha),9);
kappa(2) = round(alpha(2)/max(alpha),9);
kappa(3) = round(alpha(3)/max(alpha),9);
kappa(4) = round(alpha(4)/max(alpha),9);
N = 500;
mu = zeros(1,N);
for t = 1:125
    mu(1,t) = kappa(1);

end
for t = 126:250
    mu(1,t) = kappa(2);
end
for t = 251:375
    mu(1,t) = kappa(3);
end
for t = 376:500
    mu(1,t) = kappa(4);
end
%dlmwrite('mu4.txt', mu, 'delimiter ', '\t')
```

Using the coe cients generated, we then built the matrix whose entries were given in Eqs. (3.8-10).

Listing A.2: Code for Generating Matrix

```
function [ D ] = Dmatrix_500( mu, kappaJ1, kappaJ2,
   kappaJ3)
%digits(6)
[~,n] = size(mu);
k0 = mu;
%n = m-1;
mid = zeros(1,n);
mid(1) = -2*k0(1);
%mid(n) = -k0(n-1) - k0(n);
mid(n) = -2*k0(n);
for j = 2:n-1
    mid(j) = -(k0(j-1) + k0(j));

end
```

```
%Rright(1:n-1) = R(2:n);
%Rright(n) = 0;
up = k0(1:n-1);
%Rleft(2:n) = R(1:n-1);
%Rleft(1) = 0;

D = diag(up, 1) + diag(mid, 0) + diag(up, -1);
D(125,124) = k0(24);
D(125,125) = -kappaJ1(2);
D(125,126) = kappaJ1(1);
D(126,125) = kappaJ1(4);
D(126,126) = -kappaJ1(3);
D(126,127) = k0(26);
D(250,249) = k0(49);
D(250,250) = -kappaJ2(2);
D(250,251) = kappaJ2(1);
D(251,250) = kappaJ2(4);
D(251,251) = -kappaJ2(3);
D(251,252) = k0(51);
D(375,374) = k0(74);
D(375,375) = -kappaJ3(2);
D(375,376) = kappaJ3(1);
D(376,375) = kappaJ3(4);
D(376,376) = -kappaJ3(3);
D(376,377) = k0(76);
dlmwrite( 'Dmatrix.txt ', D, 'delimiter ', '\t ')
end
```

Finally, using this matrix, we created our Verlet Integration algorithm to approximate solutions to our system of di erential equations.

Listing A.3: Verlet Integration Code

```
function [ y ] = verlet1( zeta, alphaM )
M= 500; % number of timesteps

N = 500; %number of pendula
s = 0.03; %spacestep
x=linspace(0, N*s, N);
dx = x(2) - x(1);
dt= (1/10)*(dx/alphaM); %CFL Condition
```

```
beta = zeros(N,N);

D = dlmread('Dmatrix.txt');
D = ((dt^2)/(dx^2))*D;

for t=1:125

    beta(t,t) = zeta(1);
end

for t = 126:250

    beta(t,t) = zeta(2);
end

for t = 251:375

    beta(t,t) = zeta(3);
end

for t = 76:100

    beta(t,t) = zeta(4);
end

dlmwrite('betaTEST.txt', beta, 'delimiter', '\t')

y = zeros(N,M);
A = zeros(N,M);
y0 = 1/2*exp(-((x-(N*s/2)).^2/(5*s)^2));
%v0 = exp(-((x-(N*a)).^2/(10*a)^2));
v0=0;
y0=y0';
v0=v0';
y(:,1) = y0;
y(:,2) = y0 + v0*dt + 0.5*(D*y0 - (dt^2)*beta*sin(y0));
y(1,2) = 0;
y(end, 2) = 0;
for i = 2:M-1
    A(:,i) = D*y(:,i) - (dt^2)*beta*sin(y(:,i));
    y(:,i+1) = 2*y(:,i) - y(:,i-1) +A(:,i);
```

```
    y(1,i+1) = 0; %Boundary Condition
    y(end,i+1) = 0; %Boundary Condition
    dlmwrite('V_ydata_MOC.txt',y, 'delimiter', '\t')

end

load handel
sound(y,Fs)
end
```

A.2 Inhomogeneous Model

Next, we created a code that modeled a chain of 500 pendula where each pendulum were randomly selected. To do this, we again begin with the code for the coe cients. In this code, we generated a 500 1 vector where each entry was a randomly selected number which represented one of the four base pairs. The coe cients were then generated based o of these entries.

Listing A.4: Coe cient Code for Inhomogeneous Model

```
%%%%%%%%%%%%Coefficients%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%Realistic Coefficients%%%%%%%%%%%%
IA = 7.61*10^-44;
IT = 5.86*10^-44;
IG = 8.22*10^-44;
IC = 4.11*10^-44;
KAp = 2.27*10^-18; %KA*RA^2
KTp = 1.56*10^-18;
KGp = 2.20*10^-18;
KCp = 1.50*10^-18;
RA = 5.8*10^-10;
RT = 4.8*10^-10;
RG = 5.7*10^-10;
RC = 4.7*10^-10;
k_AT = 0.062;
k_GC = 0.096;
KA = round(KAp/(RA^2),9);
KT = round(KTp/(RT^2),9);
```

```
KC = round(KCp/(RC^2),9);
KG = round(KGp/(RG^2),9);
KAT = (KA + KT)/2;
KAG = (KA + KG)/2;
KAC = (KA + KC)/2;
KTC = (KT + KC)/2;
KTG = (KT + KG)/2;
KGC = (KG + KC)/2;
mA = 2.26*(10^-25);
mT = 2.10938*(10^-25);
mG = 2.53001*(10^-25);
mC = 1.86057*(10^-25);
%%%%%%%%% Generating Random Chain of DNA %%%%%%%%%%%%%

Bases = [1 2 3 4];
A = Bases(1);
T = Bases(2);
C = Bases(3);
G = Bases(4);
N = 200; %Number of Bases/Pendula
DNAChain = zeros(1, N);
for i=1:N
  DNAChain(i) = randperm(length(Bases), 1);

end
dlmwrite( 'DNAChain.txt', DNAChain, 'delimiter ', '\t ')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Kright = zeros(1, N-1);%%Vector for Covalent Bond Pairs


alpha_right = zeros(1, N-1);
alpha_left = zeros(1, N-1);
alpha_center = zeros(1, N);
beta = zeros(1, N);
m = zeros(1, N);
R = zeros(1, N);
k = zeros(1, N);

for i=1:N-1
    if DNAChain(i)==A && DNAChain(i+1)==A
        Kright(i) = KA;
```

```
    elseif DNAChain(i)==A && DNAChain(i+1)==T
        Kright(i) = KAT;
    elseif DNAChain(i)==A && DNAChain(i+1)==G
       Kright(i) = KAG;
    elseif DNAChain(i)==A && DNAChain(i+1)==C
        Kright(i) = KAC;
    elseif DNAChain(i)==T && DNAChain(i+1)==T
        Kright(i) = KT;
    elseif DNAChain(i)==T && DNAChain(i+1)==A
        Kright(i) = KAT;
    elseif DNAChain(i)==T && DNAChain(i+1)==G
        Kright(i) = KTG;
    elseif DNAChain(i)==T && DNAChain(i+1)==C
        Kright(i) = KTC;
    elseif DNAChain(i)==G && DNAChain(i+1)==G
        Kright(i) = KG;
    elseif DNAChain(i)==G && DNAChain(i+1)==A
        Kright(i) = KAG;
    elseif DNAChain(i)==G && DNAChain(i+1)==T
        Kright(i) = KTG;
    elseif DNAChain(i)==G && DNAChain(i+1)==C
        Kright(i) = KGC;
    elseif DNAChain(i)==C && DNAChain(i)==C
        Kright(i) = KC;
    elseif DNAChain(i)==C && DNAChain(i)==A
        Kright(i)=KAC;
    elseif DNAChain(i)==C && DNAChain(i)==T
        Kright(i)=KTC;
    else
        Kright(i) = KGC;
    end
end
for i = 1:N
    if DNAChain(i)==A
        m(i) = mA;
        R(i) = RA;
        k(i) = k_AT;
    elseif DNAChain(i)==T
        m(i) = mT;
        R(i) = RT;
        k(i) = k_AT;
    elseif DNAChain(i)==C
```

```
            m(i) = mC;
            R(i) = RC;
            k(i) = k_GC;
        else
            m(i) = mG;
            R(i) = RG;
            k(i) = k_GC;
        end
end
alpha_right(1) = (Kright(1)*R(2))/(m(1)*R(1));
alpha_left(1) = (Kright(1)*R(1))/(m(2)*R(2));
alpha_left(N-1) = (Kright(N-1)*R(N-1))/(m(N)*R(N));
alpha_right(N-1) = (Kright(N-1)*R(N))/(m(N-1)*R(N-1));
alpha_center(N-1) = (Kright(N-2) + Kright(N-1))/m(N-1);
alpha_center(1) = Kright(1)/m(1);
alpha_center(N) = Kright(N-1)/m(N);
beta(1) = k(1)/m(1);
beta(N) = k(N)/m(N);
for j=2:N-2
    alpha_left(j) = (Kright(j)*R(j))/(m(j+1)*R(j+1));
    alpha_right(j) = (Kright(j)*R(j+1))/(m(j)*R(j));
    alpha_center(j) = (Kright(j-1) + Kright(j))/m(j);
    beta(j) = k(j)/m(j);
end
temp = [alpha_right alpha_left alpha_center];
alphaM = max(temp);
alpha_right = (1/alphaM).*alpha_right;
alpha_left = (1/alphaM).*alpha_left;
alpha_center = (1/alphaM).*alpha_center;
beta = (1/alphaM).*beta;
```

After the coe cients were generated, we made the code for the Verlet Integration algorithm. Within this code, we also generated our matrix.

Listing A.5: Verlet Integration Code for Inhomogeneous Model

```
function [ y ] = verlet3( alpha_right, alpha_center, beta,
    N )
M= 500; % number of timesteps
 %number of pendulum
s = 0.03; %space between bases
x=linspace(0, N*s, N);
dx = x(2) - x(1);%spacester
```

```
%%%%%%%%%%%%Matrix for u_xx%%%%%%%%%%%%%%%%%%%%%%%%%%%%
D = diag(alpha_right,1)- diag(alpha_center,0)+diag(
   alpha_right,-1); %matrix for u_xx
temp = [alpha_right alpha_center];
alphaM = max(temp);
dt= (1/10)*(dx/alphaM); %timestep  - CFL
D = ((dt^2)/(dx^2))*D;
dlmwrite('Dmatrix.txt', D, 'delimiter', '\t')
%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%Coefficients In Front of Sine Term
beta = diag(beta);
dlmwrite('betaTEST.txt', beta, 'delimiter', '\t')
%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%Initializing%%%%%%%%%%%%%%%%%%%%%%%%%%
y = zeros(N,M);
A = zeros(N,M);
y0 = 1/2*exp(-((x-(N*s/2)).^2/(5*s)^2));%Initial Profile
%v0 = exp(-((x-(N*a)).^2/(10*a)^2));
v0=0;
y0=y0';
v0=v0';
y(:,1) = y0;
y(:,2) = y0 + v0*dt + 0.5*(D*y0 - (dt^2)*beta*sin(y0));
y(1,2) = 0;
y(end, 2) = 0;
for i = 2:M-1
     A(:,i) = D*y(:,i) - (dt^2)*beta*sin(y(:,i));
     y(:,i+1) = 2*y(:,i) - y(:,i-1) + A(:,i);
     y(1,i+1) = 0; %Boundary Condition
     y(end,i+1) = 0; %Boundary Condition
     dlmwrite('V_ydata_200chain_ihomo-symmetric.txt',y, '
        delimiter', '\t')

end

load handel
sound(y,Fs)
end
```

Appendix B: Centered Difference Methods

In this thesis we use a centered difference method to approximate several derivatives. In the Hamiltonian section in Chapter 2, we use a centered difference approximation for our velocity terms. We also use a centered difference method for $\frac{\partial^3 y}{\partial t^3}$ and $\frac{\partial^3 y}{\partial x^3}$ to derive our numerical scheme in the Verlet Integration section. Finally, to analyze the results of our model, we take the derivative of our Hamiltonian using a centered difference method. So, it is important for us to understand the derivation of this method for approximating the derivative.

First, let us derive an approximation for the second derivative. Letting h denote our step size, we will assume this to be small, $|h| \ll 1$. Next, we will that $y(x)$ is at least three times continuously differentiable and look at the Taylor expansion for $y(x + h)$ and $y(x - h)$. So, we have;

$$y(x + h) = y(x) + y'(x)h + y''(x)\frac{h^2}{2} + y'''(x)\frac{h^3}{6} + O(h^4) \qquad \text{(B.1)}$$

$$y(x - h) = y(x) - y'(x)h + y''(x)\frac{h^2}{2} - y'''(x)\frac{h^3}{6} + O(h^4) \qquad \text{(B.2)}$$

Note that $O(h^4)$ is the approximate error term and denotes that the approximation has error proportional to h^4. Now, to get our relevant approximation, we add A.1 and A.2 together to obtain;

$$y(x + h) + y(x - h) = 2y(x) + y''(x)h^2 + O(h^4) \qquad \text{(B.3)}$$

From here, we solve for $y''(x)$ and obtain;

$$y''(x) = \frac{y(x + h) - 2y(x) + y(x - h)}{h^2} + O(h^2) \qquad \text{(B.4)}$$

Now, it is important to note that this approximation can also be used for a function of two variables, $y(x;t)$. Then, the step size h will correspond to either the t or x variable depending on which variable you are taking the derivative with respect to. For example, in the Verlet integration section, we use A.4 for $\frac{\partial^2 y}{\partial t^2}$ and $\frac{\partial^2 y}{\partial x^2}$. So, assuming that our error terms are sufficiently small, we have;

$$\frac{\partial^2 y}{\partial t^2} \quad \frac{y(x;t+\ t) \quad 2y(x;t) + y(x;t \quad t)}{t^2} \quad h = \quad t \tag{B.5}$$

$$\frac{\partial^2 y}{\partial t^2} \quad \frac{y(x+\ x;t) \quad 2y(x;t) + y(x \quad x;t)}{x^2} \quad h = \quad x \tag{B.6}$$

Now, note that to solve for $y''(x)$ in A.3 we divide through by h^2 to obtain A.4. This changes are approximate error term to $O(h^2)$ so that the error is now proportional to h^2. Thus, A.4 is a second-order approximation. Now, because we use A.4 in our numerical scheme to solve our system of differential equations, this will dictate that all other derivative approximations used must also be second-order.

Next, we derive an approximation for the first derivative. Because this must be a second-order approximation, we again use the Taylor expansion for $y(x+h)$ and $y(x \quad h)$. So, subtracting A.2 from A.1, we obtain;

$$y(x+h) \quad y(x \quad h) \quad 2y'(x)h + O(h^3)$$

$$y'(x) \quad \frac{y(x+h) \quad y(x \quad h)}{2h} + O(h^2) \tag{B.7}$$

Once again, this approximation can also be used for a function of two variables. So, given $y(x;t)$, we have;

$$\frac{\partial y}{\partial t} \quad \frac{y(x;t+\ t) \quad y(x;t \quad t)}{2 \quad t} \quad h = \quad t \tag{B.8}$$

$$\frac{\partial y}{\partial x} \quad \frac{y(x+\ x;t) \quad y(x \quad x);t)}{2 \quad x} \quad h = \quad x \tag{B.9}$$

&XUULFXOXP 9LWD
Susan Rogowski

C: 804-572-3195 | E: rogosc17@wfu.edu

Education

Wake Forest University, Winston-Salem, NC — Expected Graduation: May 2019
M.A. Mathematics

James Madison University, Harrisonburg, VA — May 2015
B.S. Mathematics
Second Major: B.S. in Media, Arts, & Design with a concentration in Web Design

Academic Awards and Honors

Graduate Ambassadorship, Women in Math Program at the Institute for Advanced Study — Aug 2018
Graduate Teaching Assistantship, Wake Forest University — June 2018
Summer Research Award, Wake Forest University — Mar 2018

Research

Research Interests:
Mathematical Modeling, bio math, partial differential equations

0DVWHU¶V7KHVLV Modeling DNA Replication using the sine-Gordon Equation.

Presentations

1. Nov 2018 ±Regional Math and Stats Conference, Greensboro, NC
2. Nov 2018 ±Wake Forest AWM Brown Bag Series, Winston-Salem, NC
3. Aug 2018 ±Brown Bag Summer Research Series, Winston-Salem, NC

Teaching Experience

Teaching Assistant, Math Center, Wake Forest University — May 2018 ±Present

- Assist the Director of the Math Center to prepare the tutoring center for Summer, Fall, and Spring semesters
- Collaborate with the department faculty to obtain class materials to assist with tutoring assignments
- Help with training and orientation of new tutors
- Gained expert knowledge in Calculus I, II, and Linear Algebra by tutoring students one-on-one in these subjects

Teaching Assistant, Linear Algebra, Wake Forest University — July 2018 ±Aug 2018

Refine independent learning skills through teaching myself linear algebra concepts in order to tutor students
Increased understanding of the subject by grading homework assignments

Private Tutor, High School Algebra Nov 2017 – May 2018

Service

Co-organized Piedmont-Triad Association of Women in Mathematics (AWM) conference
Organized Career Panel event for Association of Women in Mathematics
Organized Association of Women in Mathematics Brown Bag Talks
Volunteered for Math Counts program

Professional Organizations

Association of Women in Mathematics – national member and president of campus chapter

American Mathematical Society – member

Mathematical Software

Matlab, LaTeX